CORK INSTITUTE OF TECHNOLOGY
INSTITIÚID TEICNEOLAÍOCHTA CHORCAÍ

Computer Science

http://www.cit.ie

Programming for Data Analytics

Week11: Bayesian Classification

Dr. Haithem Afli

Haithem. afli@cit.ie

@AfliHaithem

2018/2017

Contents

Conditional Probability

- Unconditional probability deals with the independent probability of a proposition. That is the probability of a proposition that is not contingent on any other probability.

- Conditional Probability:
  - Represents the probability that one **event will occur given that a second event has already occurred**

```
P(cavity | toothache) = 0.8
```

"Prob. of cavity is 0.8, given that all you know is you have toothache"

Product Rule

- The product rule can be applied

$$P(a \wedge b) = P(a \mid b)\, P(b) = P(b \mid a)\, P(a)$$

- What is the probability of drawing two kings from a deck of card (note the first king is not replaced before picking the second)

- a = first card is a king
- b = second card is a king
- P(a and b) = P(b|a) P(a)

Bayes' Rule

$$P(c \mid d) = (P(d \mid c)\ P(c)) \ / \ P(d)$$

Bayes' Rule can be easily derived from the product rule

$P(c \wedge d) =$ **$P(c \mid d)\ P(d) = P(d \mid c)\ P(c)$**

Divide by P(d): **$P(c \mid d) = P(d \mid c)\ P(c)/P(d)$**

Let’s Apply it

Example

$$P(c \mid d) = (P(d \mid c)\ P(c)) / P(d)$$

- A prison has a population of 600 inmates. It has two cellblocks.
- Cell block A has 500 prisoners with the remainder of the prisoners in cell block B.
- Half the prisoners in cell black A wear a grey uniform. The other half wears a blue uniform.
- All prisoners in cell block B wear a blue uniform.
- A prisoner wearing a blue uniform is seen escaping. What is the probability that the prisoner is from cell block A?

P(CellBlockA | BlueU)

=

Example

$$P(c \mid d) = (P(d \mid c)\ P(c)) / P(d)$$

- A prison has a population of 600 inmates. It has two cellblocks.
- Cell block A has 500 prisoners with the remainder of the prisoners in cell block B.
- Half the prisoners in cell black A wear a red uniform. The other half wears a blue uniform.
- All prisoners in cell block B wear a blue uniform.
- A prisoner wearing a blue uniform is seen escaping. What is the probability that the prisoner is from cell block A?

P(CellBlockA | BlueU)

= P(BlueU | CellBlockA)*P(CellBlockA)/P(BlueU)

= (((250/500)(500/600)) / (350/600))
= 0.71

Contents

Classification Example

- The table below shows a database of names for workers that work in a particular company. If I pick an employee from the company and their name is Joe, is this individual male or female? The class in this problem is the Sex and the attribute/feature is the Name of the individual.
- This is a trivial problem but we can use Bayes to help solve it.

Name	Sex
Joe	**Female**
Jim	Male
Joe	**Male**
Ted	Male
Mary	Female
Joe	**Male**
Carol	Female
Emily	Female

Classification Example

Name	Sex
Joe	Female
Jim	Male
Joe	Male
Ted	Male
Mary	Female
Joe	Male
Carol	Female
Emily	Female

$$P(c \mid d) = \frac{P(d \mid c)P(c)}{P(d)}$$

- We must calculate the probability for each class and then adopt the class with the highest probability
- In other words calculate probability of being a male/female given that individual is called Joe.
  - P(Female | Joe)
  - P(Male | Joe)

Classification Example

Name	Sex
Joe	Female
Jim	Male
Joe	Male
Ted	Male
Mary	Female
Joe	Male
Carol	Female
Emily	Female

$$P(c \mid d) = \frac{P(d \mid c)P(c)}{P(d)}$$

- **P(Female | Joe)** = P(Joe | Female)P(Female) / P(Joe)
  = (1/4)(4/8) / (3/8) = **0.333**

- **P(Male | Joe)** = P(Joe | Male)P(Male) / P(Joe)
  = (2/4)(4/8)/(3/8) = **0.666**

- Probability of Joe being a Male is higher, therefore we can classify the individual Joe as being a Male

Classification Example

$$P(c \mid d) = \frac{P(d \mid c)P(c)}{P(d)}$$

- P(Female | Joe) = P(Joe | Female)P(Female) / P(Joe)
  = (1/4)(4/8) / **(3/8)** = 0.333

- P(Male | Joe) = P(Joe | Male)P(Male) / P(Joe)
  = (2/4)(4/8) / **(3/8)** = 0.666

You might notice that for all these calculations, the denominators are identical—P(d). Thus, they are independent of the hypotheses.

Classification Example (maximisation)

$$P(c \mid d) = \frac{P(d \mid c)P(c)}{P(d)}$$

- P(Female | Joe) = P(Joe | Female)P(Female) / P(Joe)
  = (1/4)(4/8) = 0.125

- P(Male | Joe) = P(Joe | Male)P(Male) / P(Joe)
  = (2/4)(4/8) = 0.25

More Formally

- More formally we want to identify the most likely class C_{MAP}
- For all classes the one that maximises probability of that class given the attribute d

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(c \mid d)$$

$$= \underset{c \in C}{\operatorname{argmax}} \frac{P(d \mid c)P(c)}{P(d)}$$

$$= \underset{c \in C}{\operatorname{argmax}} P(d \mid c)P(c)$$

Bayes with Multiple Attributes/Features

Bayes with Multiple Attributes/Features

- In the previous slides we considered Bayes classification when we had only a single feature (for example the name of an individual).

- But what if we have many other attributes as well such as age, height, weight etc.

Height	Weight	Name	Sex
X	X	Joe	Female
X	X	Jim	Male
X	X	Joe	Male
X	X	Ted	Male
X	X	Mary	Female
X	X	Joe	Male
X	X	Carol	Female
x	X	Emily	Female

Bayes with Multiple Attributes/Features

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(d \mid c)P(c)$$

$$= \underset{c \in C}{\operatorname{argmax}} P(x_1, x_2, \ldots, x_n \mid c)P(c)$$

- We denote the n features x_1, x_2, .., x_n

Height	Weight	Name	Sex
X	X	Joe	Female
X	X	Jim	Male
X	X	Joe	Male
X	X	Ted	Male
X	X	Mary	Female
X	X	Joe	Male
X	X	Carol	Female
x	X	Emily	Female

Naïve Bayes

$$P(x_1, x_2, \ldots, x_n \mid c)$$

- We make the naïve assumption of conditional independence
  - We assume the feature probabilities $P(x_i \mid c_j)$ are independent given a class c.
  - Whether one features occurs given a class and whether another feature occurs given a class are independent

$$P(x_1, \ldots, x_n \mid c) = P(x_1 \mid c) \bullet P(x_2 \mid c) \bullet P(x_3 \mid c) \bullet \ldots \bullet P(x_n \mid c)$$

Naïve Bayes

- Therefore, we can reformulate our Bayesian classification equation as:

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(x_1, x_2, \ldots, x_n \mid c) P(c)$$

$$c_{NB} = \underset{c \in C}{\operatorname{argmax}} P(c) \prod_{x \in X} P(x \mid c)$$

- Let's return to the previous example (with the multi-attribute dataset).
- Assume we pick an employee with the following attribute values **Name= "Joe", Weight = 13.2, Height = 6.1**.
  - Should we class this individual as a male or female? How would we work this out?

