COMP307/AIML420 INTRODUCTION TO ARTIFICIAL INTELLIGENCE

Reasoning under uncertainty:

Naïve Bayes Classifier

Outline

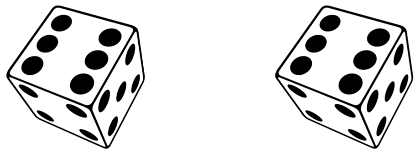
1. Review of the basic probability rules
2. Bayes Theorem
3. Naïve Bayes Classifier
   - Assumption
   - Dealing with zero count

Product Rule

$$P(A,B) = P(B) * P(A \mid B) = P(A) * P(B \mid A)$$

The probability of both events occurring is equal to the probability of the first event occurring times the probability of the second event occurring after the first event has already happened

Example: Probability that we roll two dice $D1$ and $D2$ and obtain $D1 = 4$ and $D2 = 6$.



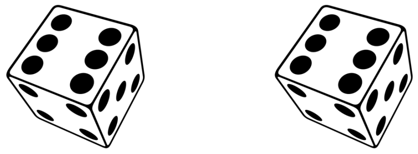
Product Rule

$$P(A, B) = P(B) * P(A \mid B) = P(A) * P(B \mid A)$$

The probability of both events occurring is equal to the probability of the first event occurring times the probability of the second event occurring after the first event has already happened

Example: Probability that when we roll two dice $D1$ and $D2$ we obtain $D1 = 4$ and $D2 = 6$.

$$P(D1 = 4, D2 = 6) = P(D1 = 4) * P(D2 = 6 \mid D1 = 4)$$



Product Rule

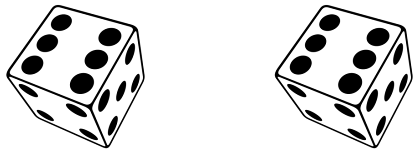
$$P(A, B) = P(B) * P(A \mid B) = P(A) * P(B \mid A)$$

The probability of both events occurring is equal to the probability of the first event occurring times the probability of the second event occurring after the first event has already happened

Example: Probability that we roll two dice $D1$ and $D2$ and obtain $D1 = 4$ and $D2 = 6$.

$P(D1 = 4, D2 = 6) = P(D1 = 4) * P(D2 = 6 \mid D1 = 4)$

$P(D1 = 4) = 1/6$
$P(D2 = 6 \mid D1 = 4) = P(D2 = 6) = 1/6$



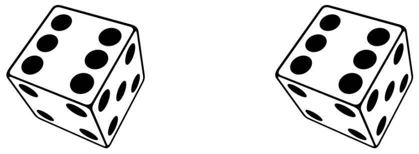
Product Rule

$$P(A,B) = P(B) * P(A \mid B) = P(A) * P(B \mid A)$$

The probability of both events occurring is equal to the probability of the first event occurring times the probability of the second event occurring after the first event has already happened

Example: Probability that we roll two dice $D1$ and $D2$ and obtain $D1 = 4$ and $D2 = 6$.

$P(D1 = 4, D2 = 6) = P(D1 = 4) * P(D2 = 6 \mid D1 = 4)$



$P(D1 = 4) = 1/6$
$P(D2 = 6 \mid D1 = 4) = P(D2 = 6) = 1/6$
The probability of $D2 = 6$ does not change by knowing that we obtained $D1 = 4$

So, $P(D1 = 4, D2 = 6) = 1/6 * 1/6 = 1/36$

Law of Total Probability

Let X and Y be random variables and let x and y be the values they can take. Then the law of total probability states:

$$P(X = x) = \sum_{y \in \Omega} P(X = x, Y = y)$$

The probability of an event A occurring can be calculated by summing the joint probability of A and other events from a set that partitions the sample space over all other events

Example: If we flip two coins (A and B), with random variables the probability that A= Tails, or $P(A = T)$*, can be expressed as

$$P(A = T) = P(A = T, B = T) + P(A = T, B = H)$$

** T = Tails, H = Heads*

Law of Total Probability

Let X and Y be random variables and let x and y be the values they can take. Then the law of total probability states:

$$P(X = x) = \sum_{y \in \Omega} P(X = x, Y = y)$$

The probability of an event A occurring can be calculated by summing the joint probability of A and other events from a set that partitions the sample space over all other events

Example: If we flip two coins (A and B), with random variables the probability that A= Tails, or $P(A = T)$*, can be expressed as

$$P(A = T) = P(A = T, B = T) + P(A = T, B = H)$$

Can also be written as

$$P(A = T) = P(A = T | B = T) * P(B = T) + P(A = T | B = H) * P(B = H)$$

* *T = Tails, H = Heads*

Normalisation Rule

$$\sum_{x} P(X = x) = 1$$

$$\sum_{x} P(X = x \mid Y = y) = 1$$

The sum of the probabilities of all values a random variable can take must sum to one. This is not changed by conditioning.

Example: If we sum the probabilities of all possible outcomes of a die, we obtain 1 (or 100%)

Normalization Rule

$$\sum_{x} P(X = x) = 1$$

$$\sum_{x} P(X = x \mid Y = y) = 1$$

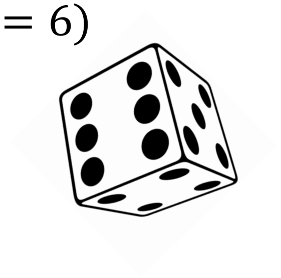
The sum of the probabilities of all values a random variable can take must sum to one. This is not changed by conditioning.

