Naïve Bayes Classifier & Decision Tree

- Although KNN is used for supervised and K-Means is used for unsupervised, both of them find the exact label for one data point
- Naïve Bayes find the probability that data point will be in each label: p(y = c|**x**) or p(c|**x**) in short, then $c = \arg\max_{c\in\{1,\dots,C\}} p(c\,|\,\mathbf{x})$
- $p(c\,|\,\mathbf{x})$ means that given data point **x**, the probability this point is in class c, it is like inference process in test set, but now in training set, we need to apply Bayes Theorem:

$$c = \arg\max_{c\in\{1,\dots,C\}} p(c\,|\,\mathbf{x}) = \arg\max_{c\in\{1,\dots,C\}} \frac{p(\mathbf{x}\,|\,c)\,p(c)}{p(\mathbf{x})} = \arg\max_{c\in\{1,\dots,C\}} p(\mathbf{x}\,|\,c)\,p(c)$$

- As mentioned in chapter 4, $p(\mathbf{x}\,|\,c)$ hardly finds out. To solve it, we assume independence among the features in one data point

$$p(\mathbf{x}\,|\,c) = p(x_1, x_2, \dots, x_d\,|\,c) = \prod_{i=1}^{d} p(x_i\,|\,c)$$

- Independence among the features assumption is reason why we call "naive" because in the reality, the features in one dataset are hardly completely independent
- Due to "Naive", NBC has fast speed while training
- When number of data points (d) is big, $p(x_i\,|\,c) \to 0$, logarithm can solve the problem

$$c = \arg\max_{c\in\{1,\dots,C\}} \left(\log\left(p(c)\right) + \sum_{i=1}^{d} \log\left(p(x_i\,|\,c)\right) \right)$$

- Finding out $p(x_i\,|\,c)$ is based on the distribution of the dataset: *Gaussian Naïve Bayes*, *Multinomial Naïve Bayes*, *Bernoulli Naïve Bayes*

Estimating Probabilities

- In the following dataset, P(Humidity= 'Low'| class= 'Yes')=P(Humidity= 'Low'| class= 'No') = 0 → P(X| class= 'Yes') = P(X| class= 'No') = 0, so you cannot categorize the X
- Estimating probabilities: $\dfrac{n_c + mp}{n + m}$
- Given the dataset, calculate Naïve Bayes to find the class of **last row**

Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes
5	Cloudy	Warm	High	Weak	Cool	Same	Yes
6	Cloudy	Cold	High	Weak	Cool	Same	No

7	Sunny	Warm	Normal	Strong	Warm	Same	?
8	Sunny	Warm	Low	Strong	Cool	Same	?

  - P(C$_i$): $\begin{cases} P(class = 'yes') = \dfrac{4}{6} = \dfrac{2}{3} \\ P(class = 'no') = \dfrac{2}{6} = \dfrac{1}{3} \end{cases}$
  - P(Sky= 'Sunny' | class = 'Yes') = $\dfrac{3}{3} = 1$
  - P(Sky= 'Sunny' | class = 'No') = 0
  - P(Sky= 'Rainy'| class= 'Yes')=0
  - P(Sky= 'Rainy'| class= 'No')= 1
  - Notice: value 'Low' is not in 'Humidity', so P(Humidity= 'Low'| class= 'Yes') = P(Humidity= 'Low'| class= 'No')=0 → Cannot decide → Apply Estimating Probabilities
  - P(Humidity= 'Low'| class= 'Yes')=

$$\frac{(\#of\ rows\ of\ class\ 'yes'\ \text{having}\ 'low') + (\#of\ rows\ of\ dataset)\left(\dfrac{1}{\#of\ unique\ class\ values}\right)}{\#of\ rows\ of\ class\ 'yes' + \#of\ rows\ of\ dataset} = \frac{0 + 6 * \frac{1}{2}}{4+6} = 0.3$$

  - P(Humidity= 'Low'| class= 'No')= $\dfrac{0 + 6 * \frac{1}{2}}{2+6} = 0.375$

- Put X = (Sunny, Warm, Low, Strong, Cool, Same)

- P(X | Class = ‘Yes’) = P(Sky= ‘Sunny’ | class = ‘Yes’) * P(AirTemp= ‘Warm’ | class = ‘Yes’) * P(Humidity= ‘Low’ | class = ‘Yes’) * P(Wind= ‘Strong’ | class = ‘Yes’) * P(Water= ‘Cool’ | class = ‘Yes’) * P(Forecast= ‘Same’ | class = ‘Yes’)

- P(X | Class = ‘No) = P(Sky= ‘Sunny’ | class = ‘No) * P(AirTemp= ‘Warm’ | class = ‘No) * P(Humidity= ‘Low’ | class = ‘No) * P(Wind= ‘Strong’ | class = ‘No) * P(Water= ‘Cool’ | class = ‘No) * P(Forecast= ‘Same’ | class = ‘No)
- Compare: P(X | Class = ‘Yes’) * P(Class = ‘Yes) and P(X | Class = ‘No’) * P(Class = ‘No’)

