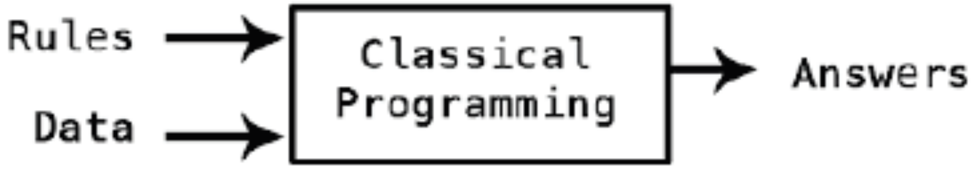
Predictive Models

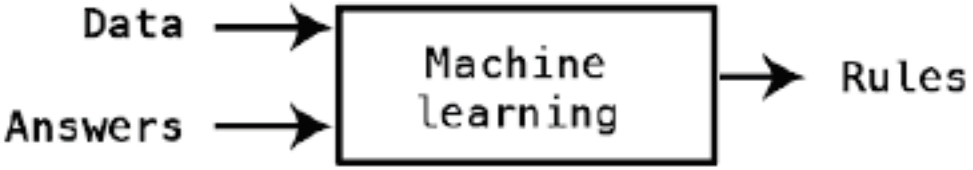
Fátima Rodrigues
mfc@isep.ipp.pt
Departamento de Engenharia Informática (DEI/ISEP)

Machine Learning a new Programming Paradigm

- In classical programming, humans input rules (a program) and data to be processed according to these rules, and out come answers



- With machine learning, humans input data as well as the answers expected from the data, and out come the rules. These rules can then be applied to new data to produce original answers



Prediction

- Prediction (forecasting) is the ability to anticipate the future

- Prediction is possible if we assume that there is some regularity in what we observe, i.e. if the observed events are not random

Prediction Models

- Objective: Fit data to a model
- Potential Result: Higher-level meta information that may not be obvious when looking at raw data

Example prediction model:

Medical Diagnosis

- Given an historical record containing the symptoms observed in several patients and the respective diagnosis
- Try to forecast the correct diagnosis for a new patient for which we know the symptoms

Entities involved in Predictive Modelling

Descriptors of the observation

- Set of variables that describe the properties (features, attributes) of the cases in the data set

Target variable

- What we want to predict/conclude regards the observations

Supervised Learning

- Prediction methods are commonly referred to as supervised learning. Supervised methods are thought to attempt the discovery of the relationships between input attributes and a target attribute

Entities involved in Predictive Modelling

The goal is to obtain an approximation of the function

$$Y = f(X_1, X_2, .., X_p)$$

where

Y is the target variable

$X_1, X_2, .., X_p$ are the variables describing the characteristics of the cases

It is assumed that Y is a variable whose values depend on the values of the variables which describe the cases. We just do not know how!

The goal of the modelling techniques is thus to obtain a good approximation of the unknown function f( )

Some working definitions….

- **Concepts**: kinds of things that can be learned
  - Aim: intelligible and operational concept description
  - Example: the relation between patient characteristics and the probability to be diabetic
- **Instances**: the individual, independent examples of a concept
  - Example: a patient, candidate drug etc.
- **Attributes**: measuring aspects of an instance
  - Example: age, weight, lab tests, microarray data etc.

How are the Models Used?

Predictive models have two main uses:

- **Prediction**

  use the obtained models to make predictions regards the target variable of new cases given their descriptors

- **Comprehensibility**

  use the models to better understand which are the factors that influence the conclusions

Types of Prediction Problems

Depending on the type of the target variable (Y) there are two different types of prediction models:

1. **Classification Problems** - the target variable Y is nominal e.g. medical diagnosis - given the symptoms of a patient try to predict the diagnosis

2. **Regression Problems** - the target variable Y is numeric e.g. forecast the market value of a certain asset given its characteristics

Types of Prediction Models

There are many techniques that can be used to obtain prediction models based on a data set. Independently of the pros and cons of each alternative, all have some key characteristics:

1. They assume a certain functional form for the unknown function f()
2. Given this assumed form the methods try to obtain the best possible model based on:
   - the given data set
   - a certain preference criterion that allows comparing the different alternative model variants

- The most common preference criteria are related with the minimization of the prediction error of the obtained models

Bias

Important choices made in a prediction model are:

- Representation language
  The language chosen to represent the patterns or models