Example: If we sum the probabilities of all possible outcomes of a die, we obtain 1 (or 100%)

$P(D = 1) + P(D = 2) + P(D = 3) + P(D = 4) + P(D = 5) + P(D = 6)$

$= 1/6 + 1/6 + 1/6 + 1/6 + 1/6 + 1/6$

$= \mathbf{6/6 = 1}$



Independence

Any of the following conditions implies independence:

$$\mathrm{P(A \mid B) = P(A)}$$
$$\mathrm{P(B \mid A) = P(B)}$$
$$\mathrm{P(A, B) = P(A) * P(B)}$$

If events A and B are independent, then their joint probability $P(A, B)$ is the product of their individual probabilities.

Independence

Any of the following conditions implies independence:

$$P(A \mid B) = P(A)$$
$$P(B \mid A) = P(B)$$
$$P(A, B) = P(A) * P(B)$$

If events A and B are independent, then their joint probability $P(A, B)$ is the product of their individual probabilities.

Example: the probability of drawing (with replacement) an Ace from a deck of cards does not influence the probability of drawing a King from the deck.

Bayes Theorem

- Describes how to update **our beliefs** about **the probability of an event based on new evidence** which we represent as $P(A \mid B)$

Remember the Product rule?

$$P(A, B) = P(B) * P(A \mid B) = P(A) * P(B \mid A)$$

Thomas Bayes (/ˈbeɪz/; c. 1701 – 7 April 1761)

Bayes Theorem

- Describes how to update **our beliefs** about **the probability of an event based on new evidence** which we represent as $P(A \mid B)$

Product rule

$$P(A,B) = \boxed{P(B)} * \boxed{P(A \mid B)} = \boxed{P(A) * P(B \mid A)}$$

$$\boxed{P(A \mid B)} = \frac{\boxed{P(B \mid A) * P(A)}}{\boxed{P(B)}}$$

Thomas Bayes (/ˈbeɪz/; c. 1701 – 7 April 1761)

Bayes Theorem Intuition

- The key idea is that **our beliefs** about the **probability of an event** should be **updated** given **new evidence**

$$\boxed{P(A \mid B) = \frac{P(B \mid A) * P(A)}{P(B)}}$$

- $P(A|B)$ is the probability of event A given that event B has occurred (this is known as the **posterior probability of event A**).
- $P(B|A)$ is the probability of event B given that event A has occurred (this is known as the **likelihood of event A**). Often known from model or data.
- $P(A)$ is the **prior probability of event A** (i.e., the total probability of A).
- $P(B)$ is the **prior probability of event B** (i.e., the total probability of B).

Bayesian Spam Filter

- Easy to measure probability of certain words in spam messages
- We want probability that a message is spam if a set of words occurs

$$P(\,spam\,|words) = \frac{P(words|spam)\,p(spam)}{P(words)}$$

- So if likelihood $P(words|spam) = 0.8$ and prior $p(spam) = 0.1$ and $P(words) = 0.5$ then posterior $P(\,spam\,|words) = 0.16$
- We can do better! Use *naïve Bayes classifiers*

Naïve Bayes Classifier

- Simple but effective probabilistic algorithm that is commonly used for classification tasks (often a baseline)
- It can be effective in practice, especially for text classification and other applications with high-dimensional feature spaces
- It is computationally efficient, making it suitable for real-time applications and large datasets
- *"Naïve"* because it makes the simplifying assumption that the *features are conditionally independent given the class label*

Principle of Naïve Bayes Classifier

- Observe a set of features $\{X_1, X_2, \cdots, X_p\}$
  - Spam words : event "click here" happens, then $X_1 = 1$, otherwise $X_1 = 0$
- Consider class C
  - Spam $C = 1$ versus not-spam $C = 0$
- Goal: to find class $C = c$ with highest probability given observations
  - Long notation: $P(C = c \mid X_1 = x_1, X_2 = x_2, \cdots, X_p = x_p)$
  - Short notation: $p(c \mid x_1, x_2, \cdots, x_p)$
  - Determine probability that an email is spam
- Naïve Bayes:

$$p(c \mid x_1, \cdots, x_p) = \frac{p(c)p(x_1, \cdots, x_p|c)}{p(x_1, \cdots, x_p)}$$

class conditional feature independence

$$= \frac{p(c)p(x_1|c) \cdots p(x_p|c)}{p(x_1, \cdots, x_p)}$$

- We don't need denominator to find c with highest probability → *score*

Example Naïve Bayes Classifier

word	class
”urgent”	**spam**
<4 examples “urgent”, spam>	
“urgent”	**spam**
“urgent”	**not spam**
<112 examples ”urgent”, not spam>	
“urgent”	**not spam**
<other data>	
“free”	**spam**
<14 examples ”urgent”, not spam>	
“free”	**spam**
“free”	**not spam**
<37 examples ”urgent”, not spam>	
“free”	**not spam**

Original dataset with 4 features and 365 entries

word	**spam**	**not spam**
“urgent”	6	114
“free”	16	39
“success”	0	171
”cheap”	12	7

Counters for a dataset with 300 entries of which 100 were spam

Counting occurrences in the dataset to find $P(X_1 = x_1|C = c), \cdots, P\left(X_p = x_p\middle|C = c\right)$ and $P(C = c)$; in short notation: $p(x_1|c), p(x_1|c), \cdots, \left(x_p\middle|c\right)$ and $p(c)$

