

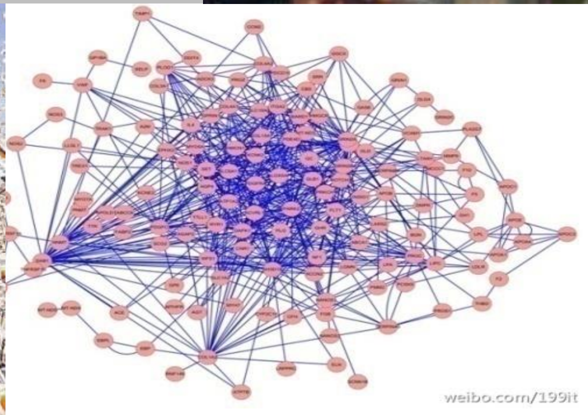

Chapter 6 Naïve Bayes

2025 Autumn

Lei Sun

01 Assumption of NB

02 Zero Probability

03 Numeric Attributes

04 Pros and Cons

Problem with likelihood

Suppose our training corpus(语料库) contains emails:

Email1: $\boldsymbol{Y}$ = spam, $\boldsymbol{X}$ ="Hi there man - feel the vitality! Nice meeting you …"

Email2: $\boldsymbol{Y}$ = ham,$\boldsymbol{X}$ =" This needs to be in production by early afternoon …"

……

Our test corpus is just one email:

Email: $\boldsymbol{X}$=" Hi! You can receive within days an approved prescription for increased vitality and stamina"

How can we estimate P($\boldsymbol{Y}$ = spam|$\boldsymbol{X}$ = "Hi! You can receive within days an approved prescription for increased vitality and stamina")?

Problem with likelihood

We can estimate the likelihood of an e-mail by pretending that the e-mail is just a bag of words (order doesn't matter).
With only a few thousand spam e-mails, we can get a pretty good estimate of these things:

$P(W = \text{"hi"} | Y = \text{spam})$, $P(W = \text{"hi"} | Y = \text{ham})$
$P(W = \text{"vitality"} | Y = \text{spam})$, $P(W = \text{"vitality"} | Y = \text{ham})$
$P(W = \text{"production"} | Y = \text{spam})$, $P(W = \text{"production"} | Y = \text{ham})$

Then we can approximate P(X| Y) by assuming that the words, W, are ***conditionally independent*** of one another given the category label:

$$P(X = x | Y = y) \approx \prod_{i=1}^{n} P(W = w_i | Y = y)$$

Naïve Bayes

$$P(B_i|A) = \frac{P(B_i)P(A|B_i)}{\sum_{j=1}^{k} P(B_j)P(A|B_j)}$$

- ✓ **Use for classification**
  - Suppose want to use X to classify. Y is class label
  - Use Bayes rule to estimate

$$P(Y = y|X = x) = \frac{P(X = x|Y = y)\, P(Y = y)}{P(X = x)}$$

- Given a record with X=x, classify target Y as the value Y that maximizes

$$P(Y=y \mid X=x)P(Y=y)$$

- Naive Bayes Classifiers (NBC) are simple yet powerful Machine Learning algorithms. They are based on and Bayes's Theorem.

Naïve Bayes

A and B is independent → $P(AB) = P(A)P(B)$

A and B is dependent → $P(AB)=P(B|A)*P(A)$

We have k classes $C_1, C_2, ..., C_k$, and a vector of n features $X = <x_1, x_2, ..., x_n>$, we want to find the class C_i that maximizes numerator

Notice : the denominator is constant and it does not depend on the class y_i.
So, we can ignore it and just focus on the numerator.

Assuming all the features X_i are independent and using Bayes's Theorem, we can calculate the conditional probability as follows:

$$P(x_1, x_2, ..., x_n)| C_i) = P(x_1 | C_i)P(x_2 | C_i)...P(x_n | C_i)$$

Naïve Bayes

Assumption of Naive Bayes

- **Feature independence:** The features of the data are conditionally independent of each other, given the class label.
- **Features are equally important:** All features are assumed to contribute equally to the prediction of the class label.

$$P(C_k|x) = \frac{P(x|C_k)P(C_k)}{P(x)} = \frac{\prod_{i=1}^{n} p(x_i|C_k)P(C_k)}{\sum_{i=1}^{n} p(x_i|C_k)P(C_k)}$$

the joint probability distribution: $\prod_{i=1}^{n} p(x_i|C_k) = P(x_1|C_k)\,P(x_2|C_k)\dots P(x_n|C_k)$

01

Naïve Bayes

Example

Suppose

outlook = sunny,
temp. = cool,
humidity = high,
wind = strong,

what's the play?

