

# Naive-Bayes Classification Algorithm

## 1. Introduction to Bayesian Classification

The Bayesian Classification represents a supervised learning method as well as a statistical method for classification. Assumes an underlying probabilistic model and it allows us to capture uncertainty about the model in a principled way by determining probabilities of the outcomes. It can solve diagnostic and predictive problems.

This Classification is named after Thomas Bayes ( 1702-1761), who proposed the Bayes Theorem.

Bayesian classification provides practical learning algorithms and prior knowledge and observed data can be combined. Bayesian Classification provides a useful perspective for understanding and evaluating many learning algorithms. It calculates explicit probabilities for hypothesis and it is robust to noise in input data.

### **Uses of Naive Bayes classification:**

#### 1. Naive Bayes text classification

(<http://nlp.stanford.edu/IR-book/html/htmledition/naive-bayes-text-classification-1.html>)

The Bayesian classification is used as a probabilistic learning method (Naive Bayes text classification). Naive Bayes classifiers are among the most successful known algorithms for learning to classify text documents.

#### 2. Spam filtering ([http://en.wikipedia.org/wiki/Bayesian\\_spam\\_filtering](http://en.wikipedia.org/wiki/Bayesian_spam_filtering))

Spam filtering is the best known use of Naive Bayesian text classification. It makes use of a naive Bayes classifier to identify spam e-mail.

Bayesian spam filtering has become a popular mechanism to distinguish illegitimate spam email from legitimate email (sometimes called "ham" or "bacn").<sup>[4]</sup> Many modern mail clients implement Bayesian spam filtering. Users can also install separate email filtering programs. Server-side email filters, such as DSPAM, SpamAssassin, SpamBayes, Bogofilter and ASSP, make use of Bayesian spam filtering techniques, and the functionality is sometimes embedded within mail server software itself.

#### 3. Hybrid Recommender System Using Naive Bayes Classifier and Collaborative Filtering (<http://eprints.ecs.soton.ac.uk/18483/>)

Recommender Systems apply machine learning and data mining techniques for filtering unseen information and can predict whether a user would like a given resource.

It is proposed a unique switching hybrid recommendation approach by combining a Naive Bayes classification approach with the collaborative filtering. Experimental results on two different data sets, show that the proposed algorithm is scalable and provide better performance—in terms of accuracy and coverage—than other algorithms while at the same time eliminates some recorded problems with the recommender systems.

4. [Online applications \(http://www.convo.co.uk/x02/\)](http://www.convo.co.uk/x02/)

This online application has been set up as a simple example of supervised machine learning and affective computing. Using a training set of examples which reflect nice, nasty or neutral sentiments, we're training Ditto to distinguish between them.

Simple Emotion Modelling, combines a statistically based classifier with a dynamical model. The Naive Bayes classifier employs single words and word pairs as features. It allocates user utterances into nice, nasty and neutral classes, labelled +1, -1 and 0 respectively. This numerical output drives a simple first-order dynamical system, whose state represents the simulated emotional state of the experiment's personification, Ditto the donkey.

## ***1.1. Independence***

### ***1.1.1. Example:***

Suppose there are two events:

- M: Manuela teaches the class (otherwise it's Andrew)
- S: It is sunny

“The sunshine levels do not depend on and do not influence who is teaching.”

### ***1.1.2. Theory:***

From  $P(S | M) = P(S)$ , the rules of probability imply:

- $P(\sim S | M) = P(\sim S)$
- $P(M | S) = P(M)$
- $P(M \wedge S) = P(M) P(S)$
- $P(\sim M \wedge S) = P(\sim M) P(S)$
- $P(M \wedge \sim S) = P(M) P(\sim S)$
- $P(\sim M \wedge \sim S) = P(\sim M) P(\sim S)$

### ***1.1.3. Theory applied on previous example:***

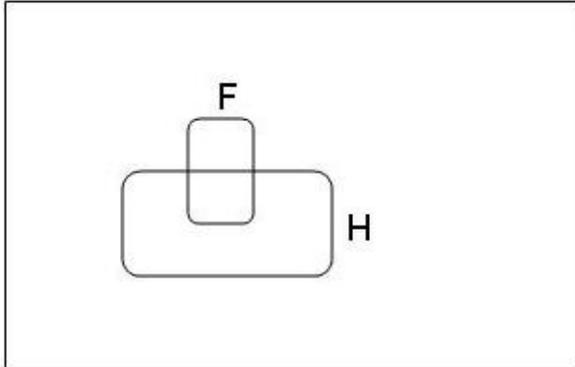
“The sunshine levels do not depend on and do not influence who is teaching.” can be specified very simply:

$$P(S | M) = P(S)$$

“Two events A and B are statistically independent if the probability of A is the same value when B occurs, when B does not occur or when nothing is known about the occurrence of B”