Example Naïve Bayes Classifier

word	class
”urgent”	spam
<4 examples “urgent”, spam>	
“urgent”	spam
“urgent”	not spam
<112 examples ”urgent”, not spam>	
“urgent”	not spam
<other data>	
“free”	spam
<14 examples ”urgent”, not spam>	
“free”	spam
“free”	not spam
<37 examples ”urgent”, not spam>	
“free”	not spam

Original dataset with 4 features and 365 entries

word	spam	not spam
“urgent”	6	114
“free”	16	39
“success”	0	171
”cheap”	12	7

Counters for a dataset with 300 entries of which 100 were spam

Counting occurrences in the dataset to find $P(X_1 = x_1|C = c), \cdots, P\left(X_p = x_p \middle| C = c\right)$ and $P(C = c)$; in short notation: $p(x_1|c), p(x_1|c), \cdots, \left(x_p \middle| c\right)$ and $p(c)$

Example Naïve Bayes Classifier

- What do the occurrence tables of the example on previous slides show?
  - $P(\text{“}urgent\text{”} = 1 \mid C = spam)$
  - $P(\text{“}urgent\text{”} = 1 \mid C = not\ spam)$
- What the occurrence tables do not show explicitly, but is known:
  - $P(\text{“}urgent\text{”} = 0 \mid C = spam) = 1 - P(\text{“}urgent\text{”} = 1 \mid C = spam)$
  - $P(\text{“}urgent\text{”} = 0 \mid C = not\ spam) = 1 - P(\text{“}urgent\text{”} = 1 \mid C = not\ spam)$

- Hence, we can now compute for any email message the numerator of

$$P\left(C = spam \mid X_1 = x_1, \cdots, X_p = x_p\right) = \frac{P(C=spam)p(X_1=x_1|C=spam) \cdots p(X_p=x_p|C=spam)}{p(x_1,\cdots,x_p)}$$

by computing the probabilities from the occurrence rates in the tables and, then filling out the x_i observed in the table and looking up the computed probabilities. **The denominator is usually not needed**; here we can compare spam and not-spam numerators.

- Is the conditional independence assumption reasonable here?

Computing the Probabilities

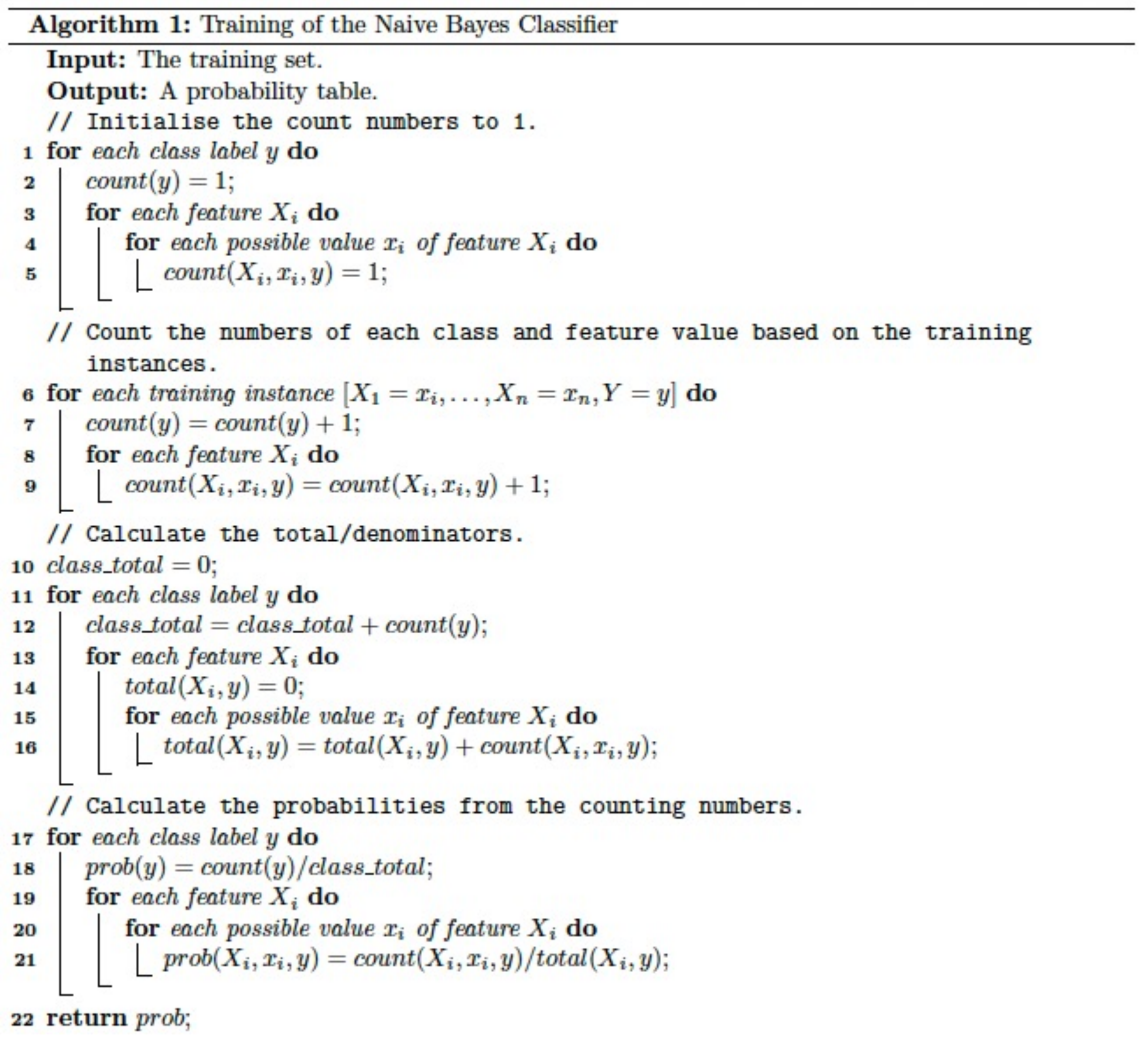
- Data set with 300 entries of which 100 were spam and the counts of the table

