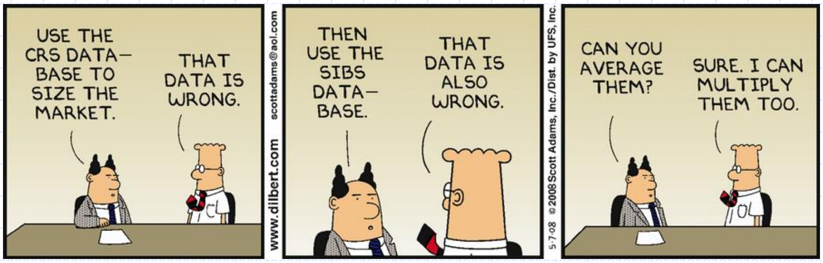
INM431 Machine Learning

Artur S. d'Avila Garcez
a.garcez@city.ac.uk
http://www.staff.city.ac.uk/~aag/
@AvilaGarcez

Based on G. Hinton's slides and C. Bishop's book

What is Machine Learning?


USE THE CRS DATA-BASE TO SIZE THE MARKET.
THAT DATA IS WRONG.
THEN USE THE SIBS DATA-BASE.
THAT DATA IS ALSO WRONG.
CAN YOU AVERAGE THEM?
SURE. I CAN MULTIPLY THEM TOO.
www.dilbert.com
scottadams@aol.com
5-7-08 ©2008 Scott Adams, Inc./Dist. by UFS, Inc.

© Artur Garcez

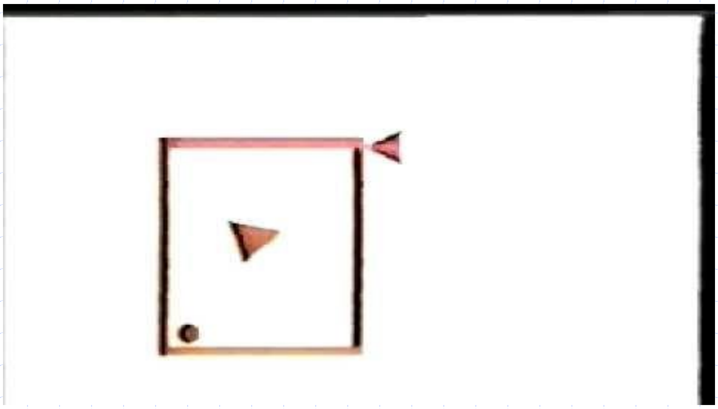
What is Learning?

Rote learning
Episodic learning
Observational learning
Associative learning
Active learning
Etc.

Experiential vs. passive learning

© Artur Garcez

Heider & Simmel (1944)



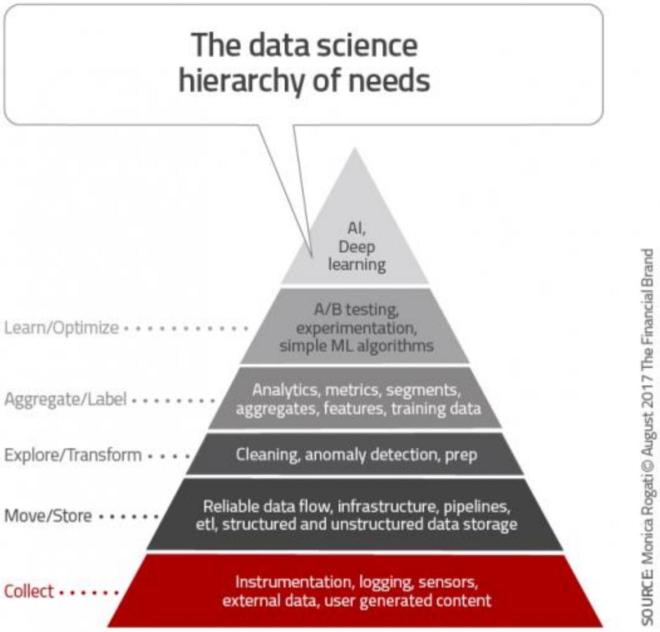
https://www.youtube.com/watch?v=wp8ebj_yRI4

© Artur Garcez

Why is ML important?