## 1.2. Conditional Probability

### 1.2.1. Simple Example:



H = "Have a headache"

F = "Coming down with Flu"

$$P(H) = 1/10$$

$$P(F) = 1/40$$

$$P(H \cap F) = 1/2$$

$$P(A|B) + P(\neg A|B) = 1$$

"Headaches are rare and flu is rarer, but if you're coming down with 'flu there's a 50-50 chance you'll have a headache."

$$\begin{aligned}
 P(H|F) &= \frac{\text{Fraction of flu-inflicted worlds in which you have a headache}}{\text{\#worlds with flu and headache}} = \frac{\text{Area of "H and F" region}}{\text{Area of "F" region}} = \frac{P(H \cap F)}{P(F)} \\
 &= \frac{\text{\#worlds with flu and headache}}{\text{\#worlds with flu}} = \frac{\text{Area of "H and F" region}}{\text{Area of "F" region}} = \frac{P(H \cap F)}{P(F)}
 \end{aligned}$$

### 1.2.2. Theory:

$P(A|B)$  = Fraction of worlds in which B is true that also have A true

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

Corollary:

$$P(A \cap B) = P(A|B) P(B)$$

$$P(A|B) + P(\neg A|B) = 1$$

$$\sum_{k=1}^n P(A = v_k | B) = 1$$

### 1.2.3. Detailed Example

- M : Manuela teaches the class
- S : It is sunny
- L : The lecturer arrives slightly late.

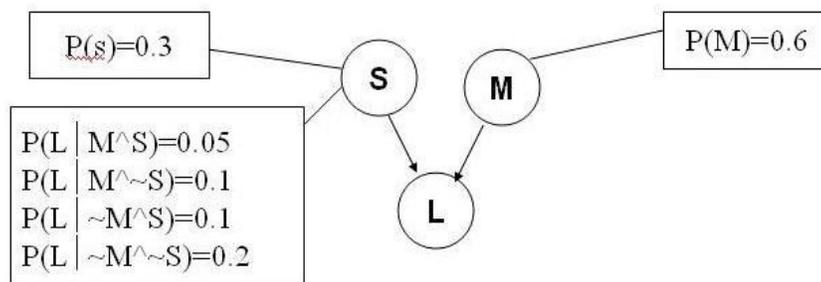
Assume both lecturers are sometimes delayed by bad weather. Andrew is more likely to arrive late than Manuela.

Let's begin with writing down the knowledge:

$$P(S \mid M) = P(S), \quad P(S) = 0.3, \quad P(M) = 0.6$$

Lateness is not independent of the weather and is not independent of the lecturer. Therefore Lateness is *dependant* on both weather and lecturer

$P(S \mid M) = P(S)$	$P(L \mid M \wedge S) = 0.05$
$P(S) = 0.3$	$P(L \mid M \wedge \sim S) = 0.1$
$P(M) = 0.6$	$P(L \mid \sim M \wedge S) = 0.1$
	$P(L \mid \sim M \wedge \sim S) = 0.2$



Read the absence of an arrow between S and M to mean "it would not help me predict M if I knew the value of S"

Read the two arrows into L to mean "that if I want to know the value of L it may help me to know M and to know S."

## 1.3. Conditional Independence

### 1.3.1. Example:

Suppose we have these three events:

- M : Lecture taught by Manuela
- L : Lecturer arrives late
- R : Lecture concerns robots

Suppose:

- Andrew has a higher chance of being late than Manuela.
- Andrew has a higher chance of giving robotics lectures.

Once you know who the lecturer is, then whether they arrive late doesn't affect whether the lecture concerns robots.

**1.3.2. Theory:**

R and L are conditionally independent given M if for all x,y,z in {T,F}:

$$P(R=x \mid M=y \wedge L=z) = P(R=x \mid M=y)$$

More generally:

Let S1 and S2 and S3 be sets of variables.

