Bayesian Learning: An Introduction

João Gama

LIAAD-INESC Porto, University of Porto, Portugal

September 2008

1. Motivation: Information Processing

2. Introduction

3. Bayesian Network Classifiers

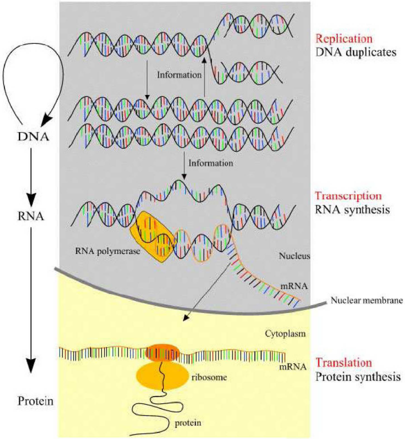
4. k-Dependence Bayesian Classifiers

5. Links and References

Outline

1. Motivation: Information Processing
2. Introduction
3. Bayesian Network Classifiers
4. k-Dependence Bayesian Classifiers
5. Links and References

Life as information processing:



Motivation: Micro Array

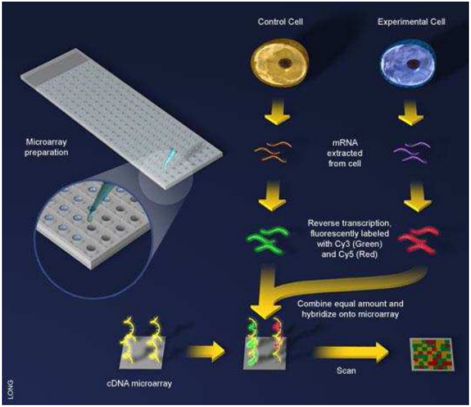

Control Cell
Experimental Cell
Microarray preparation
mRNA extracted from cell
Reverse transcription, fluorescently labeled with Cy3 (Green) and Cy5 (Red)
Combine equal amount and hybridize onto microarray
cDNA microarray
Scan
LONG

Fig. 13.2. Schematised experimental process of cDNA technique for microarray construction.

Motivation: Supervised Learning Task I

- Given:
  - a set of micro-array experiments, each done with mRNA from a different patient (same cell type from every patient)
  - Patient's expression values for each gene constitute the features, and
  - patient's disease constitutes the class
- Goal: Learn a model that accurately predicts class based on features

Micro-array data: Data Points are Samples

Person	A28202_ac	AB00014_at	AB00015_at	...
Person1	1144.0	321.0	2567.2	...
Person2	105.2	586.1	759.2	...
Person3	586.3	559.0	3210.2	...
Person4	42.8	692.0	812.2	...

Learning Problem:

Find groups of genes particularly active for groups of Persons.

Supervision: Add Class Values

Person	A28202_ac	AB00014_at	AB00015_at	. . .	Class
Person1	1144.0	321.0	2567.2	. . .	normal
Person2	105.2	586.1	759.2	. . .	cancer
Person3	586.3	559.0	3210.2	. . .	normal
Person4	42.8	692.0	812.2	. . .	cancer

Learning Problems:

- Find a function: Class = f(A28202_ac, AB00014_at, AB00015_at, . . .)
- Given the expression level of genes of a Person, predict if he has cancer or not.

Splice junctions: Supervised Learning Task II

DNA: T G C A G C T C C G G A C T C C A T
mRNA: A C G U C G A G G C C U G A G G U A

- Exons: Sequences of nucleotides that are expressed (translated to proteins)
- Introns: Intercalated sequences eliminated during the translation Non-coding regions
- Splice-junctions: Frontiers between an exon and an intron
    - Donors: border exon-intron
    - Acceptors: border intron-exon

Learning Problem: Given a sequence of DNA, recognize the boundaries between exons and introns.

Illustrative Example: patient monitoring in ICU ...

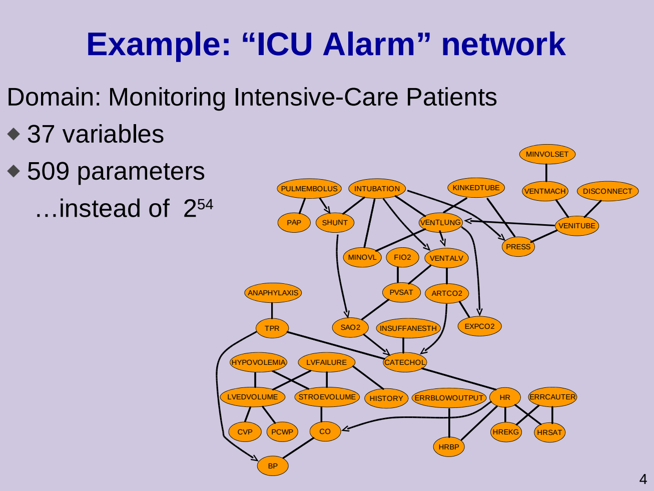
ALARM is a diagnostic application used to explore probabilistic reasoning techniques in belief networks. ALARM implements an alarm message system for patient monitoring; it calculates probabilities for a differential diagnosis based on available evidence. The medical knowledge is encoded in a graphical structure connecting 8 diagnoses, 16 finding and 13 intermediate variables.

Three types of variables are represented in ALARM. Diagnoses and other qualitative information are at the top level of the network. These variables have no predecessors and they are assumed to be mutually independent a priori. All nodes are associated with a set of mutually exclusive and exhaustive values representing the presence or absence or the severity of a particular disease. Measurements represent any available quantitative information. All continuous variables are represented categorically with sets of discrete intervals dividing the value range. Depending on the necessary level of detail, three to five categories are used per node. Intermediate variables are inferred entities that cannot be measured directly.