Assume we pick an employee with the following attribute values **Name= "Joe", Weight = 13.2, Height = 6.1**.

Should we class this individual as a male or female? How would we solve this?

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(x_1, x_2, \ldots, x_n \mid c) P(c)$$

$$c_{NB} = \underset{c \in C}{\operatorname{argmax}} P(c) \prod_{x \in X} P(x \mid c)$$

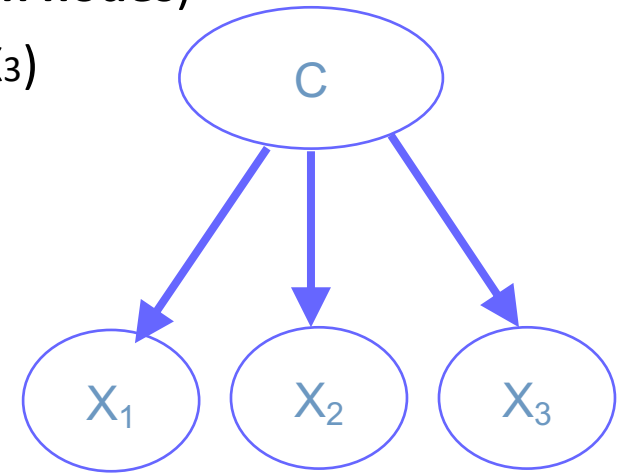
Height	Weight	Name	Sex
X	X	Joe	Female
X	X	Jim	Male
X	X	Joe	Male
X	X	Ted	Male
X	X	Mary	Female
X	X	Joe	Male
X	X	Carol	Female
x	X	Emily	Female

P(Female | Name= "Joe", Weight = 13.2, Height = 6.1) = P(Name = "Joe" | Female) * P(**Weight = 13.2** | Female) * P(Height = 6.1 | Female) * P(Female)

P(Male | Name= "Joe", Weight = 13.2, Height = 6.1) = P(Name = "Joe" | Male) * P(**Weight = 13.2** | Male) * P(Height = 6.1 | Male) * P(Male)

Naïve Bayes Model

- Simplest form of Bayesian classifier (assume **boolean nodes**)
  - A node for each feature in the domain (X_1, X_2, X_3)
  - C is a Boolean class node



- For each arc between the class variable and an evidence node we need a set of four probabilities. These are:
  - P(X1=true | C=true)
  - P(X1=true | C=false)
  - P(X1=false | C=true)
  - P(X1=false | C=false)

Training a Naïve Bayes Classifier

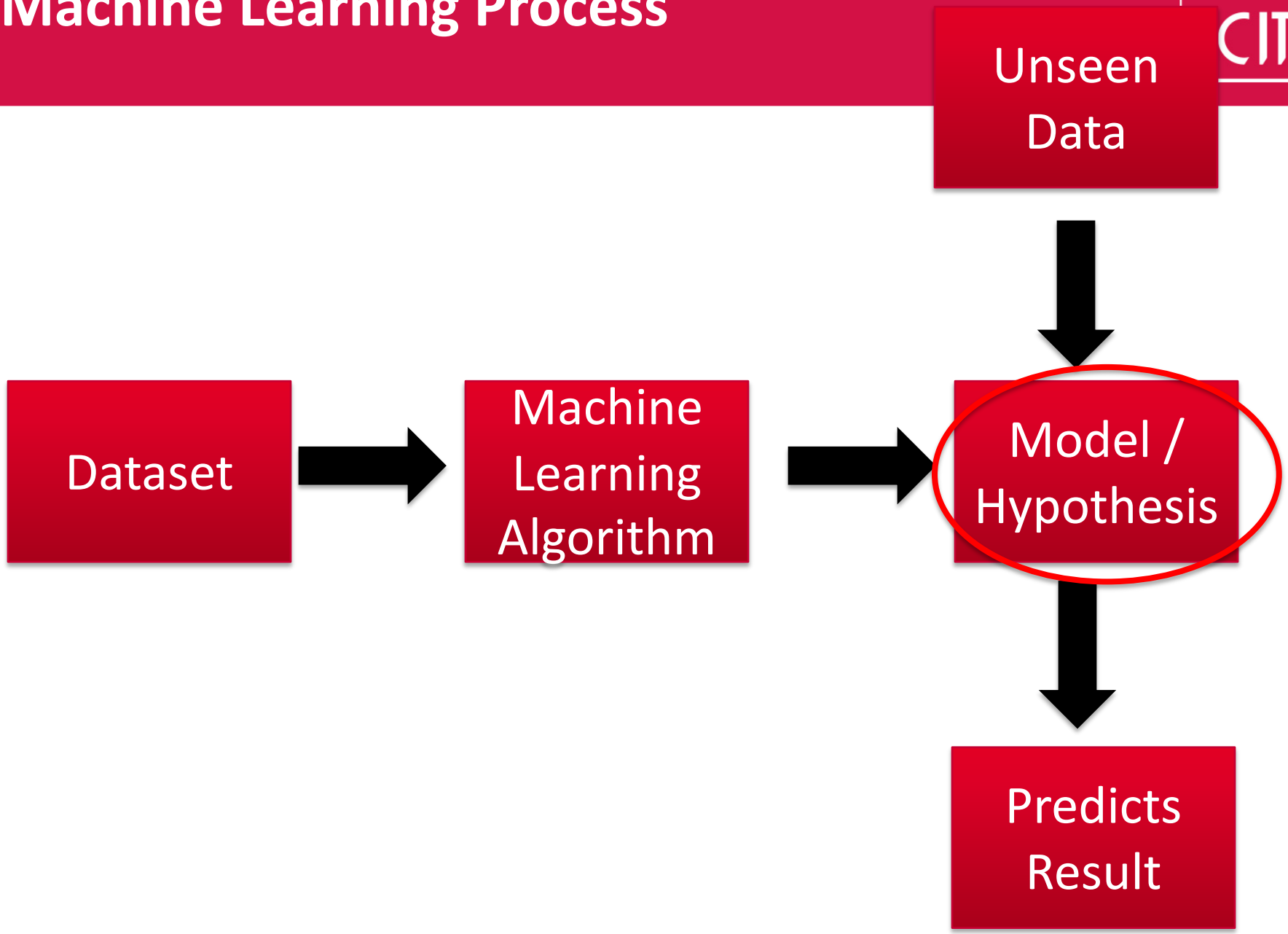
- We estimate conditional probabilities for each arc from the training data!

