# Naïve Bayes Classifier

## From Last Time

- Estimating parameters
- Using Bayes Rule to incorporate prior beliefs
  - smoothing

## Estimating Parameters

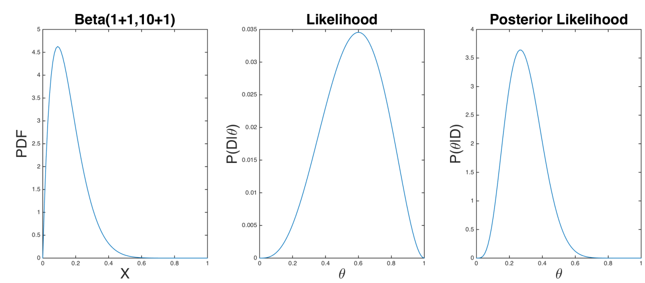
- Maximum Likelihood Estimate (MLE): choose θ that maximizes probability of observed data

$$\hat{\theta} = \arg\max_{\theta} \; P(\mathcal{D} \mid \theta)$$

- Maximum a Posteriori (MAP) estimate: choose θ that is most probable given **prior probability and the data**

$$\hat{\theta} = \arg\max_{\theta} \; P(\theta \mid \mathcal{D})$$
$$= \arg\max_{\theta} = \frac{P(\mathcal{D} \mid \theta) P(\theta)}{P(\mathcal{D})}$$

A wonderful tutorial:
http://www.cs.ubc.ca/~murphyk/Teaching/CS340-Fall06/reading/bernoulli.pdf




## Syntax

- random variable = attribute, e.g.
  - Weather is one of <sunny,rainy,cloudy,snow>
  - Windy is one of <windy,¬windy>
- Weather and Windy are discrete random variables
  - Domain values must be
    - exhaustive and
    - mutually exclusive
- Elementary propositions:
  - Weather = sunny (or simply sunny)
  - Windy = ¬windy (or simply ¬windy)

## Prior probability and distribution

- Prior or unconditional probability of a proposition is the **degree of belief** accorded to it in the absence of any other information.

P(Weather = sunny) = 0.7 (or abbrev. P(sunny) = 0.7)

- Probability distribution gives values for all possible assignments:
  - P(Weather = sunny) = 0.7
  - P(Weather = rain) = 0.2
  - P(Weather = cloudy) = 0.08
  - P(Weather = snow) = 0.02

# Conditional probability

- P(sunny | windy) = 0.8
  i.e., probability of sunny given that *windy* is all I know

- Interpreted as
  P(if Weather=sunny then Windy=windy)

- **Definition** of conditional probability:
  $P(a \mid b) = P(a \wedge b) / P(b) \qquad \text{if } P(b) > 0$

- Alternative formulation:
  $P(a \wedge b) = P(a \mid b)\, P(b) = P(b \mid a)\, P(a)$

# Bayes' Rule

- From $P(a \wedge b) = P(a \mid b)\, P(b) = P(b \mid a)\, P(a)$
  we get
  Bayes' rule:
  $P(a \mid b) = P(b \mid a)\, P(a) / P(b)$

- Useful for assessing class probability from evidence probability as:
  P(Class|Evidence) = P(Evidence|Class) P(Class) / P(Evidence)

# Onto Naïve Bayes

# Bayes' rule -- more vars

$$P(c \mid e_1, e_2) = \frac{P(c, e_1, e_2)}{P(e_1, e_2)} = \alpha P(c, e_1, e_2)$$

$\alpha = 1 / P(e_1, e_2)$

$$\begin{aligned} &= \alpha P(e_1, e_2, c) \\ &= \alpha P(e_1 \mid e_2, c) P(e_2, c) \\ &= \alpha P(e_1 \mid e_2, c) P(e_2 \mid c) P(c) \\ &= \alpha P(e_1 \mid c) P(e_2 \mid c) P(c) \end{aligned}$$

Conditional Independence

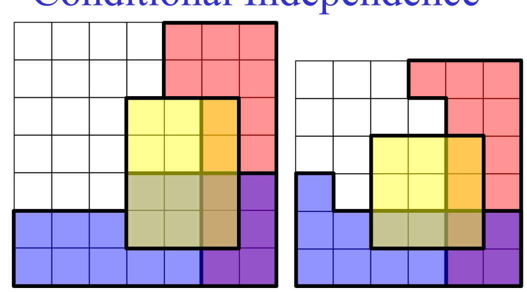
- Although the $e_1$ might not be independent of $e_2$, it might be that given the *class* they are independent.
- E.g.
  $e_1$ is 'abilityInReading'
  $e_2$ is 'lengthOfArms'
- There is indeed a dependence of abilityInReading to lengthOfArms. People with longer arms read better than those with short arms….
- However, given a *class variable*, say 'Age', the abilityInReading is independent of lengthOfArms.