- Search method
  The order in which the space is searched

- Model pruning method
  The way overfitting to the training data is avoided

This means, each prediction scheme involves

- Language bias
- Search bias
- Overfitting-avoidance bias

Search Bias

- An exhaustive search over the search space is computationally expensive
- Search is speeded up by using heuristics, but by definition heuristics cannot guarantee optimum patterns or models
- Complex search strategies possible
  - those that pursue several alternatives in parallel
  - those that allow backtracking

- A high-level search bias
  - General-to-specific: start with a root node and grow the decision tree to fit the specific data
  - Specific-to-general: choose specific examples in each class and then generalize the class by including k-nearest neighbour examples

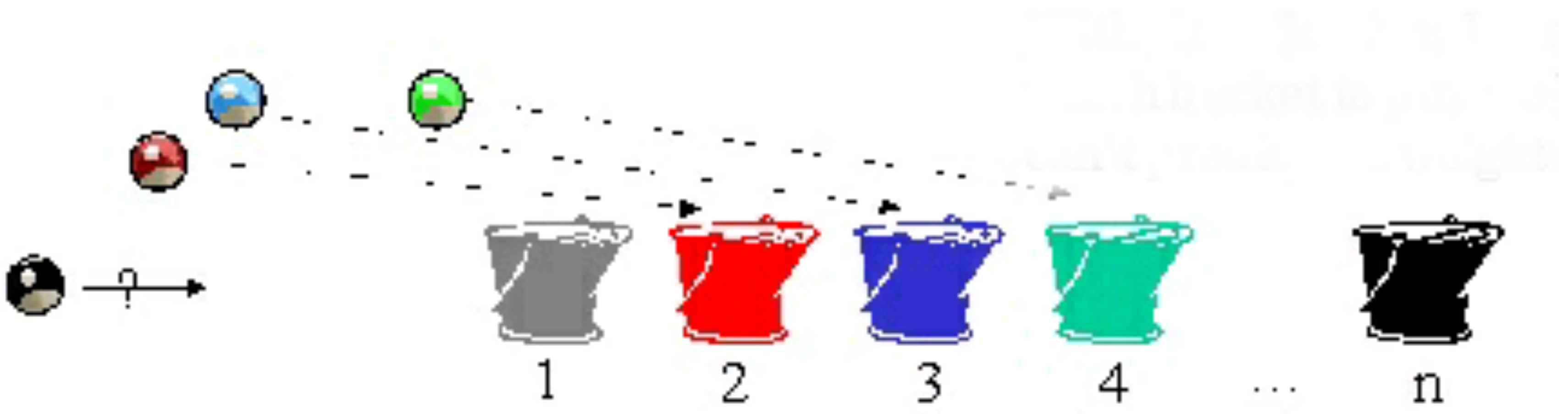
Overfitting-avoidance bias

- We want to search for ‘best’ patterns and models
- Simple models are the best
- Two strategies
  - Start with the simplest model and stop building model when it starts to become complex
  - Start with a complex model and prune it to make it simpler
- Each strategy biases search in a different way
- Biases are unavoidable in practice
  - Each data mining scheme might involve a configuration of biases
  - These biases may serve some problems well
- There is no universal best learning scheme!

Classification

Classification

It is a learning function that divides (or classifies) the data according to a predefined number of classes



The goal of classification is to organize and distribute data in different classes

- Classification: prediction of nominal discrete values
- Regression: Prediction of continuous values

Classification - Definition

Given a database $D = \{t_1, t_2, ..., t_n\}$

A set of classes $C = \{C_1, ..., C_m\}$

The goal of classification is to define a mapping $f : D \rightarrow C$ where each t_i is assigned to a class C_j

The database D is divided in m equivalence classes

Regression is similar, but can be regarded as having an infinite number of classes