The probabilities in a belief network can represent objective as well as subjective knowledge. ALARM contains statistical data on prior probabilities, logical conditional probabilities computed from equations relating variables, and a number of subjective assessments. It is necessary to obtain conditional probabilities for the states of a node, given all different states of the parent nodes.

Illustrative Example: Qualitative model ...

A DAG that reflects the conditional independence between variables.



Example: “ICU Alarm” network

Domain: Monitoring Intensive-Care Patients

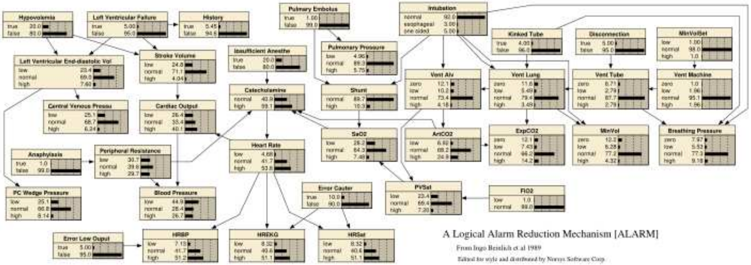
- 37 variables
- 509 parameters

…instead of 2^{54}

4

Illustrative Example: .. and Quantitative model

A set of contingency tables between dependent variables.



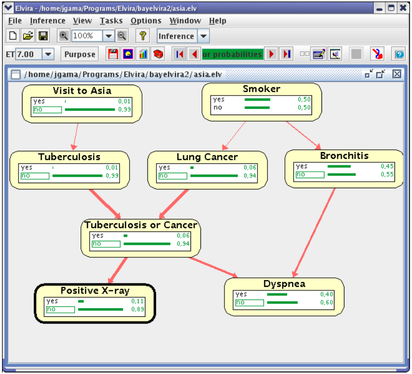
A Logical Alarm Reduction Mechanism [ALARM]

From Ingo Beinlich et al 1989

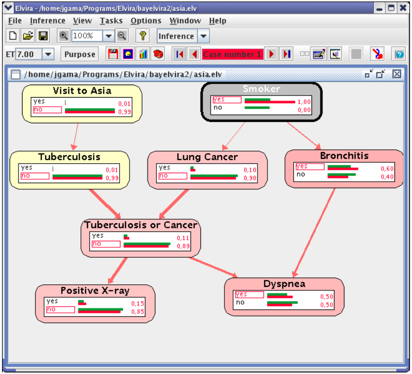
Edited for style and distributed by Norsys Software Corp.

Illustrative Example: Propagating Evidence ...

A Bayesian net:

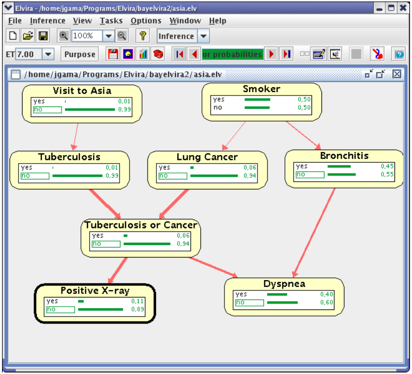


Prediction: Observing `Smoking` and propagating this evidence

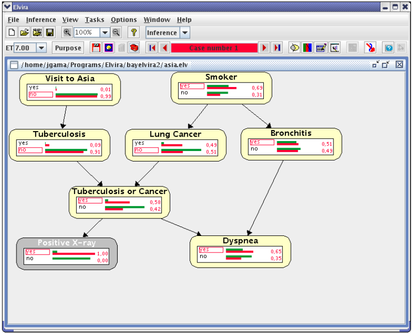


Illustrative Example: Propagating Evidence ...

A Bayesian net:



Diagnosis: Observing Positive X-ray and propagating this evidence

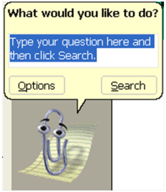


Another Interface ...

ERIC HORVITZ, a researcher at Microsoft and a guru in the field of Bayesian statistics, feels bad about the paperclip, but he hopes his latest creation will make up for it. The paperclip in question, as even casual users of Microsoft's Office software will be aware, is a cheery character who pops up on the screen to offer advice on writing a letter or formatting a spreadsheet. That was the idea, anyway. But many people regard the paperclip as annoyingly over-enthusiastic, since it appears without warning and gets in the way.

Mobile Manager evaluates incoming e-mails on a user's PC and decides which are important enough to forward to a pager, mobile phone or other e-mail address. Its Bayesian innards give it an almost telepathic ability to distinguish junk mail from genuinely important messages.

The *Clip* is an interface for a Bayesian Network:


What would you like to do?

Type your question here and then click Search.

Why Learn Bayesian Networks?

- Join Probability Distribution of a set of Variables.
- Conditional independences & graphical language capture structure of many real-world distributions
- Graph structure provides much insight into domain
- Learned model can be used for many tasks:
  - **Prediction**: Given the Inputs: which Outputs?
  - **Diagnosis**: Given the Outputs: which Inputs?
  - **Unsupervised**: Given Inputs and Outputs: Which structure?

Outline

1. Motivation: Information Processing
2. Introduction
3. Bayesian Network Classifiers
4. k-Dependence Bayesian Classifiers
5. Links and References

Bayes Theorem

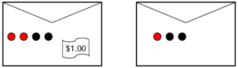
What is the most probable hypothesis h, given training data D?
A method to calculate the probability of a hypothesis based on

- its prior probability,
- the probability of observing the data given the hypothesis,
- The data itself

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

- $P(h) =$ 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

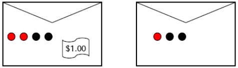
Illustrative Example



- The Win envelope has 1 dollar and four beads in it.
- The Lose envelope has three beads in it.

