Presentation about decision tree、neural network、Bayes classifier

BY Yuxuan Luo

Include:

1.Decision tree

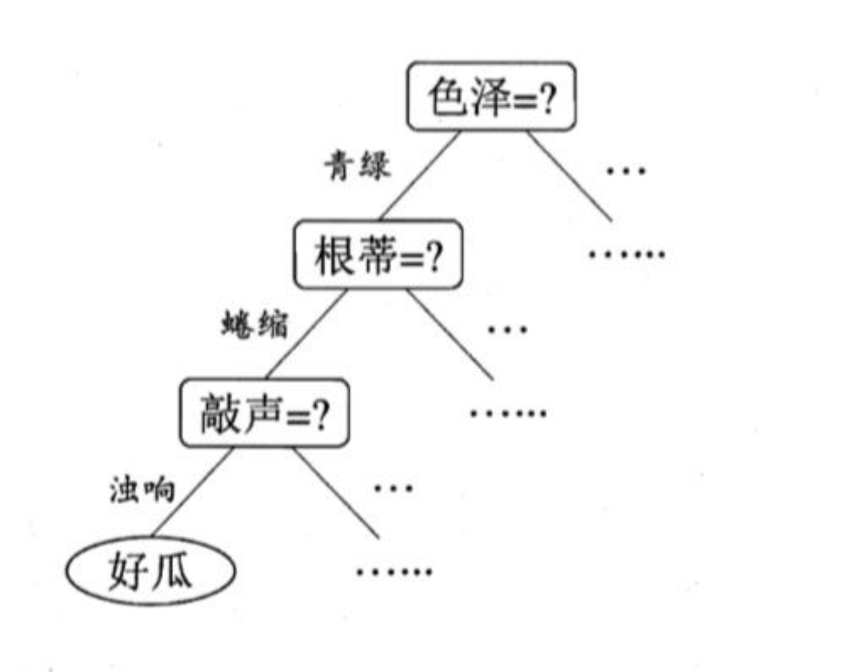
2.Neural network

3.Bayes classifier

4.Which is difficult to realize？

1.Decision tree

What is decision tree?



How decision tree works?

Like a watermelon,it has several features such as color、stripe、root ,etc.Through these features,we can generally judge whether a good watermelon or not.From the picture beside,if the watermelon color is green,the root is curve and the patting sound is not clear we consider this watermelon is good.

Create a decision tree(ID 3)

In order to create the most suitable decision tree,we should find out the best feature as the tree root node.

Calculate information entropy、information gain

$$Ent(D) = -\sum_{k=1}^{|y|} p_k \log_2 p_k$$

$$Gain(D,a) = Ent(D) - \sum_{v=1}^{V} \frac{|D^v|}{|D|} Ent(D^v)$$

$$Ent(D) = -\sum_{k=1}^{2} p_k \log_2 p_k = -\left(\frac{8}{17}\log_2\frac{8}{17} + \frac{9}{17}\log_2\frac{9}{17}\right) = 0.998$$

表 4.1 西瓜数据集 2.0

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	是
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	否
10	青绿	硬挺	清脆	清晰	平坦	软粘	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	否
17	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	否

We need to calculate all features' information entropy and information gain

Select feature $a_* = \arg max\, Gain(D, a)$ as decision attribute

Ent(color):

$$Ent(D^1) = -\left(\frac{3}{6}\log_2\frac{3}{6} + \frac{3}{6}\log_2\frac{3}{6}\right) = 1$$

$$Ent(D^2) = -\left(\frac{4}{6}\log_2\frac{4}{6} + \frac{2}{6}\log_2\frac{2}{6}\right) = 0.918$$

$$Ent(D^3) = -\left(\frac{1}{5}\log_2\frac{1}{5} + \frac{4}{5}\log_2\frac{4}{5}\right) = 0.722$$

$$Gain(D, color) = Ent(D) - \sum_{v=1}^{3}\frac{|D^v|}{|D|}Ent(D^v) = 0.998 - (\frac{6}{17} * 1 + \frac{6}{17} * 0.918 + \frac{5}{17} * 0.722) = 0.109$$

Ent(root):

$$Ent(D^1) = -\left(\frac{3}{8}\log_2\frac{3}{8} + \frac{5}{8}\log_2\frac{5}{8}\right) = 0.955$$

$$Ent(D^2) = -\left(\frac{4}{7}\log_2\frac{4}{7} + \frac{3}{7}\log_2\frac{3}{7}\right) = 0.985$$