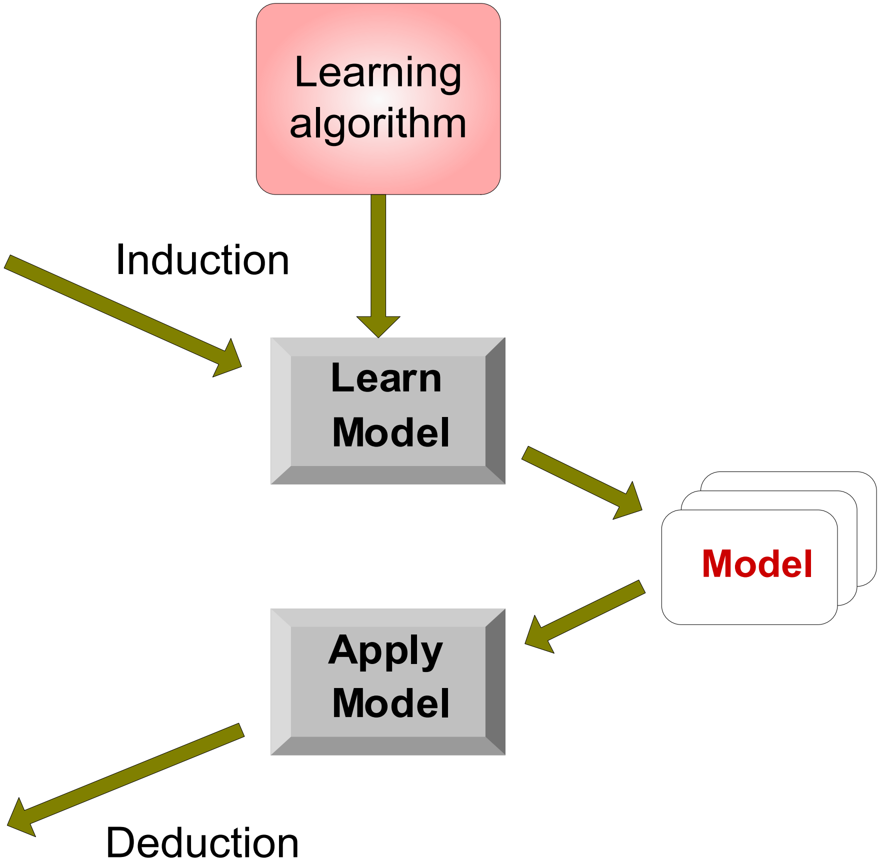
Applications

- Predicting tumor cells as benign or malignant
- Classifying credit card transactions as legitimate or fraudulent
- Classifying secondary structures of proteins
- Categorizing news stories as finance, weather, entertainment, sports, etc.

Classification - Illustration

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set



Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

Classification Techniques

- *K*-nearest neighbours
- Decision tree based methods
- Neural Networks
- Naïve Bayes and Bayesian Belief Networks
- Support Vector Machines
- ….

K-Nearest Neighbors

Instance-based Learning

- One way of solving tasks of approximating discrete or real valued target functions
- Have training examples: (x_n, f(x_n)), n=1..N
- Key idea:
  - just store the training examples
  - when a test example is given then find the closest matches

Nearest Neighbour Rule

- 1-Nearest neighbour:

  Given a query instance x_q

  - first locate the nearest training example x_n
  - then $f(x_q)$ <- $f(x_n)$

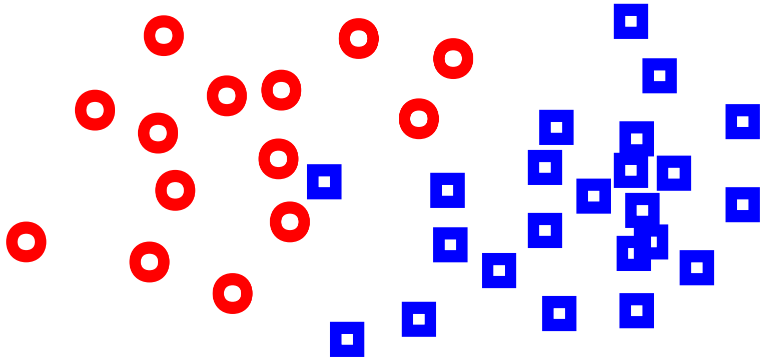
- K-Nearest neighbour:

  Given a query instance x_q

  - first locate the k nearest training examples
  - if discrete value target function then take vote among its k nearest neighbours
  - if real value target function then take the mean of the f values of the k nearest neighbours

$$f(x_q) := \frac{\sum_{i=1}^{k} f(x_i)}{k}$$

Nearest Neighbour Rule

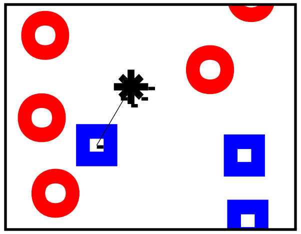


Non-parametric pattern classification

Consider a two class problem where each sample consists of two measurements (*x*,*y*).