Set-of-variables S1 and set-of-variables S2 are conditionally independent given S3 if for all assignments of values to the variables in the sets,  $P(S1's \text{ assignments} \mid S2's \text{ assignments} \ \& \ S3's \text{ assignments}) = P(S1's \text{ assignments} \mid S3's \text{ assignments})$

$$P(A|B) = P(A \wedge B)/P(B)$$

Therefore  $P(A \wedge B) = P(A|B).P(B)$  – also known as Chain Rule

$$\text{Also } P(A \wedge B) = P(B|A).P(A)$$

$$\text{Therefore } P(A|B) = P(B|A).P(A)/P(B)$$

$$P(A,B|C) = P(A \wedge B \wedge C)/P(C)$$

$$= P(A|B,C).P(B \wedge C)/P(C) \text{ – applying chain rule}$$

$$= P(A|B,C).P(B|C)$$

$$= P(A|C).P(B|C) \text{ , if A and B are conditionally independent given C.}$$

This can be extended for n values as  $P(A1,A2...An|C) = P(A1|C).P(A2|C)...P(An|C)$  if A1, A2...An are conditionally independent given C.

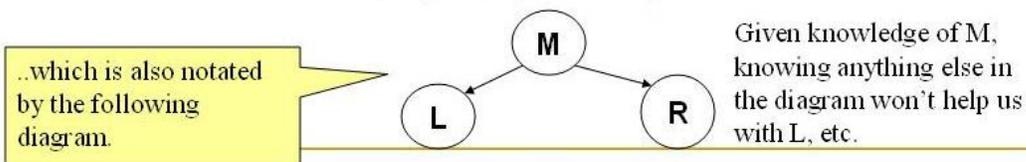
**1.3.3. Theory applied on previous example:**

For the previous example, we can use the following notations:

$$P(R \mid M,L) = P(R \mid M) \text{ and } P(R \mid \sim M,L) = P(R \mid \sim M)$$

We express this in the following way:

“R and L are conditionally independent given M”



## 2. Bayes Theorem

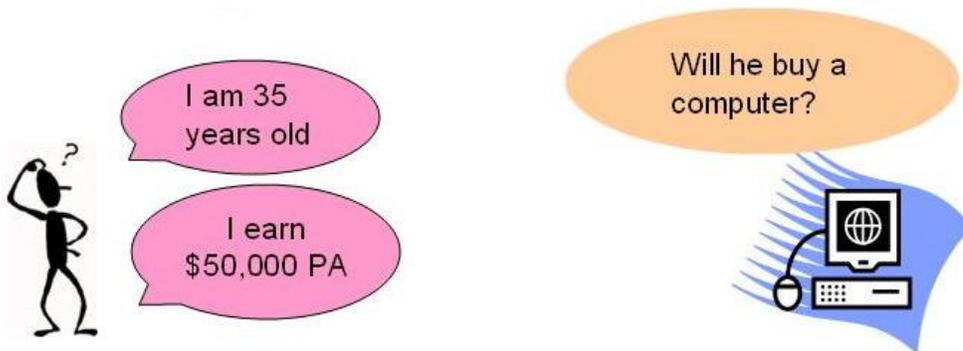
Bayesian reasoning is applied to decision making and inferential statistics that deals with probability inference. It is used the knowledge of prior events to predict future events.

Example: Predicting the color of marbles in a basket

### 2.1. Example:

rec	Age	Income	Student	Credit_rating	Buys_computer
r1	<=30	High	No	Fair	No
r2	<=30	High	No	Excellent	No
r3	31...40	High	No	Fair	Yes
r4	>40	Medium	No	Fair	Yes
r5	>40	Low	Yes	Fair	Yes
r6	>40	Low	Yes	Excellent	No
r7	31...40	Low	Yes	Excellent	Yes
r8	<=30	Medium	No	Fair	No
r9	<=30	Low	Yes	Fair	Yes
r10	>40	Medium	Yes	Fair	Yes
r11	<=30	Medium	Yes	Excellent	Yes
r12	31...40	Medium	No	Excellent	Yes
r13	31...40	High	Yes	Fair	Yes
r14	>40	Medium	No	Excellent	No

Table1: Data table



### 2.2. Theory:

*The Bayes Theorem:*

$$P(h/D) = \frac{P(D/h) P(h)}{P(D)}$$

$P(h)$  : Prior probability of hypothesis  $h$   
 $P(D)$  : Prior probability of training data  $D$   
 $P(h/D)$  : Probability of  $h$  given  $D$   
 $P(D/h)$  : Probability of  $D$  given  $h$

### ***2.3. Theory applied on previous example:***

- $D$  : 35 year old customer with an income of \$50,000 PA
- $h$  : Hypothesis that our customer will buy our computer

$P(h/D)$  : Probability that customer  $D$  will buy our computer given that we know his age and income

$P(h)$  : Probability that any customer will buy our computer regardless of age (Prior Probability)