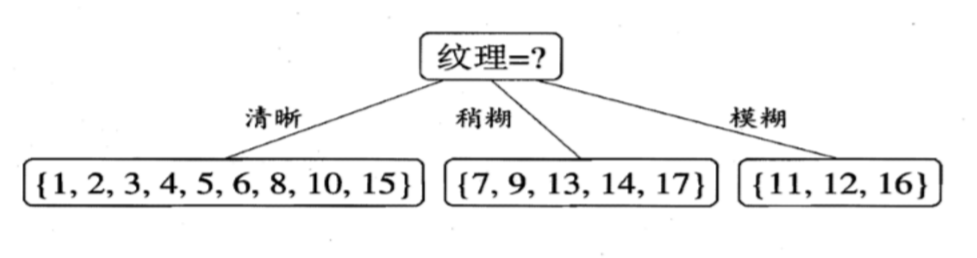
$$Ent(D^3) = -\left(\frac{2}{2}\log_2\frac{2}{2} + 0\log_2 0\right) = 0$$

$$Gain(D, root) = Ent(D) - \sum_{v=1}^{3}\frac{|D^v|}{|D|}Ent(D^v) = 0.998 - (\frac{8}{17} * 0.955 + \frac{7}{17} * 0.985 + \frac{2}{17} * 0) = 0.142$$

$Gain(D, patting\ sound) = 0.141$ $\quad$ $Gain(D, stripe) = 0.381$ → Selected as decision attribute

$Gain(D, belly) = 0.0.289$ $\quad$ $Gain(D, touch) = 0.006$

When we got the root ,need further calculation to get second decision attribute



$$\text{Ent}(D^1)=-\left(\frac{7}{9}\log_2\frac{7}{9}+\frac{2}{9}\log_2\frac{2}{9}\right)=0.764$$

$$\text{Ent}(D^1,\text{green})=-\left(\frac{3}{4}\log_2\frac{3}{4}+\frac{1}{4}\log_2\frac{1}{4}\right)=0.811$$

$$\text{Ent}(D^1,\text{white})=-(0\log_2 0+1\log_2 1)=0$$

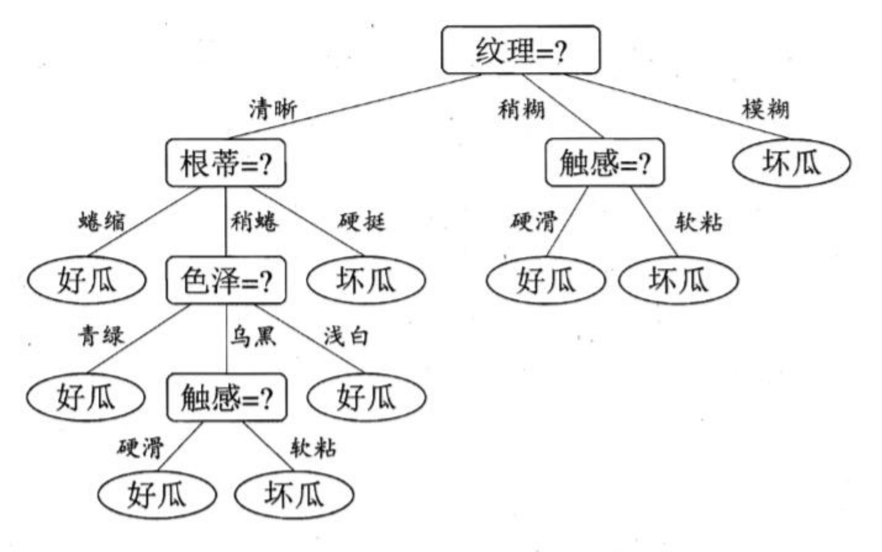
$$\text{Ent}(D^1,\text{dark})=-\left(\frac{3}{4}\log_2\frac{3}{4}+\frac{1}{4}\log_2\frac{1}{4}\right)=0.811$$

$$Gain(D^1,color)=Ent(D)-\sum_{v=1}^{3}\frac{|D^v|}{|D|}Ent(D^v)=0.764-\left(\frac{4}{9}*0.811+\frac{1}{9}*0.918+\frac{4}{9}*0.811\right)=0.044$$

$Gain(D^1,root)=0.458$ $\quad$ $Gain(D^1,patting\ sound)=0.331$

$Gain(D^1,belly)=0.458$ $\quad$ $Gain(D^1,touch)=0.458$

One of three attributes can be the decision one
And after calculation,finally get a decision tree.



Other method generate decision tree

1 C4.5 decision tree

C4.5 decision tree use Gain_ratio instead Gain to decide decision attribute

$$Gain\_ratio(D,a) = \frac{Gain(D,a)}{IV(a)} \qquad IV(a) = -\sum_{v=1}^{V} \frac{|D^v|}{|D|} \log_2 \frac{|D^v|}{|D|}$$

Select Gain(D,a) higher than average and max Gain_ratio(D,a) as decision attribute.