Common distribution in NBC

Gaussian Naïve Bayes

- For feature *i* and class *c*, x_i follows the Gaussian distribution of μ_{ci} and σ_{ci}

$$p(x_i \mid c) = p(x_i \mid \mu_{ci}, \sigma_{ci}^2) = \frac{1}{\sqrt{2\pi\sigma_{ci}^2}} \exp\left(-\frac{(x_i - \mu_{ci})^2}{2\sigma_{ci}^2}\right)$$

in which parameter $\theta = \{\mu_{ci}, \sigma_{ci}^2\}$ is determined based on the data points of class *c*

Multinomial Naïve Bayes (Example in CBD/Week06)

- This model is usually used in text classification in which data point is built on BoW ideas (Bag of words)
- Each data point has number of feature of *d* (number of words in dictionary). x_i in the data point is number of word *i* (in the dictionary) appearing in the sentence
- $p(x_i \mid c)$ ratio of frequency of word *i* (or feature *i* in general case) appearing in the sentence of class *c*

$$p(x_i \mid c) = \lambda_{ci} = \frac{N_{ci}}{N_c}$$

in which: N_{ci}: total number of word *i* (feature *i*) appearing in the sentence of class *c*, N_c: total number of all words appearing in the sentence of class *c*

- But if the word does not appear in any sentence of class *c*, $p(\text{word} \mid c) = 0$, so if we apply $p(\mathbf{x} \mid c) = \prod_{i=1}^{d} p(x_i \mid c)$ will be 0.

Laplace Smoothing can solve it

$p(x_i \mid c) = \hat{\lambda}_{ci} = \frac{N_{ci} + \alpha}{N_c + d\alpha}$ in which α > 0 and usually = 1, *d*: number of words in dictionary, $\sum_{i=1}^{d} \hat{\lambda}_{ci} = 1$, so each sentence of class c will have $\hat{\boldsymbol{\lambda}}_c = \{\hat{\lambda}_{c1}, \hat{\lambda}_{c2}, \ldots, \hat{\lambda}_{cd}\}$

Bernoulli Naïve Bayes

- This model is applied when each element has value of 0 and 1.E.g. Rather than BoW, we just consider whether the word appears in the sentence or not

$$p(x_i \mid c) = p(i \mid c) 1\{x_i = c\} + (1 - p(i \mid c))(1 - 1\{x_i \neq c\})$$

in which $p(i \mid c)$: (the meaning is like $p(x_i \mid c)$ in *Multinomial Naïve Bayes*)

Example

North or South

- In training set, we have corpus including 4 documents d1, d2, d3, d4

	Document	Content	Class
Training set	d1	hanoi pho chaolong hanoi	N

	d2	hanoi buncha pho omai	N
	d3	pho banhgio omai	N
	d4	saigon hutiu banhbo pho	S
Test set	d5	hanoi hanoi buncha hutiu	?

- Intuitively, we can predict d5 is class of N
- The problem can be solved by *Multinomial Naïve Bayes* or *Bernoulli Naïve Bayes*. We should test 2 models to choose the best one. Now *Multinomial Naïve Bayes* is implemented
- First we find $p(c)$: $\begin{cases} p(N) = \dfrac{3}{4} \\ p(S) = \dfrac{1}{4} \end{cases}$
- We got dictionary V = {hanoi, pho, chaolong, buncha, omai, banhgio, saigon, hutiu, banhbo} → $d = |V| = 9$
- TRAINING
  - class = N

	hanoi	pho	chaolong	buncha	omai	banhgio	saigon	hutiu	banhbo	
d1: $\mathbf{x}_1$	2	1	1	0	0	0	0	0	0	
d2: $\mathbf{x}_2$	1	1	0	1	1	0	0	0	0	
d3: $\mathbf{x}_3$	0	1	0	0	1	1	0	0	0	
Total	3	3	1	1	2	1	0	0	0	$N_N = 11$
$\rightarrow \hat{\lambda}_N$	4/20	4/20	2/20	2/20	3/20	2/20	1/20	1/20	1/20	$N_N + \lvert V\rvert\alpha = 20$

  - class = S

d4: $\mathbf{x}_4$	0	1	0	0	0	0	1	1	1	$N_S = 4$
$\rightarrow \hat{\lambda}_S$	1/13	1/13	1/13	1/13	1/13	1/13	2/13	2/13	2/13	$N_S + \lvert V\rvert\alpha = 13$

