Bayesian Learning: An Introduction

João Gama

LIAAD-INESC TEC, University of Porto, Portugal
jgama@fep.up.pt

September 2020

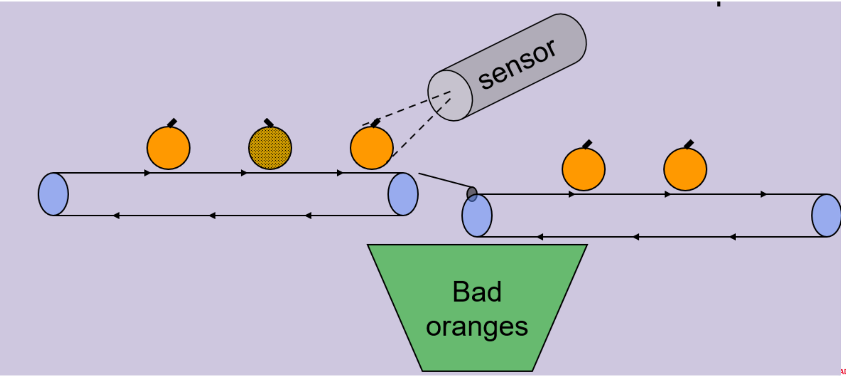
1. Motivation
2. Introduction
3. Bayesian Network Classifiers
4. k-Dependence Bayesian Classifiers
5. Links and References

Outline

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

A Case Study

- Suppose we are required to build a controller that removes bad oranges from a packaging line
- Decision are made based on a sensor that reports the overall color of the orange



Classifying Oranges I

Suppose we know all the aspects of the problem:

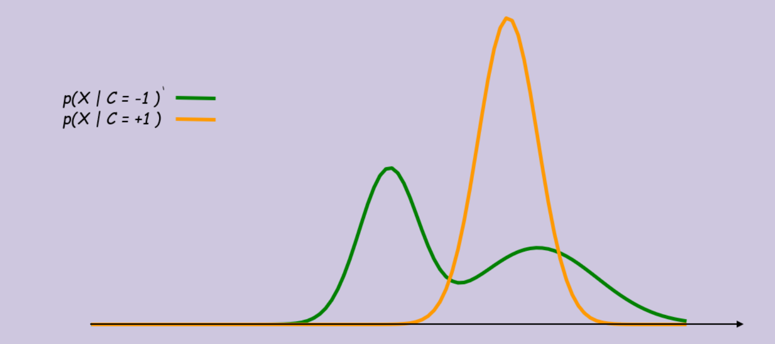
Prior Probabilities:

- Probability of good (+1) and bad (-1) oranges
  - $P(C = +1)$ = probability of a good orange
  - $P(C = -1)$ = probability of a bad orange
    - Note: $P(C = +1) + P(C = -1) = 1$
- Assumption: oranges are independent
  The occurrence of a bad orange does not depend on previous

Classifying Oranges II

Sensor performance:

- Let X denote sensor measurement from each type of oranges



Bayes Rule

- Given this knowledge, we can compute the posterior probabilities

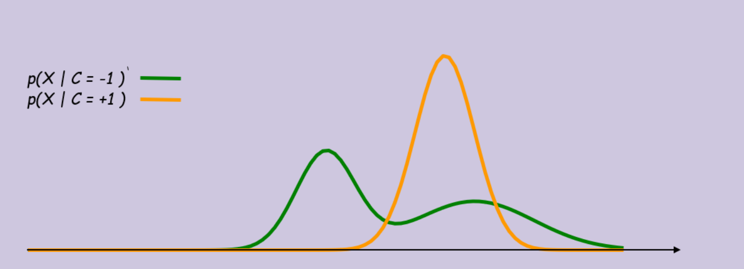
Bayes Rule:

$$P(C|X = x) = \frac{P(C) \times P(X = x|C)}{P(X = x)}$$

$$P(X = x) = P(C = +1) \times P(X = x|C = +1) + P(C = -1) \times P(X = x|C = -1)$$

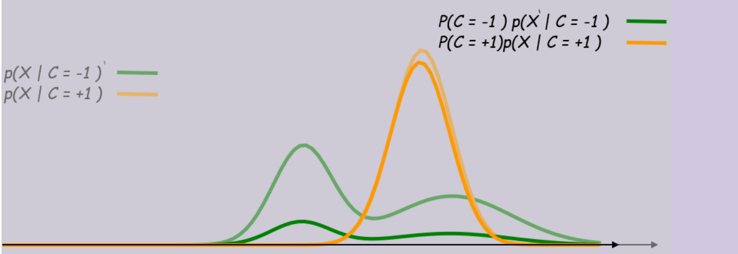
Posterior of Oranges

Data likelihood ...