$P(D/h)$  : Probability that the customer is 35 yrs old and earns \$50,000, given that he has bought our computer (Posterior Probability)

$P(D)$  : Probability that a person from our set of customers is 35 yrs old and earns \$50,000

### ***2.4. Maximum A Posteriori (MAP) Hypothesis***

#### ***2.4.1. Example:***

$h_1$ : Customer buys a computer = Yes

$h_2$  : Customer buys a computer = No

where  $h_1$  and  $h_2$  are subsets of our Hypothesis Space 'H'

$P(h/D)$  (Final Outcome) =  $\arg \max \{ P(D/h_1) P(h_1), P(D/h_2) P(h_2) \}$

$P(D)$  can be ignored as it is the same for both the terms

#### ***2.4.2. Theory:***

Generally we want the most probable hypothesis given the training data

$h_{MAP} = \arg \max P(h/D)$  (where  $h$  belongs to  $H$  and  $H$  is the hypothesis space)

$$h_{MAP} = \arg \max \frac{P(D/h) P(h)}{P(D)}$$

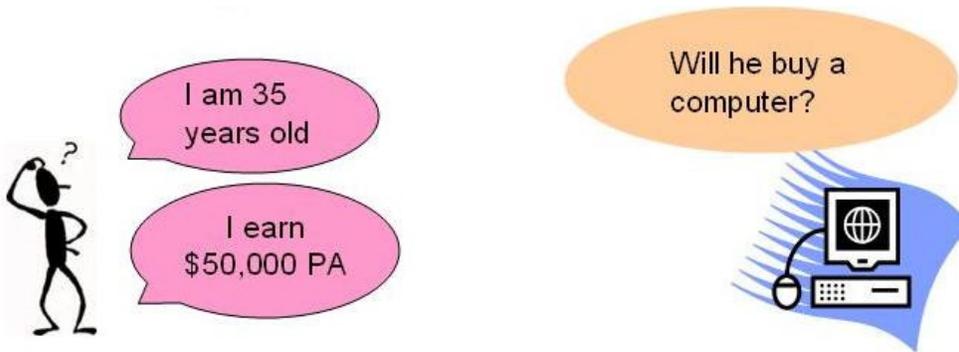
$$h_{MAP} = \arg \max P(D/h) P(h)$$

## 2.5. Maximum Likelihood (ML) Hypothesis

### 2.5.1. Example:

Sl No	Age	Income	Student	Credit rating	Buys computer
1	35	Medium	Yes	Fair	Yes
2	30	High	No	Average	No
3	40	Low	Yes	Good	No
4	35	Medium	No	Fair	Yes
5	45	Low	No	Fair	Yes
6	35	High	No	Excellent	Yes
7	35	Medium	No	Good	No
8	25	Low	No	Good	No
9	28	High	No	Average	No
10	35	Medium	Yes	Average	Yes

Table 2



### 2.5.2. Theory:

If we assume  $P(h_i) = P(h_j)$  where the calculated probabilities amount to the same  
Further simplification leads to:

$hML = \arg \max P(D/h_i)$  (where  $h_i$  belongs to  $H$ )

### 2.5.3. Theory applied on previous example:

$P(\text{buys computer} = \text{yes}) = 5/10 = 0.5$

$P(\text{buys computer} = \text{no}) = 5/10 = 0.5$

$P(\text{customer is 35 yrs \& earns } \$50,000) = 4/10 = 0.4$

$P(\text{customer is 35 yrs \& earns } \$50,000 / \text{buys computer} = \text{yes}) = 3/5 = 0.6$

$P(\text{customer is 35 yrs \& earns } \$50,000 / \text{buys computer} = \text{no}) = 1/5 = 0.2$

Customer buys a computer  $P(h_1/D) = P(h_1) * P(D/h_1) / P(D) = 0.5 * 0.6 / 0.4$

Customer does not buy a computer  $P(h_2/D) = P(h_2) * P(D/h_2) / P(D) = 0.5 * 0.2 / 0.4$

Final Outcome =  $\arg \max \{P(h1/D), P(h2/D)\} = \max(0.6, 0.2)$   
 $\Rightarrow$  Customer buys a computer

### 3. Naïve Bayesian Classification

It is based on the Bayesian theorem. It is particularly suited when the dimensionality of the inputs is high. Parameter estimation for naive Bayes models uses the method of maximum likelihood. In spite of over-simplified assumptions, it often performs better in many complex real-world situations.

