Big Data Analysis: Classification – Naïve Bayesian Classfier

Younghoon Kim

Classification

- Given
  - A set of d-dimensional vectors: $x^{(i)} = \langle x_1^{(i)}, x_2^{(i)}, \ldots, x_d^{(i)} \rangle$
  - O_j: domain of x_j
    - Categorical: e.g., O_j = {red, blue, orange, green}
    - Numerical: $O_j = \{1, 2, \ldots, 10\}$ or R
  - y: a label in a categorical domain O_Y
    - E.g., {yes, no} or { +, - }
  - Data D: $\{x^{(i)}, y^{(i)}\}_{i=1,\ldots,n}$
- Goal
  - Given a d-dimensional vector x, whose y is unknown,
  - Predict y

"Naïve Bayes" method

- Opposite strategy: use all the attributes
  - OneR: One attribute does all the work
- Two assumptions: attributes are
  - ~~equally important a priori~~
  - statistically independent (given the class value)
    - i.e., knowing the value of one attribute says nothing about the value of another (if the class is known)

* Ian H. Witten's slide

OneR

Based on these four attributes

We want to predict this attribute!

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	

x_1 x_2 x_3 x_4 y

Review of Probability Theory

- Random variables
  - $V_1, V_2, \ldots, V_k$
- Joint probability
  - $P(V_1 = v_1, V_2 = v_2, \ldots, V_k = v_k)$
- Conditional probability

$$P(V_i \mid V_j) = \frac{P(V_i, V_j)}{P(V_j)}$$

Review of Probability Theory

- Chain rule

$$P(V_1, V_2, \ldots, V_k) = \prod_{i=1}^{k} P(V_i \mid V_{i-1}, \ldots, V_1)$$

  - e.g., P (A=a, B=b, C=c)
    - P(abc) = P(a)P(b|a)P(c|ab)

Review of Probability Theory

- Independence

$$P(V_1, V_2, \ldots V_k) = \prod_{i=1}^{k} P(V_i \mid V_{i-1}, \ldots, V_1) = \prod_{i=1}^{k} P(V_i)$$

  - e.g., P (A=a, B=b)
    - P(a,b)=P(ab)=P(a)P(b)

- Conditional independence

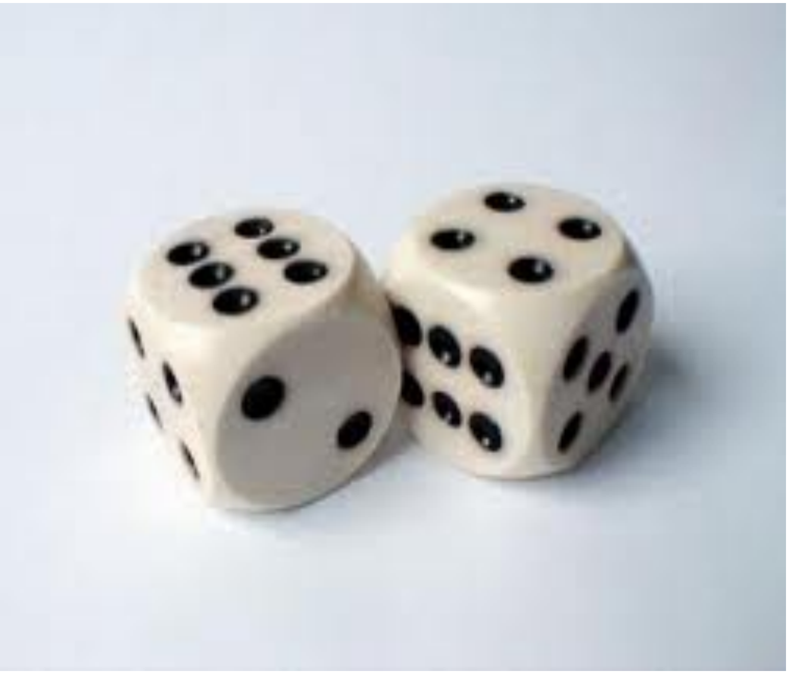
$$P(V_1, V_2, \ldots, V_k \mid V) = \prod_{i=1}^{k} P(V_i \mid V_{i-1}, \ldots V_1, V) = \prod_{i=1}^{k} P(V_i \mid V)$$