Someone draws an envelope at random and offers to sell it to you. Which one should we choose?

Illustrative Example



Before deciding, you are allowed to see one bead in one envelope.

- If it is black, Which one should we choose?
- And if it is red?

Illustrative Example

Prior Probabilities:
$P(Win) =$
$P(Lose) =$
$P(red) =$
$P(black) =$
$P(black|Win) =$
$P(red|Win) =$
After seeing the bead:
$P(Win|black) =$
$P(Win|red) =$

Illustrative Example

Prior Probabilities:

$P(Win) = 1/2$

$P(Lose) = 1/2$

$P(red) = 3/7$

$P(black) = 4/7$

$P(black|Win) = 1/2$

$P(red|Win) = 1/2$

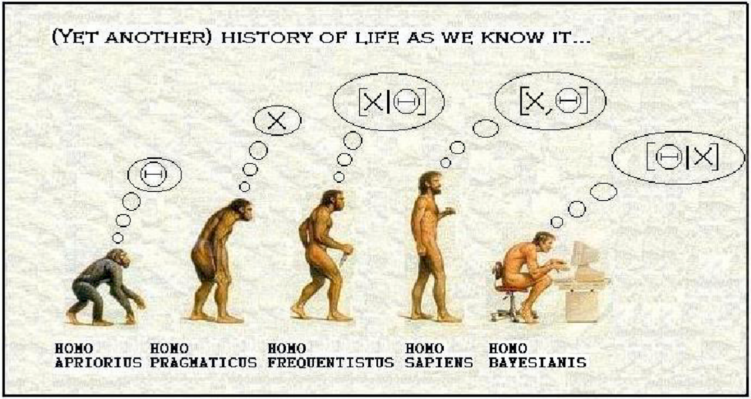
After seeing the bead:

If bead = black:

$$P(Win|black) = \frac{P(Win)P(black|Win)}{P(black)} = \frac{1/2*1/2}{4/7} = 0.4375$$

If bead = red: $P(Win|red) = \frac{P(Win)P(red|Win)}{P(red)} = \frac{1/2*1/2}{3/7} = 0.583$

Bayesians ...


(YET ANOTHER) HISTORY OF LIFE AS WE KNOW IT...
Θ
X
[X|Θ]
[X,Θ]
[Θ|X]
HOMO APRIORIUS
HOMO PRAGMATICUS
HOMO FREQUENTISTUS
HOMO SAPIENS
HOMO BAYESIANIS

Bayesians ...

- The Homo apriorius establishes the probability of an hypothesis, no matter what data tell. *Astronomical example: H0=25. What the data tell is uninteresting.*
- The Homo pragamiticus establishes that it is interested by the data only. *Astronomical example: In my experiment I found H0=66.6666666, full stop.*
- The Homo frequentistus measures the probability of the data given the hypothesis. *Astronomical example: if H0=66.666666, then the probability to get an observed value more different from the one I observed is given by an opportune expression. Don't ask me if my observed value is near the true one, I can only tell you that if my observed values is the true one, then the probability of observing data more extreme than mine is given by an opportune expression.*
- The Homo sapients measures the probability of the data and of the hypothesis. *Astronomical example: [missing]*
- The Homo bayesianis measures the probability of the hypothesis, given the data. *Astronomical example: the true value of H0 is near 72 with +-3 uncertainty.*

Stefano Andreon Homepage

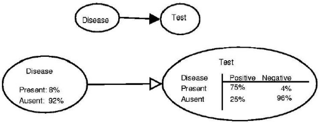
Outline

1. Motivation: Information Processing
2. Introduction
3. Bayesian Network Classifiers
4. k-Dependence Bayesian Classifiers
5. Links and References

An Illustrative Problem:

A patient takes a lab test and the result comes back positive. The test returns a correct positive result in only 75% of the cases in which the disease is actually present, and a correct negative result in only 96% of the cases in which the disease is not present. Furthermore, 8% of the entire population have this cancer.
How to represent that information?

Representation:



Disease	Positive	Negative
Present	75%	4%
Ausent	25%	96%

It is useful to represent this information in a graph.

- The graphical information is qualitative
- The nodes represent variables.
- Arcs specify the (in)dependence between variables. Direct arcs represent influence between variables.

The direction of the arc tell us that the value of the variable *disease* influences the value of the variable *test*.

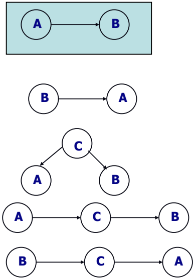
Inference:

Inference using Bayes Theorem:

$$P(Disease|Test =' Positive') =$$

$$= P(Disease)P(Test =' Positive'|Disease)$$

The Semantics of Arrows



Direction of Arrow indicates *Influence* not *causality*.
The ground truth might be different!

Naive Bayes Classifier

Assume target function $f : X \rightarrow Y$, where each instance x described by attributes $\langle x_1, x_2 \ldots x_n \rangle$.
Most probable value of $f(x)$ is:

$$
\begin{aligned}
Y_{MAP} &= \underset{y_j \in Y}{\operatorname{argmax}} P(y_j | x_1, x_2 \ldots x_n) \\
Y_{MAP} &= \underset{y_j \in Y}{\operatorname{argmax}} \frac{P(x_1, x_2 \ldots x_n | y_j) P(y_j)}{P(x_1, x_2 \ldots x_n)} \\
&= \underset{y_j \in Y}{\operatorname{argmax}} P(x_1, x_2 \ldots x_n | y_j) P(y_j)
\end{aligned}
$$

Naive Bayes Classifier

Naive Bayes assumption: Attributes are independent given the class.