- Simple frequency estimates:
  - $\mathbf{P(X=x_1 \mid C=c_1) = N_{x1c1} / N_{c1}}$
  - N_{x1c1} = counts of cases where $X=x_1$ and $C=c_1$
  - N_{c1} = count of cases where $C=c_1$

P(name = “Joe” | class = male) =

How many of male class are called Joe / Total number male class

Machine Learning Process



Classification

Let us understand Naive Bayes with the help of an example

All possible weather combinations

- Summer, Monsoon, Winter
- Sunny, No Sun
- Windy, No Wind

Naïve Bayes Classifier Example (Weather Dataset)

In order to see the probability estimates in action we will look at the weather dataset. We will need to answer questions such as the following. Given the information I know, will I go playing tennis?

Outlook =sunny, Temp = cold, Humidity = high, Windy = true: **Play = ?**

Anyone for Tennis?

ID	Outlook	Temp	Humidity	Windy	Play?
A	sunny	hot	high	false	no
B	sunny	hot	high	true	no
C	overcast	hot	high	false	yes
D	rainy	mild	high	false	yes
E	rainy	cool	normal	false	yes
F	rainy	cool	normal	true	no
G	overcast	cool	normal	true	yes
H	sunny	mild	high	false	no
I	sunny	cool	normal	false	yes
J	rainy	mild	normal	false	yes
K	sunny	mild	normal	true	yes
L	overcast	mild	high	true	yes
M	overcast	hot	normal	false	yes
N	rainy	mild	high	true	no

Outlook =sunny, Temp = cold, Humidity = high, Windy = true: **Play = ?**

Naïve Bayes Classifier Example (Weather Dataset)

In order to see the probability estimates in action we will look at the weather dataset.

Outlook =sunny, Temp = cold, Humidity = high, Windy = true: Play = ?

To answer this question we will first need to work out a whole set of probabilities.

$$c_{NB} = \underset{c \in C}{\operatorname{argmax}} P(c) \prod_{x \in X} P(x \mid c)$$

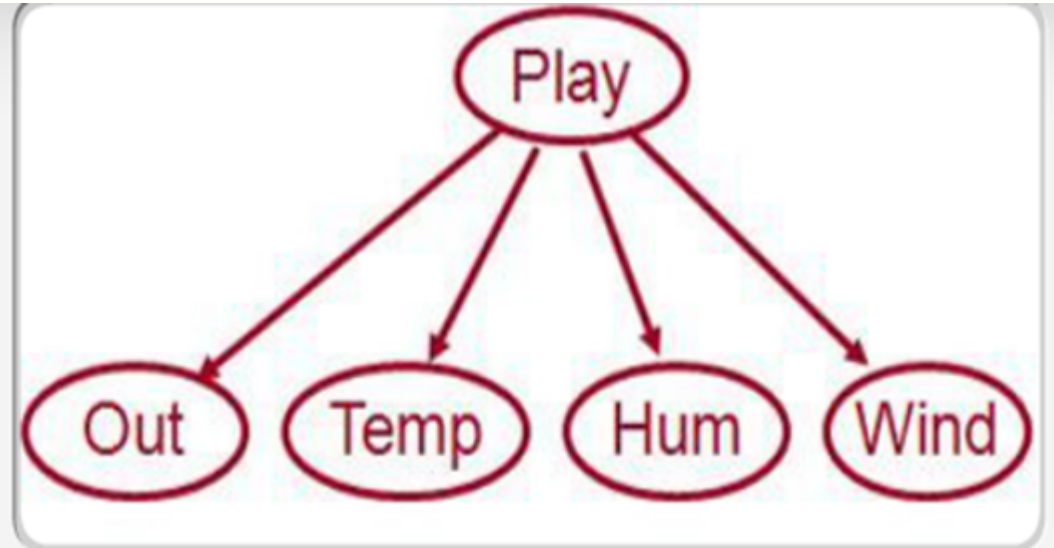
P(Play = y **| Outlook** = sunny, **Temp** = cold, **Humidity** = high, **Windy** = true**)** =

P(Play = y) * P(Outlook=s | Play=y)*P(Temp=c | Play=y)*P(Humidity = h | Play=y)*P(Windy=t | Play=y)

P(Play = n **| Outlook** = sunny, **Temp** = cold, **Humidity** = high, **Windy** = true**)** =

P(Play = n) * P(Outlook=s | Play=n)*P(Temp=c | Play=n)*P(Humidity = h | Play=n)*P(Windy=t | Play=n)

Naïve Bayes Classifier Example



P(Out = Sunny | Play = Y)

P(Out = Overcast | Play = Y)

P(Out = Rainy | Play = Y)

P(Temp = Hot | Play = Y)

P(Temp = Mild | Play = Y)

P(Temp = Cool | Play = Y)

P(Hum= High | Play = Y)

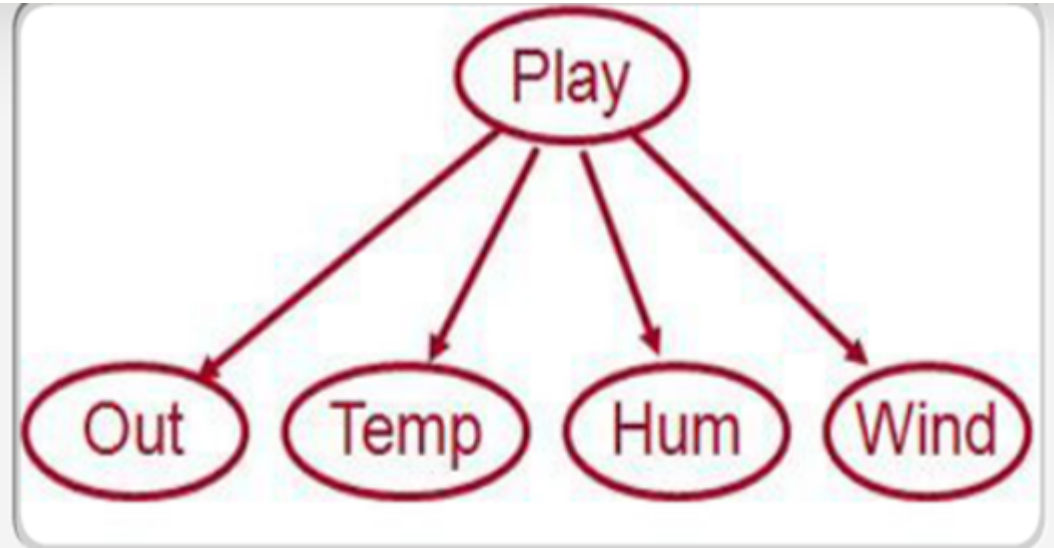
P(Hum= Normal | Play = Y)

P(Win= False | Play = Y)

P(Win= True | Play = Y)

Naïve Bayes Classifier Example

Conditional Probabilities also need to be worked out for Play = N



P(Out = Sunny | Play = N)