# Conditional Independence

- We say A and B are conditionally independent given C if

  – P(A and B | C) = P(A|C) * P(B|C)
  – P(A|B and C) = P(A|C)

# Conditional Independence



P(R and B|Y) = P(R|Y)P(B|Y)
2/12 = (4/12)(6/12)

P(R and B|Y) = P(R|Y)P(B|Y)
1/9 = (3/9)(3/9)

Exercise: Are R and B independent given not Y?
P(R and B| not Y) = P(R| not Y)P(B| not Y) ?

# Naive Bayes

$$P(c \mid e_1,\dots,e_n) = \alpha P(e_1 \mid c)\dots P(e_n \mid c) P(c)$$

- Assumption:
  Attributes are conditionally independent (given the class value)

- Although based on assumption that is almost never correct, this scheme works well in practice!

# Weather Data

| Outlook | Temp. | Humidity | Windy | Play |
|---|---|---|---|---|
| Sunny | Hot | High | False | No |
| Sunny | Hot | High | True | No |
| Overcast | Hot | High | False | Yes |
| Rainy | Mild | High | False | Yes |
| Rainy | Cool | Normal | False | Yes |
| Rainy | Cool | Normal | True | No |
| Overcast | Cool | Normal | True | Yes |
| Sunny | Mild | High | False | No |
| Sunny | Cool | Normal | False | Yes |
| Rainy | Mild | Normal | False | Yes |
| Sunny | Mild | Normal | True | Yes |
| Overcast | Mild | High | True | Yes |
| Overcast | Hot | Normal | False | Yes |
| Rainy | Mild | High | True | No |

- A new day:

| Outlook | Temp. | Humidity | Windy | Play |
|---|---|---|---|---|
| Sunny | Cool | High | True | ? |

# Naïve Bayes for classification

- Classification learning: what's the probability of the class given an instance?
- **Instance-Tuple (Evidence):** $E_1=e_1$, $E_2=e_1$,…, $E_n=e_n$
- **Class** C = {c, …}

- Naïve Bayes assumption: evidence can be split into independent parts (i.e. attributes of instance!)

$P(c|E) = P(c \mid e_1,e_2,\dots, e_n)$
$= P(e_1|c)\, P(e_2|c)\dots P(e_n|c)\, P(c) / P(e_1,e_2,\dots, e_n)$

# The weather data example

| Outlook | Temp. | Humidity | Windy | Play |
|---|---|---|---|---|
| Sunny | Cool | High | True | ? |

← *Evidence E*

P(Play=yes | E) =
P(Outlook=Sunny | play=yes) *
P(Temp=Cool | play=yes) *
P(Humidity=High | play=yes) *
P(Windy=True | play=yes) *
P(play=yes) / P(E) =
= (2/9) *
(3/9) *
(3/9) *
(3/9) *
(9/14) / P(E) = 0.0053 / P(E)

Don't worry for the 1/P(E); It's alpha, the normalization constant.

| Outlook | Temp. | Humidity | Windy | Play |
|---|---|---|---|---|
| Sunny | Hot | High | False | No |
| Sunny | Hot | High | True | No |
| Overcast | Hot | High | False | Yes |
| Rainy | Mild | High | False | Yes |
| Rainy | Cool | Normal | False | Yes |
| Rainy | Cool | Normal | True | No |
| Overcast | Cool | Normal | True | Yes |
| Sunny | Mild | High | False | No |
| Sunny | Cool | Normal | False | Yes |
| Rainy | Mild | Normal | False | Yes |
| Sunny | Mild | Normal | True | Yes |
| Overcast | Mild | High | True | Yes |
| Overcast | Hot | Normal | False | Yes |
| Rainy | Mild | High | True | No |

# The weather data example

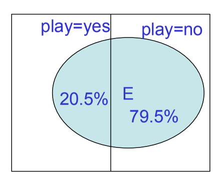
| Outlook | Temp. | Humidity | Windy | Play |
|---|---|---|---|---|
| Sunny | Cool | High | True | ? |