$P(x_1, x_2 \ldots x_n | y_j) = \prod_i P(x_i | y_j)$ which gives Naive Bayes classifier:

$$Y_{NB} = \underset{y_j \in V}{\operatorname{argmax}} P(y_j) \prod_i P(x_i | y_j)$$

Naive Bayes

- Assume a decision problem with p variables.
- Each variable assume k values.
- The joint probability requires to estimate k^p probabilities.
- Assuming that variables are conditionally independent given the class, only requires to estimate $k \times p$ probabilities.

Naive Bayes Formulae

- Naive Bayes can be expressed in addictive form:

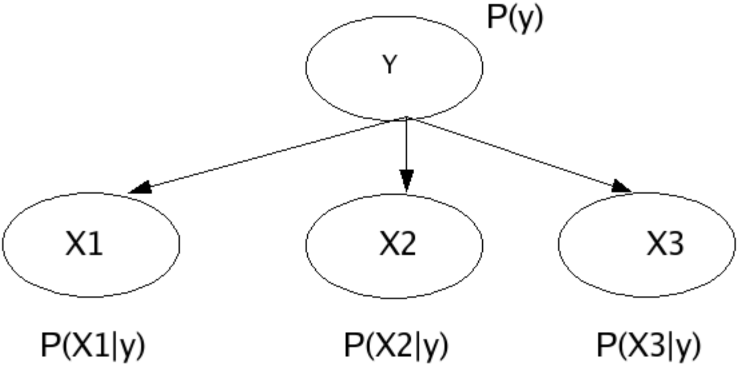
$$P(y_i|\vec{x}) \propto ln(P(y_i)) + \sum ln(P(x_j|y_i))$$

  - Points out the contribution of each attribute to the decision.
- Two class problems:

$$ln\frac{P(y_+|\vec{x})}{P(y_-|\vec{x})} \propto ln\frac{P(y_+)}{P(y_-)} + \sum ln\frac{P(x_j|y_+)}{P(x_j|y_-)}$$

The sign of each term indicates the class the attribute contributes to.

Naive Bayes as a Bayesian Net



$$p(Y|X1,X2,X3)=P(Y)P(X1|Y)P(X2|Y)P(X3|Y)$$

Naive Bayes Algorithm

```
Naive_Bayes_Learn($examples$)
    Initialize all counts to zero
    For each example $\{X, y_j\}$
        $Nr = Nr + 1$
        $Count(y_j) = Count(y_j) + 1$
        For each attribute value $x_i \in X$
            $Count(x_i, y_j) = Count(x_i, y_j) + 1$
```

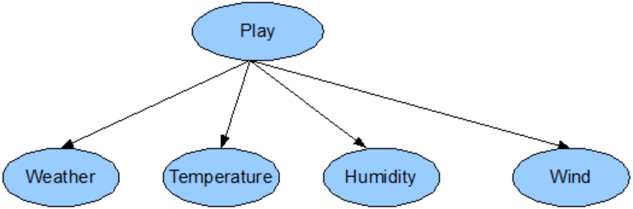
Classify_New_Instance(x)

$$y_{NB} = \operatorname*{argmax}_{y_j \in Y} \hat{P}(y_j) \prod_{x_i \in X} \hat{P}(x_i | y_j)$$

Naive Bayes: Example

Weather	Temperature	Humidity	Wind	Play
Rainy	71	91	Yes	No
Sunny	69	70	No	Yes
Sunny	80	90	Yes	No
Overcast	83	86	No	Yes
Rainy	70	96	No	Yes
Rainy	65	70	Yes	No
Overcast	64	65	Yes	Yes
Overcast	72	90	Yes	Yes
Sunny	75	70	Yes	Yes
Rainy	68	80	No	Yes
Overcast	81	75	No	Yes
Sunny	85	85	No	No
Sunny	72	95	No	No
Rainy	75	80	No	Yes

Two representations:



$$P(play|weather, temperature, humidity, wind) =$$

$$= P(Play)P(Weather|Play)P(Temperature|Play)P(Humidity|Play)P(Wind|Pla$$

Naive Bayes: Counts

Nr. examples: 14

$Play =' Yes' : 9$

$Play =' No' : 5$

Weather			Temperature			Humidity			Wind		
	Yes	No		Yes	No		Yes	No		Yes	No
Sunny	2	3	Mean	73	74.6	Mean	79.1	86.2	False	6	2
Overcast	4	0	SD	6.2	7.9	SD	10.2	9.3	True	3	3
Rainy	3	2									

Naive Bayes: Distribution Tables

Nr. examples: 14

$P(Play =' Yes') = 9/14$
$P(Play =' No') = 5/14$

Weather			Temperature			Humidity			Wind		
	Yes	No		Yes	No		Yes	No		Yes	No
Sunny	2/9	3/5	Mean	73	74.6	Mean	79.1	86.2	False	6/9	2/5
Overcast	4/9	0/5	SD	6.2	7.9	SD	10.2	9.3	True	3/9	3/5
Rainy	3/9	2/5									

Continuous attributes can be approximated using normal distribution: $N(x) = \frac{1}{\sqrt{2\pi\sigma}} e - \frac{(x-\mu)^2}{2\sigma^2}$

Naive Bayes: Test Example

Weather	Temperature	Humidity	Wind	Play
Sunny	66	90	Yes	?