word	**spam**	**not spam**
“urgent”	6	114
“free”	16	39
“success”	0	171
”cheap”	12	7

- $P(\text{“}free\text{” } occurs \mid C = spam) = \frac{p(\text{“}free\text{” } occurs, C=spam)}{P(C=spam)} = \frac{16/300}{100/300} = 0.16$
- $P(no \text{ “}free\text{”} \mid C = spam) = 1 - 0.16 = 0.84$
- Similarly for other values of other features
- $P(\, C = spam) = \frac{100}{300} = \frac{1}{3}$

Naïve Bayes Classifier

- The specific training algorithm (See assignment 3):

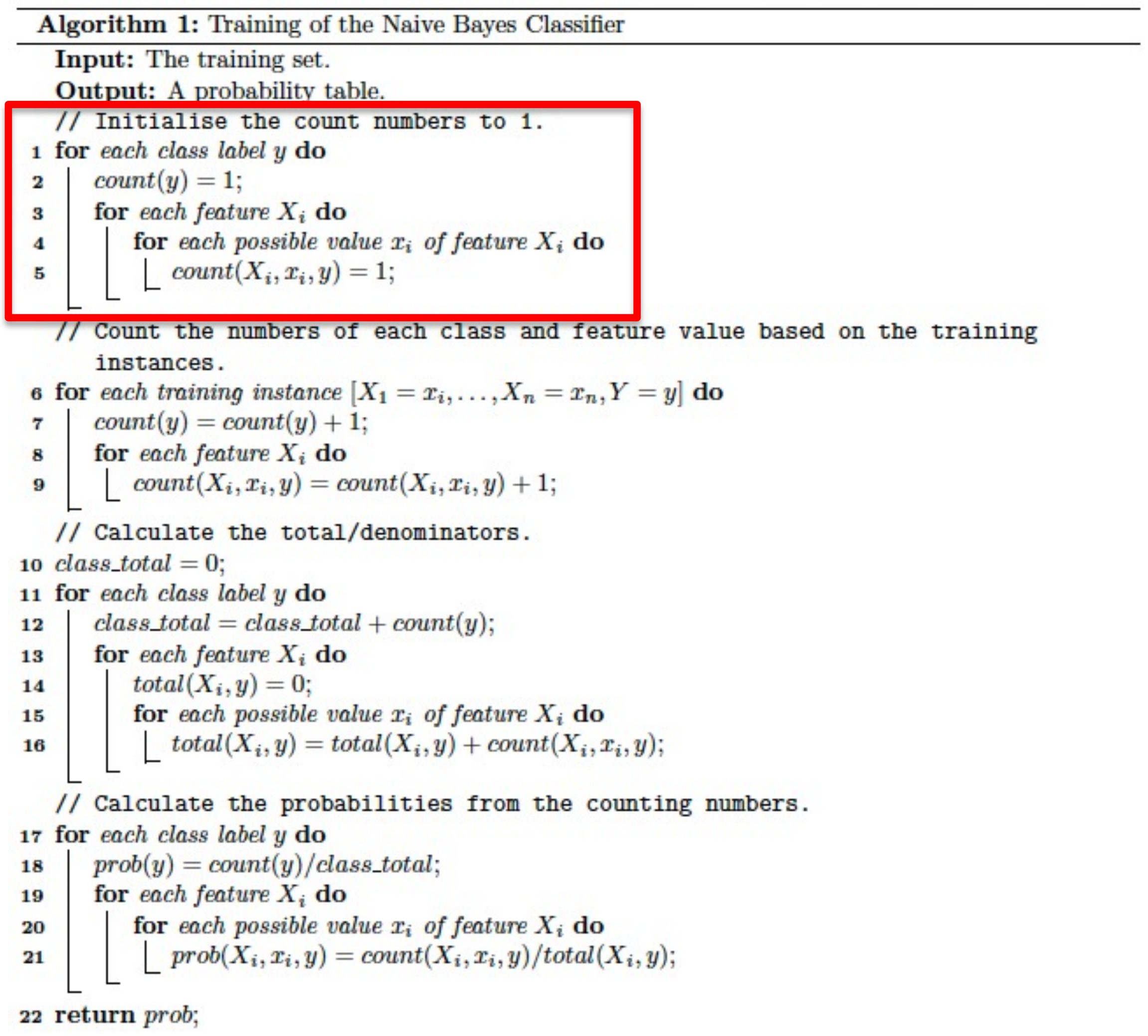


Algorithm 1: Training of the Naive Bayes Classifier

```
Input: The training set.
Output: A probability table.
// Initialise the count numbers to 1.
for each class label $y$ do
    $count(y) = 1$;
    for each feature $X_i$ do
        for each possible value $x_i$ of feature $X_i$ do
            $count(X_i, x_i, y) = 1$;
// Count the numbers of each class and feature value based on the training
   instances.
for each training instance $[X_1 = x_i, \ldots, X_n = x_n, Y = y]$ do
    $count(y) = count(y) + 1$;
    for each feature $X_i$ do
        $count(X_i, x_i, y) = count(X_i, x_i, y) + 1$;
// Calculate the total/denominators.
$class\_total = 0$;
for each class label $y$ do
    $class\_total = class\_total + count(y)$;
    for each feature $X_i$ do
        $total(X_i, y) = 0$;
        for each possible value $x_i$ of feature $X_i$ do
            $total(X_i, y) = total(X_i, y) + count(X_i, x_i, y)$;
// Calculate the probabilities from the counting numbers.
for each class label $y$ do
    $prob(y) = count(y)/class\_total$;