Day	Outlook	Temperature	Humidity	Wind	Play?
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Cloudy	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Cloudy	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	Cloudy	Mild	High	Strong	Yes
D13	Cloudy	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Naïve Bayes

$P(p = yes) = 9/14 = 0.64$

$P(p = no) = 5/14 = 0.36$

$P(wind = strong \mid p = yes) = 3/9 = 0.33$

$P(wind = strong \mid p = no) = 3/5 = 0.60$

$P(yes)P(sunny \mid yes)P(cool \mid yes)\ P(high|yes)P(strong|yes) = 0.0053$

$P(no)P(sunny|no)P(cool|no)P(high|no)P(strong|no) = 0.0206$

Therefore, the play tennis value would be “no”

02

Numeric attributes

- ✓ Assume Normal or Gaussian distribution

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- ✓ For each numeric attribute, calculate Mean and Std Dev for each class

$$\sigma^2 = \frac{1}{N-1} \Sigma(X_i - \bar{X})$$

where mu and sigma are the mean and variance of the continuous X

Numeric attributes

outlook			temperature			humidity			windy			play	
	yes	no		yes	no		yes	no		yes	no	yes	no
sunny	2	3		83	85		86	85	false	6	2	9	5
overcast	4	0		70	80		96	90	true	3	3		
rainy	3	2		68	65		80	70					
				64	72		65	95					
				69	71		70	91					
				75			80						
				75			70						
				72			90						
				81			75						
sunny	2/9	3/5	mean	73	74.6	mean	79.1	86.2	false	6/9	2/5	9/14	5/14
overcast	4/9	0/5	std dev	6.2	7.9	std dev	10.2	9.7	true	3/9	3/5		
rainy	3/9	2/5											

numeric

02

Numeric attributes

- Considering "yes" outcome for an example with Temperature=66:

$$f(\text{temp.} = 66|\text{yes}) = \frac{1}{\sqrt{2\pi} \times 6.2} e^{-\frac{(66-73)^2}{2\times 6.2^2}} = 0.034$$

- Similarly, for a "yes" outcome with Humidity = 90:

$$f(\text{humidity} = 90|\text{yes}) = 0.0221$$

- Other calculations as usual.

$$P(X|\text{yes}) \times P(\text{yes}) = \frac{2}{9} \times 0.034 \times 0.0221 \times \frac{3}{9} \times \frac{9}{14}$$
$$= 0.000036$$

Zero probability

✓ Suppose that the training data for the tennis example was different:

- outlook=sunny had been always associated with play=no
  (i.e. outlook=sunny had never occurred together with play=yes)
- then P(yes|outlook=sunny)=0 and P(no|outlook=sunny)=1

$$P(yes|x) = \frac{\underbrace{\boxed{P(x_1|yes)}}_{=0}P(x_2|yes)P(x_3|yes)P(x_4|yes)P(yes)}{P(x)}$$

- final probability P(yes|)=0 no matter of the other probabilities,
  i.e. zero probability hold a veto over the other probabilities

✓ Solution: Laplace estimator (correction)

- Add 1 to the numerator and K to the denominator, when K is the number of attribute values for a given attribute

03

Zero probability

Laplace smoothing 拉普拉斯平滑

The dataset is large enough that adding one row of each class will not make a difference in the estimated probability. This will overcome the issue of probability values to zero.

$$p(c) = \frac{|D_c| + 1}{|D| + N}$$

N is the number of class

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

N_i is the number of distinct values of attribute x_i

This will lead to the removal of all the zero values from the classes and, at the same time, will not impact the overall relative frequency of the classes.

03

Zero probability

Laplace smoothing

$$p(play = yes) = \frac{9}{14}$$

$$p(play = no) = \frac{5}{14}$$

P(sunny|yes)=0/7

P(overcast|yes)=4/7

P(rainy|yes)=3/7

	outlook		
	yes	no	...
sunny	0	5	...
overcast	4	0	...
rainy	3	2	...
			...
sunny	0/7	5/7	...
overcast	4/7	0/7	...
rainy	3/7	2/7	...

$$p(play = yes) = \frac{9+1}{14+2}$$

$$p(play = no) = \frac{5+1}{14+2}$$

$$p(sunny|yes) = \frac{0+1}{7+3}$$

$$p(overcast|yes) = \frac{4+1}{7+3}$$

Summary

Assumption of Naive Bayes

- **Feature independence:** The features of the data are conditionally independent of each other, given the class label.
- **Features are equally important:** All features are assumed to contribute equally to the prediction of the class label.
- **Continuous features are normally distributed:** If a feature is continuous, then it is assumed to be normally distributed within each class.
- **Discrete features have multinomial distributions(多项式分布):** If a feature is discrete, then it is assumed to have a multinomial distribution within each class.

We use Naïve Bayes a lot because, even though we know it is wrong, it gives us computationally efficient algorithms that work remarkably well in practice.

04

Summary

Pros	Cons
Easy to implement and computationally efficient	Assumes that features are independent, which may not always hold in real-world data
It performs well in the presence of categorical features.	It may struggle with estimating probabilities when there are no occurrences within the training data (zero-frequency problem).
Effective in cases with a large number of features.	For numerical features data is assumed to come from normal distributions