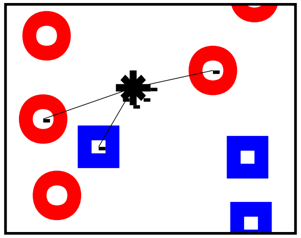
For a given query point q, assign the class of the nearest neighbour

k = 1



Compute the *k* nearest neighbours and assign the class by majority vote

k = 3



Distance between examples

- It is necessary a measure of distance in order to know who are the neighbours
- If the dataset has *T* attributes for the learning problem. Then one example point ***x*** has elements $x_t \in \Re$, *t=1,…T*
- The distance between two points $\boldsymbol{x_i}\,\boldsymbol{x_j}$ is often defined as the Euclidean distance:

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{t=1}^{T} [x_{ti} - x_{tj}]^2}$$

Characteristics of Instance-based Learning

- An instance-based learner is a *lazy-learner* and does all the work when the test example is presented. This is opposed to so-called *model-based learners*, which build a parameterised compact model of the target

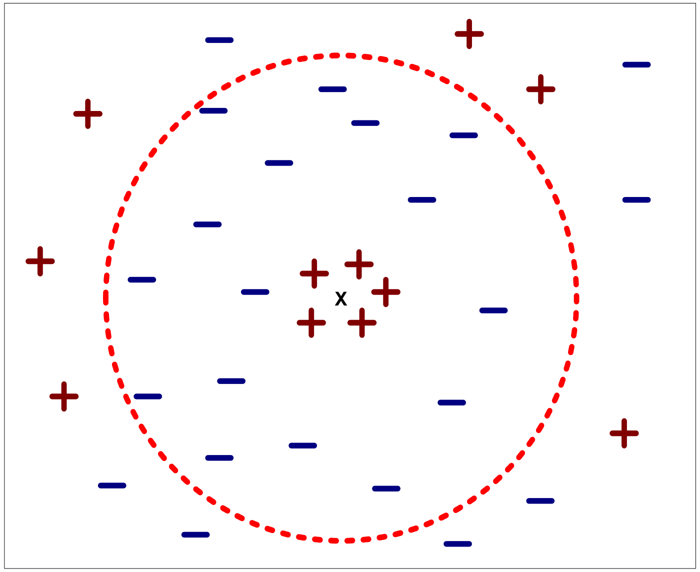
- It produces a *local* approximation to the target function (*different* with each test instance)

When to consider Nearest Neighbour algorithms?

- Instances map to points in $\Re^n$
- Not more then say 20 attributes per instance
- Lots of training data
- Advantages:
  - Training is very fast
  - Can learn complex target functions
  - Don’t lose information
- Disadvantages:
  - ? (will see them shortly…)

Choosing the value of k

- If k is too small, sensitive to noise points
- If k is too large, neighborhood may include points from other classes



- Frequent values are 3, 5 and 7 - Odd numbers to avoid draws!
- It can be estimated experimentally:
  - Global estimation searches for the ideal k for a given data set
  - Local estimation methods try to estimate the ideal k for each test case (computationally very demanding!)

Difficulties with k-nearest neighbour algorithms

- **Expensive**
  - To determine the nearest neighbour of a test instance t, must compute the distance to all *N* training examples

- **Storage Requirements**
  - Must store all training data

- **High Dimensional Data**
  - required amount of training data increases exponentially with dimension
  - computational cost also increases dramatically
  - There may be irrelevant attributes amongst the attributes – curse of dimensionality

K-Nearest Neighbors - Summary

- The k-nearest neighbors are known as lazy learners as they do not learn any model of the data - learning consists simply in storing the training data
- They do not make any assumption on the unknown functional form we are trying to approximate, which means that with sufficient data they are applicable to any problem
- They usually achieve good results but...
- They require a proper distance metric to be defined - issues like data scaling, irrelevant variables, unknown values, etc., may have a strong impact on their performance
- They have fast training time, but slow prediction time

Naive Bayes Classifier

Probabilistic Learning

- Probabilistic methods are methods concerned with describing uncertainty. They use data on past events to extrapolate future events

- Naive Bayes uses principles of probability for classification
- Naive Bayes uses data about prior events to estimate the probability of future events – requires **Priori Probabilities**

- The Bayesian theory makes two assumptions:
  - The attributes are equally important
  - The attributes are statistically independent (knowing its class)

- Assumptions of independence are not usually correct
- However ... the Bayesian learning scheme it works well !