    for each feature $X_i$ do
        for each possible value $x_i$ of feature $X_i$ do
            $prob(X_i, x_i, y) = count(X_i, x_i, y)/total(X_i, y)$;
return $prob$;
```

Naïve Bayes Classifier

- The specific training algorithm (See assignment 3):



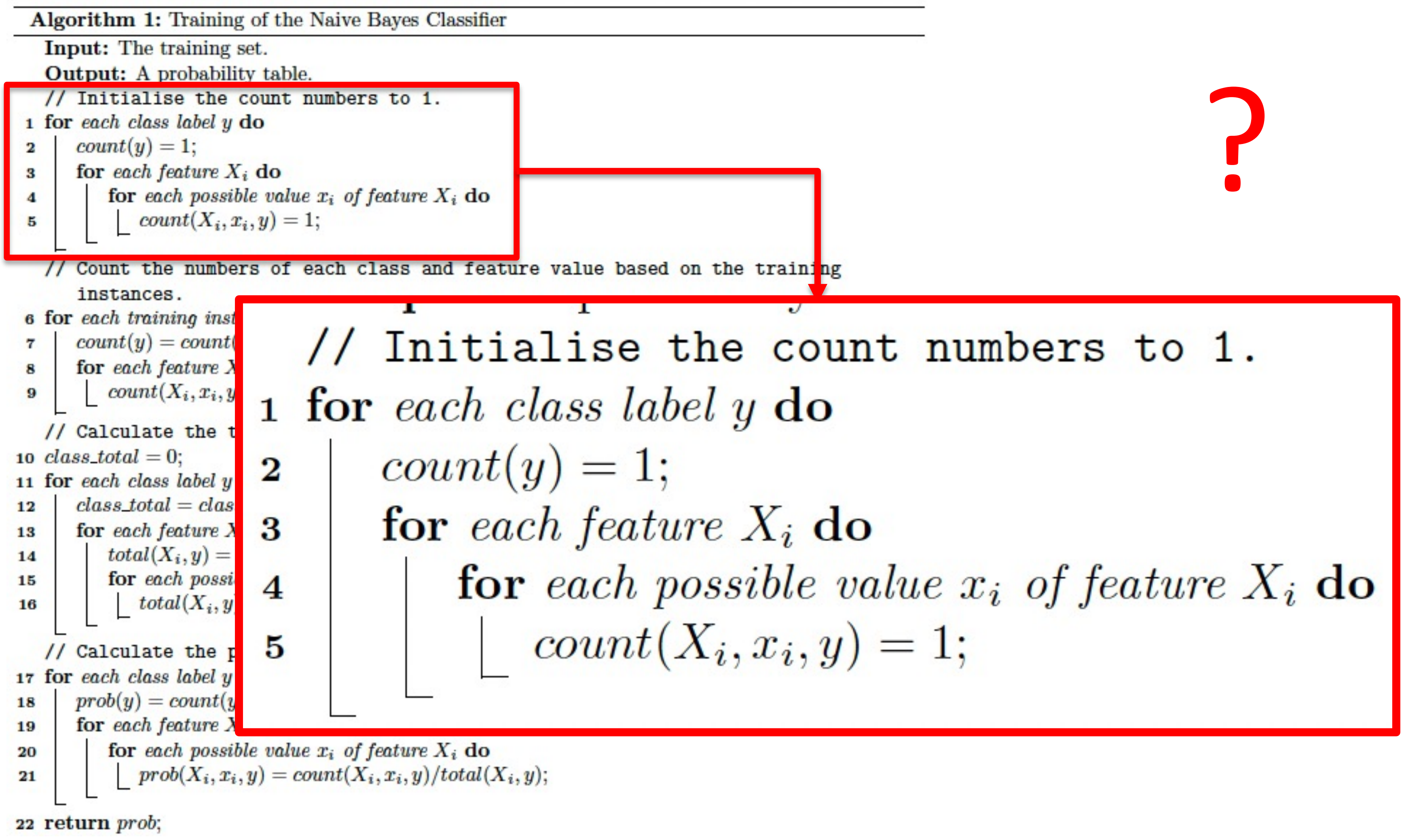
Algorithm 1: Training of the Naive Bayes Classifier

```
Input: The training set.
Output: A probability table.
// Initialise the count numbers to 1.
for each class label $y$ do
    $count(y) = 1$;
    for each feature $X_i$ do
        for each possible value $x_i$ of feature $X_i$ do
            $count(X_i, x_i, y) = 1$;
// Count the numbers of each class and feature value based on the training
   instances.
for each training instance $[X_1 = x_i, \ldots, X_n = x_n, Y = y]$ do
    $count(y) = count(y) + 1$;
    for each feature $X_i$ do
        $count(X_i, x_i, y) = count(X_i, x_i, y) + 1$;
// Calculate the total/denominators.
$class\_total = 0$;
for each class label $y$ do
    $class\_total = class\_total + count(y)$;
    for each feature $X_i$ do
        $total(X_i, y) = 0$;
        for each possible value $x_i$ of feature $X_i$ do
            $total(X_i, y) = total(X_i, y) + count(X_i, x_i, y)$;
// Calculate the probabilities from the counting numbers.
for each class label $y$ do
    $prob(y) = count(y)/class\_total$;
    for each feature $X_i$ do
        for each possible value $x_i$ of feature $X_i$ do
            $prob(X_i, x_i, y) = count(X_i, x_i, y)/total(X_i, y)$;
return $prob$;
```