- Data Science: https://hbr.org/2012/10/data-scientist-the-sexiest-job-of-the-21st-century
- Big Data, a fact of life
- How to obtain insight and derive value from data?
- Learning from data + descriptions
- ML in the "radar" of most companies

© Artur Garcez


The data science hierarchy of needs
AI, Deep learning
Learn/Optimize
A/B testing, experimentation, simple ML algorithms
Aggregate/Label
Analytics, metrics, segments, aggregates, features, training data
Explore/Transform
Cleaning, anomaly detection, prep
Move/Store
Reliable data flow, infrastructure, pipelines, etl, structured and unstructured data storage
Collect
Instrumentation, logging, sensors, external data, user generated content

SOURCE: Monica Rogati© August 2017 The Financial Brand

© Artur Garcez

The Team

Prof Artur d'Avila Garcez, FBCS
Director of the Research Centre for Machine Learning
http://www.city.ac.uk/machine-learning

Dr Oleksandr Galkin (Teaching Associate)
Dr Adam White (Research Associate)
Simon Odense (PhD student)
Benedikt Wagner (PhD student)

© Artur Garcez

The Schedule

Weeks 1 to 5, 11am-1pm, Poynton
Labs 1pm-3pm
Reading week (no lectures or labs)
Weeks 7 to 10, 11am-1pm, Poynton
Labs 1pm-3pm
Coursework submission: week 10
Week 11 Revision

© Artur Garcez

The Labs

1-2pm Lab 1 ELG06 (surnames starting: W, S, H, P, L)

1-2pm Lab 2 ELG07 (surnames: J, B, T, G, D)

2-3pm Lab 3 ELG06 (surnames: M, R, C, A, K)

2-3pm Lab 4 ELG07 (surnames: E, Y, F, N, Other)

© Artur Garcez

The References

Chris Bishop “Pattern Recognition and Machine Learning”, Springer, 2006 (main textbook)

Tom Mitchell, “Machine Learning”, McGraw Hill, 1997

Kevin Murphy, “Machine Learning: A Probabilistic Perspective”, MIT Press, 2012 (advanced)

© Artur Garcez

The Content

Probabilities (Bishop, Ch1, Ch2), Curve fitting (Bishop, Ch1)
Decision Trees (Mitchell, Ch3)
Bayesian inference (Bishop, Ch2), Bayes Theorem (Bishop, Ch2)
Expectation and covariance (Bishop, Ch2)
Gaussian maximum likelihood (Bishop, Ch2, Ch3)
Regression with least squares (Bishop, Ch3)
Naive Bayes (Bishop, Ch4, Ch8)
Bayesian networks and K2 algorithm (lecture notes)
Random Forests for classification and regression (lecture notes)
K-means (Bishop, Ch. 9)
Mixture models and Expectation Maximization (Bishop Ch2, Ch.9)
Hidden Markov models (Bishop Ch.13)
PCA (Bishop Ch.12)
K-nearest neighbours (Bishop, Ch2)

© Artur Garcez

The Assessment

30% coursework (in groups of two): comparing two ML methods on a data set of your choice

Submission deadline: 25 Nov 2019, 5pm (Matlab code + poster), all through Moodle

More about the coursework next, including marking scheme, and on Moodle, including an example poster

70% exam (in January)

© Artur Garcez

Coursework task

- Specify a Machine Learning (ML) solution for a data analysis problem:
- Choose a dataset (e.g. from UCI ML repository: http://archive.ics.uci.edu/ml/)
- Apply, compare and contrast two ML solutions: adjust the ML model parameters and explain how ML seeks to solve the problem
- Submit your Matlab code and a poster through Moodle describing: the problem (data set), ML methods (and how they were adapted to solve the problem), results, conclusion and future work, references
- See example of a poster on Moodle

© Artur Garcez

Marking scheme

- Code: syntactic correctness (5%), organization and clarity of comments (10%), appropriate use and sophistication of methods (10%). Poster: description and motivation of the problem (5%), initial analysis of the data set including basic statistics (10%), brief summary of the two ML models with their pros and cons (10%), hypothesis statement (5%), description of choice of training and evaluation methodology (5%), choice of parameters and experimental results (10%), analysis and critical evaluation of results (25%), lessons learned and future work (5%).
- Marks will be adjusted according to the self and peer evaluation (c.f. coursework file on Moodle)

© Artur Garcez

Matlab training

Every Data Science student has free access to Matlab training leading to MathWorks certificates, including:

MATLAB Fundamentals
MATLAB Programming Techniques
MATLAB for Data Processing and Visualisation

You need to be registered to use Matlab first and then request access to the training.

Please raise a service request on www.city.ac.uk/itservicedesk to create a 'Student/Home use software request' for Matlab and ask to be registered on the Matlab online training

Any problems, please contact Paul Roberts P.Roberts-1@city.ac.uk

Check also the matlab primer file on Moodle, which is a quick start guide to Matlab for use in the lab tutorials

© Artur Garcez

Modus Operandi (1)

Ask questions during or immediately after the lectures and labs.