Probability Theory

The main tool of uncertainty is the probability theory, which assigns to each sentence a numerical degree of belief between *0* and *1*. It provides a way of summarizing the uncertainty

Probability

The probability of an event can be estimated from observed data by dividing the number of trials in which an event occurred by the total number of trials.

Example:

If 10 out of 50 email messages are spam, then the probability of spam can be estimated as P(spam) = 20%

P(ham) = 1 – 0.20 = 80%

This works because the events spam and ham are **mutually exclusive and exhaustive**

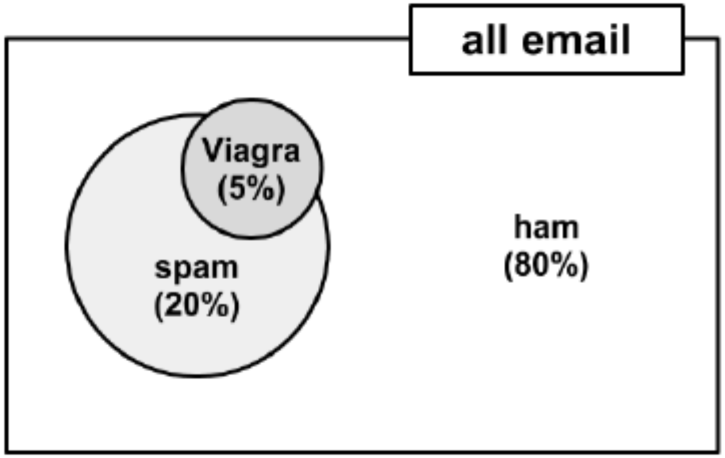
Joint probability

Often, we are interested in monitoring **several non-mutually exclusive events** for the same trial

If some events occur with the event of interest, we may be able to use them to make predictions

Example:

Consider, for instance, a second event based on the outcome that the email message contains the word Viagra



P(spam) = 20%
P(ham) = 80%
P(viagra) = 5%

Joint probability

We want to estimate the probability of both P(spam) and P(Viagra) occurring, which can be written as P(spam ∩ Viagra)

Calculating P(spam ∩ Viagra) depends on the joint probability of the two events, or how the probability of one event is related to the probability of the other

If the two events are totally unrelated, or independent events the probability of both happening is P(A ∩ B) = P(A) x P(B)

P(spam ∩ Viagra) = 0.05 * 0.20 = 0.01 (1% of all messages are spam containing the word Viagra)

In reality, it is far more likely that **P(spam) and P(Viagra) are highly dependent**, which means that **this calculation is incorrect**

Conditional probability with Bayes' theorem

The relationships between dependent events can be described using **Bayes' Theorem**

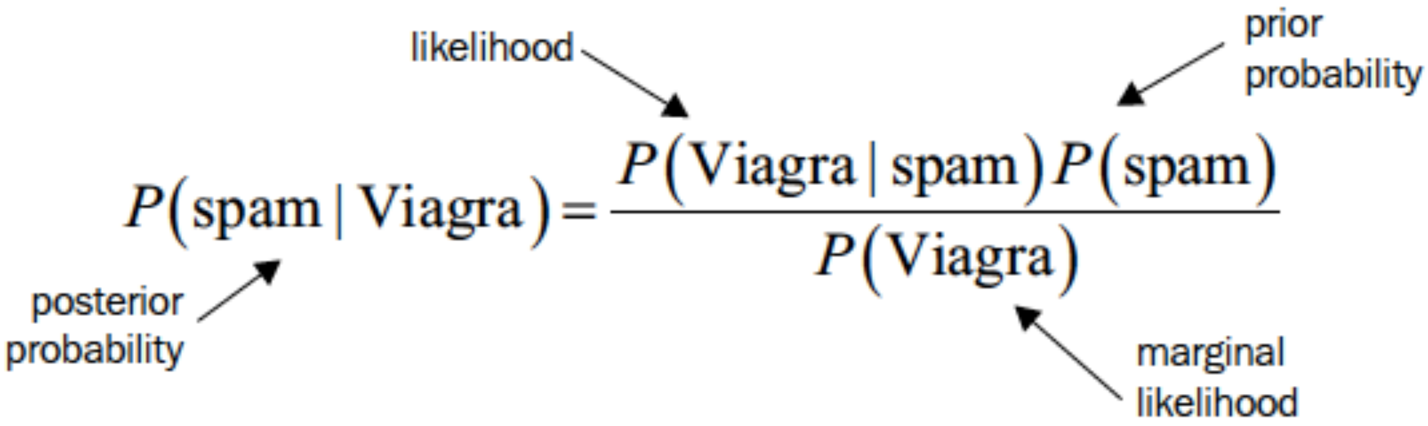
$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} = \frac{P(A \cap B)}{P(B)}$$