P(Out = Overcast | Play = N)

P(Out = Rainy | Play = N)

P(Temp = Hot | Play = N)

P(Temp = Mild | Play = N)

P(Temp = Cool | Play = N)

P(Hum= High | Play = N)

P(Hum= Normal | Play = N)

P(Win= False | Play = N)

P(Win= True | Play = N)

ID	Outlook	Temp	Humidity	Windy	Play?
A	sunny	hot	high	false	no
B	sunny	hot	high	true	no
C	overcast	hot	high	false	yes
D	rainy	mild	high	false	yes
E	rainy	cool	normal	false	yes
F	rainy	cool	normal	true	no
G	overcast	cool	normal	true	yes
H	sunny	mild	high	false	no
I	sunny	cool	normal	false	yes
J	rainy	mild	normal	false	yes
K	sunny	mild	normal	true	yes
L	overcast	mild	high	true	yes
M	overcast	hot	normal	false	yes
N	rainy	mild	high	true	no

- We will first look at calculating the probabilities needed for the class 'play' and the attribute '**windy**'. Lets work out the P(Wind = t | Play = y)

  - $P(X=x_1 \mid C=c_1) = N_{x1c1} / N_{c1}$
  - N_{x1c1} = counts of cases where $X=x_1$ and $C=c_1$
  - N_{c1} = count of cases where $C=c_1$

ID	Outlook	Temp	Humidity	Windy	Play?
A	sunny	hot	high	false	no
B	sunny	hot	high	true	no
C	overcast	hot	high	false	yes
D	rainy	mild	high	false	yes
E	rainy	cool	normal	false	yes
F	rainy	cool	normal	true	no
G	overcast	cool	normal	true	yes
H	sunny	mild	high	false	no
I	sunny	cool	normal	false	yes
J	rainy	mild	normal	false	yes
K	sunny	mild	normal	true	yes
L	overcast	mild	high	true	yes
M	overcast	hot	normal	false	yes
N	rainy	mild	high	true	no

- We will first look at calculating the probabilities needed for the class 'play' and the attribute '**windy**'. Lets work out the P(Wind = t | Play = y)

As we can see from the image, there are 9 cases where **Play=y**. In 3 of these, **Wind=t**, and in the other 6, **Wind=f**. Therefore, the probability of **Wind=t** given that **Play=y** is 3/9, according to these observations.

P(Wind=t | Play=y) = 3/9
P(Wind=f | Play=y) = 6/9

ID	Outlook	Temp	Humidity	Windy	Play?
A	sunny	hot	high	false	no
B	sunny	hot	high	true	no
C	overcast	hot	high	false	yes
D	rainy	mild	high	false	yes
E	rainy	cool	normal	false	yes
F	rainy	cool	normal	true	no
G	overcast	cool	normal	true	yes
H	sunny	mild	high	false	no
I	sunny	cool	normal	false	yes
J	rainy	mild	normal	false	yes
K	sunny	mild	normal	true	yes
L	overcast	mild	high	true	yes
M	overcast	hot	normal	false	yes
N	rainy	mild	high	true	no

Next work out P(Wind = t | Play = n) and P(Wind = f | Play = n)

As we can see from the image, there are 5 cases where **Play=n**. In 3 of these, **Wind=t**, and in the other 2, **Wind=f**. Therefore:

P(Wind=t | Play=n) = 3/5
P(Wind=f | Play=n) = 2/5

We now have all four probabilities we need for the arc between play and windy. Next we need to apply the same method to calculate the probabilities for each of the other arcs.

Calculate probabilities for Humidity Attribute

Start with:

P(Humidity =high| Play = yes

ID	Outlook	Temp	Humidity	Windy	Play?
A	sunny	hot	high	false	no
B	sunny	hot	high	true	no
C	overcast	hot	high	false	yes
D	rainy	mild	high	false	yes
E	rainy	cool	normal	false	yes
F	rainy	cool	normal	true	no
G	overcast	cool	normal	true	yes
H	sunny	mild	high	false	no
I	sunny	cool	normal	false	yes
J	rainy	mild	normal	false	yes
K	sunny	mild	normal	true	yes
L	overcast	mild	high	true	yes
M	overcast	hot	normal	false	yes
N	rainy	mild	high	true	no

P(Humidity =high| Play = yes) = 3/9
P(Humidity =normal| Play = yes) = 6/9

P(Humidity =high| Play = no) = 4/5
P(Humidity =normal| Play = no) = 1/5

Naïve Bayes Example

Using the same approach we can calculate the probabilities associated with each of the other arcs.

Probabilities associated with play and evidence arc Outlook.

```
P(Outlook=s | Play=y) = 2/9  P(Outlook=s | Play=n) = 3/5
P(Outlook=o | Play=y) = 4/9  P(Outlook=o | Play=n) = 0/5
P(Outlook=r | Play=y) = 3/9  P(Outlook=r | Play=n) = 2/5
```

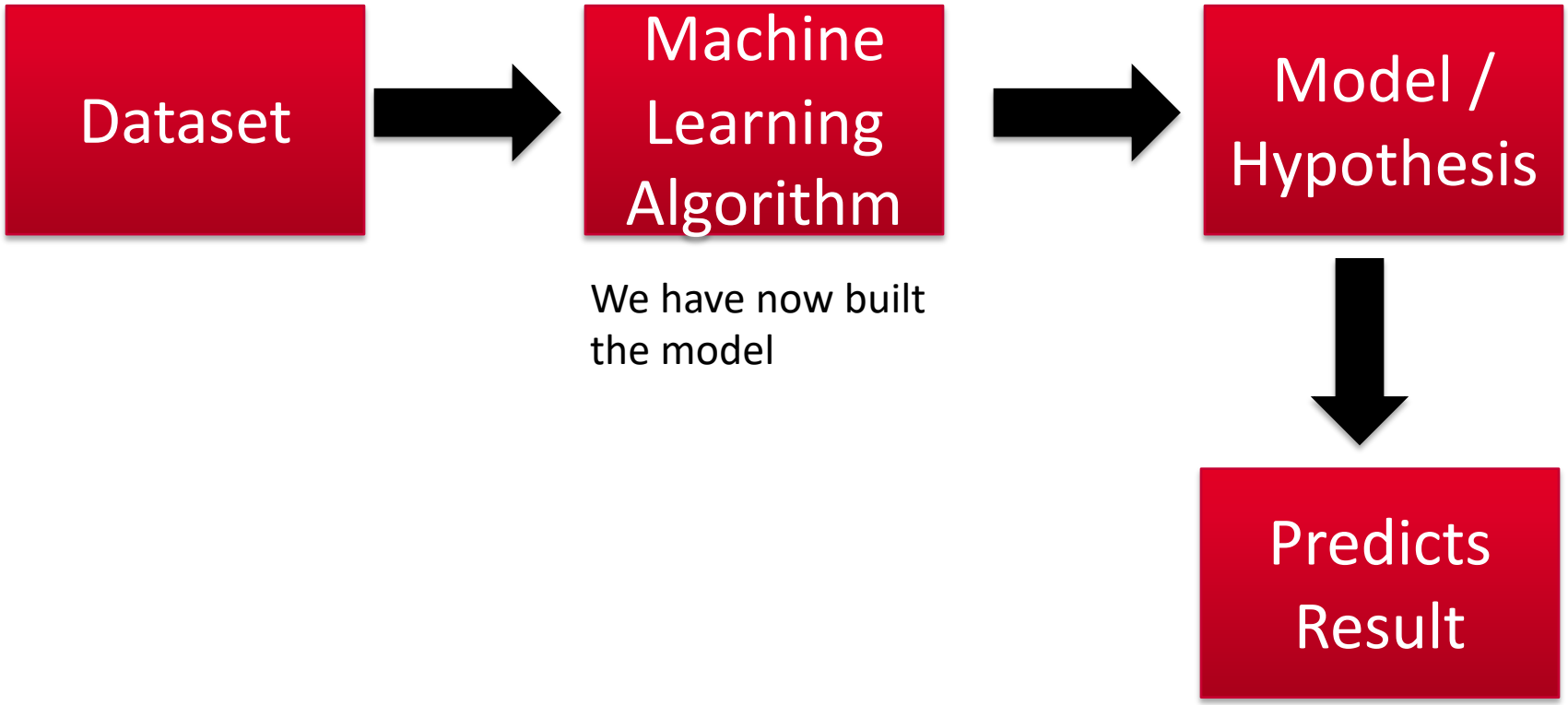
Probabilities associated with play and evidence arc Temp.

```
P(Temp=h | Play=y) = 2/9  P(Temp=h | Play=n) = 2/5
P(Temp=m | Play=y) = 4/9  P(Temp=m | Play=n) = 2/5
P(Temp=c | Play=y) = 3/9  P(Temp=c | Play=n) = 1/5
```