  - e.g., P(A=a,B=b|C=c)
    - P(ab|c)=P(a|c)P(b|c)

Probability Basics

- Quiz: We have two six-sided dice. When they are tolled, it could end up with the following occurrence: (*A*) dice 1 lands on side “3”, (*B*) dice 2 lands on side “1”, and (*C*) Two dice sum to eight. Answer the following questions:

1) $P(A) = ?$
2) $P(B) = ?$
3) $P(C) = ?$
4) $P(A \mid B) = ?$
5) $P(C \mid A) = ?$
6) $P(A, B) = ?$
7) $P(A, C) = ?$
8) Is $P(A, C)$ equal to $P(A) * P(C)$?



[Slide of Ke Chen, Univ. of Manchester]

Probability of event H given evidence E

- Thomas Bayes, British mathematician, 1702–1761

$$\Pr[\underset{\text{class}}{H} \mid \underset{\text{instance}}{E}] = \frac{\Pr[E \mid H]\Pr[H]}{\Pr[E]}$$

- Pr[H] is a priori probability of H
  - Probability of event before evidence is seen
- Pr[H | E] is a posteriori probability of H
  - Probability of event after evidence is seen
- "Naïve" assumption:
  - Evidence splits into parts that are independent

$$\Pr[H \mid E] = \frac{\Pr[E_1 \mid H]\Pr[E_2 \mid H] \ldots \Pr[E_n \mid H]\Pr[H]}{\Pr[E]}$$

Ian H. Witten's slide

A word about the Bayesian frame work

- Allows us to combine observed data and prior knowledge
- Provides practical learning algorithms
- It is a generative (model based) approach, which offers a useful conceptual framework
  - This means that any kind of objects (e.g. time series, trees, etc.) can be classified, based on a probabilistic model specification

[Ata Kaban's slide]

Bayes' Rule

$$p(h \mid d) = \frac{P(d \mid h)P(h)}{P(d)}$$

Understanding Bayes' rule

d = data

h = hypothesis

Proof. Just rearrange:

$p(h \mid d)P(d) = P(d \mid h)P(h)$

$P(d,h) = P(d,h)$

the same joint probability

on both sides

Who is who in Bayes' rule

$P(h)$: prior belief (probability of hypothesis *h* before seeing any data)

$P(d \mid h)$: likelihood (probability of the data if the hypothesis *h* is true)

$P(d) = \sum_h P(d \mid h)P(h)$: data evidence (marginal probability of the data)

$P(h \mid d)$: posterior (probability of hypothesis *h* after having seen the data *d*)

Marginal Probability

- For any events X and Y, $P(X,Y)=P(X|Y)P(Y)$
- If we know $P(X,Y)$, then the so-called ***marginal probability*** $P(X)$ can be computed as
  - $P(X) = \Sigma_Y P(X, Y) = \Sigma_Y P(X|Y)P(Y)$
- Probabilities sum to 1. Conditional probabilities sum to 1 **provided that their conditions are the same**.
  - $\Sigma_X P(X|Y) = 1$

Example:
Does patient have cancer or not?

- A patient takes a lab test and **the result comes back positive**. It is known that the test returns **a correct positive result in only 98% of the cases** and **a correct negative result in only 97% of the cases**. Furthermore, **only 0.008 of the entire population has this disease.**

1. What is the probability that this patient has cancer?
2. What is the probability that he does not have cancer?
3. What is the diagnosis?

Example:
Does patient have cancer or not?