Advantage: Requires a small amount of training data to estimate the parameters

#### 3.1. Example

rec	Age	Income	Student	Credit_rating	Buys_computer
r1	<=30	High	No	Fair	No
r2	<=30	High	No	Excellent	No
r3	31...40	High	No	Fair	Yes
r4	>40	Medium	No	Fair	Yes
r5	>40	Low	Yes	Fair	Yes
r6	>40	Low	Yes	Excellent	No
r7	31...40	Low	Yes	Excellent	Yes
r8	<=30	Medium	No	Fair	No
r9	<=30	Low	Yes	Fair	Yes
r10	>40	Medium	Yes	Fair	Yes
r11	<=30	Medium	Yes	Excellent	Yes
r12	31...40	Medium	No	Excellent	Yes
r13	31...40	High	Yes	Fair	Yes
r14	>40	Medium	No	Excellent	No

X = ( age= youth, income = medium, student = yes, credit\_rating = fair)

A person belonging to tuple X will buy a computer?

#### 3.2.Theory:

Derivation:

D : Set of tuples

- Each Tuple is an 'n' dimensional attribute vector
- X : (x1,x2,x3,... xn)

Let there be 'm' Classes : C1,C2,C3...Cm

Naïve Bayes classifier predicts X belongs to Class Ci iff

$$\square P(C_i/X) > P(C_j/X) \text{ for } 1 \leq j \leq m, j \neq i$$

Maximum Posteriori Hypothesis

- $\square P(C_i/X) = P(X/C_i) P(C_i) / P(X)$
- $\square$  Maximize  $P(X/C_i) P(C_i)$  as  $P(X)$  is constant

With many attributes, it is computationally expensive to evaluate  $P(X/C_i)$ .

Naïve Assumption of “class conditional independence”

$$P(X / C_i) = \prod_{k=1}^n P(x_k / C_i)$$

$$P(X/C_i) = P(x_1/C_i) * P(x_2/C_i) * \dots * P(x_n/ C_i)$$

### 3.3. Theory applied on previous example:

$$P(C_1) = P(\text{buys\_computer} = \text{yes}) = 9/14 = 0.643$$

$$P(C_2) = P(\text{buys\_computer} = \text{no}) = 5/14 = 0.357$$

$$P(\text{age}=\text{youth} / \text{buys\_computer} = \text{yes}) = 2/9 = 0.222$$

$$P(\text{age}=\text{youth} / \text{buys\_computer} = \text{no}) = 3/5 = 0.600$$

$$P(\text{income}=\text{medium} / \text{buys\_computer} = \text{yes}) = 4/9 = 0.444$$

$$P(\text{income}=\text{medium} / \text{buys\_computer} = \text{no}) = 2/5 = 0.400$$

$$P(\text{student}=\text{yes} / \text{buys\_computer} = \text{yes}) = 6/9 = 0.667$$

$$P(\text{student}=\text{yes} / \text{buys\_computer} = \text{no}) = 1/5 = 0.200$$

$$P(\text{credit rating}=\text{fair} / \text{buys\_computer} = \text{yes}) = 6/9 = 0.667$$

$$P(\text{credit rating}=\text{fair} / \text{buys\_computer} = \text{no}) = 2/5 = 0.400$$

$$P(X/\text{Buys a computer} = \text{yes}) = P(\text{age}=\text{youth} / \text{buys\_computer} = \text{yes}) * P(\text{income}=\text{medium} / \text{buys\_computer} = \text{yes}) * P(\text{student}=\text{yes} / \text{buys\_computer} = \text{yes}) * P(\text{credit rating}=\text{fair} / \text{buys\_computer} = \text{yes}) = 0.222 * 0.444 * 0.667 * 0.667 = 0.044$$

$$P(X/\text{Buys a computer} = \text{No}) = 0.600 * 0.400 * 0.200 * 0.400 = 0.019$$

Find class  $C_i$  that Maximizes  $P(X/C_i) * P(C_i)$

$$\Rightarrow P(X/\text{Buys a computer} = \text{yes}) * P(\text{buys\_computer} = \text{yes}) = 0.028$$

$$\Rightarrow P(X/\text{Buys a computer} = \text{No}) * P(\text{buys\_computer} = \text{no}) = 0.007$$