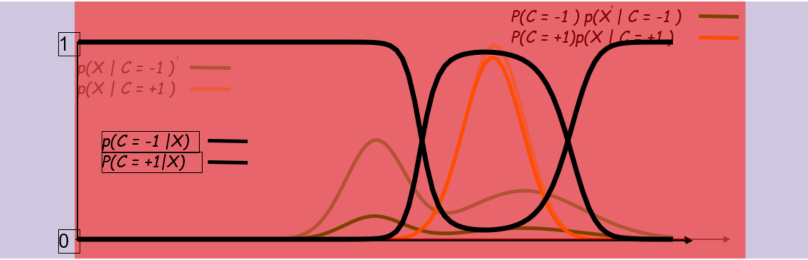
Posterior of Oranges

Data likelihood ...Combined with prior



Posterior of Oranges

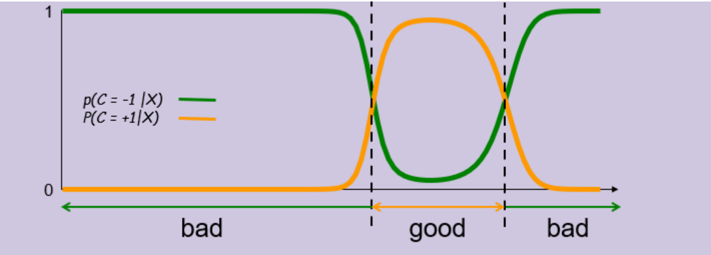
Data likelihood ...Combined with prior...and Normalized



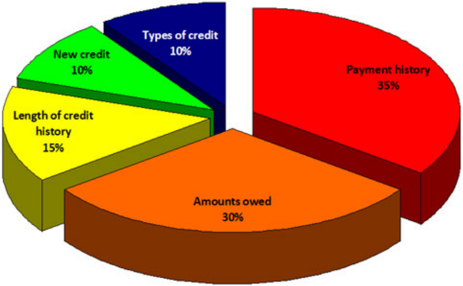
Decision making

Intuition:

- Predict `Good` if $P(C = +1|X) > P(C = -1|X)$
- Predict `Bad`, otherwise



Credit applications:



Motivation: Supervised Learning Task

- Given:
  - Historical data about clients credit operations
    - Characteristics of the clients
    - Characteristics of the credit application
  - The success of the operation constitutes the class
- Goal: Learn a model that accurately predicts the success (or not) of new credit applications

Credit data: Clients are Data Points

Historical data:

Client	amount	age	salary	account
Client 1	medium	junior	low	yes
Client 2	medium	junior	low	no
Client 3	low	junior	low	yes
Client 4	high	medium	low	yes
Client 5	high	senior	high	yes
Client 6	high	senior	high	no

Supervision: Add Class Values

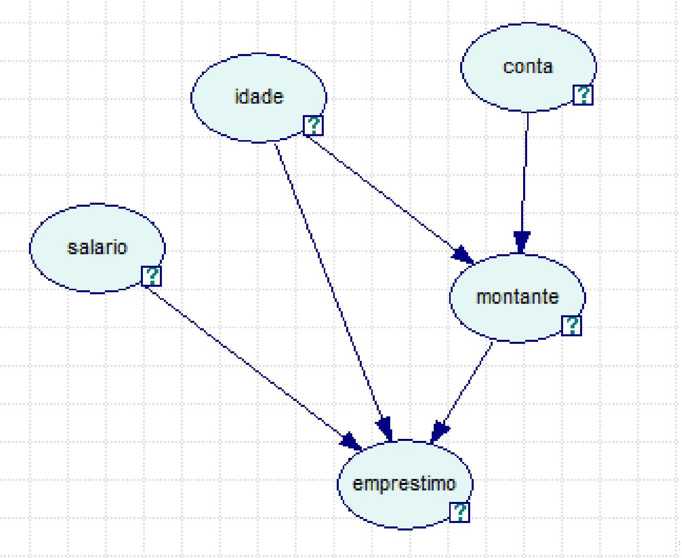
Client	amount	age	salary	account	Loan
Client 1	medium	junior	low	yes	no
Client 2	medium	junior	low	no	no
Client 3	low	junior	low	yes	yes
Client 4	high	medium	low	yes	yes
Client 5	high	senior	high	yes	yes
Client 6	high	senior	high	no	no