2 CART decision tree

CART decision tree use Gini_index instead Gain to decide decision attribute

$$Gini(D) = \sum_{k=1}^{|y|} \sum_{k' \neq k} p_k p_{k'} = 1 - \sum_{k=1}^{|y|} p_k^2 \qquad Gini\_index(D,a) = \sum_{v=1}^{V} \frac{|D^v|}{|D|} Gini(D^v)$$

The lower Gini(D),the more purity of Dataset D.Select mini $Gini\_index(D,a)$ as decision attribute.

In order to avoid overfitting problem,we use pruning to improve accuracy.

Continuous attributes —— bi-partition

$$T_a = \left\{ \frac{a^i + a^{i+1}}{2} \middle| 1 \leq i \leq n-1 \right\}$$

T_a as a node to calculate entropy and gain

$$Gain(D,a) = \max_{t \in T_a} Gain(D,a,t) = \max_{t \in T_a} Ent(D) - \sum_{\lambda \in \{-,+\}} \frac{|D_t^\lambda|}{|D|} Ent(D_t^\lambda)$$

表 4.3 西瓜数据集 3.0

编号	色泽	根蒂	敲声	纹理	脐部	触感	密度	含糖率	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	0.697	0.460	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	0.774	0.376	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	0.634	0.264	是
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	0.608	0.318	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	0.556	0.215	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	0.403	0.237	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	0.481	0.149	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	0.437	0.211	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	0.666	0.091	否
10	青绿	硬挺	清脆	清晰	平坦	软粘	0.243	0.267	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	0.245	0.057	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	0.343	0.099	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	0.639	0.161	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	0.657	0.198	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	0.360	0.370	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	0.593	0.042	否
17	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	0.719	0.103	否

e.g. $T = \frac{0.243 + 0.245}{2} = 0.244$ $\quad T = \frac{0.245 + 0.343}{2} = 0.294$

$$T_{\text{density}} = \{0.244, 0.294, 0.351, 0.381, 0.420, 0.459, 0.518, 0.574, 0.600, 0.621, 0.636, 0.648, 0.661, 0.681, 0.708, 0.746\}$$

$$Ent(D,a,-0.244) = -(0\log_2 0 + 1\log_2 1) = 0 \qquad Ent(D,a,+0.244) = -\left(\frac{8}{16}\log_2\frac{8}{16} + \frac{8}{16}\log_2\frac{8}{16}\right) = 1$$

$$Gain(D, density, 0.244) = Ent(D) - \left(\frac{16}{17} * 1 + \frac{1}{17} * 0\right) = 0.998 - 0.941 = 0.057$$

Missing value

We give those not missing value a weight,and modify the function

$$\rho = \frac{\sum_{\boldsymbol{x} \in \tilde{D}} w_{\boldsymbol{x}}}{\sum_{\boldsymbol{x} \in D} w_{\boldsymbol{x}}} ,$$

$$\tilde{p}_k = \frac{\sum_{\boldsymbol{x} \in \tilde{D}_k} w_{\boldsymbol{x}}}{\sum_{\boldsymbol{x} \in \tilde{D}} w_{\boldsymbol{x}}} \quad (1 \leqslant k \leqslant |\mathcal{Y}|) ,$$

$$\tilde{r}_v = \frac{\sum_{\boldsymbol{x} \in \tilde{D}^v} w_{\boldsymbol{x}}}{\sum_{\boldsymbol{x} \in \tilde{D}} w_{\boldsymbol{x}}} \quad (1 \leqslant v \leqslant V) .$$

$$\begin{aligned} \mathrm{Gain}(D, a) &= \rho \times \mathrm{Gain}(\tilde{D}, a) \\ &= \rho \times \left(\mathrm{Ent}\left(\tilde{D}\right) - \sum_{v=1}^{V} \tilde{r}_v \, \mathrm{Ent}\left(\tilde{D}^v\right) \right) \end{aligned}$$

$$\mathrm{Ent}(\tilde{D}) = -\sum_{k=1}^{|\mathcal{Y}|} \tilde{p}_k \log_2 \tilde{p}_k .$$

表 4.4 西瓜数据集 2.0α

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
1	–	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	–	是
3	乌黑	蜷缩	–	清晰	凹陷	硬滑	是
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	–	蜷缩	浊响	清晰	凹陷	硬滑	是
6	青绿	稍蜷	浊响	清晰	–	软粘	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是
8	乌黑	稍蜷	浊响	–	稍凹	硬滑	是
9	乌黑	–	沉闷	稍糊	稍凹	硬滑	否
10	青绿	硬挺	清脆	–	平坦	软粘	否
11	浅白	硬挺	清脆	模糊	平坦	–	否
12	浅白	蜷缩	–	模糊	平坦	软粘	否
13	–	稍蜷	浊响	稍糊	凹陷	硬滑	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否
15	乌黑	稍蜷	浊响	清晰	–	软粘	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	否
17	青绿	–	沉闷	稍糊	稍凹	硬滑	否