- P(A|B) can be read as the probability of event A given that event B occurred
- This is known as conditional probability, since the probability of A is dependent (that is, conditional) on what happened with event B

Conditional probability

If a message was randomly selected from the mailbox and the message contains the word Viagra (event B), what is the likelihood of the message being spam (event A)?

Applying Bayes theorem it is possible to calculate the posterior probability that measures how likely the message may be spam



$$P(\text{spam} \mid \text{Viagra}) = \frac{P(\text{Viagra} \mid \text{spam})\,P(\text{spam})}{P(\text{Viagra})}$$

Bayes Theorem

Assuming we have 100 messages in our mailbox:

- 5 messages contain the word Viagra, and these 4 are Spam

P (Viagra | Spam) = 4/20 = 0.20

P(spam) = 20%

P(viagra) = 5%

$$P(Spam \mid Viagra) = \frac{P(Viagra \mid Spam)P(Spam)}{P(Viagra)}$$

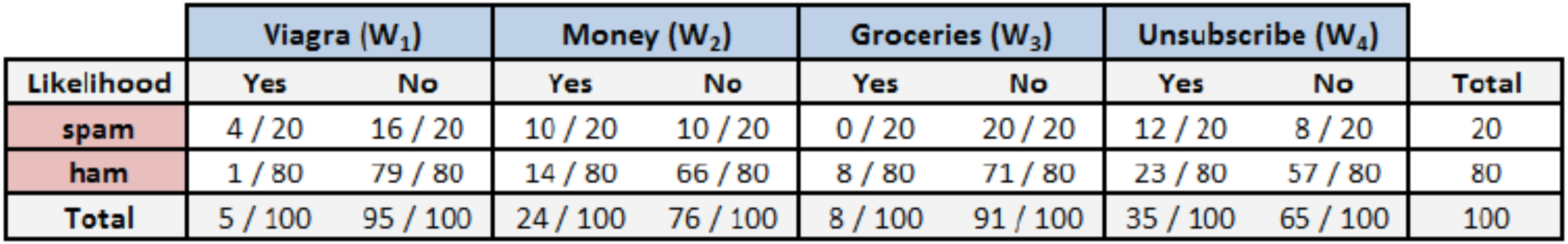
$$P(Spam \mid Viagra) = \frac{0.2 \times 0.2}{0.05} = 0.8$$

Therefore, 80% is the probability of a message being spam since it contains the word Viagra

Naïve Bayes Classification

Let's extend our spam filter by adding a few additional terms to be monitored: money, groceries, and unsubscribe

The naive Bayes learner is trained by constructing a likelihood table for the appearance of these four words (W1, W2, W3, and W4), as shown in the following diagram for 100 emails:



	Viagra (W_1)		Money (W_2)		Groceries (W_3)		Unsubscribe (W_4)		
Likelihood	**Yes**	**No**	**Yes**	**No**	**Yes**	**No**	**Yes**	**No**	**Total**
spam	4 / 20	16 / 20	10 / 20	10 / 20	0 / 20	20 / 20	12 / 20	8 / 20	20
ham	1 / 80	79 / 80	14 / 80	66 / 80	8 / 80	71 / 80	23 / 80	57 / 80	80
Total	5 / 100	95 / 100	24 / 100	76 / 100	8 / 100	91 / 100	35 / 100	65 / 100	100

Likelihood

Prior

$$P(Y|X_1,\ldots,X_n) = \frac{P(X_1,\ldots,X_n|Y)P(Y)}{P(X_1,\ldots,X_n)}$$

Normalization Constant

Model Parameters

Using Bayes' theorem, we can define the problem as shown in the following formula, which captures the probability that a message is spam, given that Viagra = Yes, Money = No, Groceries = No, and Unsubscribe = Yes:

$$P(Spam \mid W_1 \cap \neg W_2 \cap \neg W_3 \cap W_4) = \frac{P(W_1 \cap \neg W_2 \cap \neg W_3 \cap W_4 \mid Spam)P(Spam)}{P(W_1 \cap \neg W_2 \cap \neg W_3 \cap W_4)}$$

