The Math Behind

Monday 15 April 2024 7:19 PM

→ Parametric ML Algo

e.g.

x_1	x_2	x_3	$x_4 \cdots x_n$	y

Assume we have a multi-class classification problem with k classes.

query : $\langle x_1\ x_2\ x_3 \ldots x_n \rangle \rightarrow x_q$

Naive bayes calculates k different probabilities

$P(y_1|x_q) = P(x_q|y_1)P(y_1) / P(x_q)$

$P(y_2|x_q)$

⋮

$P(y_k|x_q) = P(x_q|y_k)P(y_k) / P(x_q)$

→ can ignore this term ($P(x_q)$)

$P(y_k|x_q) = P(x_q|y_k)P(y_k)$

$= P(x_1 \cap x_2 \cap \ldots \cap x_n | y_k) P(y_k)$

$P(x_1, x_2, x_3, \ldots x_n | y_k) P(y_k)$

$\dfrac{P(x_1, x_2, x_3, \ldots, x_n, y_k)}{P(y_k)} P(y_k)$

$P(A|B) = \dfrac{P(A \cap B)}{P(B)}$

$P(\underbrace{x_1}_{A}, \underbrace{x_2, x_3, \ldots, x_n, y_k}_{B})$

$P(A \cap B) = P(A|B)P(B)$

↓

$P(x_1|x_2, x_3, \ldots, x_n, y_k)\, P(\underbrace{x_2}_{A}, \underbrace{x_3, \ldots, x_n, y_k}_{B})$

↓

$P(x_2|x_3, x_4, \ldots, x_n, y_k)\, P(\underbrace{x_3}_{A}, \underbrace{x_4, \ldots, x_n, y_k}_{B})$

↓

$P(x_3|x_4, x_5, \ldots, x_n, y_k)\, P(x_4, x_5, \ldots, x_n, y_k)$

$P(x_1|x_2, x_3, \ldots, x_n, y_k) P(x_2|x_3, x_4, \ldots, x_n, y_k) P(x_3|x_4, x_5, \ldots, x_n, y_k) \ldots P(x_{n-1}|x_n, y_k) P(x_n|y_k) P(y_k)$

"Naive bayes Assumes that the features are independent of each other"

if A & B are independent $P(A|B) = P(A)$

likewise $P(A|B \cap C) = P(A|B, C) = P(A|C)$, where A is dependent on C but not B.

By this Naive Assumption,

$P(x_1|x_2, x_3, \ldots, x_n, y_k) P(x_2|x_3, x_4, \ldots, x_n, y_k) P(x_3|x_4, x_5, \ldots, x_n, y_k) \ldots P(x_{n-1}|x_n, y_k) P(x_n|y_k) P(y_k)$

↓

foreach class → $\underbrace{P(y_k)}_{\text{Prior}}\, P(x_1|y_k) P(x_2|y_k) P(x_3|y_k) \ldots P(x_n|y_k)$

Compare the probabilities for each class and choose one with the maximum probability.

Training phase → Calc probabilities

Testing phase → Use the calculated conditional probabilities to calculate the product & pick class with max probability.

`from sklearn.naive_bayes import ...`

How naive bayes works with numerical data?

age	married
36	Y
40	N
29	Y
⋮	⋮

query = 55

$P(Y|55) = P(55|Y) P(Y)$

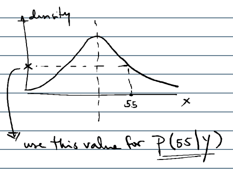
$P(N|55) = P(55|N) P(N)$

→ What if we cant find a point where someone who is married is 55, then this becomes 0.

with numerical data, we assume it follows gaussian distribution.

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

acts like probability

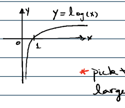


use this value for $P(55|Y)$

* Separate the classes in the target column & calc μ & σ for each class, treat them as different gaussian distributions.

What if data is not gaussian?
- Data transformation
- Alternative Distribution
- Discretization
- Kernel density estimation (when distribution is unknown)
- use other algorithms

log probabilities : $\log_e(ab) = \log_e(a) + \log_e(b)$
- underflow
- Numerical stability



* pick the class with larger log probability.

laplace additive smoothing [on conditional probabilities]
- helps avoid zero probabilities by adding a small constant (α) to the numerator and ($n\alpha$) to the denominator of each probability estimate, default $\alpha = 1$
- helps control bias variance tradeoff