$P(Yes|Weather = \text{‘}Sunny', Temperature = 66, Humidity = 90, Wind = \text{‘}Yes') =$
$= P(Yes)P(Weather = \text{‘}Sunny'|Yes)P(Temperature = 66|Yes)P(Humidity =$
$90|Yes)P(Wind = \text{‘}Yes'|Yes)$

$P(No|Weather = \text{‘}Sunny', Temperature = 66, Humidity = 90, Wind = \text{‘}Yes') =$

$= P(No)P(Weather = \text{“}Sunny''|No)P(Temperature = 66|No)P(Humidity =$

$90|No)P(Wind = \text{“}Yes''|No)$

Naive Bayes: Subtleties

- what if none of the training instances with target value y_j have attribute value x_i? Then $\hat{P}(x_i|y_j) = 0$ and $\hat{P}(y_j) \prod_i \hat{P}(x_i|y_j) = 0$

Typical solution is Bayesian estimate for $\hat{P}(x_i|y_j)$
$\hat{P}(x_i|y_j) \leftarrow \frac{n_c+mp}{n+m}$ where

- n is number of training examples for which $y = y_j$,
- n_c number of examples for which $Y = y_j$ and $X = x_i$
- p is prior estimate for $\hat{P}(x_i|y_j)$
- m is weight given to prior (i.e. number of "virtual" examples)

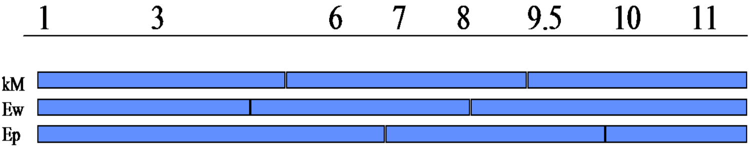
Discretization

Transform a continuous variable into a set of ordered intervals (bins).
Two Main Problems:

- How many Intervals?
  Generic: Use 10 intervals or $min(10, nr.\ of\ different\ values)$
  Bioinformtics: few intervals (2, 3, ...)
- How to define to borders of each Interval?

Discretization: Basic Methods

- **Equal width discretization**. Divides the range of observed values for a feature into k equally sized bins. Pos: simplicity, Neg: the presence of outliers.
- **Equal frequency discretization**. Divides the range of observed values into k bins, where (considering n instances) each bin contains n/k values.
- **k-means**. An iterative method that begins with an equal-width discretization, iteratively adjust the boundaries to minimize a squared-error function and only stops when it can not change any value to improve the previous criteria.



Naive Bayes: Analysis

1. Conditional independence assumption is often violated

$$P(x_1, x_2 \ldots x_n | y_j) = \prod_i P(x_i | y_j)$$

- ...but it works surprisingly well anyway. Note don't need estimated posteriors $\hat{P}(y_j|x)$ to be correct; need only that

$$\operatorname*{argmax}_{y_j \in Y} \hat{P}(y_j) \prod_i \hat{P}(x_i|y_j) = \operatorname*{argmax}_{y_j \in Y} P(y_j) P(x_1 \ldots, x_n|y_j)$$

- see [Domingos & Pazzani, 1996] for analysis
- Naive Bayes posteriors often unrealistically close to 1 or 0

Naive Bayes: Analysis

- Robust to the presence of irrelevant attributes.
  Suppose a two class problem, where X_i is an irrelevant attribute: $p(x_i|y_1) = p(x_i|y_2)$.
  $p(Y|x_1, ..., x_i, ..., x_n) \propto$
  $p(Y)p(x_i|c)\prod_{l=1}^{i-1} p(x_l|Y)\prod_{l=i+1}^{n} p(x_l|Y)$ and
  $p(Y|x_1, ..., x_i, ..., x_n) \propto p(Y|x_1, ..., x_{i-1}, x_{i+1}, ..., x_n)$
- Redundant variables must be taken into account.
  Suppose that $X_i = X_{i-1}$ then $p(x_{i-1}|Y) = p(x_i|Y)$
  $p(Y|x_1, ..., x_{i-1}, x_i, ..., x_n) \propto$
  $p(Y)p(x_{i-1}|Y)p(x_i|Y)\prod_{l=1}^{i-2} p(x_l|Y)\prod_{l=i+1}^{n} p(x_l|Y)$
  and
  $p(Y|x_1, ..., x_{i-1}, x_i, ..., x_n) \propto$
  $p(Y)p(x_i|Y)^2\prod_{l=1}^{i-2} p(x_l|Y)\prod_{l=i+1}^{n} p(x_l|Y)$

Naive Bayes: Summary

- The variability of a dataset is summarized in contingency tables.
    - Requires a single scan over the dataset.
    - The algorithm is Incremental (incorporation of new examples) and decremental (forgetting old examples).
- The dimension of the decision model is independent of the number of examples.
    - Low variance: stable with respect to small perturbations of the training set.
    - High bias: the number of possible states is finite.

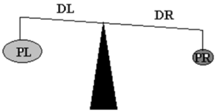
Naive Bayes: Extensions

- Techniques that apply different naive Bayes classifiers to different regions of the input space. Recursive Bayes (Langley, 93), Naive Bayes Tree (Kohavi,96).
- Techniques that built new attributes that reflect interdependencies between original attributes. Semi-naive Bayes (Kononenko, 91).
- Processing continuous attributes: Flexible Bayes (G.John), Linear Bayes (Gama, 01)
- Search for better Parameters: Iterative Bayes (Gama, 99)

Successful Stories

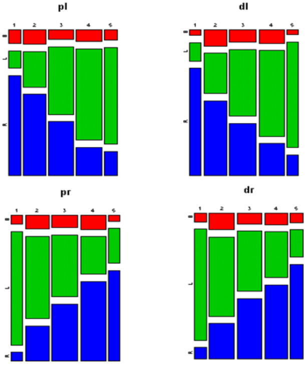
- KDDCup 1998: Winner - Boosting naive Bayes;
- Coil 1999: Winner - Simple naive Bayes;
- The most used classifier in Text Mining;