← *Evidence E*

P(Play=no | E) =
P(Outlook=Sunny | play=no) *
P(Temp=Cool | play=no) *
P(Humidity=High | play=no) *
P(Windy=True | play=no) *
P(play=no) / P(E) =
= (3/5) *
(1/5) *
(4/5) *
(3/5) *
(5/14) / P(E) = 0.0206 / P(E)

| Outlook | Temp. | Humidity | Windy | Play |
|---|---|---|---|---|
| Sunny | Hot | High | False | No |
| Sunny | Hot | High | True | No |
| Overcast | Hot | High | False | Yes |
| Rainy | Mild | High | False | Yes |
| Rainy | Cool | Normal | False | Yes |
| Rainy | Cool | Normal | True | No |
| Overcast | Cool | Normal | True | Yes |
| Sunny | Mild | High | False | No |
| Sunny | Cool | Normal | False | Yes |
| Rainy | Mild | Normal | False | Yes |
| Sunny | Mild | Normal | True | Yes |
| Overcast | Mild | High | True | Yes |
| Overcast | Hot | Normal | False | Yes |
| Rainy | Mild | High | True | No |

# Normalization constant




P(play=yes | E) + P(play=no | E) = 1 i.e.
0.0053 / P(E) + 0.0206 / P(E) = 1 i.e.
P(E) = 0.0053 + 0.0206
So,
P(play=yes | E) = 0.0053 / (0.0053 + 0.0206) = **20.5%**
P(play=no | E) = 0.0206 / (0.0053 + 0.0206) = **79.5%**

# The "zero-frequency problem"

- What if an attribute value doesn't occur with every class value (e.g. "Humidity = High" for class "Play=Yes")?
  - Probability P(Humidity=High| play=yes) will be zero!
- P(Play="Yes"|E) will also be zero!
  - No matter how likely the other values are!
- Remedy:
  - Add 1 to the count for every attribute value-class combination (Laplace estimator);
  - Add *k* (# of possible attribute values) to the denominator. (see example on the right).

P(play=yes | E) =
P(Outlook=Sunny | play=yes) *
P(Temp=Cool | play=yes) *
P(Humidity=High | play=yes) *
P(Windy=True | play=yes) *
P(play=yes) / P(E) =
= (2/9) * (3/9) * (3/9) * (3/9) *(9/14) / P(E) = 0.0053 / P(E)

It will be instead:

Number of possible values for 'Outlook'

= ((2+1)/(9+3)) * ((3+1)/(9+3)) * ((3+1)/(9+2)) * ((3+1)/(9+2)) *((9+1)/(14+2)) / P(E) = 0.0069 / P(E)

Number of possible values for 'Windy'

# The "zero-frequency problem"

- That's called smoothing
- "Hallucinating" training data, as for MAP

# Missing values

- **Training**: instance is not included in the frequency count for attribute value-class combination
- **Classification**: attribute will be omitted from calculation
- Example:

| Outlook | Temp. | Humidity | Windy | Play |
|---|---|---|---|---|
| ? | Cool | High | True | ? |

P(play=yes | E) =
P(Temp=Cool | play=yes) *
P(Humidity=High | play=yes) *
P(Windy=True | play=yes) *
P(play=yes) / P(E) =
= (3/12) * (3/11) * (3/11) *(10/16) / P(E) = 0.0116 / P(E)

P(play=no | E) =
P(Temp=Cool | play=no) *
P(Humidity=High | play=no) *
P(Windy=True | play=no) *
P(play=no) / P(E) =
= (2/8) * (5/7) * (4/7) *(6/16) / P(E) = 0.0382 / P(E)

After normalization: P(play=yes | E) = **23%**, P(play=no | E) = **77%**

# Dealing with numeric attributes

- Usual assumption: attributes have a normal or Gaussian probability distribution (given the class).
- Probability density function for the normal distribution is:

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

- We approximate $\mu$ by the sample mean:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

- We approximate $\sigma^2$ by the sample variance:

