Course 395: Machine Learning – Lectures

- Lecture 1-2: Concept Learning (*M. Pantic*)
- ➢ Lecture 3-4: Decision Trees & CBC Intro (*M. Pantic*)
- Lecture 5-6: Artificial Neural Networks (*THs*)
- Lecture 7-8: Instance Based Learning (*M. Pantic*)
- Lecture 9-10: Genetic Algorithms (*M. Pantic*)
- Lecture 11-12: Evaluating Hypotheses (*THs*)
- Lecture 13-14: Guest Lectures on ML Applications
- Lecture 15-16: Inductive Logic Programming (*S. Muggleton*)
- Lecture 17-18: Inductive Logic Programming (*S. Muggleton*)

Decision Trees & CBC Intro – Lecture Overview

- Problem Representation using a Decision Tree
- ID3 algorithm
- The problem of overfitting
- Research on affective computing, natural HCI, and ambient intelligence
- Facial expressions and Emotions
- Overview of the CBC
- Group forming

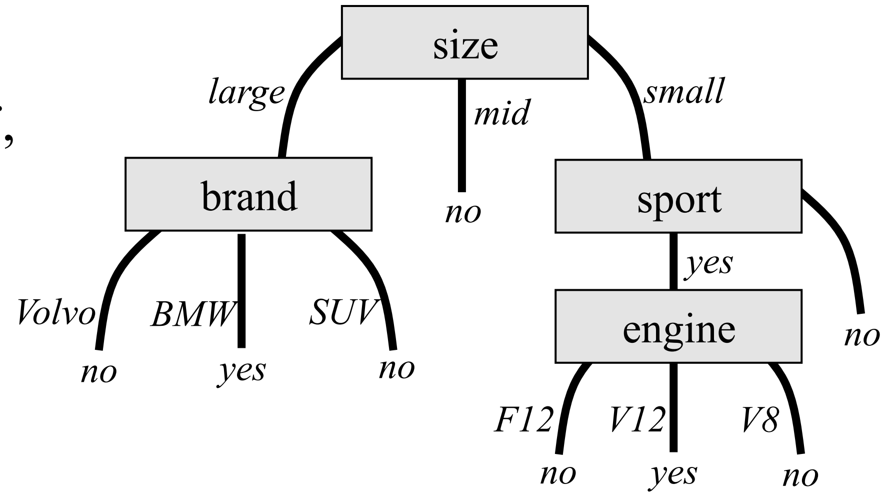
Problem Representation using a Decision Tree

- Decision Tree learning is a method for approximating discrete classification functions by means of a tree-based representation

- A learned Decision Tree classifies a new instance by sorting it down the tree
  - tree node ↔ classification OR test of a specific attribute of the instance
  - tree branch ↔ possible value for the attribute in question

- Concept: *Good Car*

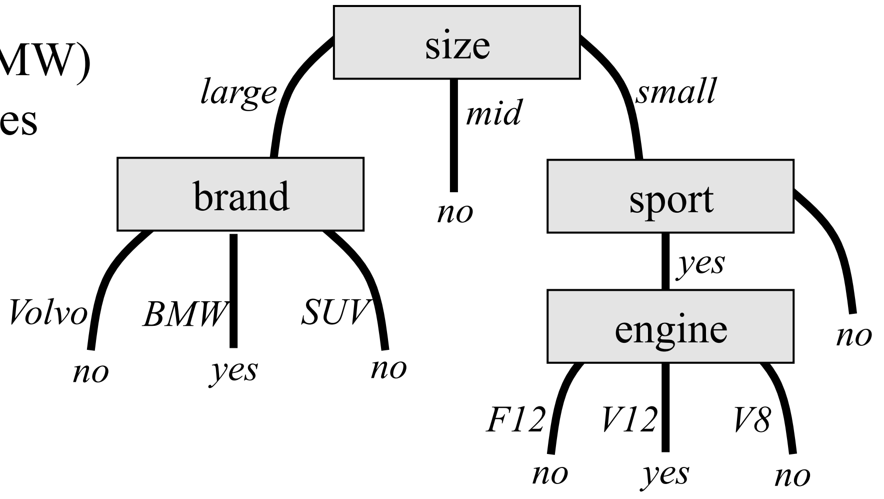
  ‹size = *small*, brand = *Ferari*, model = *Enzo*, sport = yes, engine = *V12*, colour = *red*›