- A patient takes a lab test and **the result comes back positive**. It is known that the test returns **a correct positive result in only 98% of the cases** and **a correct negative result in only 97% of the cases**. Furthermore, **only 0.008 of the entire population has this disease.**

1. What is the probability that this patient has cancer?
2. What is the probability that he does not have cancer?
3. What is the diagnosis?

"병이 있는 경우,
이 중 98%는 양성으로 나오며
나머지 2%는
음성으로 판별된다"

Example:
Does patient have cancer or not?

- A patient takes a lab test and **the result comes back positive**. It is known that the test returns **a correct positive result in only 98% of the cases** and **a correct negative result in only 97% of the cases**. Furthermore, **only 0.008 of the entire population has this disease.**

1. What is the probability that this patient has cancer?
2. What is the probability that he does not have cancer?
3. What is the diagnosis?

"병이 없는 경우,
이중 97%는 음성으로 나오며
나머지 3%는 양성이다"

Example

$$\left.\begin{array}{l} hypothesis1 : 'cancer' \\ hypothesis2 : '\neg cancer' \end{array}\right\} \quad hypothesis\ space\ H$$

$$-\,data : '+'$$

$$1. P(cancer \mid +) = \frac{P(+ \mid cancer)P(cancer)}{P(+)} = \frac{\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots}{\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots} = \ldots\ldots\ldots$$

$$P(+ \mid cancer) = 0.98$$

$$P(cancer) = 0.008$$

$$P(+) = P(+ \mid cancer)P(cancer) + P(+ \mid \neg cancer)P(\neg cancer)$$

$$= \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$P(+ \mid \neg cancer) = 0.03$$

$$P(\neg cancer) = \ldots\ldots\ldots$$

$$2. P(\neg cancer \mid +) = \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$3. Diagnosis ??$$

Example

$$\left.\begin{array}{r} hypothesis1:'cancer' \\ hypothesis2:'\neg cancer' \end{array}\right\} \quad hypothesis\ space\ H$$

$$-data:'+'$$

$$1.P(cancer\,|\,+) = \frac{P(+\,|\,cancer)P(cancer)}{P(+)} = \frac{\text{0.00784}}{\text{0.0376}} = \text{0.2085}$$

$$P(+\,|\,cancer) = 0.98$$

$$P(cancer) = 0.008$$

$$P(+) = P(+\,|\,cancer)P(cancer) + P(+\,|\,\neg cancer)P(\neg cancer)$$

$$= \text{0.98*0.008 + 0.03 * 0.992 = 0.0376}$$

$$P(+\,|\,\neg cancer) = 0.03$$

$$P(\neg cancer) = \text{0.992}$$

$$2.P(\neg cancer\,|\,+) = \text{0.7915}$$

$$3.Diagnosis??$$

Choosing Hypotheses

- *Maximum Likelihood* hypothesis:

$$h_{ML} = \arg\max_{h \in H} P(d \mid h)$$

- Generally we want the most probable hypothesis given training data. This is the *maximum a posteriori* hypothesis:
  - Useful observation: it does not depend on the denominator P(d)

$$h_{MAP} = \arg\max_{h \in H} P(h \mid d)$$

Now we compute the diagnosis

- To find the Maximum Likelihood hypothesis, we evaluate P(d|h) for the data d, which is the positive lab test and chose the hypothesis (diagnosis) that maximises it:

$$P(+|cancer) = \ldots\ldots\ldots\ldots$$
$$P(+|\neg cancer) = \ldots\ldots\ldots\ldots$$
$$\Rightarrow Diagnosis : h_{ML} = \ldots\ldots\ldots\ldots\ldots\ldots$$

암 인데 +가 나올 확률과 암이 아닌데 +가 나올 확률을 비교

- To find the Maximum A Posteriori hypothesis, we evaluate P(d|h)P(h) for the data d, which is the positive lab test and chose the hypothesis (diagnosis) that maximises it. This is the same as choosing the hypotheses gives the higher posterior probability.