There is a discussion board on Moodle; post questions (and answers) there too.

The TAs and I will monitor the board and respond when appropriate.

Email me only if absolutely necessary

© Artur Garcez

Modus Operandi (2)

Lecture notes and lab exercises will be made available on Moodle before the lectures.

Read the material before the lectures and labs; this is the best way to prepare for the exam!

Sample solutions for the lab exercises will be made available later…

An example exam will be made available and at a revision lecture we will go through the exam answers so you know what is expected of you in terms of assessment.

© Artur Garcez

What is Machine Learning?

Machine Learning is the study of computer algorithms that improve automatically through experience according to a performance measure

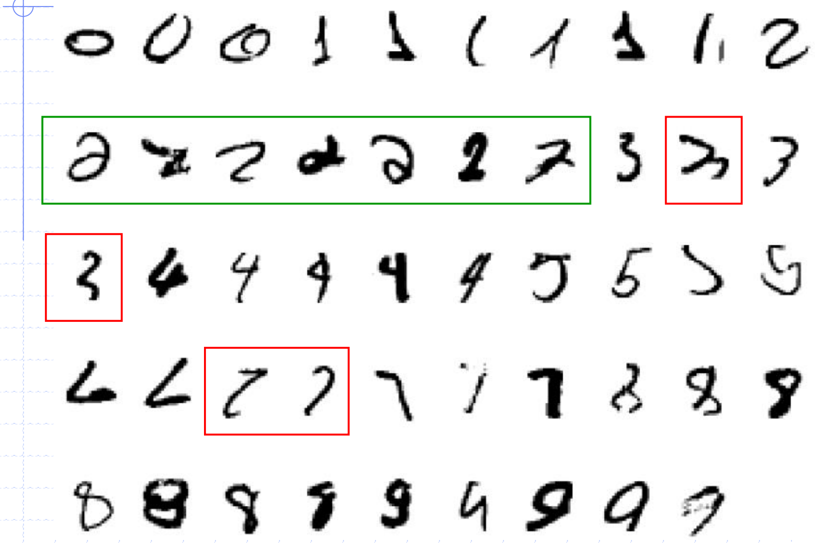
"Instead of writing a program by hand, we collect lots of examples that specify the correct output for a given input.

A machine learning algorithm then takes these examples and produces a program that does the job.

This program may look very different from a typical hand-written program. It may contain millions of numbers (parameters).

If we do it right, the program works for new cases (generalization) as well as the ones we trained it on". Geoff Hinton

A classic example of a task that requires machine learning: It is very hard to say what makes a 2



Examples of tasks that are best solved by a learning algorithm

- Recognizing patterns:
  - Facial identities or facial expressions
  - Handwritten or spoken words
  - Medical images
- Generating patterns:
  - Generating images or motion sequences (gap filling)
- Recognizing anomalies:
  - Unusual sequences of credit card transactions
  - Unusual patterns of sensor readings in a nuclear power plant or unusual sound in your car engine.
- Prediction:
  - Future stock prices or currency exchange rates

Examples of tasks that are best solved by a learning algorithm (cont.)

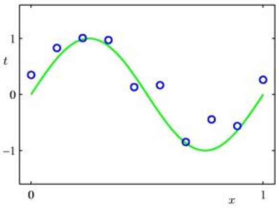
- The web contains lots of data (at service and infrastructure levels). Big datasets often use/need Machine Learning, especially if the data is noisy.
- Spam filtering, fraud detection:
  - The enemy adapts so we must adapt too.
- Recommendation systems:
  - Lots of noisy data. Million dollar prize!
- Video understanding:
  - Content-based classification of YouTube and Facebook videos
- Information retrieval:
  - Find documents or images with similar content
- Data Visualization:
  - Display a huge database in a revealing way

Generalization

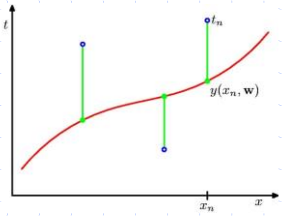
- The real aim of machine learning is to do well on test data that is not known during learning.
- Choosing the values for the parameters that minimize a loss/error function on the training data is not necessarily the best policy.
- We want the learning machine to model the true regularities in the data and to ignore the noise in the data.
  - But the learning machine does not know which regularities are real and which are accidental quirks of the particular set of training examples
- So how can we be sure that the machine will generalize correctly to new data?