Problem Representation using a Decision Tree

- A learned Decision Tree can be represented as a set of *if-then* rules
- To 'read out' the rules from a learned Decision Tree
  - tree ↔ disjunction (∨) of sub-trees
  - sub-tree ↔ conjunction (∧) of constraints on the attribute values
- Rule: *Good Car*

IF (size = large AND brand = BMW)
OR (size = small AND sport = yes
AND engine = V12)
THEN Good Car = yes
ELSE Good Car = no;



Decision Tree Learning Algorithm

- Decision Tree learning algorithms employ top-down greedy search through the space of possible solutions.

- A general Decision Tree learning algorithm:
  1. perform a statistical test of each attribute to determine how well it classifies the training examples when considered alone;
  2. select the attribute that performs best and use it as the root of the tree;
  3. to decide the descendant node down each branch of the root (parent node), sort the training examples according to the value related to the current branch and repeat the process described in steps 1 and 2.

- ID3 algorithm is one of the most commonly used Decision Tree learning algorithms and it applies this general approach to learning the decision tree.

ID3 Algorithm

- ID3 algorithm uses so-called *Information Gain* to determine how informative an attribute is (i.e., how well it alone classifies the training examples).

- *Information Gain* is based on a measure that we call *Entropy*, which characterizes the impurity of a collection of examples *S* (i.e., impurity↑ → *E(S)*↑):

  $$E(S) \equiv \mathrm{abs}(-\,p_\oplus \log_2 p_\oplus - \; p_\otimes \log_2 p_\otimes),$$

  where $p_\oplus$ ($p_\otimes$) is the proportion of positive (negative) examples in *S*.

  (Note: $E(S) = 0$ if *S* contains only positive or only negative examples
  $\leftrightarrow p_\oplus = 1,\ p_\otimes = 0,\ E(S) = \mathrm{abs}(-\,1 \cdot 0 - \; 0 \cdot \log_2 p_\otimes) = 0$)

  (Note: $E(S) = 1$ if *S* contains equal amount of positive and negative examples
  $\leftrightarrow p_\oplus = ½,\ p_\otimes = ½,\ E(S) = \mathrm{abs}(-\,½ \cdot 1 - \; ½ \cdot 1) = 1$)

  In the case that that the target attribute can take *n* values:

  $$E(S) \equiv \textstyle\sum_i \mathrm{abs}(-\,p_i \log_2 p_i),\ i = [1..n]$$

  where p_i is the proportion of examples in *S* having the target attribute value *i*.

ID3 Algorithm

- *Information Gain* is based on a measure that we call *Entropy*, which characterizes the impurity of a collection of examples S (impurity↑ → $E(S)$↑):

  $E(S) \equiv \text{abs}(-\, p_\oplus \log_2 p_\oplus - \, p_\otimes \log_2 p_\otimes)$,

  where $p_\oplus$ ($p_\otimes$) is the proportion of positive (negative) examples in S.

  (Note: $E(S) = 0$ if S contains only positive or only negative examples
  $\leftrightarrow p_\oplus = 1, p_\otimes = 0, E(S) = \text{abs}(-\, 1 \cdot 0 - \, 0 \cdot \log_2 p_\otimes) = 0$)

  (Note: $E(S) = 1$ if S contains equal amount of positive and negative examples
  $\leftrightarrow p_\oplus = ½, p_\otimes = ½, E(S) = \text{abs}(-\, ½ \cdot 1 - \, ½ \cdot 1) = 1$)

  In the case that that the target attribute can take n values:

  $E(S) \equiv \sum_i \text{abs}(-\, p_i \log_2 p_i),\ i = [1..n]$

  where p_i is the proportion of examples in S having the target attribute value i.