$$P(+|cancer)P(cancer) = \ldots\ldots\ldots\ldots\ldots\ldots$$
$$P(+|\neg cancer)P(\neg cancer) = \ldots\ldots\ldots\ldots\ldots$$
$$\Rightarrow Diagnosis : h_{MAP} = \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

+가 나왔는데 암일 확률과 +가 나왔는데 암이 아닐 확률을 비교

Towards Naïve Bayesian Classifier

- Let D be a training set of tuples and their associated class labels, and each tuple is represented by an n-ary attribute vector $X = \langle x_1, x_2, \ldots, x_n \rangle$
- Suppose there are m classes $C_1, C_2, \ldots, C_m$.
- Classification is to derive the maximum posteriori, i.e., the maximal $P(C_i|X)$
- This can be derived from Bayes' theorem

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)}$$

- Since P(X) is constant for all classes, only

$$P(C_i|X) \approx P(X|C_i)P(C_i)$$

  - needs to be maximized

Derivation of Naïve Bayes Classifier

- A simplified assumption: attributes are conditionally independent (i.e., no dependence relation between attributes):

$$P(X|C_i)P(C_i)$$
$$= P(C_i)\prod_{k=1}^{n} P(x_k|C_i) = P(x_1|C_i) \times P(x_2|C_i) \times \cdots \times P(x_n|C_i)$$

- This greatly reduces the computation cost: Only counts the class distribution

Derivation of Naïve Bayes Classifier

- If A_k is categorical, $P(x_k|C_i)$ is the # of tuples in C_i having value x_k for A_k divided by $|C_{i,D}|$ (= # of tuples of C_i in D)
- If A_k is continous-valued, $P(x_k|C_i)$ is usually computed based on Gaussian distribution with a mean μ and standard deviation σ

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

where $P(x_k|C_i) = g\left(x_k, \mu_{C_i}, \sigma_{C_i}\right)$

Training Dataset

Class:

C1:buys_computer = 'yes'

C2:buys_computer = 'no'

Data sample

X = (age <=30,

Income = medium,

Student = yes

Credit_rating = Fair)

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Calculation…

- $P(C_i)$: P(buys_computer = "yes") = 9/14 = 0.643
  P(buys_computer = "no") = 5/14= 0.357

- Compute $P(X|C_i)$ for each class
  P(age = "<=30" | buys_computer = "yes") = 2/9 = 0.222
  P(age = "<= 30" | buys_computer = "no") = 3/5 = 0.6
  P(income = "medium" | buys_computer = "yes") = 4/9 = 0.444
  P(income = "medium" | buys_computer = "no") = 2/5 = 0.4
  P(student = "yes" | buys_computer = "yes) = 6/9 = 0.667
  P(student = "yes" | buys_computer = "no") = 1/5 = 0.2
  P(credit_rating = "fair" | buys_computer = "yes") = 6/9 = 0.667
  P(credit_rating = "fair" | buys_computer = "no") = 2/5 = 0.4

- **X = (age <= 30 , income = medium, student = yes, credit_rating = fair)**

$P(X|C_i)$: P(X|buys_computer = "yes") = 0.222 x 0.444 x 0.667 x 0.667 = 0.044
P(X|buys_computer = "no") = 0.6 x 0.4 x 0.2 x 0.4 = 0.019
$P(X|C_i)*P(C_i)$: P(X|buys_computer = "yes") * P(buys_computer = "yes") = 0.028
P(X|buys_computer = "no") * P(buys_computer = "no") = 0.007

Therefore, X belongs to class ("buys_computer = yes")

Play-tennis Example

- Example: Play Tennis

PlayTennis: training examples

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

Estimating $P(x_i|C)$

Outlook	Temperature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

P(class=p) = 9/14	
P(class=n) = 5/14	

outlook	
P(sunny|p) = 2/9	P(sunny|n) = 3/5
P(overcast|p) = 4/9	P(overcast|n) = 0
P(rain|p) = 3/9	P(rain|n) = 2/5
temperature	
P(hot|p) = 2/9	P(hot|n) = 2/5
P(mild|p) = 4/9	P(mild|n) = 2/5
P(cool|p) = 3/9	P(cool|n) = 1/5
humidity	
P(high|p) = 3/9	P(high|n) = 4/5
P(normal|p) = 6/9	P(normal|n) = 2/5
windy	
P(true|p) = 3/9	P(true|n) = 3/5
P(false|p) = 6/9	P(false|n) = 2/5