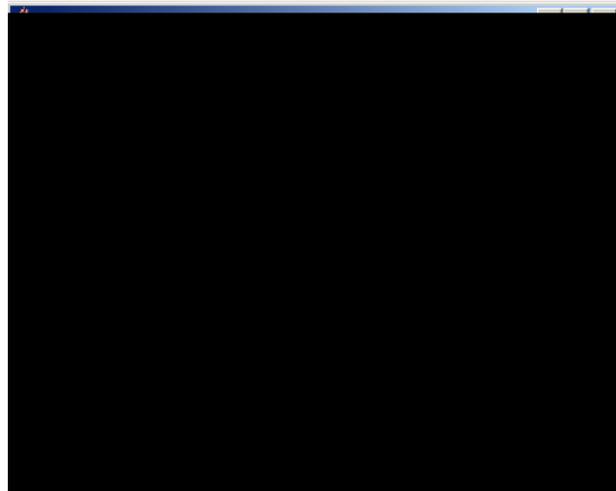
Prediction : Buys a computer for Tuple X

## 4. Sample running example with weka

### 4.1. Bayesian Network Classifiers in Weka

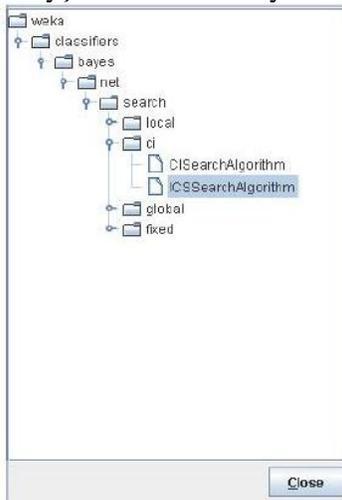
Let  $U = \{x_1, \dots, x_n\}$ ,  $n \geq 1$  be a set of variables. A Bayesian network  $B$  over a set of variables  $U$  is a network structure  $BS$ , which is a directed acyclic graph (DAG) over  $U$  and a set of probability tables  $BP = \{p(u|pa(u)) \mid u \in U\}$  where  $pa(u)$  is the set of parents of  $u$  in  $BS$ . A Bayesian network represents a probability distributions  $P(U) = \prod_{u \in U} p(u|pa(u))$ .

Below, a Bayesian network is shown for the variables in the iris data set. Note that the links between the nodes class, petalength and petalwidth do not form a directed cycle, so the graph is a proper DAG.



### 4.2. Conditional independence test based structure learning

Conditional independence tests in Weka are slightly different from the standard tests described in the literature. To test whether variables  $x$  and  $y$  are conditionally independent given a set of variables  $Z$ , a network structure with arrows  $\forall z \in Z z \rightarrow y$  is compared with one with arrows  $\{x \rightarrow y\} \cup \forall z \in Z z \rightarrow y$ . A test is performed.



At the moment, only the ICS [9] and CI algorithm are implemented.

The ICS algorithm makes two steps, first find a skeleton (the undirected graph with edges if there is an arrow in network structure) and second direct all the edges in the skeleton to get a DAG.

Starting with a complete undirected graph, we try to find conditional independencies  $\langle x, y | Z \rangle$  in the data. For each pair of nodes  $x, y$ , we consider sets  $Z$  starting with cardinality 0, then 1 up to a user defined maximum. Further-more, the set  $Z$  is a subset of nodes that are neighbors of both  $x$  and  $y$ . If an independency is identified, the edge between  $x$  and  $y$  is removed from the skeleton.

The first step in directing arrows is to check for every configuration  $x - z - y$  where  $x$  and  $y$  not connected in the skeleton whether  $z$  is in the set  $Z$  of variables that justified removing the link between  $x$  and  $y$  (cached in the first step). If  $z$  is not in  $Z$ , we can assign direction  $x \rightarrow z \leftarrow y$ .

Finally, a set of graphical rules is applied to direct the remaining arrows.

Rule 1:  $i \rightarrow j - k \ \& \ i - / - k \Rightarrow j \rightarrow k$

Rule 2:  $i \rightarrow j \rightarrow k \ \& \ i - - k \Rightarrow i \rightarrow k$

Rule 3 m

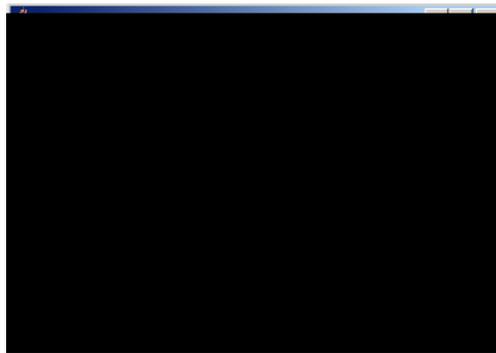
$$\begin{array}{c}
 \quad \quad \quad / \backslash \\
 \quad \quad \quad i \quad | \quad k \\
 i \rightarrow j \leftarrow k \quad \backslash \quad / \\
 \quad \quad \quad \quad \quad j
 \end{array}
 \Rightarrow m \rightarrow j$$

Rule 4 m

$$\begin{array}{c}
 \quad \quad \quad / \quad \backslash \\
 \quad \quad \quad i \quad - - - \quad k \\
 i \rightarrow j \quad \quad \quad \backslash \quad / \\
 \quad \quad \quad \quad \quad j
 \end{array}
 \Rightarrow i \rightarrow m \ \& \ k \rightarrow m$$

Rule 5: if no edges are directed then take a random one (first we can find)

The ICS algorithm comes with the following options.