- *Information Gain* – Reduction in $E(S)$ caused by partitioning S according to attribute A

  $IG(S, A) = E(S) - \sum_{v \in values(A)} (|S_v| \,/\, |S|)\, E(S_v)$

  where *values(A)* are all possible values for attribute A, $S_v \in S$ contains all examples for which attribute A has the value v, and $|S_v|$ is the cardinality of set S_v.

ID3 Algorithm – Example

1. For each attribute *A* of the training examples in set *S* calculate:
   $IG(S, A) = E(S) - \sum_{v \in values(A)} (|S_v| / |S|)\, E(S_v)$, $E(S_v) \equiv \sum_v \text{abs}(- p_v \log_2 p_v)$, $v = [1..n]$.
2. Select the attribute with the maximal *IG(S, A)* and use it as the root of the tree.
3. To decide the descendant node down each branch of the root (i.e., parent node), sort the training examples according to the value related to the current branch and repeat the process described in steps 1 and 2.

Target concept: *Play Tennis* (Mitchell's book, p. 59)

	PlayTennis(d)	*outlook*	*temperature*	*humidity*	*wind*
1	0	sunny	hot	high	weak
2	0	sunny	hot	high	strong
…	…	…	…	…	…
13	1	overcast	hot	normal	weak
14	0	rain	mild	high	strong

$IG(D, Outlook) = E(D) - 5/14\, E(D_{sunny}) - 4/14\, E(D_{overcast}) - 5/14\, E(D_{rain})$

ID3 Algorithm – Example

	PlayTennis(d)	*outlook*	*temperature*	*humidity*	*wind*
1	0	sunny	hot	high	weak
2	0	sunny	hot	high	strong
3	1	overcast	hot	high	weak
4	1	rain	mild	high	weak
5	1	rain	cool	normal	weak
6	0	rain	cool	normal	strong
7	1	overcast	cool	normal	strong
8	0	sunny	mild	high	weak
9	1	sunny	cool	normal	weak
10	1	rain	mild	normal	weak
11	1	sunny	mild	normal	strong
12	1	overcast	mild	high	strong
13	1	overcast	hot	normal	weak
14	0	rain	mild	high	strong

ID3 Algorithm – Example

1. For each attribute A of the training examples in set S calculate:
   $IG(S, A) = E(S) - \sum_{v \in values(A)} (|S_v| / |S|)\, E(S_v)$, $E(S_v) \equiv \sum_v \mathrm{abs}(-\, p_v \log_2 p_v)$, $v = [1..n]$.
2. Select the attribute with the maximal $IG(S, A)$ and use it as the root of the tree.
3. To decide the descendant node down each branch of the root (i.e., parent node), sort the training examples according to the value related to the current branch and repeat the process described in steps 1 and 2.

Target concept: *Play Tennis* (Mitchell's book, p. 59)

	PlayTennis(d)	*outlook*	*temperature*	*humidity*	*wind*
…	…	…	…	…	…

$IG(D, Outlook) = E(D) - 5/14\, E(D_{sunny}) - 4/14\, E(D_{overcast}) - 5/14\, E(D_{rain})$

$= 0.940 - 0.357 \cdot 0.971 - 0 - 0.357 \cdot 0.971 = \boxed{0.246}$

$IG(D, Temperature) = E(D) - 4/14\, E(D_{hot}) - 6/14\, E(D_{mild}) - 4/14\, E(D_{cool})$

$= 0.940 - 0.286 \cdot 1 - 0.429 \cdot 0.918 - 0.286 \cdot 0.811 = 0.029$

$IG(D, Humidity) = E(D) - 7/14\, E(D_{high}) - 7/14\, E(D_{normal}) = 0.940 - ½ \cdot 0.985 - ½ \cdot 0.591 = 0.151$