A simple example: fitting a polynomial

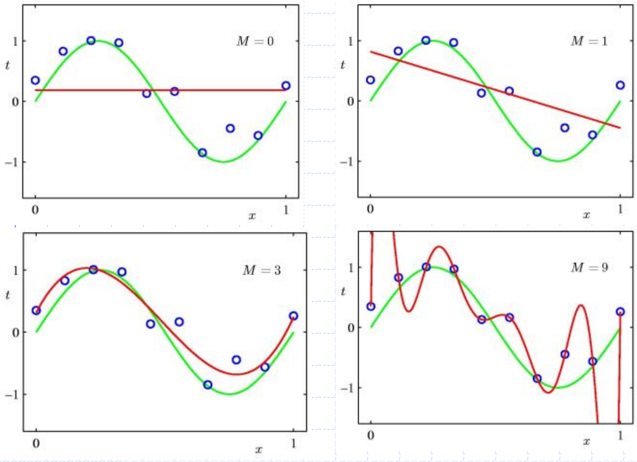
- The green curve is the true function (which is not a polynomial)
- The data points are uniform in x but have noise in y.



- We will use a loss function that measures the squared error in the prediction of y(x). The loss for the red polynomial is the sum of the squared vertical errors.



Some fits to the data: which one is best?



$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

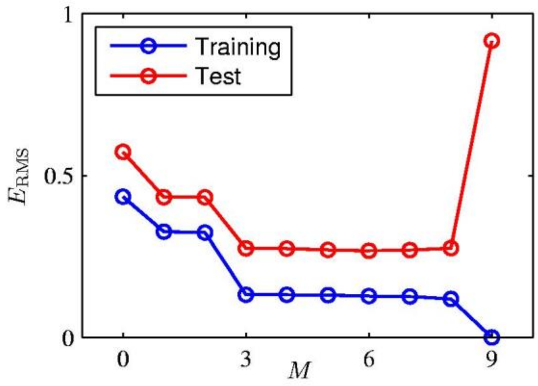
Curve fitting

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n,\mathbf{w}) - t_n\}^2$$

Curve fitting problem: choose the value (**w***) of **w** for which *E*(**w**) is as small as possible. How about the value of M (the degree of y)?

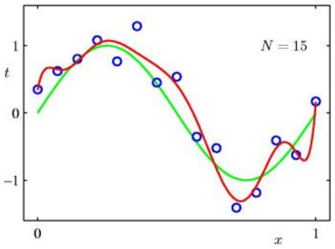
	$M=0$	$M=1$	$M=3$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

Overfitting and Model Selection



A simple way to reduce model complexity

- If we penalize polynomials that have big values for their coefficients, we will get less wiggly solutions:



penalized loss function

regularization parameter

$$\tilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

target value

Using a validation set

- Divide the total dataset into three subsets:
  - Training data is used for learning the parameters of the model.
  - Validation data is not used for learning but is used for deciding what type of model and what amount of regularization works best.
  - Test data is used to get a final, unbiased estimate of how well the learner works. We expect this estimate of the generalization error to be worse than on the validation data.
- We could then re-divide the total dataset to get another estimate of the generalization error…

Decision Trees

- Very successful ML method, but not probabilistic
- Offers a graphical representation of a Boolean function (propositional logic formula) or a discrete function, in general
- Basis for **Random Forests** which are probabilistic and state of the art in computer vision:

http://research.microsoft.com/en-us/projects/decisionforests/

© Artur Garcez

Decision trees (cont.)

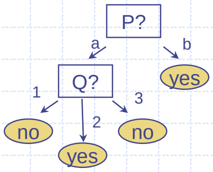
Input: set of properties describing object/situation
Output: yes/no (or more generally Output in S, where S is a finite set of classes)

Non-terminal nodes: test on values of property

Terminal nodes: Boolean value of the *goal class*

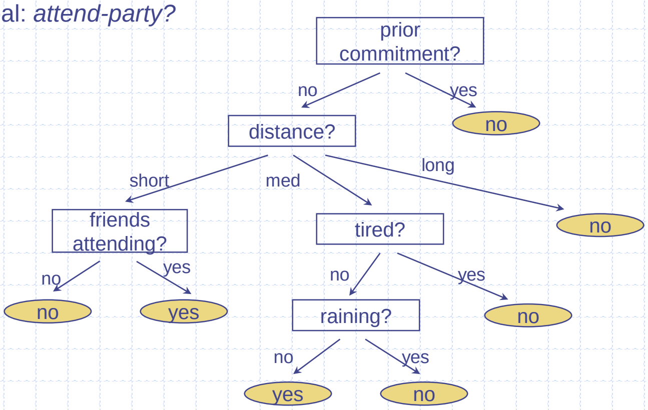
Branches: values of test

Properties (P, Q, …) = attributes



Decision trees: example

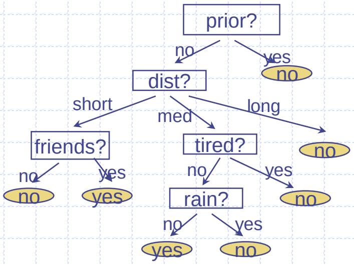
Goal: *attend-party?*



Equivalence to propositional logic

Attend-party?