- For a number of reasons the formula $P(X_1,\ldots,X_n|Y)$ is computationally difficult to solve
- As additional features are added, tremendous amounts of memory are needed to store probabilities for all of the possible intersecting events
- Enormous training datasets would be required to ensure that enough data is available to model all of the possible interactions

Naïve Bayes Assumption

Assume that all features are independent **given the class label Y**

Equationally speaking:

$$P(X_1, \ldots, X_n|Y) = \prod_{i=1}^{n} P(X_i|Y)$$

Naive Bayes assumes independence among events. Specifically, naive Bayes assumes class-conditional independence, which means that events are independent so long as they are conditioned on the same class value

$$P(\text{Spam} \mid W_1 \cap \neg W_2 \cap \neg W_3 \cap W_4) = \frac{P(W_1 \mid \text{spam}) P(\neg W_2 \mid \text{spam}) P(\neg W_3 \mid \text{spam}) P(W_4 \mid \text{spam}) P(\text{spam})}{P(W_1) P(\neg W_2) P(\neg W_3) P(W_4)}$$

Naïve Bayes

$$P(Spam \mid W_1 \cap \neg W_2 \cap \neg W_3 \cap W_4) = \frac{4}{20} \times \frac{10}{20} \times \frac{20}{20} \times \frac{12}{20} \times \frac{20}{100} = 0.012$$

$$P(Ham \mid W_1 \cap \neg W_2 \cap \neg W_3 \cap W_4) = \frac{1}{80} \times \frac{66}{80} \times \frac{72}{80} \times \frac{23}{80} \times \frac{80}{100} = 0.002$$

- Because 0.012 / 0.002 = 6, we can say that this message is six times more likely to be spam than ham

- To convert these numbers to probabilities, the probability of spam is equal to the likelihood that the message is spam divided by the likelihood that the message is either spam or ham:

$$P(spam) = \frac{0.012}{(0.012 + 0.002)} = 0.857 \qquad P(ham) = 1 - 0.857 = 0.143$$

Bayes optimal Classifier

- What is he most probable classification of a new **instance** given the training data?
    - The most probable classification of the new instance is obtained by combining the prediction of *all hypothesis*, weighted by their *posterior probabilities*
- If the classification of new example can take any value v_j from some set *V,* then the probability $P(v_j|D)$ that the correct classification for the new **instance is v_j,** is just:

$$P(v_j|D) = \sum_{h_i \in H} P(v_j|h_i)P(h_i|D)$$

Naïve Bayes Classifier

Assume a target function f : X → V, where each instance *x* described by attributes $a_1, a_2 .. a_n$

Most probable value of *f(x)* is:

$$v_{MAP} = \arg\max_{v_j \in V} P(v_j | a_1, a_2 \ldots a_n)$$

$$v_{MAP} = \arg\max_{v_j \in V} \frac{P(a_1, a_2 \ldots a_n | v_j) P(v_j)}{P(a_1, a_2 \ldots a_n)}$$

$$= \arg\max_{v_j \in V} P(a_1, a_2 \ldots a_n | v_j) P(v_j)$$

Naïve Bayes assumption

$$P(a_1, a_2 \ldots a_n | v_j) = \prod_i P(a_i | v_j)$$

which gives

$$\text{Naive Bayes classifier: } v_{NB} = \operatorname*{argmax}_{v_j \in V} P(v_j) \prod_i P(a_i | v_j)$$

Why is this useful?