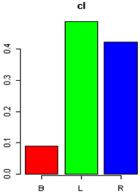
The Balance-scale Problem



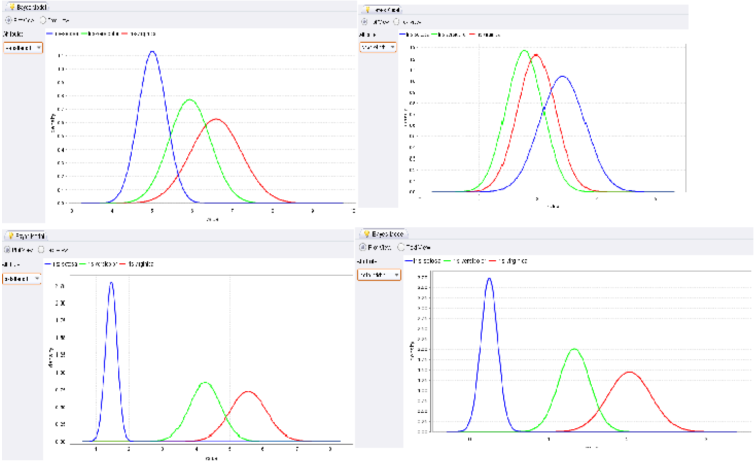
A balança pode estar em três estados:
.Equilibrada (B)
.Inclinada para a direita (R)
.Inclinada para a esquerda (L)

	pl	dl	pr	dr	cl
1	4	1	3	2	R
2	3	3	4	3	R
3	5	1	4	2	R
4	3	4	5	4	R
5	5	4	2	3	L
6	3	4	3	2	L
7	5	3	2	4	L
8	4	4	4	1	L

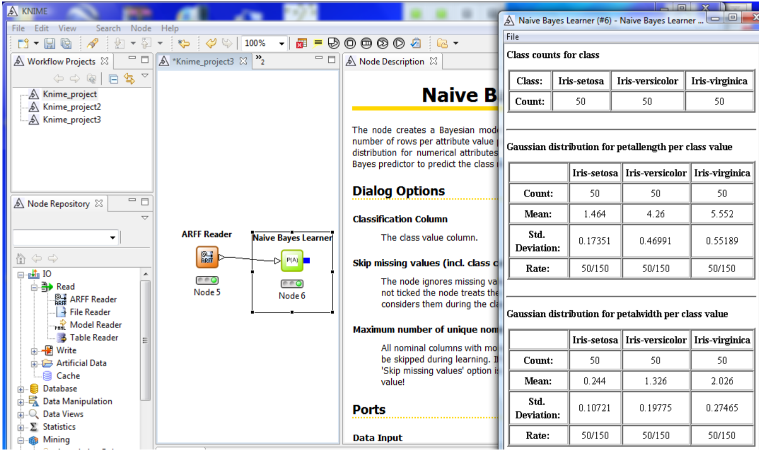




Rapid Miner: Naive Bayes



Knimer: Naive Bayes



Naive B

The node creates a Bayesian mod
number of rows per attribute value
distribution for numerical attributes
Bayes predictor to predict the class

Dialog Options

Classification Column

The class value column.

Skip missing values (incl. class c

The node ignores missing va
not ticked the node treats the
considers them during the cla

Maximum number of unique nom

All nominal columns with mo
be skipped during learning. I
'Skip missing values' option i
value!

Ports

Data Input

Class counts for class

Class:	Iris-setosa	Iris-versicolor	Iris-virginica
Count:	50	50	50

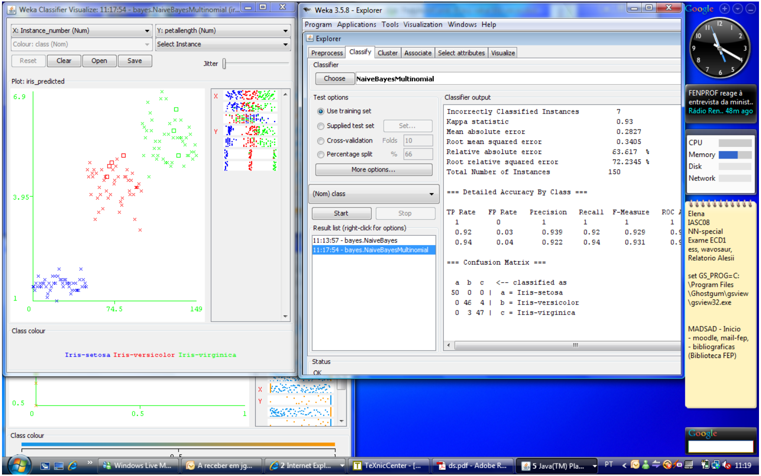
Gaussian distribution for petallength per class value

	Iris-setosa	Iris-versicolor	Iris-virginica
Count:	50	50	50
Mean:	1.464	4.26	5.552
Std. Deviation:	0.17351	0.46991	0.55189
Rate:	50/150	50/150	50/150

Gaussian distribution for petalwidth per class value

	Iris-setosa	Iris-versicolor	Iris-virginica
Count:	50	50	50
Mean:	0.244	1.326	2.026
Std. Deviation:	0.10721	0.19775	0.27465
Rate:	50/150	50/150	50/150

Weka: Naive Bayes



Classifier: NaiveBayesMultinomial

Test options

- ● Use training set
- ○ Supplied test set
- ○ Cross-validation Folds 10
- ○ Percentage split % 66

(Nom) class

Result list (right-click for options)