e.g. Ent(color): $Ent(\widetilde{D}) = -\sum_{k=1}^{2} \tilde{p}_k \log_2 \tilde{p}_k = -\left(\frac{6}{14}\log_2\frac{6}{14} + \frac{8}{14}\log_2\frac{8}{14}\right) = 0.985$

$$Ent(\widetilde{D}^1) = -\left(\frac{2}{4}\log_2\frac{2}{4} + \frac{2}{4}\log_2\frac{2}{4}\right) = 1 \qquad Ent(\widetilde{D}^2) = -\left(\frac{4}{6}\log_2\frac{4}{6} + \frac{2}{6}\log_2\frac{2}{6}\right) = 0.918$$

$$Ent(\widetilde{D}^3) = -\left(\frac{0}{4}\log_2\frac{0}{4} + \frac{4}{4}\log_2\frac{4}{4}\right) = 0$$

$$Gain(\widetilde{D}, color) = Ent(\widetilde{D}) - \sum_{v=1}^{3} \tilde{\gamma}_v Ent(\widetilde{D}^v) = 0.985 - \left(\frac{4}{14} * 1 + \frac{6}{14} * 0.918 + \frac{4}{14} * 0\right) = 0.306$$

Decision tree application

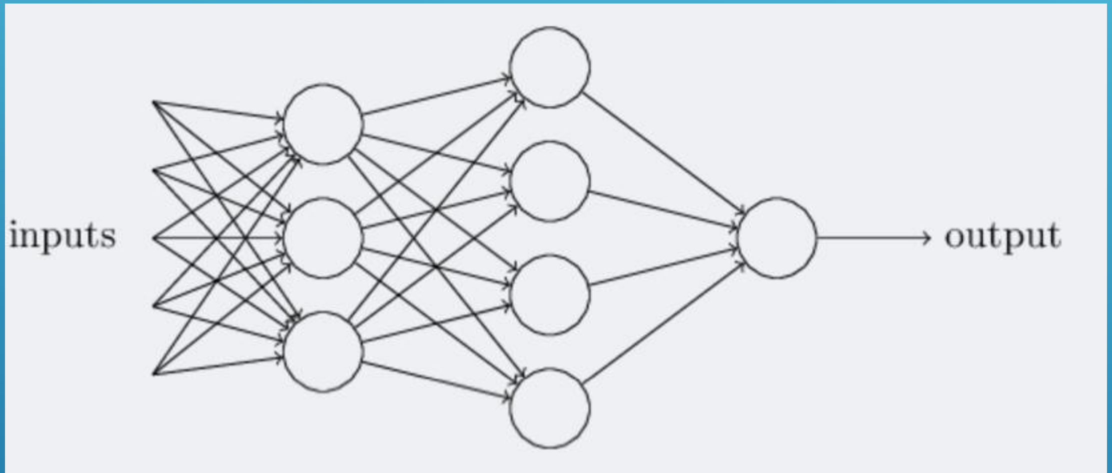
Classify something like whether a product qualify or not

Bank pass the applicant of Credit Card or not

Auxiliary medical system

2. Neural Networks

What is neural networks?

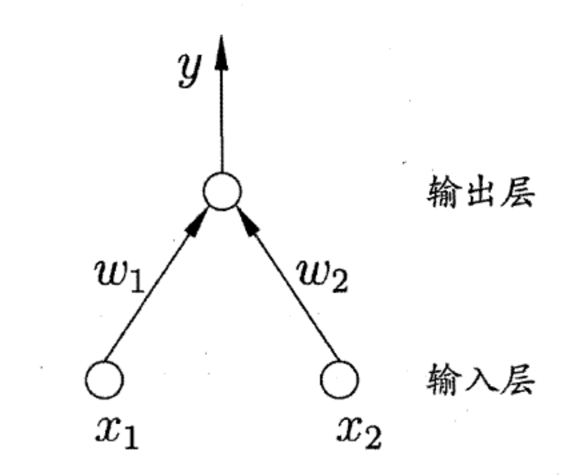


Neural networks include single-layer network call Perceptron and multi-layer networks

Neural networks include three parts:
①Architecture
②Activity function
③Learning rule

How neural networks works?

Perceptron consist of two layers: input 、output.Perceptron easily solves "∧,∨, ¬"problem.