[George Kollios (Boston Univ)'s slide]

Naive Bayesian Classifier

- Given a training set, we can compute the probabilities

Outlook	P	N		Humidity	P	N
sunny	2/9	3/5		high	3/9	4/5
overcast	4/9	0		normal	6/9	1/5
rain	3/9	2/5				
Tempreature				Windy		
hot	2/9	2/5		true	3/9	3/5
mild	4/9	2/5		false	6/9	2/5
cool	3/9	1/5				

[George Kollios (Boston Univ)'s slide]

Play-tennis Example: Classifying X

- An unseen sample X = <rain, hot, high, false>

- $P(X|p) \cdot P(p) =$
  $P(rain|p) \cdot P(hot|p) \cdot P(high|p) \cdot P(false|p) \cdot P(p) =$
  $3/9 \cdot 2/9 \cdot 3/9 \cdot 6/9 \cdot 9/14 = 0.010582$
- $P(X|n) \cdot P(n) =$
  $P(rain|n) \cdot P(hot|n) \cdot P(high|n) \cdot P(false|n) \cdot P(n) =$
  $2/5 \cdot 2/5 \cdot 4/5 \cdot 2/5 \cdot 5/14 = 0.018286$

- Sample X is classified in class n (don't play)

[George Kollios (Boston Univ)'s slide]

Example

- Test Phase
  - Given a new instance, predict its label

    $\mathbf{x}'$=(Outlook=*Sunny,* Temperature=*Cool,* Humidity=*High,* Wind=*Strong*)
  - Look up tables achieved in the learning phrase

    P(Outlook=*Sunny*|Play=*Yes*) = 2/9

    P(Temperature=*Cool*|Play=*Yes*) = 3/9

    P(Huminity=*High*|Play=*Yes*) = 3/9

    P(Wind=*Strong*|Play=*Yes*) = 3/9

    P(Play=*Yes*) = 9/14

    P(Outlook=*Sunny*|Play=*No*) = 3/5

    P(Temperature=*Cool*|Play==*No*) = 1/5

    P(Huminity=*High*|Play=*No*) = 4/5

    P(Wind=*Strong*|Play=*No*) = 3/5

    P(Play=*No*) = 5/14
  - Decision making with the MAP rule

    P(*Yes*|$\mathbf{x}'$): [P(*Sunny*|Yes)P(*Cool*|*Yes*)P(*High*|Yes)P(*Strong*|*Yes*)]P(Play=*Yes*) = 0.0053

    P(*No*|$\mathbf{x}'$): [P(*Sunny*|No) P(*Cool*|No)P(*High*|*No*)P(*Strong*|*No*)]P(Play=*No*) = 0.0206

Given the fact P(*Yes*|$\mathbf{x}'$) < P(*No*|$\mathbf{x}'$), we label $\mathbf{x}'$ to be "*No*".

Naïve Bayesian Classifier

- "Naïve Bayes": all attributes contribute equally and independently
- Works surprisingly well
  - even if independence assumption is clearly violated
- Why?
  - classification doesn't need accurate probability estimates *so long as the greatest probability is assigned to the correct class*
- Adding redundant attributes causes problems
  - (e.g. identical attributes) -> *attribute selection*

The Independence Hypothesis…

- … makes computation possible
- … yields optimal classifiers when satisfied
- … but is seldom satisfied in practice, as attributes (variables) are often correlated.
- Attempts to overcome this limitation:
  - Bayesian networks, that combine Bayesian reasoning with causal relationships between attributes

[George Kollios (Boston Univ)'s slide]