Decision trees implicitly define conjunctions of disjunctions



In Logic:

attend-party(P) IF

(not prior(P) AND dist(P,short) AND friends(P))

OR

(not prior(P) AND dist(P,med) AND not tired(P) AND not rain(P))

Learning algorithm

1) Start with a set of examples (training set), set of attributes SA, default value for goal.
2) If the set of examples is empty, then add a leaf with the default value for the goal and terminate, otherwise:
3) If all examples have the same classification, then add a leaf with that classification and terminate, otherwise:
4) If the set of attributes SA is empty, then return the default value for the goal and terminate, otherwise:
5) **Choose** an attribute A to split on.
6) Add a corresponding test to the tree.
7) Create new branches for each value of the attribute.
8) Assign each example to the appropriate branch.
9) Iterate from step 1) on each branch, with set of attributes SA \ {A} and default value the majority value for the current set of examples.

Choosing the “best” attribute

- Intuition:
  - The aim is to minimise the depth of the final tree
  - Choose attribute that provides as exact as possible a classification:

    perfect attribute: all examples are either positive or negative

    useless attribute: the proportion of positive and negative examples in the new set is roughly the same as in the old set
- Use information theory for defining *perfect/useful/useless* by computing the information gain from choosing attributes

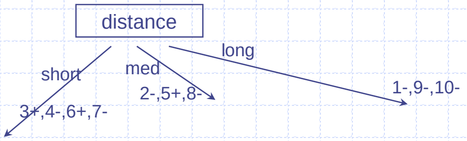
Example: training set

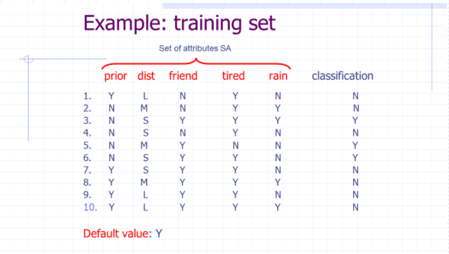
Set of attributes SA

	prior	dist	friend	tired	rain	classification
1.	Y	L	N	Y	N	N
2.	N	M	N	Y	Y	N
3.	N	S	Y	Y	Y	Y
4.	N	S	N	Y	N	N
5.	N	M	Y	N	N	Y
6.	N	S	Y	Y	N	Y
7.	Y	S	Y	Y	N	N
8.	Y	M	Y	Y	Y	N
9.	Y	L	Y	Y	N	N
10.	Y	L	Y	Y	Y	N

Default value: Y

Example: decision tree learning (1) choice in step 5 is random



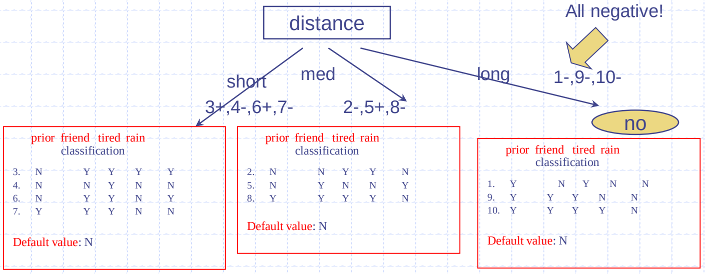

Example: training set

Set of attributes SA

	prior	dist	friend	tired	rain	classification
1.	Y	L	N	Y	N	N
2.	N	M	N	Y	Y	N
3.	N	S	Y	Y	Y	Y
4.	N	S	N	Y	N	N
5.	N	M	Y	N	N	Y
6.	N	S	Y	Y	N	Y
7.	Y	S	Y	Y	N	N
8.	Y	M	Y	Y	Y	N
9.	Y	L	Y	Y	N	N
10.	Y	L	Y	Y	Y	N

Default value: Y

Example: decision tree learning (2)



	prior	friend	tired	rain	classification
3.	N	Y	Y	Y	Y
4.	N	N	Y	N	N
6.	N	Y	Y	N	Y
7.	Y	Y	Y	N	N