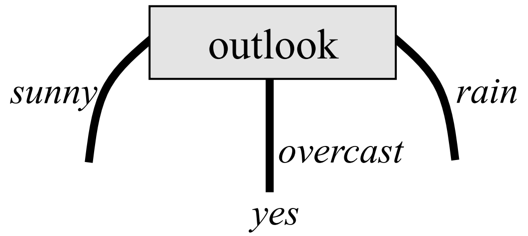
$IG(D, Wind) = E(D) - 8/14\, E(D_{weak}) - 6/14\, E(D_{strong}) = 0.940 - 0.571 \cdot 0.811 - 0.429 \cdot 1 = 0.048$

ID3 Algorithm – Example

1. For each attribute A of the training examples in set S calculate:
   $IG(S, A) = E(S) - \sum_{v \in values(A)} (|S_v| / |S|) \, E(S_v)$, $E(S_v) \equiv \sum_v \mathrm{abs}(-\, p_v \log_2 p_v)$, $v = [1..n]$.
2. Select the attribute with the maximal $IG(S, A)$ and use it as the root of the tree.
3. To decide the descendant node down each branch of the root (i.e., parent node), sort the training examples according to the value related to the current branch and repeat the process described in steps 1 and 2.

Target concept: *Play Tennis* (Mitchell' s book, p. 59)

	PlayTennis(d)	*outlook*	*temperature*	*humidity*	*wind*
…	…	…	…	…	…



ID3 Algorithm – Example

	PlayTennis(d)	*outlook*	*temperature*	*humidity*	*wind*
1	0	sunny	hot	high	weak
2	0	sunny	hot	high	strong
3	1	overcast	hot	high	weak
4	1	rain	mild	high	weak
5	1	rain	cool	normal	weak
6	0	rain	cool	normal	strong
7	1	overcast	cool	normal	strong
8	0	sunny	mild	high	weak
9	1	sunny	cool	normal	weak
10	1	rain	mild	normal	weak
11	1	sunny	mild	normal	strong
12	1	overcast	mild	high	strong
13	1	overcast	hot	normal	weak
14	0	rain	mild	high	strong

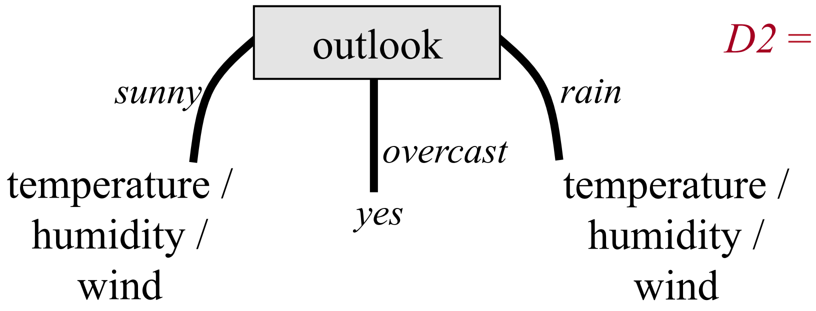
ID3 Algorithm – Example

1. For each attribute A of the training examples in set S calculate:
   $IG(S, A) = E(S) - \sum_{v \in values(A)} (|S_v| / |S|) E(S_v)$, $E(S_v) \equiv \sum_v \text{abs}(-p_v \log_2 p_v)$, $v = [1..n]$.
2. Select the attribute with the maximal $IG(S, A)$ and use it as the root of the tree.
3. To decide the descendant node down each branch of the root (i.e., parent node), sort the training examples according to the value related to the current branch and repeat the process described in steps 1 and 2.

Target concept: *Play Tennis* (Mitchell' s book, p. 59)

	PlayTennis(d)	*outlook*	*temperature*	*humidity*	*wind*
…	…	…	…	…	…

$D1 = \{d \in D \mid Outlook(d) = sunny\}$



$D2 = \{d \in D \mid Outlook(d) = rain\}$

ID3 Algorithm – Example

	PlayTennis(d)	*outlook*	*temperature*	*humidity*	*wind*	
1	0	sunny	hot	high	weak	
2	0	sunny	hot	high	strong	
8	0	sunny	mild	high	weak	***D1***
9	1	sunny	cool	normal	weak	
11	1	sunny	mild	normal	strong	
4	1	rain	mild	high	weak	
5	1	rain	cool	normal	weak	
6	0	rain	cool	normal	strong	***D2***
10	1	rain	mild	normal	weak	
14	0	rain	mild	high	strong	
3	1	overcast	hot	high	weak	
7	1	overcast	cool	normal	strong	
12	1	overcast	mild	high	strong	
13	1	overcast	hot	normal	weak	