Learning Problems:

- Find a function:
  Loan = f(amount,age,salary,account)
- Given the characteristics of a client, predict if the application will succeed or not.

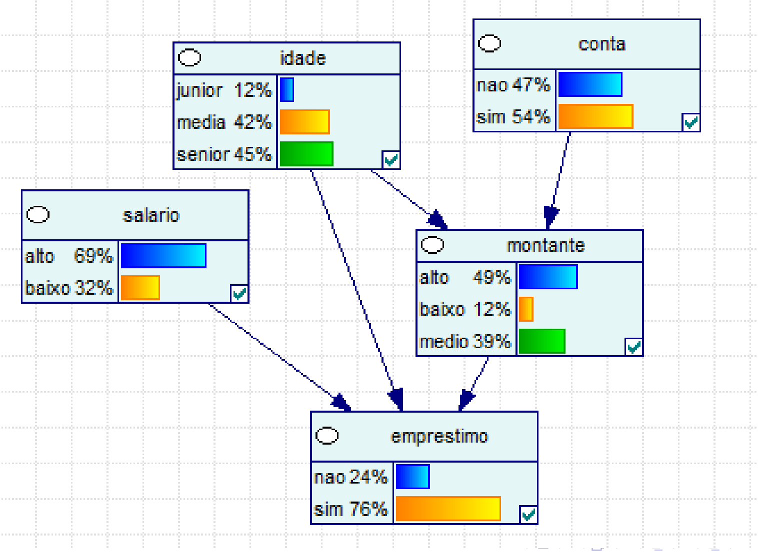
Qualitative model ...

A DAG: reflects the conditional independence between variables.

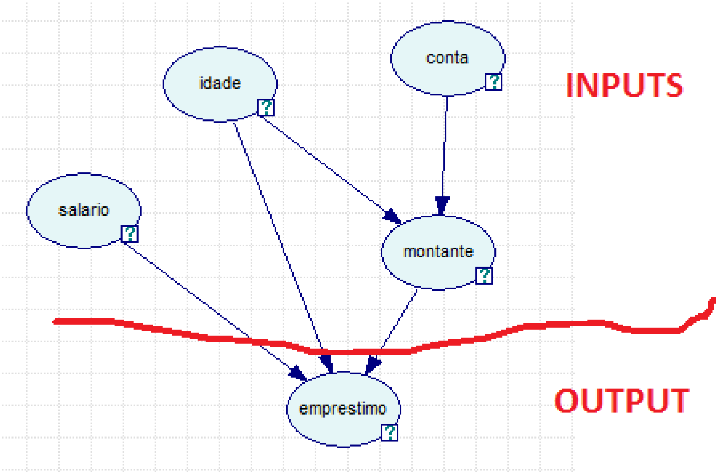


... and Quantitative model

A set of contingency tables between dependent variables.

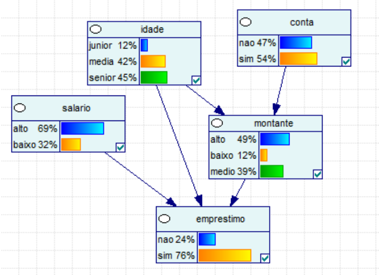


Causal Model: inputs and outputs.

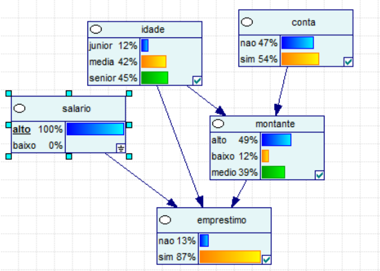


Propagating Evidence ...