Since the ICS algorithm is focused on recovering causal structure, instead of finding the optimal classifier, the Markov blanket correction can be made afterwards.

Specific options:

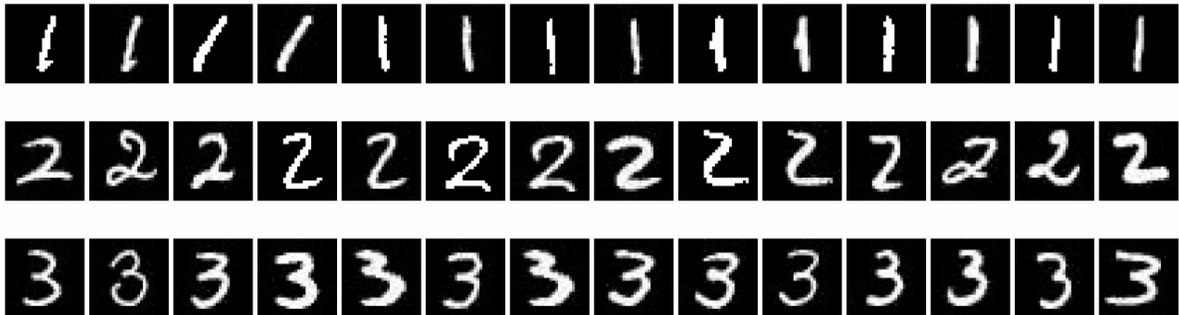
The maxCardinality option determines the largest subset of  $Z$  to be considered in conditional independence tests  $\langle x, y | Z \rangle$ .

The scoreType option is used to select the scoring metric.

## 5. Exercises

Implement, test and interpret the results for Naïve Bayes algorithm for the following problems, using the attached input files

**5.1.** Being given the following binary files (imagini.zip), which represent the classes for 1, 2 and 3, you must find out the class of a digit in an image.



There will be used as attributes white pixels (value 255) and the positions of their appearance.

Algorithm:

Step 1: It is loaded the image which will be classified as being ONE, TWO or THREE

Step 2: There are loaded the images found in the folder **images**. The name of the files belonging to class ONE are: "image1\_\*.jpg", the ones belonging to class TWO are: "image2\_\*.jpg" and the ones for class THREE are : "image3\_\*.jpg".

Step3: It is determined the a priori probability for each class:  
 $P(\text{UNU}) = \text{NrTemplateInClassONE} / \text{NumberTotalTemplates}$   
 $P(\text{DOI}) = \text{NrTemplateInClassTWO} / \text{NumberTotalTemplates}$   
 $P(\text{TREI}) = \text{NrTemplateInClassTHREE} / \text{NumberTotalTemplates}$

Step 4: It is determined the probability that the image from the Step 1 to be in class ONE, TWO or THREE. Let (i,j) be the position of a white pixel in the image. It is calculated the probability that the pixel having the coordinates (i, j) to be white for the class ONE, TWO and THREE.

```
count1i,j = 0
for k = 1,n ; n – the number of images in class ONE
if image1_k(i,j) = 255 then
count1i,j = count1i,j + 1
probability1(i,j) =count1i,j / NrTemplateInClassONE
```

```
count2i,j = 0
for k = 1,n ; n- the number of images in class TWO
```

if image2\_k(i,j) = 255 then  
 count2i,j = count2i,j + 1  
 probability2(i,j) = count2i,j / NrTemplateInClassTWO

count3i,j = 0  
 for k = 1,n ; n- the number of images in class THREE  
 if image3\_k(i,j) = 255 then  
 count3i,j = count3i,j + 1  
 probability 3(i,j) = count3i,j / NrTemplateInClassTHREE

Step 5.

The posteriori probability that the image in Step 1 to be in class ONE is:  
 $P(T|ONE) = \text{average}(\text{probabilitate1}(i,j));$  (i, j) – the position of the white pixels in the image from Step1

Step 6.