?

Naïve Bayes Classifier

- The specific training algorithm (See assignment 3):



Algorithm 1: Training of the Naive Bayes Classifier

Input: The training set.
Output: A probability table.

```
   // Initialise the count numbers to 1.
1  for each class label $y$ do
2      $count(y) = 1$;
3      for each feature $X_i$ do
4          for each possible value $x_i$ of feature $X_i$ do
5              $count(X_i, x_i, y) = 1$;
   // Count the numbers of each class and feature value based on the training
      instances.
6  for each training inst
7      $count(y) = count($
8      for each feature $X$
9          $count(X_i, x_i, y$
   // Calculate the t
10 $class\_total = 0$;
11 for each class label $y$
12     $class\_total = clas$
13     for each feature $X$
14         $total(X_i, y) =$
15         for each possi
16             $total(X_i, y$
   // Calculate the p
17 for each class label $y$
18     $prob(y) = count(y$
19     for each feature $X$
20         for each possible value $x_i$ of feature $X_i$ do
21             $prob(X_i, x_i, y) = count(X_i, x_i, y)/total(X_i, y)$;
22 return $prob$;
```

```
   // Initialise the count numbers to 1.
1  for each class label $y$ do
2      $count(y) = 1$;
3      for each feature $X_i$ do
4          for each possible value $x_i$ of feature $X_i$ do
5              $count(X_i, x_i, y) = 1$;
```

?