A Bayesian net:

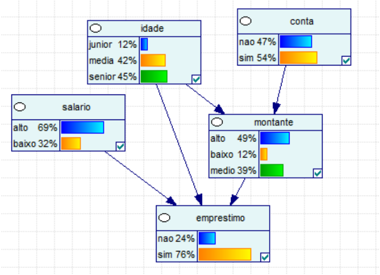


Prediction: Observing `Inputs` and propagating this evidence

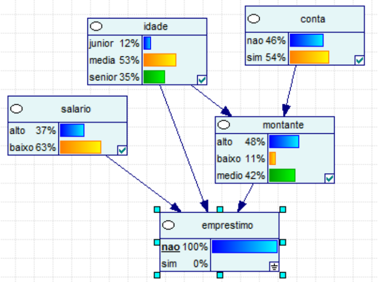


Propagating Evidence ...

A Bayesian net:



Diagnosis: Observing `Outputs` and propagating this evidence



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

2 Introduction

3 Bayesian Network Classifiers

4 k-Dependence Bayesian Classifiers

5 Links and References

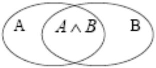
Basics on Statistics

Axioms:

- All probabilities between 0 and 1
  $0 \leq P(A) \leq 1$
- True proposition has probability 1, false has probability 0.
  $P(true) = 1$ and $P(false) = 0$
- The probability of a disjunction is:
  $P(A \vee B) = P(A) + P(B) - P(A \wedge B)$

Conditional Probability:

- $P(A|B)$ is the probability of A given B
  Assumes that B is all and only information known.
- Defined by: $P(A|B) = \frac{P(A \wedge B)}{P(B)}$



Bayes Theorem

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

Combine the prior distribution and the likelihood of the observed data in order to derive the posterior distribution.

Proof:

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

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

$$P(h \wedge D) = P(D|h)P(h)$$

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

Bayes Theorem

What is the most probable hypothesis h, given training data D?

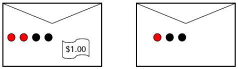
$$P(h|D) = \frac{P(h)P(D|h)}{P(D)}$$

Computing the probability of a hypothesis based on:

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

- $P(h)$ = 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

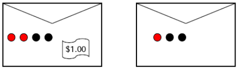
Illustrative Example



- 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:

$P(Win) = 1/2$

$P(Lose) = 1/2$

$P(red) = 3/7$

$P(black) = 4/7$

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

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

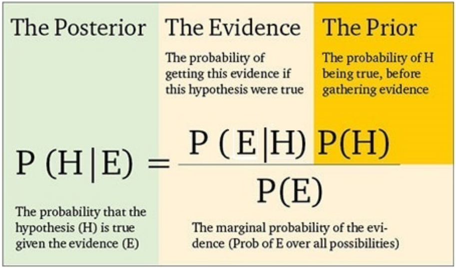
After seeing the bead:

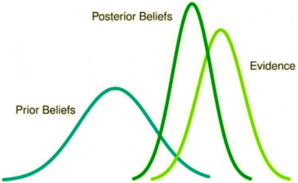
If bead = black:

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

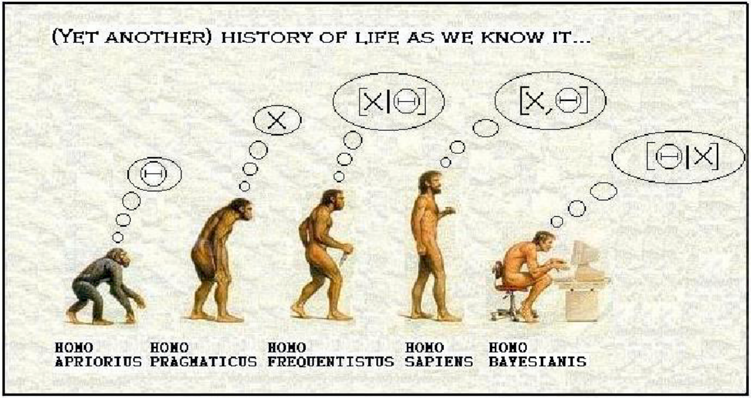
If bead = red: $P(Win|red) = \frac{P(Win)P(red|Win)}{P(red)} = \frac{1/2*1/2}{3/7} = 0.583$

Bayes Theorem ...