ID3 Algorithm – Example

	PlayTennis(d)	*outlook*	*temperature*	*humidity*	*wind*
1	0	sunny	hot	high	weak
2	0	sunny	hot	high	strong
8	0	sunny	mild	high	weak
9	1	sunny	cool	normal	weak
11	1	sunny	mild	normal	strong
…	…	…	…	…	…

D1

$E(D1) = \text{abs}\,(-2/5 \log_2 2/5 - 3/5 \log_2 3/5) = 0.971$

$IG(D1, Temperature) = E(D1) - 2/5\, E(D1_{hot}) - 2/5\, E(D1_{mild}) - 1/5\, E(D1_{cool})$
$= 0.971 - 0.4 \cdot 0 - 0.4 \cdot 1 - 0.4 \cdot 0 = 0.571$

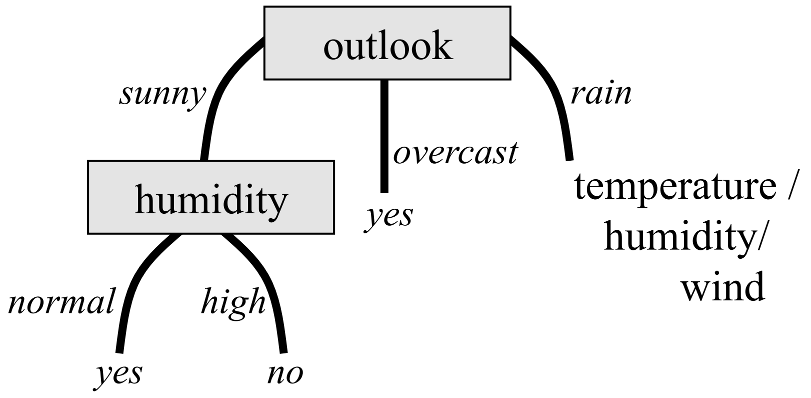
$IG(D1, Humidity) = E(D1) - 3/5\, E(D1_{high}) - 2/5\, E(D1_{normal})$
$= 0.971 - 0.6 \cdot 0 - 0.4 \cdot 0 = \boxed{0.971}$

$IG(D1, Wind) = E(D1) - 3/5\, E(D1_{weak}) - 2/5\, E(D1_{strong})$
$= 0.971 - 0.6 \cdot 0.918 - 0.4 \cdot 1 = 0.02$

ID3 Algorithm – Example

	PlayTennis(d)	*outlook*	*temperature*	*humidity*	*wind*
1	0	sunny	hot	high	weak
2	0	sunny	hot	high	strong
8	0	sunny	mild	high	weak
9	1	sunny	cool	normal	weak
11	1	sunny	mild	normal	strong
…	…	…	…	…	…

D1



ID3 Algorithm – Example

	PlayTennis(d)	*outlook*	*temperature*	*humidity*	*wind*
4	1	rain	mild	high	weak
5	1	rain	cool	normal	weak
6	0	rain	cool	normal	strong
10	1	rain	mild	normal	weak
14	0	rain	mild	high	strong
…	…	…	…	…	…

$\xrightarrow{}$ ***D2***

$E(D2) = \text{abs}\,(-3/5 \log_2 3/5 - 2/5 \log_2 2/5) = 0.971$

$IG(D2, Temperature) = E(D2) - 0/5\, E(D2_{hot}) - 3/5\, E(D2_{mild}) - 2/5\, E(D2_{cool})$
$= 0.971 - 0 - 0.6 \cdot 0.918 - 0.4 \cdot 1 = 0.02$