- TEST

d5: $\mathbf{x}_5$	2	0	0	1	0	0	0	1	0	

d5 = [hanoi, hanoi, buncha, hutiu]

$$\left[\begin{array}{l} p(c=N\mid d5) \propto p(c=N)\prod_{i=1}^{d5} p(x_i\mid N) = \dfrac{3}{4}\left(\dfrac{4}{20}\right)^2 \dfrac{2}{20}\dfrac{1}{20} \approx 1.5\times10^{-4} \\ p(c=S\mid d5) \propto p(c=S)\prod_{i=1}^{d5} p(x_i\mid \mathrm{S}) = \dfrac{1}{4}\left(\dfrac{1}{13}\right)^2 \dfrac{1}{13}\dfrac{2}{13} \approx 1.75\times10^{-5} \end{array}\right.$$

$$\rightarrow p(c=N\mid d5) > p(c=S\mid d5) \rightarrow d5 \in class(N)$$

- In above example, we use Laplace Smoothing $\alpha = 1$. 1.5×10^{-4} and 1.75×10^{-5} just helps you to find the class.
  Note: $p(d5) = p(c=N\mid d5) + p(c=S\mid d5)$

$$\begin{cases} p(\mathrm{c}=\mathrm{N}\mid d5) = \dfrac{p(c=N)\prod_{i=1}^{d5} p(x_i\mid N)}{p(d5)} = \dfrac{1.5\times10^{-4}}{1.5\times10^{-4}+1.75\times10^{-5}} = 0.8955 \\ p(\mathrm{c}=S\mid d5) = 1 - p(\mathrm{c}=\mathrm{N}\mid d5) = 0.1045 \end{cases}$$

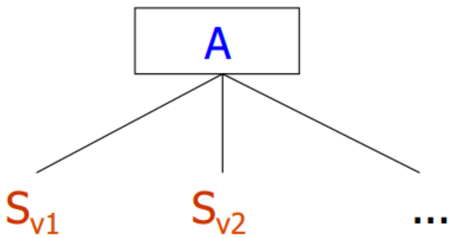
- So, probability that d5 is in class of N: 89.55% and in class of S: 10.45%

Summary

- NBC is usually applied in text classification
- Due to the independence assumption, NBC usually have fast training and testing
- If independence assumption is correct with the dataset we are considering, NBC will have better result than SVM and logistics regression
- Laplace Smoothing is used in MultinomialNB to avoid the absence of word in the dictionary in the class

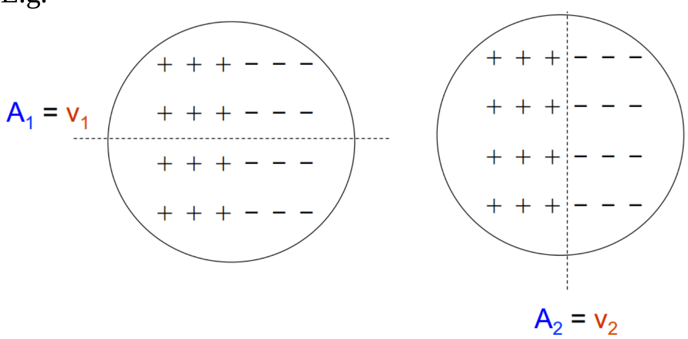
ID3

- Entropy S: đo sự hỗn loạn của S (Sample).
  - E.g. If your dataset has 2 classes, Entropy(S) = $-p_+ \log_2(p_+) - p_- \log_2(p_-) = \sum_{i \in (1,c)} -p_i \log_2(p_i)$
- Gain(S, A) trong đó A là feature: lượng thông tin A chứa để giải thích cho S, so you choose feature which has maximum Gain



Gain(S, A) = Entropy(S) - $\sum_{v \in Values(A)} (S_v / S) \cdot Entropy(S_v)$

- E.g.



  - Should choose $A_2 = v_2$ to minimize the entropy S

- Given the dataset, calculate ID3:

No.	Sky	AirTemp	Humidity	Wind	Water	Forecast	Enjoy
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes
5	Cloudy	Warm	High	Weak	Cool	Same	Yes
6	Cloudy	Cold	High	Weak	Cool	Same	No

  - Entropy(S) =

$$-p_{yes} \log_2(p_{yes}) - p_{no} \log_2(p_{no}) = -\frac{4}{6}\log_2\frac{4}{6} - \frac{2}{6}\log_2\frac{2}{6} = 0.917$$

  - Gain(S, Sky)

$$= Entropy(S) - \sum_{v \in (\text{Sunny,Rainy,Cloudy})} (S_v / S) Entropy(S_v)$$

$$= Entropy(S) - \left[\frac{3}{6} Entropy(S_{Sunny}) + \frac{1}{6} Entropy(S_{Rainy}) + \frac{2}{6} Entropy(S_{Cloudy})\right]$$