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
2. Introduction
3. Bayesian Network Classifiers
4. k-Dependence Bayesian Classifiers
5. Links and References

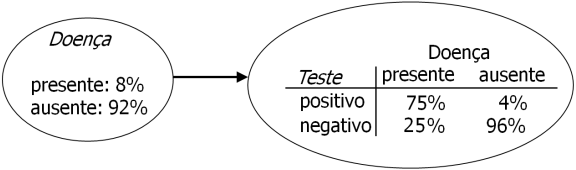
An Illustrative Problem:

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:



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*.

Naive Bayes Classifier

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

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

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.

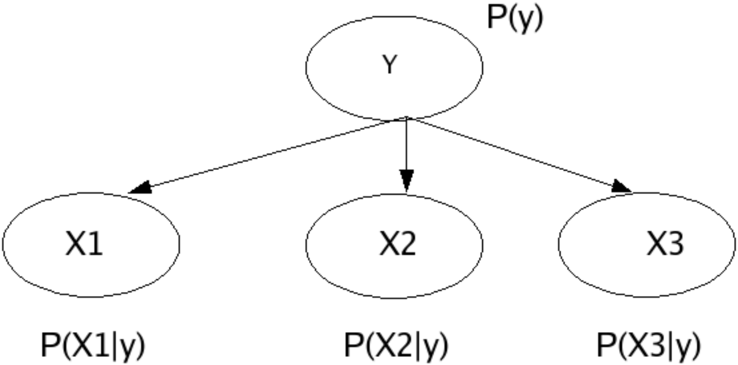
- Two class problems:

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

$$ln\frac{P(y_+|\vec{x})}{P(y_-|\vec{x})} \propto ln\frac{P(y_+)}{1-P(y_+)} + \sum ln\frac{P(x_j|y_+)}{1-P(x_j|y_+)}$$

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 $x_i \in X$

$Count(x_i, y_j) = Count(x_i, y_j) + 1$

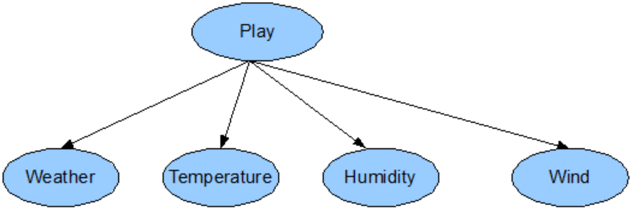
Classify_New_Instance(x)

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

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:



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

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

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: $N(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

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)

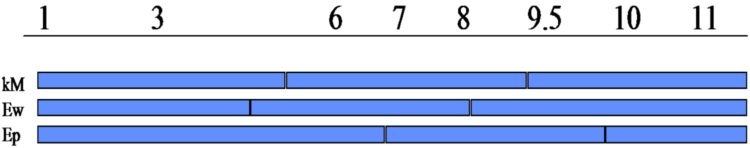
Discretization

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.

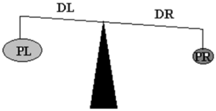
Naive Bayes: Extensions

- 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

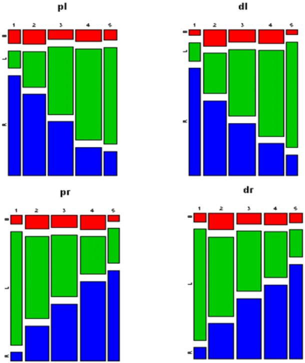
- KDDCup 1998: Winner - Boosting naive Bayes;
- Coil 1999: Winner - Simple naive Bayes;
- The most used classifier in Text Mining;

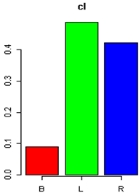
The Balance-scale Problem



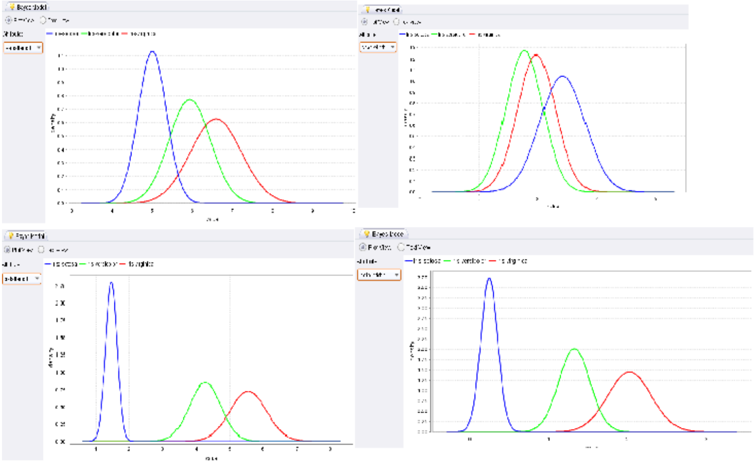
A balança pode estar em três estados:
.Equilibrada (B)
.Inclinada para a direita (R)
.Inclinada para a esquerda (L)