Default value: N

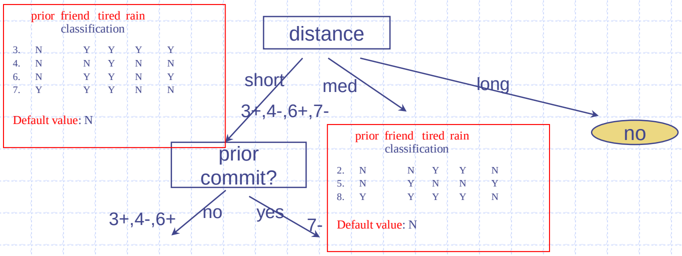
	prior	friend	tired	rain	classification
2.	N	N	Y	Y	N
5.	N	Y	N	N	Y
8.	Y	Y	Y	Y	N

Default value: N

	prior	friend	tired	rain	classification
1.	Y	N	Y	N	N
9.	Y	Y	Y	N	N
10.	Y	Y	Y	Y	N

Default value: N

Example: decision tree learning (3)



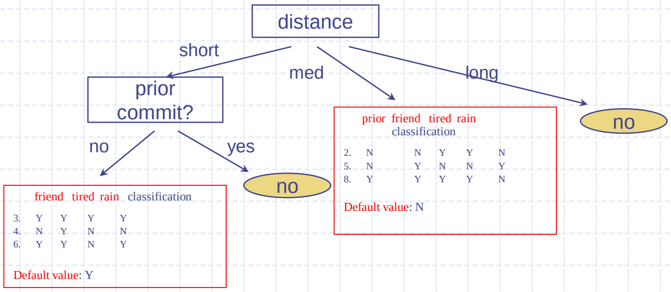
	prior	friend	tired	rain	classification
3.	N	Y	Y	Y	Y
4.	N	N	Y	N	N
6.	N	Y	Y	N	Y
7.	Y	Y	Y	N	N

Default value: N

	prior	friend	tired	rain	classification
2.	N	N	Y	Y	N
5.	N	Y	N	N	Y
8.	Y	Y	Y	Y	N

Default value: N

Example: decision tree learning (4)



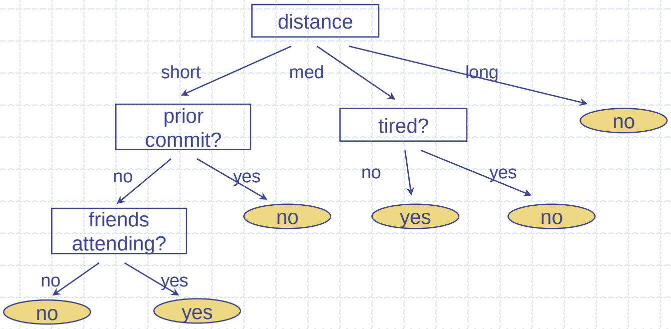
	friend	tired	rain	classification
3.	Y	Y	Y	Y
4.	N	Y	N	N
6.	Y	Y	N	Y

Default value: Y

	prior	friend	tired	rain	classification
2.	N	N	Y	Y	N
5.	N	Y	N	N	Y
8.	Y	Y	Y	Y	N

Default value: N

Example: decision tree learning – final tree



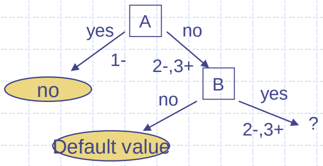
Limitations of decision trees

- Decision trees cannot represent all Predicate Logic formulas, e.g. is there another less distant party?
- Multiple trees exist for the same Boolean function, some exponentially large…
- Some functions only admit exponentially large trees (e.g. parity function, returning *yes* iff the input vector has an odd number of 1’s)

Empty set of attributes if data is noisy

	A	B	classification
1.	Y	N	N
2.	N	Y	N
3.	N	Y	Y

noise



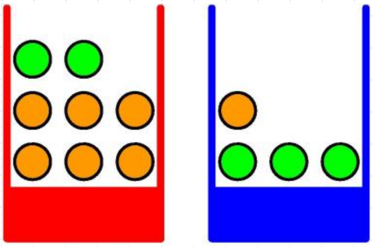
DT: Another example

Weather	Parents Visiting	Money	Decision
Sunny	Yes	Rich	Cinema
Sunny	No	Rich	Tennis
Windy	Yes	Rich	Cinema
Rainy	Yes	Poor	Cinema
Rainy	No	Rich	Stay in
Rainy	Yes	Poor	Cinema
Windy	No	Poor	Cinema
Windy	No	Rich	Shopping
Windy	Yes	Rich	Cinema
Sunny	No	Rich	Tennis

© Artur Garcez

Probability Theory

Apples and Oranges: pick an item from a box!