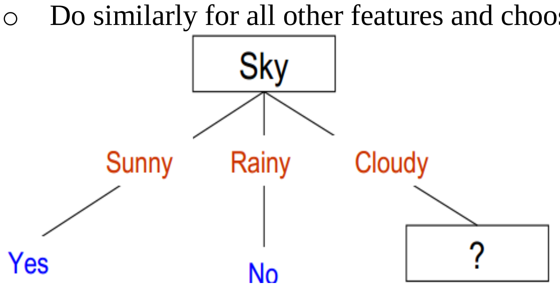
(Intuitively, you see that Entropy(S_{Sunny})=0 because Sunny just include one class: ‘Yes’ and Entropy(S_{Rainy})=0 because Rainy just include one class: ‘No’)

$$= Entropy(S) - \frac{2}{6} Entropy(S_{Cloudy})$$

$$= Entropy(S) - \frac{2}{6}\left(-\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2}\right)$$

$$= 0.584$$

  - Do similarly for all other features and choose maximum Gain, and the largest Gain is feature Sky



  - Delete all rows which does not include Cloudy in Sky
  - $Entropy(S_{Cloudy}) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$
  - Gain(S_{Cloudy}, AirTemp)

$$= Entropy(S_{Cloudy}) - \sum_{v \in (\text{Warm,Cold})} (S_v / S) Entropy(S_v)$$

$$= 1 - \left[\frac{1}{2} Entropy(S_{Warm}) + \frac{1}{2} Entropy(S_{Cold})\right]$$

$$= 1 - (0 + 0)$$

$$= 1$$

C4.5

- One weakness of ID3: if one feature in your dataset is continuous, the ID3 is highly likely to get overfitting when every edge in Decision Tree of this continuous feature will be one number
- $\text{SplitInfo}_A D = -\sum_{v \in A} p_v \log_2 p_v$
- Gain(A) = Gain(S, A)
- $\text{GainRatio(A)} = \dfrac{Gain(A)}{SplitInfo_A D}$
- Choose feature max GainRatio
- Given the dataset, Calculate C4.5

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

- $\text{SplitInfo}_{\text{Income}} S$

$$= -\frac{4}{14}\log_2\frac{4}{14} - \frac{6}{14}\log_2\frac{6}{14} - \frac{4}{14}\log_2\frac{4}{14} = 0.926$$

- Gain(Income)

$$= Entropy(S) - \sum_{v \in (low,medium,high)} (S_v / S) Entropy(S_v) = 0.94 - 0.91 = 0.03$$

- $\text{GainRatio(Income)} = \dfrac{0.03}{0.926} = 0.032$
- Do similarly for all other features and choose feature max GainRatio

Cart

- Binary Split for feature A. S_A is subset of A which has 1 or v-1 unique values of A
- $\text{Gini(D)} = 1 - \sum_{i \in Values(D)} p_i^2$
- $\text{Gini}_A D = \sum_{i \in Values(A)} p_i Gini(D_i)$
- $\Delta\text{Gini(A)} = \text{Gini(D)} - \text{Gini}_A D$
- Choose feature has min $\text{Gini}_A D$ or max ΔGini(A)
- Given the dataset, Calculate C4.5

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

- $\text{Gini(S)} = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2 = 0.46$
-

$$Gini_{i \in (\text{Low,medium}) \in income}(S) = \sum_{i \in (\text{Low,medium})} p_i Gini(D_i)$$

$$= \frac{10}{14}\left(1 - \frac{6}{10}^2 - \frac{4}{10}^2\right) + \frac{4}{14}\left(1 - \frac{1}{4}^2 - \frac{3}{4}^2\right)$$

$$= 0.45 = Gini_{i \in (\text{High})}$$

- $\text{Gini}_{\text{income} \in (\text{Low, High})} = \text{Gini}_{\text{income} \in (\text{Medium})} = 0.315$
    - $\text{Gini}_{\text{income} \in (\text{Medium, High})} = \text{Gini}_{\text{income} \in (\text{High})} = 0.3$
    - You choose $\text{Gini}_{\text{income} \in (\text{Medium, High})} = 0.3$
      $\text{Gini}_{\text{age} \in \{\text{youth,senior}\}/\{\text{middle\_aged}\}} = 0.375$
      $\text{Gini}_{\text{student}} = 0.367$
    - $\text{Gini}_{\text{credit\_rating}} = 0.429$
    - Choose Income to split, $\text{Gini}_A D$ is min

	Splitting Criteria	Attribute type	Missing values	Pruning Strategy	Outlier Detection
ID3	Information Gain	Handles only Categorical value	Do not handle missing values.	No pruning is done	Susceptible to outliers
CART	Towing Criteria	Handles both Categorical & Numeric value	Handle missing values.	Cost-Complexity pruning is used	Can handle Outliers
C4.5	Gain Ratio	Handles both Categorical & Numeric value	Handle missing values.	Error Based pruning is used	Susceptible to outliers