	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

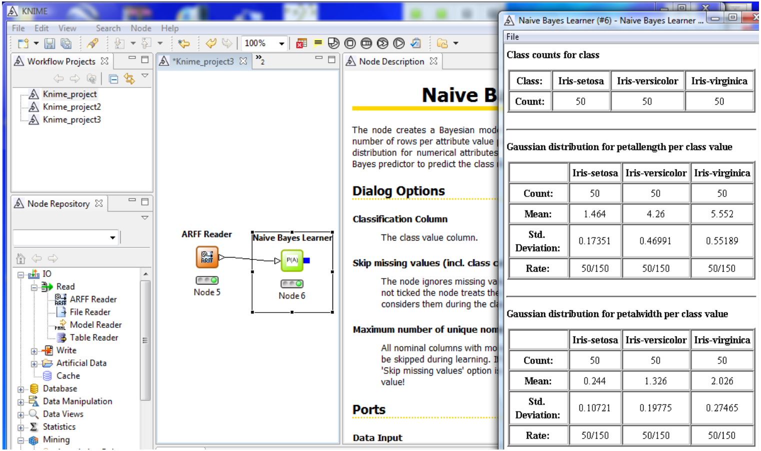




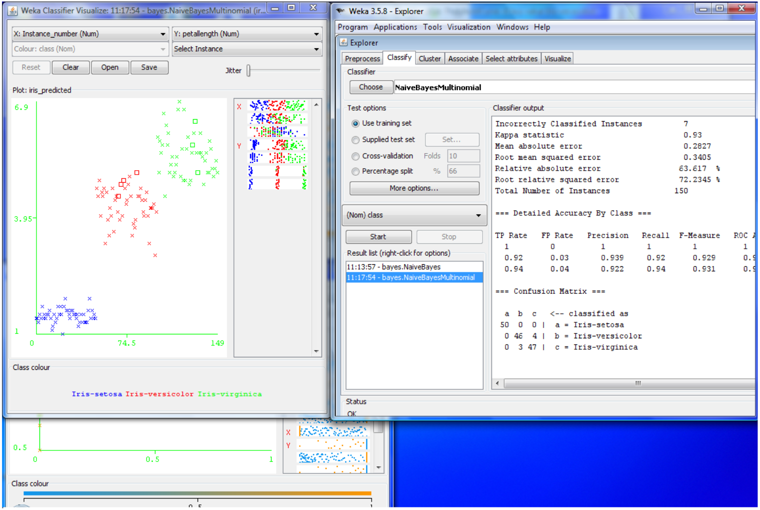
Rapid Miner: Naive Bayes



Knime: Naive Bayes



Weka: Naive Bayes



Outline

1 Motivation

2 Introduction

3 Bayesian Network Classifiers

4 k-Dependence Bayesian Classifiers

5 Links and References

Entropy

Entropy: measures the randomness of a random variable.
Uncertainty of a random discrete variable (Entropy):

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

Conditional Entropy

conditional entropy quantifies the remaining entropy (i.e. uncertainty) of a random variable Y given that the value of a second random variable X is known.

Uncertainty about X after knowing Y:

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

Mutual Information

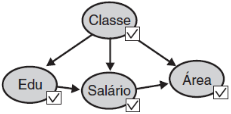
Reduction in the uncertainty of X when Y is known:

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

$$I(X,Y) = \sum_i \sum_j P(x_i, y_j) log \frac{P(x_i, y_j)}{P(x_i)p(x_j)}$$

- $I(X,Y) = 0$ if and only if X and Y are independent random variables;
- I(X,Y) increases with the increase of the degree of dependence between X and Y;
- nonnegative: $I(X,Y) \geq 0$;
- symmetric: $I(X,Y) = I(Y,X)$.