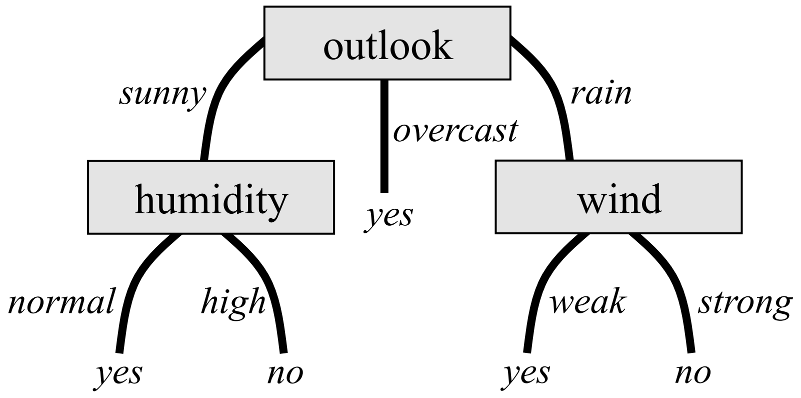
$IG(D2, Humidity) = E(D2) - 2/5\, E(D2_{high}) - 3/5\, E(D2_{normal})$
$= 0.971 - 0.4 \cdot 1 - 0.6 \cdot 0.918 = 0.02$

$IG(D2, Wind) = E(D2) - 3/5\, E(D2_{weak}) - 2/5\, E(D2_{strong})$
$= 0.971 - 0.6 \cdot 0 - 0.4 \cdot 0 = \boxed{0.971}$

ID3 Algorithm – Example

	PlayTennis(d)	*outlook*	*temperature*	*humidity*	*wind*
4	1	rain	mild	high	weak
5	1	rain	cool	normal	weak
6	0	rain	cool	normal	strong
10	1	rain	mild	normal	weak
14	0	rain	mild	high	strong
…	…	…	…	…	…

D2



ID3 Algorithm – Advantages & Disadvantages

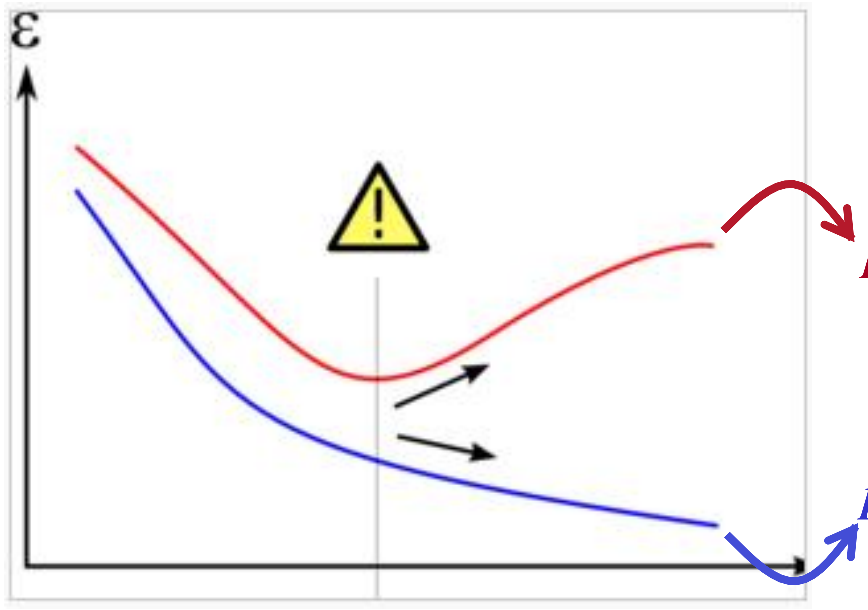
- Advantages of ID3 algorithm:
  1. Every discrete classification function can be represented by a decision tree → it cannot happen that ID3 will search an incomplete hypothesis space.
  2. Instead of making decisions based on individual training examples (as is the case by Find-S and Candidate-Elimination algorithms), ID3 uses statistical properties of all examples (information gain) → resulting search is much less sensitive to errors in individual training examples.