The posteriori probability that the image in Step 1 to be in class TWO is:  
 $P(T|TWO) = \text{average}(\text{probabilitate1}(i,j));$  (i, j) – the position of the white pixels in the image from Step1

Step 7:

The posteriori probability that the image in Step 1 to be in class THREE is:  
 $P(T|THREE) = \text{average}(\text{probabilitate1}(i,j));$  (i, j) – the position of the white pixels in the image from Step1

Step 8:

It is determined the probability P for each image class and it is assigned the image from Step1 to the class of images that has the greatest probability.

$P(ONE|T) = P(T|ONE)*P(ONE)$   
 $P(TWO|T) = P(T|TWO)*P(TWO)$   
 $P(THREE|T) = P(T|THREE)*P(THREE)$

In order to load an image and to load pixels from an image in an array, you can use the following java code:

```
import java.awt.*;
import java.awt.image.*;
import java.io.*;
import javax.swing.*;
import java.util.*;
public class CImagesLoad {
    Vector<Image> images1 = new Vector<Image>();
    Vector<Image> images2 = new Vector<Image>();
    Vector<Image> images3 = new Vector<Image>();
    public String getFile(boolean isSaveDialog)
    {
        String currentDirectoryName = new File("").getAbsolutePath() +File.separator;
        try{
            JFileChooser fc = new JFileChooser(new File(new
            File(currentDirectoryName).getParent()));
            int result = 0;
            if(!isSaveDialog)
                result = fc.showOpenDialog(null);
```

```

        else
            result = fc.showSaveDialog(null);
        if(result==JFileChooser.CANCEL_OPTION) return null;
        else { //if(result==JFileChooser.APPROVE_OPTION){
            return fc.getSelectedFile().getAbsolutePath();
        }
    }
catch(Exception e)
{
    return null;
}
}
public void load_images (int template){
    String f = getFile(false);
    if (f==null)
    {
        return;
    }
    int k = 1;
    while (true)
    {
        String curent = new java.io.File (f).getAbsolutePath ();
        int pos = curent.lastIndexOf ("\\");
        curent = curent.substring (0, pos);
        if (k < 10)
        {
            curent += "\\image" + template + "_0" + k + ".jpg";
        }
        else
        {
            curent += "\\image" + template + "_" + k + ".jpg";
        }
        Image img = null;
        img = new javax.swing.ImageIcon(curent).getImage();
        if(img==null || img.getWidth(null)<=0 ||img.getHeight(null)<=0)
        {
            System.out.println("The file \n" + f.toString() + "\nhas an unsupported image
format");
            break;
        }
        else
        {
            k++;
            switch (template)
            {
                case 1:
                    images1.add (img);
                    break;
                case 2:
                    images2.add (img);
                    break;
                case 3:
                    images3.add (img);
                    break;
                default:
                    System.out.println("Other class");
                    break;
            }
        }
    }
}

```

```

    }
    public void load_pixels (Image image)
    {
        int width = image.getWidth(null);
        int height = image.getHeight(null);
        // Allocate buffer to hold the image's pixels
        int pixels[] = new int[width * height];
        // Grab pixels
        PixelGrabber pg = new PixelGrabber (image, 0, 0, width, height,
        pixels, 0, width);
        try
        {
            pg.grabPixels();
        }
        catch (InterruptedException e)
        {
            System.out.println ("Error image loading");
        }
    }
}

```

**5.2.** Modify 5.1 in order to classify images that are belonging to class 4.

**5.3.** Implement the example 3.1 for the following tuple X:

X = ( age= youth, income = high, student = no, credit\_rating = fair)

Find out if a person belonging to tuple X will buy a computer

## **6.References**

[http://en.wikipedia.org/wiki/Bayesian\\_probability](http://en.wikipedia.org/wiki/Bayesian_probability)

[http://en.wikipedia.org/wiki/Naive\\_Bayes\\_classifier](http://en.wikipedia.org/wiki/Naive_Bayes_classifier)

[http://www.let.rug.nl/~tiedeman/ml05/03\\_bayesian\\_handout.pdf](http://www.let.rug.nl/~tiedeman/ml05/03_bayesian_handout.pdf)

<http://www.statsoft.com/textbook/stnaiveb.html>

<http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/mlbook/ch6.pdf>

Chai, K.; H. T. Hn, H. L. Chieu; “Bayesian Online Classifiers for Text Classification and Filtering”, Proceedings of the 25th annual international ACM SIGIR conference on Research and Development in Information Retrieval, August 2002, pp 97-104

DATA MINING Concepts and Techniques, Jiawei Han, Micheline Kamber Morgan Kaufman Publishers, 2003