$y = f(\sum_i w_i x_i - \theta)$

Activity function: $\mathrm{sgn}(x) = \begin{cases} 1, & x \geq 0; \\ 0, & x < 0; \end{cases}$

e.g. $x_1 \wedge x_2$ set $w_1 = w_2 = 1, \theta = 2$ $y = f(1 * x_1 + 1 * x_2 - 2)$

Only when $x_1 = x_2 = 1, y = 1$

$x_1 \vee x_2$ set $w_1 = w_2 = 1, \theta = 0.5$ $y = f(1 * x_1 + 1 * x_2 - 0.5)$

Only when $x_1 = 1 \; or \; x_2 = 1, y = 1$

$\neg x_1$ set $w_1 = -0.6, w_2 = 0, \theta = -0.5$ $y = f(-0.6 * x_1 + 0 * x_2 + 0.5)$

when $x_1 = 1, y = 0$;when $x_1 = 0, y = 1$

Error BackPropagation neural networks

Parameter update function: $v \leftarrow v + \Delta v$

$Dataset = \{(x_1, y_1), (x_2, y_2), \dots\dots, (x_m, y_m)\}, x_i \in \mathbb{R}^d, y_i \in \mathbb{R}^l$

$f(x) = sigmoid(x) = \dfrac{1}{1 + e^{-x}}$ Learing rate: η

Node h receive input $\alpha_h = \sum_{i=1}^{d} v_{ih} x_i$; threshold γ_h

Node j output $\beta_j = \sum_{h=1}^{q} w_{hj} b_h$; threshold θ_j

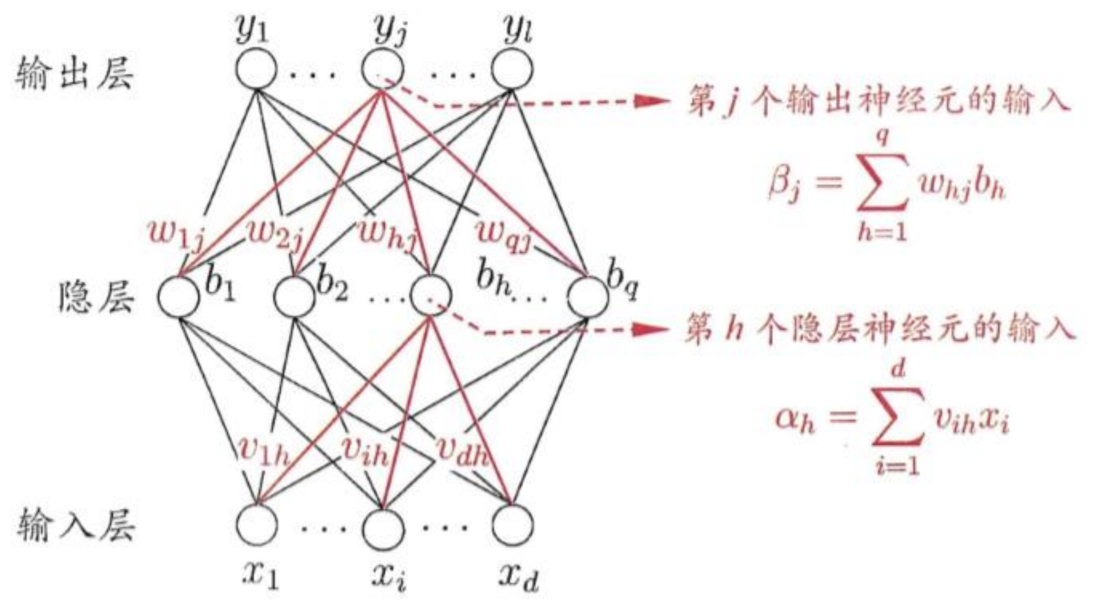


图 5.7 BP 网络及算法中的变量符号

Set neural networks output $\hat{y}_k = (\hat{y}_1^k, \hat{y}_2^k, \dots, \hat{y}_l^k)$

$$\hat{y}_j^k = f(\beta_j - \theta_j) \qquad error:\ E_k = \frac{1}{2}\sum_{j=1}^{l}(\hat{y}_j^k - y_j^k)^2$$

$$\Delta w_{hj} = -\eta \frac{\partial E_k}{\partial w_{hj}} \qquad \frac{\partial E_k}{\partial w_{hj}} = \frac{\partial E_k}{\partial \hat{y}_j^k} * \frac{\partial \hat{y}_j^k}{\partial \beta_j} * \frac{\partial \beta_j}{\partial w_{hj}} \qquad \frac{\partial \beta_j}{\partial w_{hj}} = b_h \qquad f'(x) = f(x)(1 - f(x))$$

$$g_j = -\frac{\partial E_k}{\partial \hat{y}_j^k} * \frac{\partial \hat{y}_j^k}{\partial \beta_j} = \hat{y}_j^k(1 - \hat{y}_j^k)(y_j^k - \hat{y}_j^k) \qquad \Delta w_{hj} = \eta g_j b_h$$

Other parameters update:

$$\Delta\theta_j = -\eta g_j ,$$

$$\Delta v_{ih} = \eta e_h x_i ,$$

$$\Delta\gamma_h = -\eta e_h ,$$

$$\begin{aligned} e_h &= -\frac{\partial E_k}{\partial b_h} \cdot \frac{\partial b_h}{\partial \alpha_h} && = \sum_{j=1}^{l} w_{hj} g_j f'(\alpha_h - \gamma_h) \\ &= -\sum_{j=1}^{l} \frac{\partial E_k}{\partial \beta_j} \cdot \frac{\partial \beta_j}{\partial b_h} f'(\alpha_h - \gamma_h) && = b_h(1 - b_h) \sum_{j=1}^{l} w_{hj} g_j . \end{aligned}$$