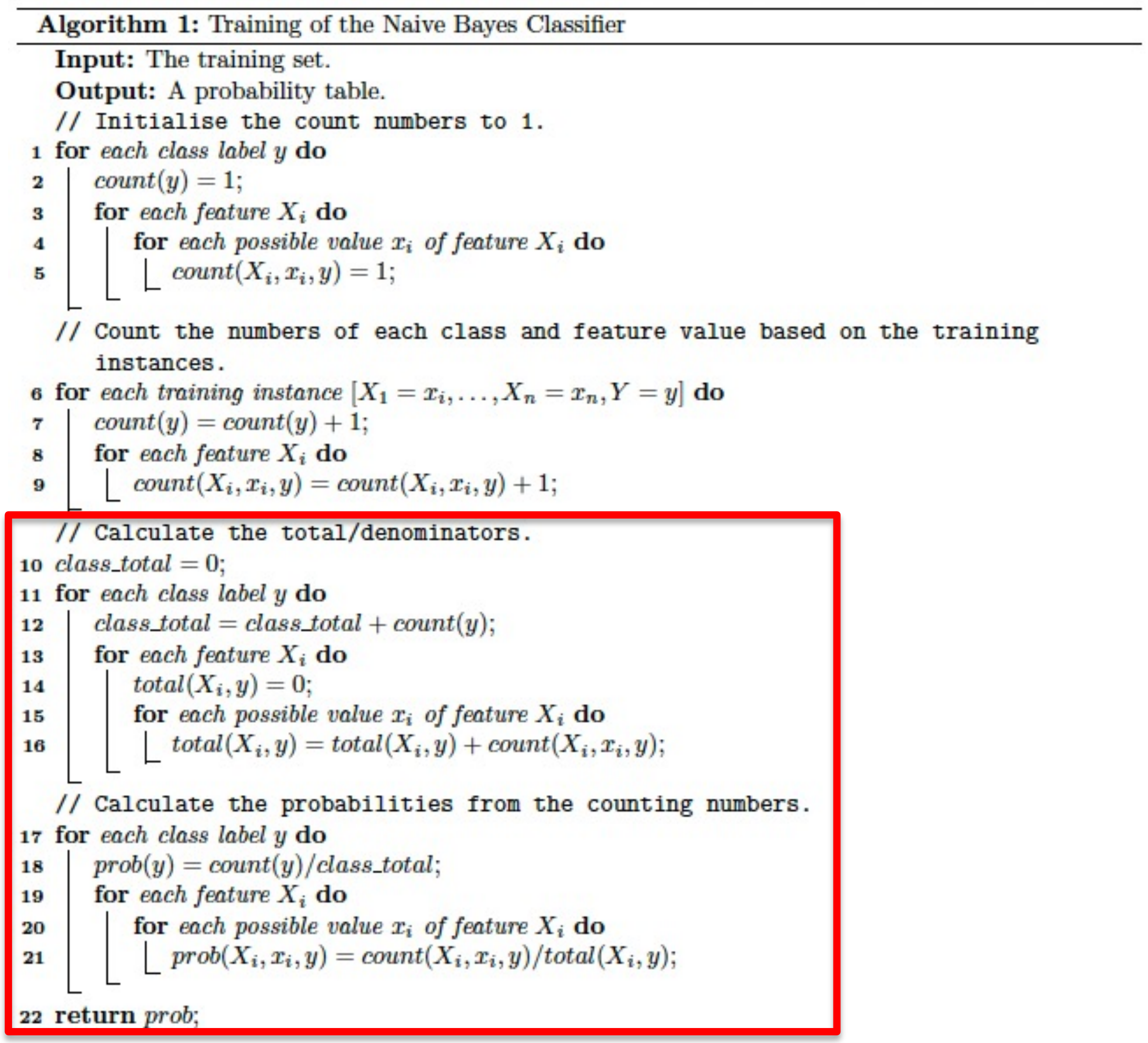
Naïve Bayes Classifier

- $p(c \mid x_1, \cdots, x_p) = \frac{p(c)p(x_1|c) \cdots p(x_p|c)}{p(x_1,\cdots,x_p)}$
- Unfortunately $p(x_i|c) = 0$ means $p(c \mid x_1, \cdots, x_p) = 0$
- We simply initialize counts with 1 to avoid 0 probabilities ("smoothing")

word	**spam**	**not spam**
"urgent"	6	114
"free"	16	39
"success"	$\boxed{0}$	171
"cheap"	12	7

Naïve Bayes Classifier

- The specific training algorithm (See assignment 3):



Algorithm 1: Training of the Naive Bayes Classifier

```
Input: The training set.
Output: A probability table.
// Initialise the count numbers to 1.
for each class label $y$ do
    $count(y) = 1$;
    for each feature $X_i$ do
        for each possible value $x_i$ of feature $X_i$ do
            $count(X_i, x_i, y) = 1$;
// Count the numbers of each class and feature value based on the training
   instances.
for each training instance $[X_1 = x_i, \ldots, X_n = x_n, Y = y]$ do
    $count(y) = count(y) + 1$;
    for each feature $X_i$ do
        $count(X_i, x_i, y) = count(X_i, x_i, y) + 1$;
// Calculate the total/denominators.
$class\_total = 0$;
for each class label $y$ do
    $class\_total = class\_total + count(y)$;
    for each feature $X_i$ do
        $total(X_i, y) = 0$;
        for each possible value $x_i$ of feature $X_i$ do
            $total(X_i, y) = total(X_i, y) + count(X_i, x_i, y)$;
// Calculate the probabilities from the counting numbers.
for each class label $y$ do
    $prob(y) = count(y)/class\_total$;
    for each feature $X_i$ do
        for each possible value $x_i$ of feature $X_i$ do
            $prob(X_i, x_i, y) = count(X_i, x_i, y)/total(X_i, y)$;
return $prob$;
```

Naïve Bayes Classifier

- The training algorithm:

```
// Calculate the total/denominators.
$class\_total = 0$;
for each class label $y$ do
    $class\_total = class\_total + count(y)$;
    for each feature $X_i$ do
        $total(X_i, y) = 0$;
        for each possible value $x_i$ of feature $X_i$ do
            $total(X_i, y) = total(X_i, y) + count(X_i, x_i, y)$;

// Calculate the probabilities from the counting numbers.
for each class label $y$ do
    $prob(y) = count(y)/class\_total$;
    for each feature $X_i$ do
        for each possible value $x_i$ of feature $X_i$ do
            $prob(X_i, x_i, y) = count(X_i, x_i, y)/total(X_i, y)$;
return $prob$;
```

Computes: $p(C = c)$

Computes: $p(X_i = x_i | C = c)$

Another Example

- Another context: approve or decline a loan
- Three features (Job, Deposit, Family)

Applicant	Job	Deposit	Family	Class
1	true	low	single	Approve
2	true	low	couple	Approve
3	true	high	single	Approve
4	true	high	single	Approve
5	false	high	couple	Approve
6	true	low	couple	Decline
7	false	low	couple	Decline
8	true	low	children	Decline
9	false	low	single	Decline
10	false	high	children	Decline

A Second Example

- Our objective in the second example is to make a program that mimics a loan officer. We use naïve Bayes to define a decision based on example data.

A Second Example

- Approve or decline a loan
- Three features, job, deposit, family, with 2, 2, 3 values, respectively

Class	Approve	Decline
Total	5	5
Job = true	4	2
Job = false	1	3
Dep = low	2	4
Dep = high	3	1
Fam = single	3	1
Fam = couple	2	2
Fam = children	0	2

	Approve	Decline
P(Class)	5/10	5/10
P(Job = true | Class)	4/5	2/5
P(Job = false | Class)	1/5	3/5
P(Dep = low | Class)	2/5	4/5
P(Dep = high | Class)	3/5	1/5
P(Fam = single | Class)	3/5	1/5
P(Fam = couple | Class)	2/5	2/5
P(Fam = children | Class)	0/5	2/5

A Second Example

Class	Approve	Decline
Total	5	5
Job = true	4	2
Job = false	1	3
Dep = low	2	4
Dep = high	3	1
Fam = single	3	1
Fam = couple	2	2
Fam = children	$\bigcirc$ 0	2