- Number of parameters for modeling $P(X_1,\ldots,X_n|Y)$:
  - $2(2^n-1)$
- Number of parameters for modeling $P(X_1|Y),\ldots,P(X_n|Y)$
  - $2n$

Naïve Bayes Training

- Training in Naïve Bayes is **easy**:
  - Estimate P(Y=v) as the fraction of records with Y=v

$$P(Y = v) = \frac{Count(Y = v)}{\#\ records}$$

  - Estimate P(X_i=u|Y=v) as the fraction of records with Y=v for which X_i=u

$$P(X_i = u|Y = v) = \frac{Count(X_i = u \wedge Y = v)}{Count(Y = v)}$$

- (This corresponds to Maximum Likelihood estimation of model parameters)

Naïve Bayes Training

- In practice, some of these counts can be zero
- Fix this by adding “virtual” counts:

$$P(X_i = u|Y = v) = \frac{Count(X_i = u \wedge Y = v) + 1}{Count(Y = v) + 2}$$

  - (This is like putting a prior on parameters and doing MAP estimation instead of MLE)
  - This is called *Smoothing*

Naïve Bayes Classification

The weather data, with counts and probabilities

outlook			temperature			humidity			windy			play	
	yes	no		yes	no		yes	no		yes	no	yes	no
sunny	2	3	hot	2	2	high	3	4	false	6	2	9	5
overcast	4	0	mild	4	2	normal	6	1	true	3	3		
rainy	3	2	cool	3	1								
sunny	2/9	3/5	hot	2/9	2/5	high	3/9	4/5	false	6/9	2/5	9/14	5/14
overcast	4/9	0/5	mild	4/9	2/5	normal	6/9	1/5	true	3/9	3/5		
rainy	3/9	2/5	cool	3/9	1/5								

A new day

outlook	temperature	humidity	windy	play
sunny	cool	high	true	?

Naïve Bayes Classification

- Likelihood of yes

$$=\frac{2}{9}\times\frac{3}{9}\times\frac{3}{9}\times\frac{3}{9}\times\frac{9}{14}=0.0053$$

- Likelihood of no

$$=\frac{3}{5}\times\frac{1}{5}\times\frac{4}{5}\times\frac{3}{5}\times\frac{5}{14}=0.0206$$

- Therefore, the prediction is No

Naïve Bayes Classification with numerical attributes

- For data sets with numerical attribute values
  - It assumes normal distributions for numerical attributes

- Let x_1, x_2, …, x_n be the values of a numerical attribute in the training data set

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

$$\sigma = \frac{1}{n-1}\sum_{i=1}^{n}\left(x_i - \mu\right)^2$$

$$P(A_i \mid c_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} e^{-\frac{(A_i-\mu_{ij})^2}{2\sigma_{ij}^2}}$$

Naïve Bayes Classification with numerical attributes

The numeric weather data with summary statistics

outlook			temperature			humidity			windy			play	
	yes	no		yes	no		yes	no		yes	no	yes	no
sunny	2	3		83	85		86	85	false	6	2	9	5
overcast	4	0		70	80		96	90	true	3	3		
rainy	3	2		68	65		80	70					
				64	72		65	95					
				69	71		70	91					
				75			80						
				75			70						
				72			90						
				81			75						
sunny	2/9	3/5	mean	73	74.6	mean	79.1	86.2	false	6/9	2/5	9/14	5/14
overcast	4/9	0/5	std dev	6.2	7.9	std dev	10.2	9.7	true	3/9	3/5		
rainy	3/9	2/5											

Naïve Bayes Classification with numerical attributes

A new day				
outlook	temperature	humidity	windy	play
sunny	66	90	true	?

$$P(temperature = 66 \mid play = Y) = \frac{1}{\sqrt{2\pi(6.2)^2}} e^{-\frac{(66-73)^2}{2\times 6.2^2}} = 0.0590$$

$$P(temperature = 66 \mid play = N) = \frac{1}{\sqrt{2\pi(7.9)^2}} e^{-\frac{(66-74,6)^2}{2\times 7.9^2}} = 0.0472$$

$$P(humidity = 90 \mid Joga = S) = 0.041$$

$$P(humidity = 90 \mid Joga = N) = 0.042$$

Naïve Bayes Classification with numerical attributes

Likelihood of Yes

$$\frac{2}{9} \times 0.059 \times 0.041 \times \frac{3}{9} \times \frac{9}{14} = 0.00012$$

- Likelihood of No

$$\frac{3}{5} \times 0.0472 \times 0.042 \times \frac{3}{5} \times \frac{5}{14} = 0.00025$$

- Therefore, the prediction is **No**

Naïve Bayes Classifier

- Along with decision trees, neural networks, nearest neighbor is one of the most practical learning methods

- When to use:
  - Moderate or large training set available
  - Attributes that describe instances are conditionally independent given classification

- Successful applications:
  - Diagnosis
  - Documents Classification