Aim to minimize $E = \frac{1}{m}\sum_{k=1}^{m} E_k$

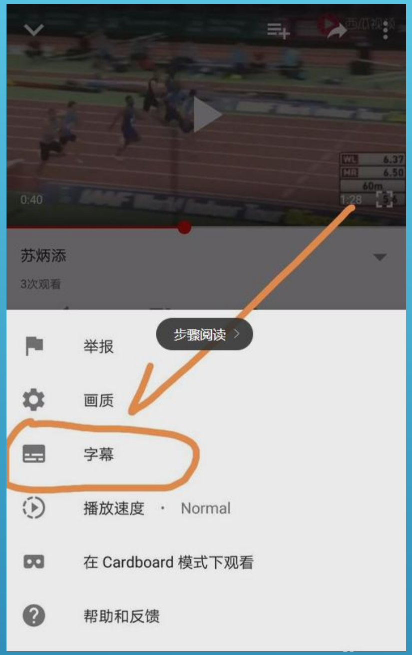
An unsolved question: how accumulated error backpropagation works？

Sometime E traps into local minimum,but it is not the global minimum.
Solution:1.use another original value and start again,after serval trial,
select the minimum E.
2.use simulated annealing technology,every step has a rate to
accept a worsen result.
3.random Stochastic Gradient Descent.
4.genetic algorithms

Neural networks Application

①Image identification

②Voice Recognition



3. Bayes classifier

①Based on Bayesian decision theory → Every sample probability is known

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

λ_{ij} *is the loss of recognize* c_j *instead of* c_i

$R(c_i|x)$ is the sample x total conditional risk.

Aim to reduce the total risk.

If aim to minimize the error rate

$$\lambda_{ij} = \begin{cases} 0, & if\ i = j; \\ 1, & otherwise \end{cases}$$

$$R(c|x) = 1 - P(c|x)$$

$$h^*(x) = arg_{c\in Y} max R(c|x)$$

So maximize P(c|x) can minimize the R(c|x)

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

If h can minimize the conditional risk,
the total risk R(h) can reduce at the same time

$$h^*(x) = arg_{c\in Y} min R(c|x)$$

$h^*(x)$ call Bayes optimal classifier, R(h^*) Bayes risk

Parameter P(c|x) need to be calculated,use Maximum Likelihood Estimation

set $P(c|x)= P(c|\theta_c)$

Sample x are in Class c, D_c include all sample x

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

But we usually use $LL(\theta_c) = \log P(D_c|\theta_c) = \sum_{x \in D_c} \log P(x|\theta_c)$

$$\hat{\theta}_c = arg_{\theta_c} \max LL(\theta_c)$$

If parameter is continuous-value,we typically assume that $P(c|x) \sim \mathrm{N}(\mu_c, \sigma_c^2)$ that

$$\hat{\mu}_c = \frac{1}{|D_c|} \sum_{x \in D_c} x \qquad \hat{\sigma}_c^2 = \frac{1}{|D_c|} \sum_{x \in D_c} (x - \hat{\mu}_c)(x - \hat{\mu}_c)^T$$

Naïve Bayes classifier

Naïve bayes classifier adopt attribute conditional independence assumption. So

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)} = \frac{P(c)}{P(x)}\prod_{i=1}^{d} P(x_i|c) \qquad h_{nb}(x) = arg_{c\epsilon Y}\max P(c)\prod_{i=1}^{d} P(x_i|c)$$

$$P(c) = \frac{|D_c|}{|D|} \qquad P(x_i|c) = \frac{|D_{c,x_i}|}{|D_c|}$$

Continuous attribute

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

In order to avoid missing some attribute,use Laplacian correction

$$\hat{P}(c) = \frac{|D_c|+1}{|D|+N},$$

$$\hat{P}(x_i \mid c) = \frac{|D_{c,x_i}|+1}{|D_c|+N_i}.$$

编号	色泽	根蒂	敲声	纹理	脐部	触感	密度	含糖率	好瓜
测1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	0.697	0.460	?

e.g.