Naïve Bayesian Classification

- The effect of class conditional independence
  - $P(v_1,v_2,\ldots,v_k)$ and each attributes have d values.
    - without conditional independency → d^k (chain rule)
    - with conditional independency →$d*k$
  - Simplify the computations
    - Considered "naïve" in this sense
  - In practice, dependencies can exist between attributes
    - Inaccuracy problem
- Relaxing the strong independence assumption
  - TAN, BAN, Bayesian Multi-net

Naïve Bayes

- Algorithm: Continuous-valued Features
  - Numberless values for a feature
  - Conditional probability often modeled with the normal distribution

$$\hat{P}(X_j \mid C = c_i) = \frac{1}{\sqrt{2\pi}\sigma_{ji}} \exp\left(-\frac{(X_j - \mu_{ji})^2}{2\sigma_{ji}^2} \right)$$

μ_{ji} : mean (avearage) of feature values X_j of examples for which $C = c_i$

σ_{ji} : standard deviation of feature values X_j of examples for which $C = c_i$

  - Learning Phase: for $\mathbf{X} = (X_1, \cdots, X_n),\ C = c_1, \cdots, c_L$
    Output: $n \times L$ normal distributions and $P(C = c_i)\ \ i = 1, \cdots, L$
  - Test Phase: Given an unknown instance $\mathbf{X}' = (a'_1, \cdots, a'_n)$
    - Instead of looking-up tables, calculate conditional probabilities with all th e normal distributions achieved in the learning phrase
    - Apply the MAP rule to make a decision

Naïve Bayes

- Example: Continuous-valued Features
  - Temperature is naturally of continuous value.

    Yes: 25.2, 19.3, 18.5, 21.7, 20.1, 24.3, 22.8, 23.1, 19.8

    No: 27.3, 30.1, 17.4, 29.5, 15.1
  - Estimate mean and variance for each class

$$\mu = \frac{1}{N}\sum_{n=1}^{N} x_n, \quad \sigma^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n - \mu)^2$$

$$\mu_{Yes} = 21.64, \ \sigma_{Yes} = 2.35$$
$$\mu_{No} = 23.88, \ \sigma_{No} = 7.09$$

  - **Learning Phase**: output two Gaussian models for P(temp|C)

$$\hat{P}(x \mid Yes) = \frac{1}{2.35\sqrt{2\pi}} \exp\left(-\frac{(x-21.64)^2}{2 \times 2.35^2}\right) = \frac{1}{2.35\sqrt{2\pi}} \exp\left(-\frac{(x-21.64)^2}{11.09}\right)$$

$$\hat{P}(x \mid No) = \frac{1}{7.09\sqrt{2\pi}} \exp\left(-\frac{(x-23.88)^2}{2 \times 7.09^2}\right) = \frac{1}{7.09\sqrt{2\pi}} \exp\left(-\frac{(x-23.88)^2}{50.25}\right)$$

Relevant Issues

- Violation of Independence Assumption
  - For many real world tasks, $P(X_1,\cdots,X_n|C) \neq P(X_1|C)\cdots P(X_n|C)$
  - Nevertheless, naïve Bayes works surprisingly well anyway!
- Zero conditional probability Problem
  - If no example contains the attribute value $X_j = a_{jk}$, $\hat{P}(X_j = a_{jk}|C = c_i) = 0$
  - In this circumstance, $\hat{P}(x_1|c_i)\cdots\hat{P}(a_{jk}|c_i)\cdots\hat{P}(x_n|c_i) = 0$ during test
  - For a remedy, conditional probabilities estimated with

Smoothing

$$\hat{P}(X_j = a_{jk}|C = c_i) = \frac{n_c + mp}{n + m}$$

n_c : number of training examples for which $X_j = a_{jk}$ and $C = c_i$

n : number of training examples for which $C = c_i$

p : prior estimate (usually, $p = 1/t$ for t possible values of X_j)

m : weight to prior (number of "virtual" examples, $m \geq 1$)