- 11:13:57 - bayes.NaiveBayes
- 11:17:54 - bayes.NaiveBayesMultinomial

Classifier output

```
Incorrectly Classified Instances         7
Kappa statistic                          0.93
Mean absolute error                      0.2827
Root mean squared error                  0.3405
Relative absolute error                 63.617  %
Root relative squared error             72.2345 %
Total Number of Instances              150

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   ROC A
  1         0          1         1         1         1
  0.92      0.03       0.939     0.92      0.929     0.9
  0.94      0.04       0.922     0.94      0.931     0.9

=== Confusion Matrix ===

  a  b  c   <-- classified as
 50  0  0 |  a = Iris-setosa
  0 46  4 |  b = Iris-versicolor
  0  3 47 |  c = Iris-virginica
```

Status
OK

Elena
IASC08
NN-special
Exame ECD1
ess, wavosaur,
Relatorio Alesii

set GS_PROG=C:
\Program Files
\Ghostgum\gsview
\gsview32.exe

MADSAD - Inicio
- moodle, mail-fep,
- bibliograficas
(Biblioteca FEP)

Outline

1 Motivation: Information Processing

2 Introduction

3 Bayesian Network Classifiers

4 k-Dependence Bayesian Classifiers

5 Links and References

Mutual Information

Uncertainty of a random Variable (Entropy):

$$H(X) = -\sum P(x) log_2(P(x))$$

Uncertainty about X after knowing Y (Conditional Entropy):

$$H(X|Y) = -\sum_{y} P(y) \sum_{x} P(x|y) log P(x|Y)$$

Reduction in the uncertainty of X when Y is known (Mutual Information):

$$I(X, Y) = H(X|Y) - H(X)$$

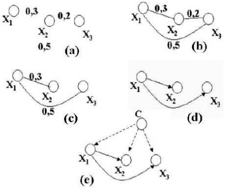
$$I(X, Y) = \sum_{i} \sum_{j} P(x_i, y_j) log \frac{P(x_i, y_j)}{P(x_i) p(x_j)}$$

Example: TAN

Tree augmented naive Bayes

- Compute the Mutual Information between all pairs of variables given the Class

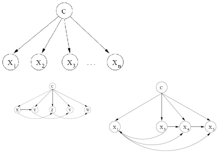
$$I(X,Y|C) = \sum_{c} P(c) I(X,Y|C=c)$$



- Construct a spanning tree maximizing MI.
- All the variables depends on the class.

$$p(c|x_1,x_2,x_3) \propto p(c)p(x_1|c)p(x_2|x_1,c)p(x_3|x_1,c)$$

k-Dependency Bayesian Networks



- All the attributes depends on the class
- Any attribute depends on k other attributes, at most.

- Restricted Bayesian Networks
- Decision Models of Increase Complexity

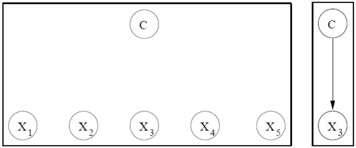
$$p(c|x_1, x_2, x_3) \propto p(c)p(x_1|c)p(x_2|x_1, c)p(x_3|x_1, c)$$

k-Dependency Bayesian Networks: a common framework for classifiers with increase (smooth) complexity.

k-Dependency Bayesian Networks

- Compute $I(X_i, C)$ and $I(X_i, X_j|C)$ for all pairs of variables
- Iterate
  - Choose the variable X_{max} not yet in the model that maximizes $I(X_i|C)$
  - Choose the k parents of X_{max}: those with greater $I(X_j, X_{max}|C)$

k-Dependency Bayesian Networks

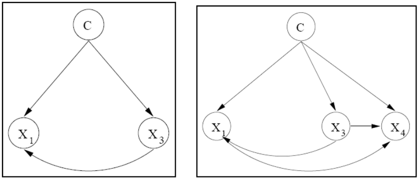


kDB $k = 2.$

$$I(X_3, C) > I(X_1, C) > I(X_4, C) > I(X_5, C) > I(X_2, C)$$

$$I(X_3, X_4|C) > I(X_2, X_5|C) > I(X_1, X_3|C) > I(X_1, X_2|C) > I(X_2, X_4|C) >$$
$$I(X_2, X_3|C) > I(X_1, X_4|C) > I(X_4, X_5|C) > I(X_1, X_5|C) > I(X_3, X_5|C)$$

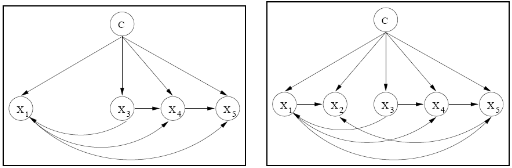
k-Dependency Bayesian Networks



kDB $\quad k = 2.$

$$I(X_3, C) > I(X_1, C) > I(X_4, C) > I(X_5, C) > I(X_2, C)$$
$$I(X_3, X_4|C) > I(X_2, X_5|C) > I(X_1, X_3|C) > I(X_1, X_2|C) > I(X_2, X_4|C) >$$
$$I(X_2, X_3|C) > I(X_1, X_4|C) > I(X_4, X_5|C) > I(X_1, X_5|C) > I(X_3, X_5|C)$$
$$p(c|x_1, x_2, x_3, x_4, x_5)$$