$\downarrow \alpha \uparrow$ variance $\uparrow \alpha \uparrow$ Bias

* Naive bayes performs well on homogeneous features in terms of datatypes

Sklearn allows us to use several variations of naive bayes
- Gaussian [all numerical columns] → Additive smoothing not applied on GNB. we use a continuous probability distribution to calculate probabilities.
- Categorical [all categorical features] → every feature should follow Multinoulli distribution
  → laplace additive smoothing applied on conditional probabilities → $P(x_i = t | y = c; \alpha) = \dfrac{N_{tic} + \alpha}{N_c + \alpha n_i}$

  where N_{tic} is the # of times t appears in feature x_i which belong to class c. α is the smoothing parameter. n_i is the # of categories available in feature x_i.
- multinomial [all features are discrete] [text classification BOW/tfid]
- Bernoulli → each feature follows Bernoulli distribution. (Binary BOW)

  https://nlp.stanford.edu/IR-book/html/htmledition/the-bernoulli-model-1.html
- out of core [Research] — partial_fit() → update model, predict, chunks
- complement ** [Research paper] [for imbalanced data] [variation of multinomial]

Table 13.1: Data for parameter estimation examples.

	docID	words in document	in c = China?	

out of core [Research]

complement ** [Research paper] [for imbalanced data] [variation of multinomial]

each feature follows Bernoulli distribution.
(Binary BOW)

https://nlp.stanford.edu/IR-book/html/htmledition/the-bernoulli-model-1.html

Table 13.1: Data for parameter estimation examples.

	docID	words in document	in c = China?
training set	1	Chinese Beijing Chinese	yes
	2	Chinese Chinese Shanghai	yes
	3	Chinese Macao	yes
	4	Tokyo Japan Chinese	no
test set	5	Chinese Chinese Chinese Tokyo Japan	?

each column represents a multinoulli distribution →

each row is following a multinomial distribution →

	Chinese	Beijin	Shanghai	Macao	Tokyo	Japan	
1.	2	1	0	0	0	0	yes
2.	2	0	1	0	0	0	yes
3.	1	0	0	1	0	0	yes
4.	1	0	0	0	1	1	no
5.	3	0	0	0	1	1	?

P(Yes | Chinese=3, Beijin=0, Shanghai=0, Macao=0, Tokyo=1, Japan=1)

→ P(yes) P(Chinese|y) P(Chinese|y) P(Chinese|y) P(Tokyo|y) P(Japan|y)

$3/4 \times \frac{5}{8} \times \frac{5}{8} \times \frac{5}{8} \times \frac{0}{8} \times \frac{0}{8}$ → perform laplace smoothing

= P(Yes) P(Chinese|Yes) P(Beijin|Yes) P(Shanghai|Yes) P(Macao|Yes) P(Tokyo|Yes) P(Japan|Yes)

$= 3/4 \times 5/8 \times 1/8 \times 1/8 \times 1/8 \times 0/8 \times 0/8$

→ laplace additive smoothing:

$$\hat{\theta}_{yi} = \frac{N_{yi} + \alpha}{N_y + \alpha n}$$

N_{yi} is the number of times feature i appears in class y in the training set.
N_y is the the total count of all features in class y.
$\alpha \geq 0$ accounts for features not present in the learning samples & prevents 0 probabilities.
$\alpha = 1$ is called laplace smoothing, $\alpha < 1$ is called lidstone smoothing.
n is the total # of vocabulary.

laplace smoothing = $\frac{3}{4} \times \frac{5+1}{(8+6)} \times \frac{1+1}{(8+6)} \times \frac{1+1}{(8+6)} \times \frac{1+1}{(8+6)} \times \frac{0+1}{(8+6)} \times \frac{0+1}{(8+6)}$

$= 3/4 \times 6/14 \times 2/14 \times 2/14 \times 2/14 \times 1/14 \times 1/14 \longrightarrow 3/4 \times \left(\frac{3}{7}\right)^3 \times (2/14)^0 \times (2/14)^0 \times (2/14)^0 \times \left(\frac{1}{14}\right)^1 \times (1/14)^1$

log proba = −3.521

P(No | Chinese=3, Beijin=0, Shanghai=0, Macao=0, Tokyo=1, Japan=1)
= P(NO) P(Chinese|NO) P(Beijin|NO) P(Shanghai|NO) P(Macao|NO) P(Tokyo|NO) P(Japan|NO)

laplace smoothing = $1/4 \times \frac{1+1}{(3+6)} \times \frac{0+1}{(3+6)} \times \frac{0+1}{(3+6)} \times \frac{0+1}{(3+6)} \times \frac{1+1}{(3+6)} \times \frac{1+1}{(3+6)}$

$= 1/4 \times 2/9 \times 1/9 \times 1/9 \times 1/9 \times 2/9 \times 2/9 \longrightarrow 1/4 \times (2/9)^3 \times (1/9)^0 \times (1/9)^0 \times (1/9)^0 \times (2/9)^1 \times (2/9)^1$

log proba = −3.8681

P(yes|d5) > P(NO|d5)

∴ yes will be assigned to d5.