Example: TAN



	Classe	Não		Sim	
	Edu	Não	Sim	Não	Sim
Salário	Baixo	0,6	0,4	0,33	0,66
	Alto	0,4	0,4	0,66	0,33

Tree augmented naive Bayes

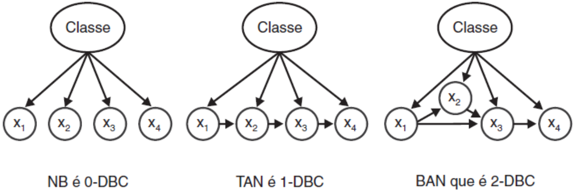
- Compute the Mutual Information between all pairs of variables given the Class

$$I(X, Y|C) = \sum_c P(c)I(X, Y|C = c)$$

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

$$p(C, E, S, A) \propto p(C)p(E|C)p(S|C, E)p(A|C, S)$$

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

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

k-Dependency Bayesian Networks

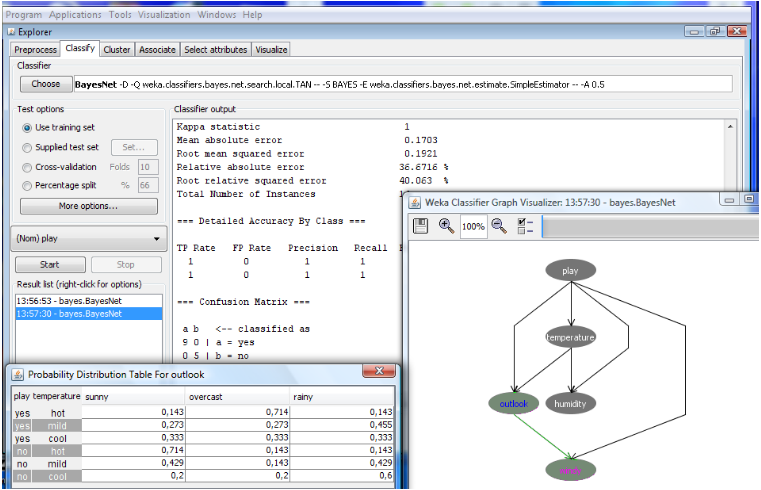
- Compute $I(X_i, C)$ and $I(X_i, X_j|C)$ for all pairs of variables
- Iterate
  - Choose the variable X_{max} not yet in the model that maximizes $I(X_i|C)$
  - Choose the k parents of X_{max}: those with greater $I(X_j, X_{max}|C)$

Factorization:

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|Parents(X_i))$$

where $Parents(X_i)$ denotes immediate predecessors of X_i in graph

Weka: TAN


Program Applications Tools Visualization Windows Help
Explorer
Preprocess Classify Cluster Associate Select attributes Visualize
Classifier
Choose
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
Set...
Cross-validation Folds 10
Percentage split % 66
More options...
(Nom) play
Start
Stop
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
=== 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
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
Weka Classifier Graph Visualizer: 13:57:30 - bayes.BayesNet
100%
play
temperature
outlook
humidity
windy

Outline

1 Motivation

2 Introduction

3 Bayesian Network Classifiers

4 k-Dependence Bayesian Classifiers

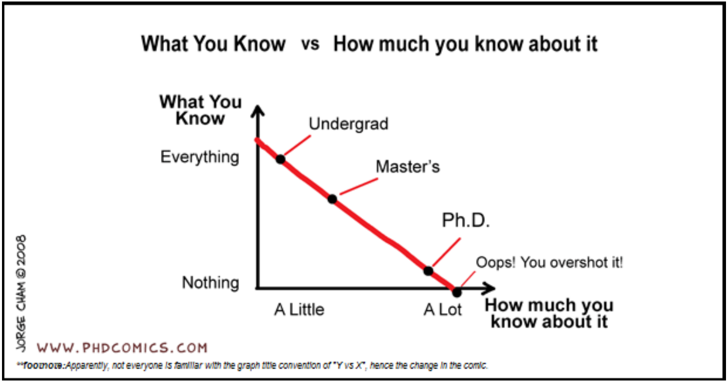
5 Links and References

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

- Gama, Carvalho, Faceli, Lorena, Oliveira *Extração de Conhecimento de Dados*, Silabo, 2012 (Cap 5)
- 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



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