P(good)=$\frac{8}{17} = 0.471$

P(good)=$\frac{7}{17} = 0.529$

P(green | good)=$\frac{3}{8} = 0.375$

P(green | bad)=$\frac{3}{9} = 0.333$

P(curve | good)=$\frac{3}{8} = 0.375$

P(curve | bad)=$\frac{3}{9} = 0.333$

P(density=0.697 | good)=$\frac{1}{\sqrt{2\pi}*0.129}\exp\left(-\frac{(0.697-0.574)^2}{2*0.129^2}\right) = 1.959$

P(density=0.697 | bad)=$\frac{1}{\sqrt{2\pi}*0.129}\exp\left(-\frac{(0.697-0.574)^2}{2*0.129^2}\right) = 1.203$

P(good)=0.471*0.375*1.959*…$= 0.063$

P(bad)=$0.529 * 0.333 * 0.333 * \cdots = 6.80 * 10^{-5}$

P(good)>P(bad),so this sample is good one

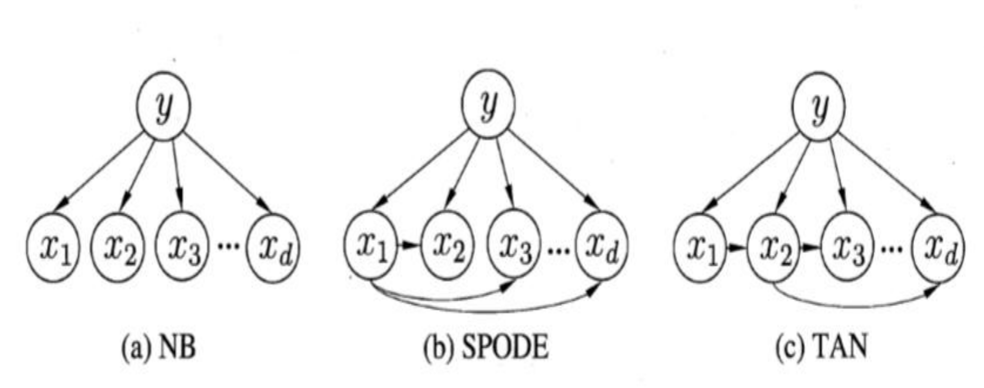
Semi-Naïve Bayes classifier

One-Dependent Estimator

Assume that every attribute depend on at most one non-class attribute.

$$P(c \mid \boldsymbol{x}) \propto P(c) \prod_{i=1}^{d} P(x_i \mid c, pa_i)$$

To solve the problem,need to find out the parent.
If each attribute depend on the same attribute,
called Super-parent ODE like the model(b).
Model (c) Tree Augemented naïve Bayes is based
on maximum weighted spanning tree,at last simplify
Into tree.



$$I(x_i, x_j \mid y) = \sum_{x_i, x_j;\, c \in \mathcal{Y}} P(x_i, x_j \mid c) \log \frac{P(x_i, x_j \mid c)}{P(x_i \mid c) P(x_j \mid c)}$$

Bayesian network

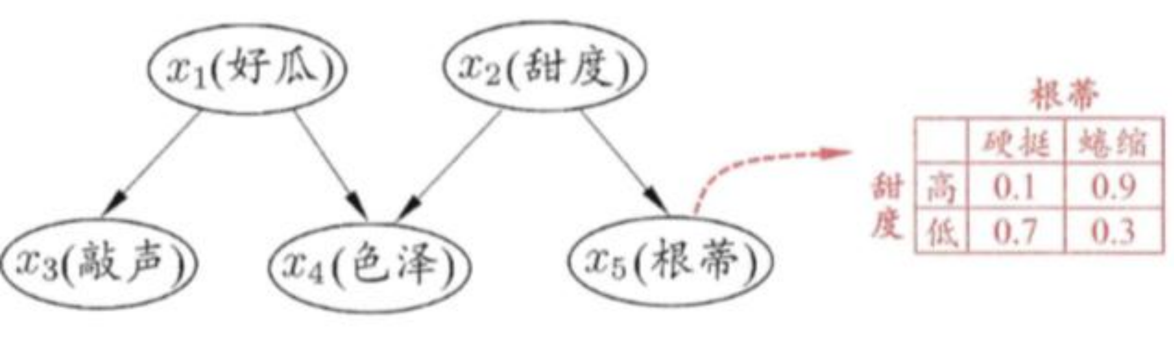
Use Directed Acyclic Graph to describe the relationship between attributes
And Conditional Probability Table display the joint distribution probability.

$B = \langle G, \Theta \rangle$ G is Directed Acyclic Graph.
Θ is parameter.

joint distribution probability: $$P_B(x_1, x_2, \dots, x_d) = \prod_{i=1}^{d} P_B(x_i|\pi_i) = \prod_{i=1}^{d} \theta_{x_i|\pi_i}$$

e.g.