We will also need to calculate the prior probabilities. That is the probability that `play = y` and `play = n`. This is easily done as follows:

```
P(Play=y) = 9/14
P(Play=n) = 5/14
```

Machine Learning Process



Classify a New Instance

- *Will I play tennis under the following conditions:*
  - ***Outlooks = sunny, Temp = cool, Humidity = high, Windy = true, Play = ?***

Play is y or n. Evaluate probability of each given data.

P(Play = y | Outlook = s, Temp =c, Humidity = h, Wind = t) =
P(Play = y) * P(Outlook = s | Play = y) * P(Temp=c | Play = y) * P(Humidity= h | Play = y) * P(Wind = t | Play = y)
= 9/14 * 2/9 * 3/9 * 3/9 * 3/9 = **0.005291**

P(Play = n | Outlook = s, Temp =c, Humidity = h, Wind = t) =
P(Play = n) * P(Outlook = s | Play = n) * P(Temp=c | Play = n) * P(Humidity= h | Play = n) * P(Wind = t | Play = n)
= 5/14 * 3/5 * 1/5 * 4/5 *3/5 = **0.020571**

Normalise the Results

$$P(c_j)\prod_{x\in X} P(x \mid c)$$

P(Play = y | data) = 0.005291
P(Play = n | data) = 0.020571

(Why do above probabilities not add to 1?)

P(Play = y | data) = (0.005291*100) / (0.005291+ 0.020571) = 20.5%
P(Play = n | data) = (0.020571 *100) / (0.005291+ 0.020571) = 79.5%

Conclusion: more likely NOT to play tennis today.

From the calculations, it is seen that the probability of `Play=yes` is `20.5%`, whereas the probability of `Play=no` is `79.5%`. Selecting the outcome with the higher probability, the classification is that Play=no.

$$P(c_j)\prod_{x\in X} P(x \mid c)$$

Consider the following data instance:

Outlook =overcast, Temp = mild, Humidity = normal, Windy = false: Play = ?

```
P(Outlook=s | Play=y) = 2/9   P(Outlook=s | Play=n) = 3/5
P(Outlook=o | Play=y) = 4/9   P(Outlook=o | Play=n) = 0/5
P(Outlook=r  | Play=y) = 3/9  P(Outlook=r  | Play=n) = 2/5
```

```
P(Wind=t | Play=y) = 3/9    P(Wind=t | Play=n) = 3/5
P(Wind=f | Play=y) = 6/9    P(Wind=f | Play=n) = 2/5
```

```
P(Temp=h | Play=y) = 2/9   P(Temp=h  | Play=n) = 2/5
P(Temp=m | Play=y) = 4/9   P(Temp=m | Play=n) = 2/5
P(Temp=c  | Play=y) = 3/9  P(Temp=c  | Play=n) = 1/5
```

P(Humidity =high| Play = yes) = 3/9
P(Humidity =normal| Play = yes) = 6/9

P(Humidity =high| Play = no) = 4/5
P(Humidity =normal| Play = no) = 1/5

```
P(Play=y) = 9/14
P(Play=n) = 5/14
```

$$P(c_j)\prod_{x\in X} P(x \mid c)$$

Consider the following data instance:

Outlook =overcast, Temp = mild, Humidity = normal, Windy = false: Play = ?

```
P(Outlook=s | Play=y) = 2/9  P(Outlook=s | Play=n) = 3/5
P(Outlook=o | Play=y) = 4/9  P(Outlook=o | Play=n) = 0/5
P(Outlook=r  | Play=y) = 3/9 P(Outlook=r  | Play=n) = 2/5
```

Problem with Using Frequencies for Probability Calculations

- So far we estimated probabilities using the following:
  - P(X=x1 | C=c1) = $(N_{x1c1}) / (N_{c1})$
  - N_{x1c1} = counts of cases where attribute X=x1 and class C=c1
  - N_{c1} = count of cases where class C=c1

- What happens when a particular features does not occur for a given class. **In other words when $N_{x1c1} = 0$?**

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(x_1, x_2, \ldots, x_n \mid c) P(c)$$

$$c_{NB} = \underset{c \in C}{\operatorname{argmax}} P(c_j) \prod_{x \in X} P(x \mid c)$$

- If we multiply all the probabilities together the result of the entire expression becomes 0.

Avoiding Zeros

- To avoid the problem outlined on the previous slide we typically use +1 or laplace smoothing.

- Often some basic softening of the equation is performed. For example (**+1 smoothing**), $(N_{x1c1} + 1) / (N_{c1} + 2)$

- ***Laplace*** **Smoothing** (m-estimate) : $(N_{x1c1} + 1) / (N_{c1} + |X|)$

  - N_{x1c1} = counts of cases where X=x1 and C=c1
  - N_{c1} = count of cases where C=c1
  - $|X|$ = count of cases of X (number of features(attributes))

Avoiding Zeros

- Remember we worked out P(Outlook = o | Play = n) = 0/5

- +1 smoothing $(N_{x1c1} + 1) / (N_{c1} + 2)$
- If we use +1 smoothing **P(Outlook = o | Play = n) would be (0+1)/(5+2) = 1/7**

- *Laplace* Smoothing (m-estimate) : $(N_{x1c1} + 1) / (N_{c1} + |X|)$
- **P(Outlook = o | Play = n) would be (0+1)/(5+4) = 1/9**

  - Remember |X| is the number of attributes

Problems with Probabilities for Naïve Bayes

$$P(c_j)\prod_{x\in X} P(x \mid c)$$

Can you see any computational problem that may occur from this formula? Hint: What might happen if you have a large amount of features?