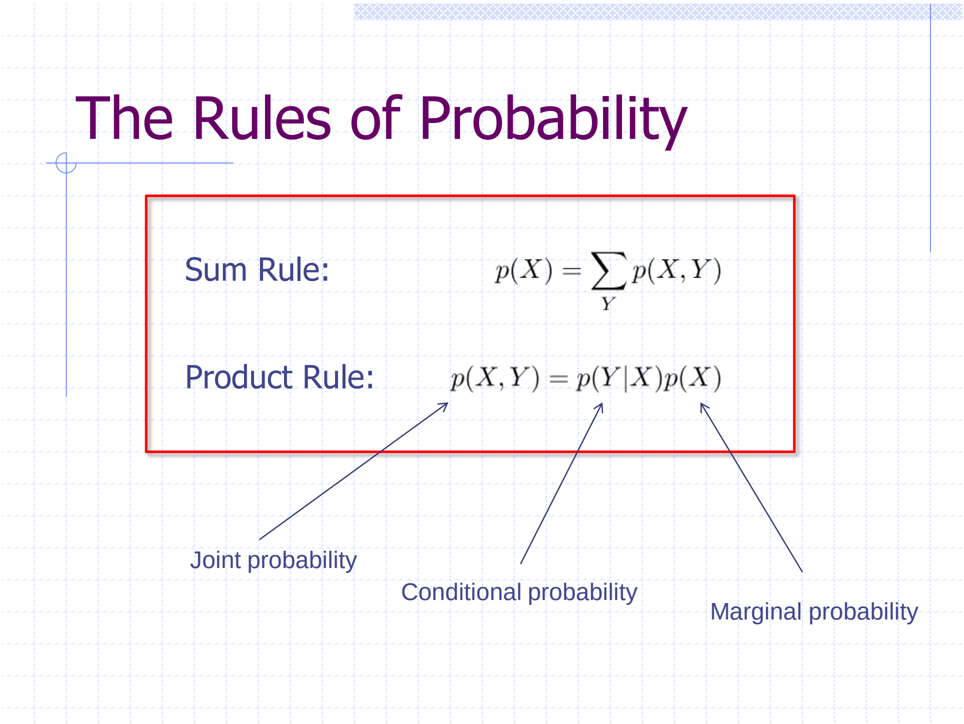


Random variables: BOX = {red, blue}, FRUIT = {apple, orange}

What's the probability of picking an apple?

P(FRUIT=apple) = P(FRUIT=apple,BOX=red) + P(FRUIT=apple,BOX=blue) = P(FRUIT=apple|BOX=red).P(BOX=red) + P(FRUIT=apple|BOX=blue).P(BOX=blue)

Try this: given that I picked an orange, what's the probability that it came from the blue box? Use the sum and product rules!