- Disadvantages of ID3 algorithm:
  1. ID3 determines a single hypothesis, not a space of consistent hypotheses (as is the case by Candidate-Elimination algorithm) → ID3 cannot determine how many different decision trees are consistent with the available training data.
  2. ID3 grows the tree to perfectly classify the training examples without performing a backtracking in its search → ID3 may overfit the training data and converge to locally optimal solution that is not globally optimal.

The Problem of Overfitting

- Def (*Mitchell 1997*):

  Given a hypothesis space H, h ∈ H overfits the training data if ∃h' ∈ H such that h has smaller error over the training examples, but h' has smaller error than h over the entire distribution of instances.



performance of h ∈ H on testing data

performance of h ∈ H on training data

The Problem of Overfitting

- Ways to avoid overfitting:
  1. Stop the training process before the learner reaches the point where it perfectly classifies the training data.
  2. Apply backtracking in the search for the optimal hypothesis. In the case of Decision Tree Learning, backtracking process is referred to as ‘post-pruning of the overfitted tree’.

- Ways to determine the correctness of the learner’s performance:
  1. Use two different sets of examples: training set and validation set.
  2. Use all examples for training, but apply a statistical test to estimate whether a further training will produce a statistically significant improvement of the learner’s performance. In the case of Decision Tree Learning, the statistical test should estimate whether expanding / pruning a particular node will result in a statistically significant improvement of the performance.
  3. Combine 1. and 2.

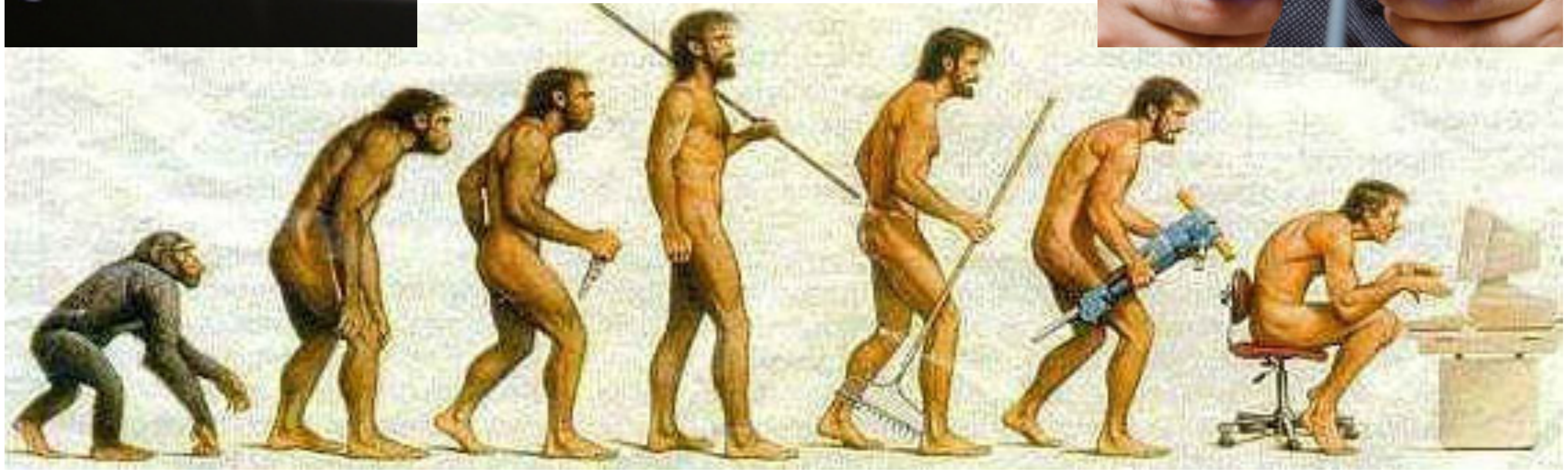
Decision Tree Learning – Exam Questions

- Tom Mitchell's book –chapter 3
- Relevant exercises from chapter 3: 3.1, 3.2, 3.3, 3.4