The computation issue is that of underflow: doing too many multiplications of small numbers.

When we go to calculate the product p(w0|ci)p(w1|ci)p(w2|ci)...p(wN|ci) and many of these numbers are very small, we'll get underflow (multiply many small numbers in a programming language and eventually it rounds off to 0.)

Using Log

- The most common solution to the problem on the previous slide is to calculate the logarithm of this product.

- Doing this allows us to avoid the underflow or round-off error problem. Why? Because we end up adding the individual probabilities rather than multiplying them (log(xy) = log(x) + log(y))

- In other word we now get the log of the Bayes equation

$$\mathbf{log}(P(c)\prod_{x\in X}P(x|c))$$

  - We now use

$$\mathbf{log}\,P(c)+\sum_{x\in X}\mathbf{log}\,P(x\mid c)$$

- Word of caution about Naïve Bayes probability estimates.

Contents

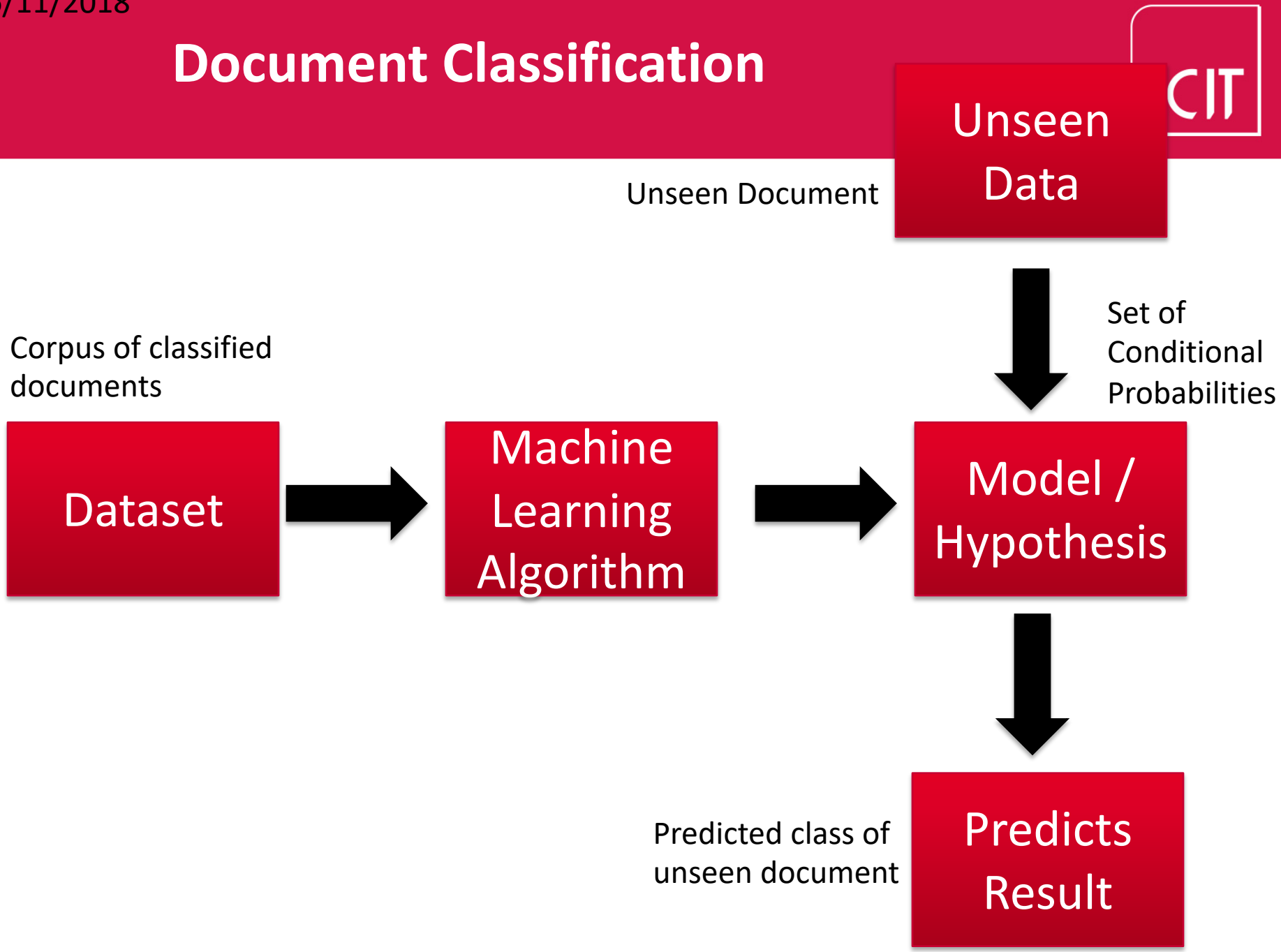
Document Classification

- Naive Bayes classifiers are among the most successful known algorithms for learning to classify text documents.

- In document classification **each word is treated as an feature/attribute**.

- Document Classification
  - Spam Filtration
  - Author Identification
  - Age or gender identification
  - Sentiment Analysis (movie review, product reviews, important applications)

Document Classification



Document Classification

- A Bayesian classifier will typically either adopt a **bag** of words or **set** of words approach.
  - (Bernoulli model) **Set of words**, counts the number of documents where a word occurs
  - (Multinomial Model) **Bag of words**, counts the total occurrences of a word across all documents.

- When classifying a test document, the Bernoulli model uses **binary occurrence** information, ignoring the number of occurrences of a word in a document , whereas the multinomial model keeps track of multiple occurrences in a single document.

- The models also differ in how **non-occurring terms** are used in classification. They do not affect the classification decision in the multinomial model; but in the Bernoulli model the probability of non-occurrence is factored in when computing probabilities

Calculating Prior Probabilities

$$c_{MAP} = argmax_{c \in C} \log P(c) + \sum_{w \in W} \log P(w \mid c)$$

- The first thing we need to do is calculate the prior probabilities (that is, the probability of the class). This calculation is the same for both multinomial and binomial.

$$P(c) = \frac{\text{Number of documents of class c}}{Total\ Number\ of\ documents}$$

Naïve Bayes - Multinomial Model

$$c_{MAP} = argmax_{c \in C} \log P(c) + \sum_{w \in W} \log P(w \mid c)$$

- Calculation of the probabilities in the multinomial model as are follows (notice we use laplace smoothing here):

- $P(w \mid c) = \frac{count(w,c)+1}{count(c)+|V|}$

count(w, c) is the number of occurrences of the word w in all documents of class c.

count(c) The total number of words in all documents of class c (including duplicates).

|V| The number of words in the vocabulary


CIT

Exercise

- The table below shows a very simple training set containing 4 documents and the words contained within those documents.
- It also contains the class of each of the document.
- Objective is to classify the new Test as either class Comp or class Politics.
  - We will use **laplace** for calculating the Multinomial probabilities
  - We will use simple **+1 smoothing** for calculating the Bernoulli probabilities

	Doc	Words	Class
Training	1	Cloud Java Cloud	Comp
	2	Cloud Cloud Spring	Comp
	3	Cloud Software	Comp
	4	Referendum Software Election	Politics
Test	5	Java Software Java Election	?