The Rules of Probability

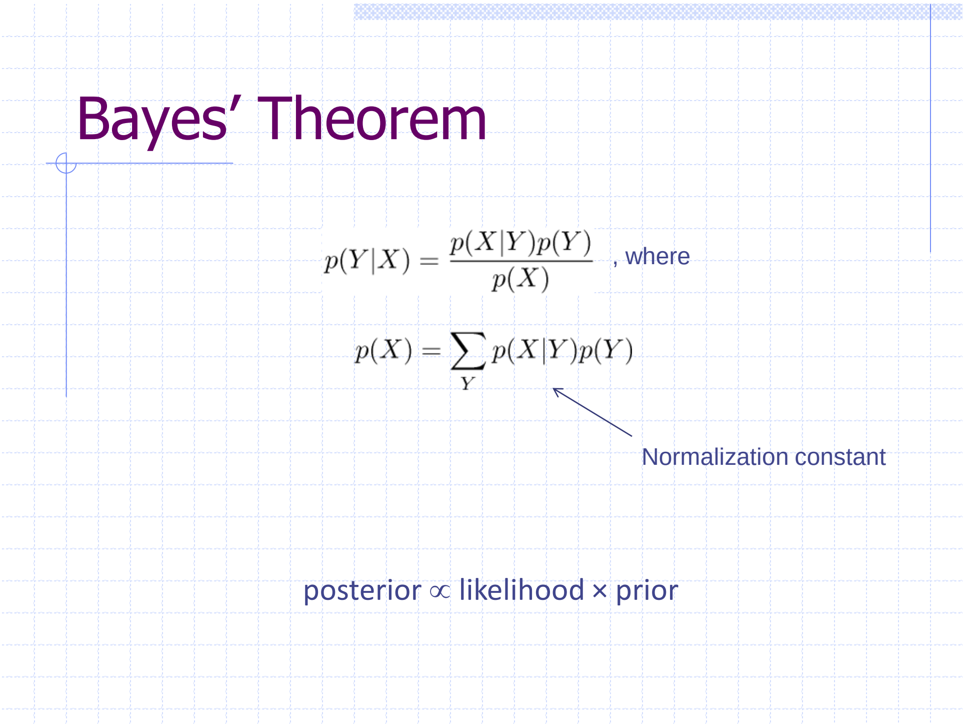
Sum Rule: $$p(X) = \sum_Y p(X, Y)$$

Product Rule: $$p(X, Y) = p(Y|X)p(X)$$

Joint probability

Conditional probability

Marginal probability



Bayes’ Theorem

$$p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)}$$, where

$$p(X) = \sum_Y p(X|Y)p(Y)$$

Normalization constant

posterior ∝ likelihood × prior

A Bayesian Problem

A test for salmonella is made available to chicken farmers. The test will correctly show a positive result for salmonella 95% of the time. However the test also shows positive results 15% of the time in salmonella free chickens. 10% of chickens have salmonella.

If a chicken tests positive, what is the probability that it has salmonella?

Example – notation and what is known

Begin by defining the notation for the facts and for the question asked:

Let A be the presence of salmonella

Let B be a positive test

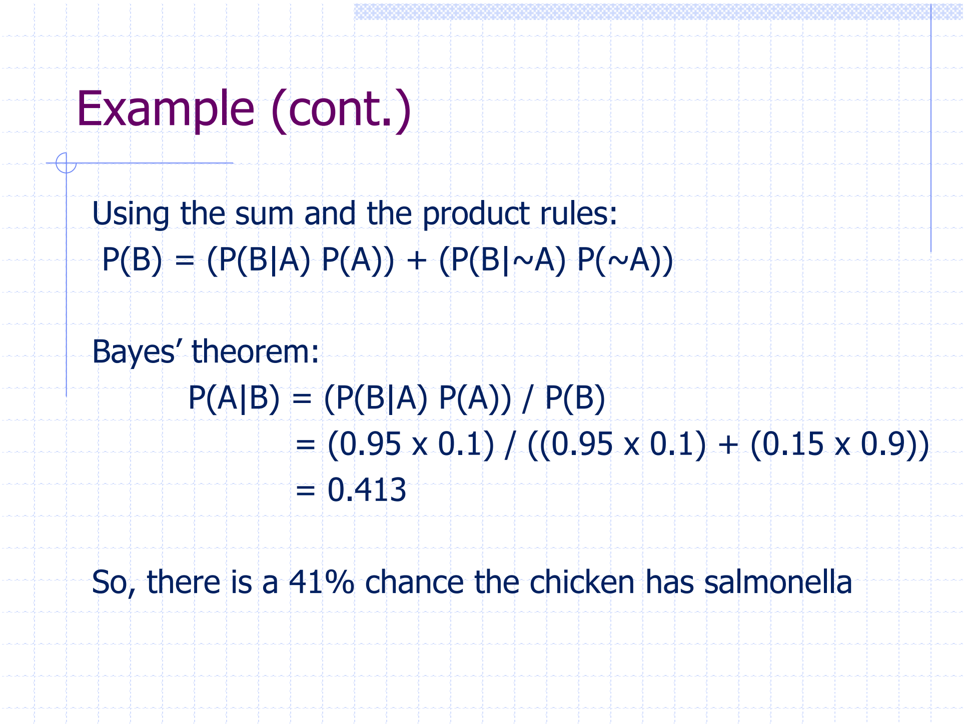
P(A|B) = ? (the probability that a chicken has salmonella given a positive test)

Establish the probabilities for the facts and fact dependencies

P(A) = 0.1

P(B|A) = 0.95

P(B|~A) = 0.15



Example (cont.)

Using the sum and the product rules:

$P(B) = (P(B|A)\ P(A)) + (P(B|{\sim}A)\ P({\sim}A))$

Bayes' theorem:

$$\begin{aligned} P(A|B) &= (P(B|A)\ P(A)) / P(B) \\ &= (0.95 \times 0.1) / ((0.95 \times 0.1) + (0.15 \times 0.9)) \\ &= 0.413 \end{aligned}$$

So, there is a 41% chance the chicken has salmonella