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

# Weather Data with Numeric Attrib.

| outlook | temperature | humidity | windy | play |
|---|---|---|---|---|
| sunny | 85 | 85 | FALSE | no |
| sunny | 80 | 90 | TRUE | no |
| overcast | 83 | 86 | FALSE | yes |
| rainy | 70 | 96 | FALSE | yes |
| rainy | 68 | 80 | FALSE | yes |
| rainy | 65 | 70 | TRUE | no |
| overcast | 64 | 65 | TRUE | yes |
| sunny | 72 | 95 | FALSE | no |
| sunny | 69 | 70 | FALSE | yes |
| rainy | 75 | 80 | FALSE | yes |
| sunny | 75 | 70 | TRUE | yes |
| overcast | 72 | 90 | TRUE | yes |
| overcast | 81 | 75 | FALSE | yes |
| rainy | 71 | 91 | TRUE | no |

| Outlook | Temp. | Humidity | Windy | Play |
|---|---|---|---|---|
| Sunny | 66 | 90 | true | ? |

f(Temperature=66 | yes)
=e(- ((66-m)^2 / 2*var) ) / sqrt(2*3.14*var)

m =
(83+70+68+64+69+75+75+72 +81)/ 9 = 73

var = ( (83-73)^2 + (70-73)^2 + (68-73)^2 + (64-73)^2 + (69-73)^2 + (75-73)^2 + (75-73)^2 + (72-73)^2 + (81-73)^2 )/ (9-1) = 38

f(Temperature=66 | yes)
=e(- ((66-73)^2 / (2*38) ) ) / sqrt(2*3.14*38) = **.034**

We compute similarly:
f(Temperature=66 | no)

# Weather Data

| outlook | temperature | humidity | windy | play |
|---|---|---|---|---|
| sunny | 85 | 85 | FALSE | no |
| sunny | 80 | 90 | TRUE | no |
| overcast | 83 | 86 | FALSE | yes |
| rainy | 70 | 96 | FALSE | yes |
| rainy | 68 | 80 | FALSE | yes |
| rainy | 65 | 70 | TRUE | no |
| overcast | 64 | 65 | TRUE | yes |
| sunny | 72 | 95 | FALSE | no |
| sunny | 69 | 70 | FALSE | yes |
| rainy | 75 | 80 | FALSE | yes |
| sunny | 75 | 70 | TRUE | yes |
| overcast | 72 | 90 | TRUE | yes |
| overcast | 81 | 75 | FALSE | yes |
| rainy | 71 | 91 | TRUE | no |

| Outlook | Temp. | Humidity | Windy | Play |
|---|---|---|---|---|
| Sunny | 66 | 90 | true | ? |

f(Humidity=90 | yes)
=e(- ((90-m)^2 / 2*var) ) / sqrt(2*3.14*var)

m =
(86+96+80+65+70+80+70+90 +75)/ 9 = 79

var = ( (86-79)^2 + (96-79)^2 + (80-79)^2 + (65-79)^2 + (70-79)^2 + (80-79)^2 + (70-79)^2 + (90-79)^2 + (75-79)^2 )/ (9-1) = 104

f(Humidity=90 | yes)
=e(- ((90-79)^2 / (2*104) ) ) / sqrt(2*3.14*104) = **.022**

We compute similarly:
f(Humidity=90 | no)

# Classifying a new day

- A new day E:

| Outlook | Temp. | Humidity | Windy | Play |
|---|---|---|---|---|
| Sunny | 66 | 90 | true | ? |

P(play=yes | E) =
P(Outlook=sunny | play=yes) *
P(Temp=66 | play=yes) *
P(Humidity=90 | play=yes) *
P(Windy=true | play=yes) *
P(play=yes) / P(E) =
= (2/9) * (0.034) * (0.022) * (3/9)
*(9/14) / P(E) = 0.000036 / P(E)

P(play=no | E) =
P(Outlook=sunny | play=no) *
P(Temp=66 | play=no) *
P(Humidity=90 | play=no) *
P(Windy=true | play=no) *
P(play=no) / P(E) =
= (3/5) * (0.0291) * (0.038) * (3/5)
*(5/14) / P(E) = 0.000136 / P(E)

After normalization: P(play=yes | E) = **20.9%**, P(play=no | E) = **79.1%**

# Probability densities

- Relationship between probability and density:

$$\Pr[c-\frac{\varepsilon}{2}<x<c+\frac{\varepsilon}{2}]\approx \varepsilon * f(c)$$

- But: this doesn't change calculation of a posteriori probabilities because ε cancels out after normalization

# Discussion of Naïve Bayes

- Naïve Bayes works surprisingly well (even if independence assumption is clearly violated)
- Because classification doesn't require accurate probability estimates as long as maximum probability is assigned to correct class