	Doc	Words	Class
Training	1	Cloud Java Cloud	Comp
	2	Cloud Cloud Spring	Comp
	3	Cloud Software	Comp
	4	Referendum Software Election	Politics
Test	5	Java Software Java Election	?

$$c_{MAP} = argmax_{c \in C} \log P(c) + \sum_{w \in W} \log P(w \mid c)$$

	Doc	Words	Class
Training	1	Cloud Java Cloud	Comp
	2	Cloud Cloud Spring	Comp
	3	Cloud Software	Comp
	4	Referendum Software Election	Politics
Test	5	Java Software Java Election	?

$$c_{MAP} = argmax_{c \in C} \log P(c) + \sum_{w \in W} \log P(w \mid c)$$

$$P(Comp) = \frac{3}{4}$$

$$P(Politics) = \frac{1}{4}$$

	Doc	Words	Class
Training	1	Cloud Java Cloud	Comp
	2	Cloud Cloud Spring	Comp
	3	Cloud Software	Comp
	4	Referendum Software Election	Politics
Test	5	Java Software Java Election	?

$$c_{MAP} = argmax_{c \in C} \log P(c) + \sum_{w \in W} \log P(w \mid c)$$

	Doc	Words	Class
Training	1	Cloud Java Cloud	Comp
	2	Cloud Cloud Spring	Comp
	3	Cloud Software Java	Comp
	4	Referendum Software Election	Politics
Test	5	Java Software Java Election	?

$$P(w \mid c) = \frac{count(w, c) + 1}{count(c) + |V|}$$

Notice we use Laplace smoothing here

	Doc	Words	Class
Training	1	Cloud Java Cloud	Comp
	2	Cloud Cloud Spring	Comp
	3	Cloud Software Java	Comp
	4	Referendum Software Election	Politics
Test	5	Java Software Java Election	?

$$P(w \mid c) = \frac{count(w,c)+1}{count(c)+|V|}$$

Notice we use Laplace smoothing here

$$P(Cloud \mid Comp) = \frac{5+1}{9+6}$$

$$P(Java \mid Comp) = \frac{2+1}{9+6}$$

$$P(Software \mid Comp) = \frac{1+1}{9+6}$$

$$P(Spring \mid Comp) = \frac{1+1}{9+6}$$

$$P(Referendum \mid Comp) = \frac{0+1}{9+6}$$

$$P(Election \mid Comp) = \frac{0+1}{9+6}$$

	Doc	Words	Class
Training	1	Cloud Java Cloud	Comp
	2	Cloud Cloud Spring	Comp
	3	Cloud Software Java	Comp
	4	Referendum Software Election	Politics
Test	5	Java Software Java Election	?

$$P(w \mid c) = \frac{count(w,c)+1}{count(c)+|V|}$$

Notice we use Laplace smoothing here

$$P(Cloud \mid Politics) = \frac{0+1}{3+6}$$

$$P(Java \mid Politics) = \frac{0+1}{3+6}$$

$$P(Software \mid Politics) = \frac{1+1}{3+6}$$

$$P(Spring \mid Politics) = \frac{0+1}{3+6}$$

$$P(Referendum \mid Politics) = \frac{1+1}{3+6}$$

$$P(Election \mid Politics) = \frac{1+1}{3+6}$$

	Doc	Words	Class
Test	5	Java Software Java Election	?

$$P(Cloud \mid Comp) = \frac{6}{15}$$

$$P(Java \mid Comp) = \frac{3}{15}$$

$$P(Software \mid Comp) = \frac{2}{15}$$

$$P(Spring \mid Comp) = \frac{2}{15}$$

$$P(Election \mid Comp) = \frac{1}{15}$$

$$P(Referendum \mid Comp) = \frac{1}{15}$$

$$P(Cloud \mid Politics) = \frac{1}{9}$$

$$P(Java \mid Politics) = \frac{1}{9}$$

$$P(Software \mid Politics) = \frac{2}{9}$$

$$P(Spring \mid Politics) = \frac{1}{9}$$

$$P(Election \mid Politics) = \frac{2}{9}$$

$$P(Referendum \mid Politics) = \frac{2}{9}$$

$$P(Comp) = \frac{3}{4} \qquad P(Politics) = \frac{1}{4}$$

	Doc	Words	Class
Test	5	Java Software Java Election	?

$$c_{MAP} = argmax_{c \in C} \log P(c) + \sum_{w \in W} \log P(w \mid c)$$

	Doc	Words	Class
Test	5	Java Software Java Election	?

$$P(c \mid W) = \log P(c) + \sum_{w \in W} \log P(w \mid c)$$

$P(Comp \mid Test)$ = log(3/4) + log(3/15) + log(2/15) + log (3/15) + log (1/15) = **-3.57**

$P(Politics \mid Test)$ = log(1/4) + log(1/9) + log(2/9)+log(1/9)+log(2/9)= **-3.81**

Classify the document as being of class Comp

Naïve Bayes: Text Classification for Multinomial

Examples are a set of training documents.

V is the set of classes (ex. Spam / NotSpam)

Learn_naive_Bayes_text(*Examples*, *V*)

1. collect all words that occur in *Examples*
   Vocabulary ← all distinct words in *Examples*
2. calculate the required $P(v_j)$ and $P(w_k|v_j)$ probability terms
   For each target value v_j in V do
   - $docs_j \leftarrow$ subset of *Examples* for which the target value is v_j
   - $P(v_j) \leftarrow \frac{|docs_j|}{|Examples|}$
   - $Text_j \leftarrow$ a single document created by concatenating all members of $docs_j$
   - $n \leftarrow$ total number of words in $Text_j$ (counting duplicate words multiple times)
   - for each word w_k in *Vocabulary*
     - $n_k \leftarrow$ number of times word w_k occurs in $Text_j$
     - $P(w_k|v_j) \leftarrow \frac{n_k+1}{n+|Vocabulary|}$

Document Classification

- Classify_naive_Bayes_text(newDoc)
  - allWords <- all words in Doc that are found in Vocabulary (note the same word may appear multiple times)
  - Return V_{NB}, where:

$$V_{NB} = \underset{v_j \in V}{\arg max} \quad logP(v_j) + \sum_{x \in allWords} \log P(x \mid v_j)$$

Bernoulli Model

- We will next look at the application of the Bernoulli model to the same example. (Note we will use **plus 1 smoothing**)

- $$P(w \mid c) = \frac{Number\ of\ Docments\ of\ class\ c\ containing\ word\ w+1}{Total\ number\ of\ documents\ of\ class\ c+2}$$

	Doc	Words	Class
Training	1	Cloud Java Cloud	Comp
	2	Cloud Cloud Spring	Comp
	3	Cloud Software Java	Comp
	4	Referendum Software Election	Politics
Test	5	Java Software Java Election	?