	Approve	Decline
P(Class)	5/10	5/10
P(Job = true | Class)	4/5	2/5
P(Job = false | Class)	1/5	3/5
P(Dep = low | Class)	2/5	4/5
P(Dep = high | Class)	3/5	1/5
P(Fam = single | Class)	3/5	1/5
P(Fam = couple | Class)	2/5	2/5
P(Fam = children | Class)	0/5	2/5

Problem case! Feature = Fam
Value = children
Class = Approve

A Second Example

- **Solving** the problem of the 0 probabilities: add one to all values

Class	Approve	Decline
Total	6	6
Job = true	5	3
Job = false	2	4
Dep = low	3	5
Dep = high	4	2
Fam = single	4	2
Fam = couple	3	3
Fam = children	1	3

	Approve	Decline
P(Class)	6/12	6/12
P(Job = true | Class)	5/7	3/7
P(Job = false | Class)	2/7	4/7
P(Dep = low | Class)	3/7	5/7
P(Dep = high | Class)	4/7	2/7
P(Fam = single | Class)	4/8	2/8
P(Fam = couple | Class)	3/8	3/8
P(Fam = children | Class)	1/8	3/8

Solved! Initialized the counters with 1

A Second Example

- Given a test instance (Job=true,Dep=high,Fam=children)
- Calculate P(**Decline** | Job=true, Dep=high, Fam=children)
- Calculate P(Approve | Job=true, Dep=high, Fam=children)
- See which probability is higher

$$
\begin{aligned}
& P(Decline|Job = true, Dep = high, Fam = children) \\
&= \frac{P(Decline) * P(Job = true, Dep = high, Fam = children|Decline)}{P(Job = true, Dep = high, Fam = children)} \\
&= \frac{P(Decline) * P(Job = true|Decline) * P(Dep = high|Decline) * P(Fam = children|Decline)}{P(Job = true, Dep = high, Fam = children)} \\
&= \frac{3/7 \times 2/7 \times 3/8 \times 1/2}{P(Job = true, Dep = high, Fam = children)} \\
&= \frac{0.0230}{P(Job = true, Dep = high, Fam = children)}
\end{aligned}
$$

Second Example

- Given a test instance (Job=true,Dep=high,Fam=children)
- Calculate P(Decline | Job=true, Dep=high, Fam=children)
- Calculate P(**Approve** | Job=true, Dep=high, Fam=children)
- See which probability is higher

$$
\begin{aligned}
&P(Approve|Job = true, Dep = high, Fam = children)\\
=&\frac{P(Approve) * P(Job = true, Dep = high, Fam = children|Approve)}{P(Job = true, Dep = high, Fam = children)}\\
=&\frac{P(Approve) * P(Job = true|Approve) * P(Dep = high|Approve) * P(Fam = children|Approve)}{P(Job = true, Dep = high, Fam = children)}\\
=&\frac{5/7 \times 4/7 \times 1/8 \times 1/2}{P(Job = true, Dep = high, Fam = children)}\\
=&\frac{0.0255}{P(Job = true, Dep = high, Fam = children)}
\end{aligned}
$$

A Second Example

$$
\begin{aligned}
&P(Approve|Job = true, Dep = high, Fam = children) \\
&= \frac{P(Approve) * P(Job = true, Dep = high, Fam = children|Approve)}{P(Job = true, Dep = high, Fam = children)} \\
&= \frac{P(Approve) * P(Job = true|Approve) * P(Dep = high|Approve) * P(Fam = children|Approve)}{P(Job = true, Dep = high, Fam = children)} \\
&= \frac{5/7 \times 4/7 \times 1/8 \times 1/2}{P(Job = true, Dep = high, Fam = children)} \\
&= \frac{\boxed{0.0255}}{P(Job = true, Dep = high, Fam = children)}
\end{aligned}
$$

$$
\begin{aligned}
&P(Decline|Job = true, Dep = high, Fam = children) \\
&= \frac{P(Decline) * P(Job = true, Dep = high, Fam = children|Decline)}{P(Job = true, Dep = high, Fam = children)} \\
&= \frac{P(Decline) * P(Job = true|Decline) * P(Dep = high|Decline) * P(Fam = children|Decline)}{P(Job = true, Dep = high, Fam = children)} \\
&= \frac{3/7 \times 2/7 \times 3/8 \times 1/2}{P(Job = true, Dep = high, Fam = children)} \\
&= \frac{\boxed{0.0230}}{P(Job = true, Dep = high, Fam = children)}
\end{aligned}
$$

Naïve Bayes Classifier

- Making predictions:

- *For the prediction of each test instance, you need to calculate the score (numerator) of the test instance for each class and find the class with the* ***largest score****.*

- *The* ***score*** *of a class y is calculated as follows.*

Algorithm 2: Calculation of the class score.

```
Input: A test instance $[X_1 = x_1, \ldots, X_n = x_n]$, a class label $y$, the probability table $prob$.
Output: The score.
$score = prob(y)$;
for each feature $X_i$ do
    $score = score * prob(X_i, x_i, y)$;
return $score$;
```

- If we have *n* classes, we will obtain *n* scores. The class prediction is the class with the highest score.

Summary

- **Introduced** the Bayes Theorem
- **Discussed** the intuition behind the Bayes Theorem
- **Introduced** the Naive Bayes Classifier
  - The naïve assumption: conditional independence
  - Algorithm and some details

Coming up next…

- Bayesian Networks