$$P(x_1, x_2, x_3, x_4, x_5) = P(x_1)P(x_2)P(x_3|x_1)P(x_4|x_1, x_2)P(x_5|x_2)$$



Determine the structure

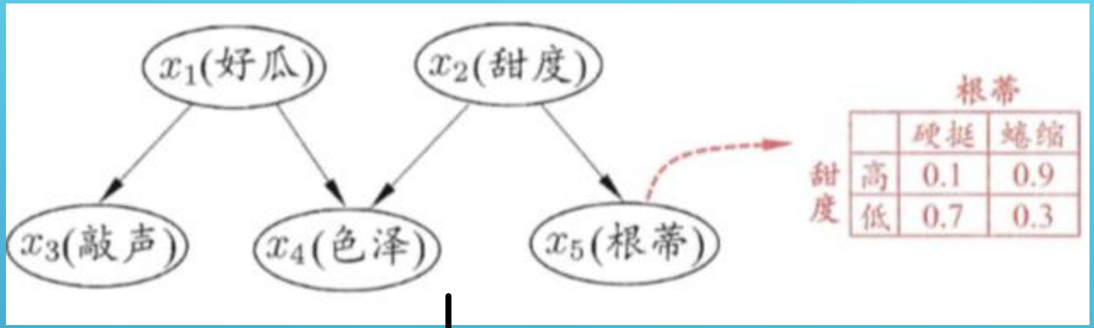
Method：

1.Create a moral graph

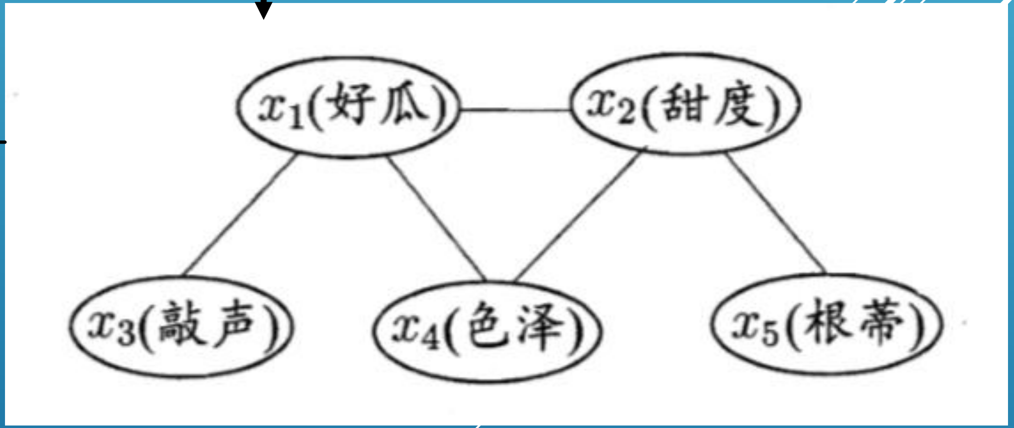
Add edges between all pairs of nodes having a common child.

Remove all directions

2.Find out all conditional independent ralationship



$$x_3 \perp x_4 | x_1, x_4 \perp x_5 | x_2, x_3 \perp x_2 | x_1, x_3 \perp x_5 | x_1, x_3 \perp x_5 | x_2$$



Moral graph

Learning

① Determining the graphical structure
② Determining the conditional probabilities
③ Define a score function to assess beyesian network

Minimal Description Length Criterion

Dataset D=$\{x_1, x_2, \ldots, x_m\}$,Beyesian network B=$\langle G, \Theta \rangle$

Score function: $s(B|D) = f(\theta)|B| - LL(B|D)$

| B | is the number of parameters
$f(\theta)$is the length of each θ

$$LL(B|D) = \sum_{i=1}^{m} \log P_B(x_i)$$

If $f(\theta)$=1 , $\mathrm{AIC(B|D)} = s(B|D) = |B| - LL(B|D)$
If $f(\theta)$=$\frac{1}{2}\log m$, $\mathrm{BIC(B|D)} = s(B|D) = \frac{1}{2}\log m\,|B| - LL(B|D)$
$f(\theta)$ is constant, $\hat{\theta}_{x_i|\pi_i} = \hat{P}_D(x_i|\pi_i)$, to minimize $s(B|D)$ is to search the structure,
But it is hard to solve,use Gibbs sampling to solve.

Gibbs sampling

1: n_q = 0
2: q^0=对 Q 随机赋初值
3: for t = 1,2, . . . , T do
4: for $Q_i \in Q$ do
5: Z = E U Q \ $\{Q_i\}$;
6: $z = e \cup q^{t-1} \backslash \{q_i^{t-1}\}$;
7: 根据 B 计算分布 $P_B(Q_i$ | Z = z);
8: q^t =根据 $P_B(Q_i$ | Z = z)采样所获Q_i取值;
9: q^t =将 q^{t-1}中的 q_i^{t-1} 用 q_i^t 替换
10: end for
11: if q^t = q then
12: n_q = n_q + 1
13: end if
14: end for 输出: P(Q = q | E = e)≅ $\frac{n_q}{T}$

Application

Auto classifier
words filter/corrector
Medical application
Rank system

4.Which is difficult to realize？

That’s all thank you!