In this example we will use +1 smoothing.

	Doc	Words	Class
Training	1	Cloud Java Cloud	Comp
	2	Cloud Cloud Spring	Comp
	3	Cloud Software Java	Comp
	4	Referendum Software Election	Politics
Test	5	Java Software Java Election	?

Notice we use +1 smoothing here

$$P(Cloud \mid Comp) = \frac{3+1}{3+2}$$

$$P(Java \mid Comp) = \frac{2+1}{3+2}$$

$$P(Software \mid Comp) = \frac{1+1}{3+2}$$

$$P(Spring \mid Comp) = \frac{1+1}{3+2}$$

$$P(Referendum \mid Comp) = \frac{0+1}{3+2}$$

$$P(Election \mid Comp) = \frac{0+1}{3+2}$$

	Doc	Words	Class
Training	1	Cloud Java Cloud	Comp
	2	Cloud Cloud Spring	Comp
	3	Cloud Software Java	Comp
	4	Referendum Software Election	Politics
Test	5	Java Software Java Election	?

Notice we use +1 smoothing here

$$P(Cloud \mid Politics) = \frac{0+1}{1+2}$$

$$P(Java \mid Politics) = \frac{0+1}{1+2}$$

$$P(Software \mid Politics) = \frac{1+1}{1+2}$$

$$P(Spring \mid Politics) = \frac{0+1}{1+2}$$

$$P(Referendum \mid Politics) = \frac{1+1}{1+2}$$

$$P(Election \mid Politics) = \frac{1+1}{1+2}$$

	Doc	Words	Class
Test	5	Java Software Java Election	?

$$P(Cloud \mid Comp) = \frac{4}{5}$$

$$P(Java \mid Comp) = \frac{3}{5}$$

$$P(Software \mid Comp) = \frac{2}{5}$$

$$P(Spring \mid Comp) = \frac{2}{5}$$

$$P(Election \mid Comp) = \frac{1}{5}$$

$$P(Referendum \mid Comp) = \frac{1}{5}$$

$$P(Cloud \mid Politics) = \frac{1}{3}$$

$$P(Java \mid Politics) = \frac{1}{3}$$

$$P(Software \mid Politics) = \frac{2}{3}$$

$$P(Spring \mid Politics) = \frac{1}{3}$$

$$P(Election \mid Politics) = \frac{2}{3}$$

$$P(Referendum \mid Politics) = \frac{2}{3}$$

	Doc	Words	Class
Test	5	Java Software Java Election	?

$$P(c \mid W) = \log P(c) + \sum_{w \in W} \log P(w \mid c)$$

In Bernoulli we go through every word in the **vocabulary** and we incorporate the probability of the word **occurring** and the word **not occurring** given the class.

$P(Comp \mid Test)$ = log(3/4) + log(1-(4/5))+log(3/5)+log(2/5)+log(1-(2/5))+log(1/5)+log(1-(1/5)) = -2.46

$P(Politics \mid Test)$ = log(1/4) + log(1-(1/3))+log(1/3)+log(2/3)+log(1-(1/3))+log(2/3)+log(1-(2/3)) = -2.26

Classify the document as being of class Politics

Bernoulli v's Multinomial Model

- Empirical evaluations tend to show that the multinomial model typically outperforms the Bernoulli model as the vocabulary size increases in size.

- Please note that this is not always the case and it can be dependent on the data you use and the appropriate choice of features (pre-processing steps such as stop word removal etc.).

- In practice you should evaluate both a multinomial and Bernoulli approach to your text classification problem.

Twitter Sentiment Analysis using Bayes

- Performing sentiment analysis on Twitter is very challenging
  - Tweets are short messages, restricted to **140** characters in length.
  - Due to the nature of this microblogging service (quick and short messages), people use **acronyms**, make spelling mistakes, use **emoticons** and other characters that express special meanings
  - They are prone to **negation**, **idioms** and **sarcasm**.
  - All of which make accurate sentiment analysis much more difficult to achieve.

Pre-processing in Document Classification

- A range of pre-processing steps are often taken to clean the dataset. Basic steps include removal of punctuation and lower-casing all words. The objective of many of these techniques is reducing the number of features (words) in the dataset.

- Other techniques include:

- Stemming and Lemmatization
- Negation
- Emoticon substitution and Stop-word removal
- Acronym Replacement
- N-Grams

Strengths of Naïve Bayes

- Training Set Size and Speed
  - Naïve Bayes is a **very fast algorithm**
    - The process of calculating the probabilities is the only potentially time consuming component.
  - Another advantage of Naïve Bayes is that it is a probabilistic classifier so unlike many other algorithms it provides **some degree of certainty** in it's conclusions.
    - For example, we may only wish to classify the polarity of a tweet if we are more than 75% confident that the tweet is positive or negative (see previous slides).

Strengths of Naïve Bayes

- Naïve Bayes is **less sensitive to irrelevant features**...
  - Suppose we are trying to classify a persons gender based on several features, including eye colour (Of course, eye colour is completely irrelevant to a persons gender)
  - How would Naïve Bayes deal with such an irrelevant attribute.

Strengths of Naïve Bayes

- Naïve Bayes is **less sensitive to irrelevant features**...
  - Suppose we are trying to classify a persons gender based on several features, including eye colour (Of course, eye colour is completely irrelevant to a persons gender)
  - How would Naïve Bayes deal with such an irrelevant attribute.

p(eye = brown | female) * p(wears_dress= yes | female) * …..

p(eye = brown | male) * p(wears_dress= yes | male) * …..

p(eye = brown | female) * p(wears_dress= yes | female) * …..
=> 5,000/10000 * 9,975/10000

p(eye = brown | male) * p(wears_dress= yes | male) * …..
=> 5000/10000 * 25/10000

Weakness of Naïve Bayes

- The "Naive" attribute comes from the fact that the model **assumes that all features are fully independent** given the class, which in real problems they almost never are.

- In practice this approach still works reasonably well for many real-world problems.

- However, we can adopt a more realistic approach that will incorporate certain dependencies amongst the variables in our domain using Bayesian Networks.

Discussion

CORK INSTITUTE OF TECHNOLOGY
INSTITIÚID TEICNEOLAÍOCHTA CHORCAÍ

Computer Science

http://www.cit.ie

Thank you

Haithem. afli@cit.ie

@AfliHaithem