# Tax Data – Naive Bayes

| Tid | Refund | Marital Status | Taxable Income | Evade |
|---|---|---|---|---|
| 1 | Yes | Single | 125K | No |
| 2 | No | Married | 100K | No |
| 3 | No | Single | 70K | No |
| 4 | Yes | Married | 120K | No |
| 5 | No | Divorced | 95K | Yes |
| 6 | No | Married | 60K | No |
| 7 | Yes | Divorced | 220K | No |
| 8 | No | Single | 85K | Yes |
| 9 | No | Married | 75K | No |
| 10 | No | Single | 90K | Yes |

Classify: (_, No, Married, 95K, ?)

(Apply also the Laplace normalization)

# Tax Data – Naive Bayes

| Tid | Refund | Marital Status | Taxable Income | Evade |
|---|---|---|---|---|
| 1 | Yes | Single | 125K | No |
| 2 | No | Married | 100K | No |
| 3 | No | Single | 70K | No |
| 4 | Yes | Married | 120K | No |
| 5 | No | Divorced | 95K | Yes |
| 6 | No | Married | 60K | No |
| 7 | Yes | Divorced | 220K | No |
| 8 | No | Single | 85K | Yes |
| 9 | No | Married | 75K | No |
| 10 | No | Single | 90K | Yes |

Classify: (_, No, Married, 95K, ?)

P(Yes) = 3/10 = 0.3
P(Refund=No|Yes) = (3+1)/(3+2) = 0.8
P(Status=Married|Yes) = (0+1)/(3+3) = 0.17

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

Approximate μ with: (95+85+90)/3 =90
Approximate σ² with:
( (95-90)^2+(85-90) ^2+(90-90) ^2 )/(3-1) = 25
f(income=95|Yes) =
e(- ( (95-90)^2 / (2*25)) ) / sqrt(2*3.14*25) = .048
P(Yes | E) = α*.8*.17*.048*.3= α*0.0019584

# Tax Data

| Tid | Refund | Marital Status | Taxable Income | Evade |
|---|---|---|---|---|
| 1 | Yes | Single | 125K | No |
| 2 | No | Married | 100K | No |
| 3 | No | Single | 70K | No |
| 4 | Yes | Married | 120K | No |
| 5 | No | Divorced | 95K | Yes |
| 6 | No | Married | 60K | No |
| 7 | Yes | Divorced | 220K | No |
| 8 | No | Single | 85K | Yes |
| 9 | No | Married | 75K | No |
| 10 | No | Single | 90K | Yes |

Classify: (_, No, Married, 95K, ?)

P(No) = 7/10 = .7
P(Refund=No|No) = (4+1)/(7+2) = .556
P(Status=Married|No) = (4+1)/(7+3) = .5

$$f(income\,|\,No)=\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Approximate μ with:
(125+100+70+120+60+220+75)/7 =110
Approximate σ² with:
((125-110)^2 + (100-110)^2 + (70-110)^2 + (120-110)^2 + (60-110)^2 + (220-110)^2 + (75-110)^2 )/(7-1) = 2975
f(income=95|No) =
e( -((95-110)^2 / (2*2975)) ) / sqrt(2*3.14* 2975) = .00704
P(No | E) = α*.556*.5* .00704*0.7= α*. 00137

## Tax Data

| Tid | Refund | Marital Status | Taxable Income | Evade |
|---|---|---|---|---|
| 1 | Yes | Single | 125K | No |
| 2 | No | Married | 100K | No |
| 3 | No | Single | 70K | No |
| 4 | Yes | Married | 120K | No |
| 5 | No | Divorced | 95K | Yes |
| 6 | No | Married | 60K | No |
| 7 | Yes | Divorced | 220K | No |
| 8 | No | Single | 85K | Yes |
| 9 | No | Married | 75K | No |
| 10 | No | Single | 90K | Yes |

Classify: (_, No, Married, 95K, ?)

P(Yes | E) = $\alpha$*.0019584
P(No | E) = $\alpha$*.00137

$\alpha$ = 1/(.0019584 + .00137)=300.44

P(Yes|E) = 300.44 *.0019584 = 0.59
P(No|E) = 300.44 *.00137 = 0.41

We predict "Yes."