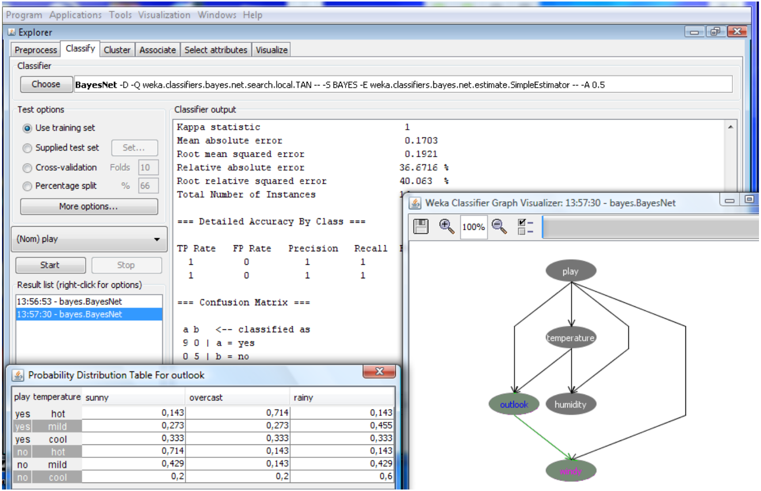
k-Dependency Bayesian Networks



kDB $k = 2.$

$$I(X_3, C) > I(X_1, C) > I(X_4, C) > I(X_5, C) > I(X_2, C)$$
$$I(X_3, X_4|C) > I(X_2, X_5|C) > I(X_1, X_3|C) > I(X_1, X_2|C) > I(X_2, X_4|C) >$$
$$I(X_2, X_3|C) > I(X_1, X_4|C) > I(X_4, X_5|C) > I(X_1, X_5|C) > I(X_3, X_5|C)$$
$$p(c|x_1, x_2, x_3, x_4, x_5) \propto$$
$$p(c)p(x_1|x_3, c)p(x_2|x_1, x_5, c)p(x_3|c)p(x_4|x_1, x_3, c)p(x_5|x_1, x_4, c)$$

Weka: TAN



Classifier: BayesNet -D -Q weka.classifiers.bayes.net.search.local.TAN -- -S BAYES -E weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A 0.5

Test options

- ● Use training set
- ○ Supplied test set
- ○ Cross-validation Folds 10
- ○ Percentage split % 66

(Nom) play

Result list (right-click for options)

- 13:56:53 - bayes.BayesNet
- 13:57:30 - bayes.BayesNet

Classifier output

```
Kappa statistic                          1
Mean absolute error                      0.1703
Root mean squared error                  0.1921
Relative absolute error                 36.6716 %
Root relative squared error             40.063  %
Total Number of Instances               1

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall
  1         0          1          1
  1         0          1          1

=== Confusion Matrix ===

 a b   <-- classified as
 9 0 | a = yes
 0 5 | b = no
```

Weka Classifier Graph Visualizer: 13:57:30 - bayes.BayesNet

Probability Distribution Table For outlook

play	temperature	sunny	overcast	rainy
yes	hot	0,143	0,714	0,143
yes	mild	0,273	0,273	0,455
yes	cool	0,333	0,333	0,333
no	hot	0,714	0,143	0,143
no	mild	0,429	0,143	0,429
no	cool	0,2	0,2	0,6

Outline

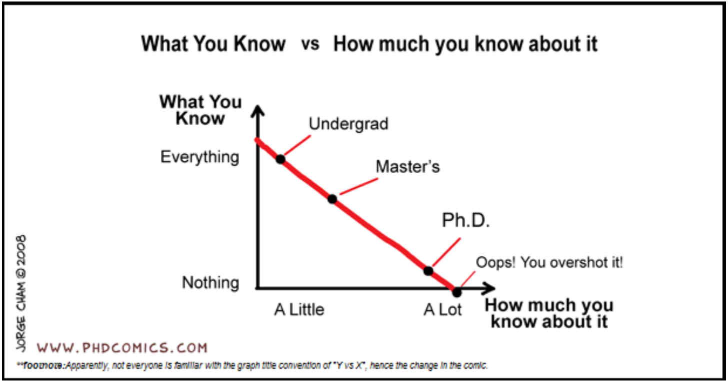
1. Motivation: Information Processing
2. Introduction
3. Bayesian Network Classifiers
4. k-Dependence Bayesian Classifiers
5. Links and References

Software Available

- R (package e1071)
- Weka (naive Bayes, TAN models, k-dependence Bayesian classifiers)
- Genie: Bayesian Networks, Influence Diagrams, Dynamic Bayesian Networks (http://genie.sis.pitt.edu/)
- Hugin (http://www.hugin.com/)
- Elvira (http://www.ia.uned.es/ elvira)
- Kevin Murphy's MATLAB toolbox - supports dynamic BNs, decision networks, many exact and approximate inference algorithms, parameter estimation, and structure learning
  (http://www.ai.mit.edu/ murphyk/Software/BNT/bnt.html)
- Free Windows software for creation, assessment and evaluation of belief networks.
  (http://www.research.microsoft.com/dtas/msbn/default.htm)
- Open source package for the technical computing language R, developed by Aalborg University
  (http://www.math.auc.dk/novo/deal)
- http://www.snn.ru.nl/nijmegen

Bibliography

- Tom Mitchell, *Machine Learning*, (chapter 6), McGraw-Hill, 1997
- R. Duda, P. Hart, D. Stork; *Pattern Classification*, J. Willey & Sons, 2000
- J.Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, 1988
- P. Domingos, M.Pazzani; *On the Optimality of the Simple Bayes Classifier under zero-one loss*, Machine Learning, 29
- R. Neapolitan, *Learning Bayesian Networks*, Prentice Hall, 2004


What You Know vs How much you know about it
What You Know
Everything
Nothing
Undergrad
Master's
Ph.D.
Oops! You overshot it!
A Little
A Lot
How much you know about it
JORGE CHAM © 2008
WWW.PHDCOMICS.COM
**footnote: Apparently, not everyone is familiar with the graph title convention of "Y vs X", hence the change in the comic.

Would you like to learn more? Wait for SAD ...