## Benefits

- Why use Naïve Bayes
  - works well
  - fewer parameters
- How many params for binary classification with N binary features?
  - $P(e_1 \mid C) \ldots P(e_N \mid C) * P(C)$
- For the full joint distribution?
  - $P(e_1 \ldots e_N \mid C) * P(C)$
- In practice, take the log before multiplying

## Text Classification

- Assign a document to a category (e.g. spam/not spam)
- Naïve Bayes models are often used for this task.
- Evidence variables are the presence or absence of each word in the language.
  - Bag of words

## Bernoulli Naïve Bayes

- Training: Given **training data** a set of documents that **have been** assigned to categories, training is just counting

of course $P(\neg W|C) = 1 - P(W|C)$

- Prior probability **P**(Category)
  - Fraction of all the training documents that are of that category
- Conditional probabilities **P**($Word_i$ | Category)
  - Fraction of docs of category c that contain word $Word_i$.
- Conditional probabilities **P**($\neg Word_i$ | Category)
  - Fraction of docs of category c that **don't** contain word $Word_i$.
- Also, P($Word_i$ | Category = $\neg$c)
  - Fraction of docs **not** of category c that contain word $Word_i$.
- P($\neg Word_i$ | Category = $\neg$c)
  - Fraction of docs **not** of category c that **don't** contain word $Word_i$.

## Bernoulli Naïve Bayes

- Now we can use Naïve Bayes for classifying a new document:

$P(Category = c \mid Word_1 = true, \ldots, Word_n = false) =$
$\alpha * P(Category = c) \prod^{n}_{i=1} P(Word_i\ ? \mid Category = c)$

$P(Category = \neg c \mid Word_1 = true, \ldots, Word_n = false) =$
$\alpha * P(Category = \neg c) \prod^{n}_{i=1} P(Word_i ? \mid Category = \neg c)$

- $Word_1, \ldots, Word_n$ are all of the words in **any** document
  - (i.e. all the words we know about)
- P($Word_i$ ? | Category = c) is
  - P($Word_i$ | Category = c) if word i does appear in test doc
  - P($\neg Word_i$ | Category = c) if word i does not appear in test doc
- $\alpha$ is the normalization constant.

## Multinomial Naïve Bayes

- Don't use probability for words that don't appear in a document

$P(Category = c \mid Word_1 = true, \ldots, Word_m = true) =$
$\alpha * P(Category = c) \prod^{m}_{i=1} P(Word_i \mid Category = c)$

- Now the product is over **only the m words that do appear** in the document we are testing on
  - (all words will be equal to true)
- If the same word appears more than one time (say t times), we count its probability more than one time (t times).

# Multinomial Naïve Bayes

- We also redefine the $P(w_i|y)$
- $N_{yi}$ is the total number of times feature i appeared for any instance with label y

$$P(w_i|y) = \frac{N_{yi} + \alpha_s}{N_y + p * \alpha_s}$$

- $N_y = \sum_{i=1}^{p} N_{yi}$ is the total number of times any feature appeared in an instance label y
- p is the total number of features,
- and $\alpha_s$ is the additive smoothing parameter.
  - Different from normalization constant

Binomial cares if a word doesn't occur, but not how many times it occurs.
Multinomial cares if a word occurs >1 time, but ignores absent words.

| Test Document | Bernoulli | Multinomial |
|---|---|---|
| X Y Y Z | Calculate<br>P(X\|C)P(Y\|C)P(Z\|C)P(C)<br>and<br>P(X\|¬C)P(Y\|¬C)<br>P(Z\|¬C)P(¬C) | Calculate<br>P(X\|C)P(Y\|C) P(Y\|C)P(Z\|C)P(C)<br>and<br>P(X\|¬C)P(Y\|¬C) P(Y\|¬C)<br>P(Z\|¬C)P(¬C) |
| X Y | Calculate<br>P(X\|C)P(Y\|C)P(¬Z\|C)P(C)<br>and<br>P(X\|¬C)P(Y\|¬C)P(¬Z\|¬C)P(¬C) | Calculate<br>P(X\|C)P(Y\|C)P(C)<br>and<br>P(X\|¬C)P(Y\|¬C)P(¬C) |
| Y Y Z | Calculate<br>P(¬X\|C)P(Y\|C)P(Z\|C)P(C)<br>and<br>P(¬X\|¬C)P(Y\|¬C)P(Z\|¬C)P(¬C) | Calculate<br>P(Y\|C)P(Y\|C)P(C)<br>and<br>P(Y\|¬C)P(Y\|¬C)P(¬C) |