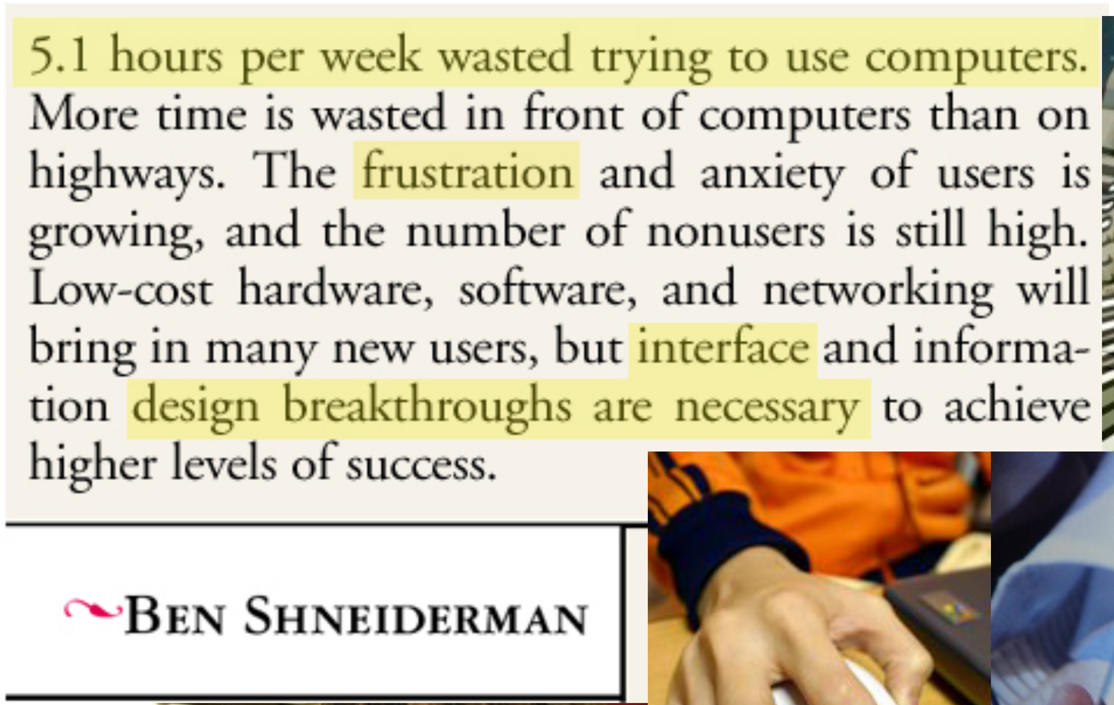
Decision Trees & CBC Intro – Lecture Overview

- Problem Representation using a Decision Tree
- ID3 algorithm
- The problem of overfitting
- ➢ Research on affective computing, natural HCI, and ambient intelligence
- ➢ Facial expressions and Emotions
- Overview of the CBC
- Group forming

Importance of Computing Technology

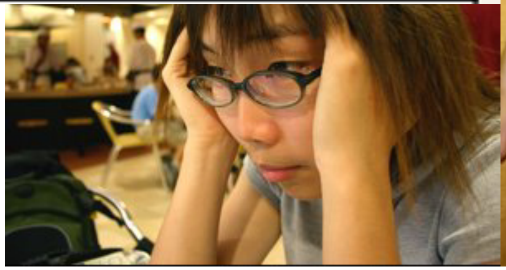


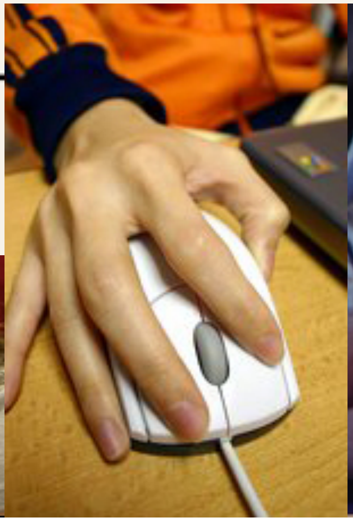
Current Human-Computer Interfaces



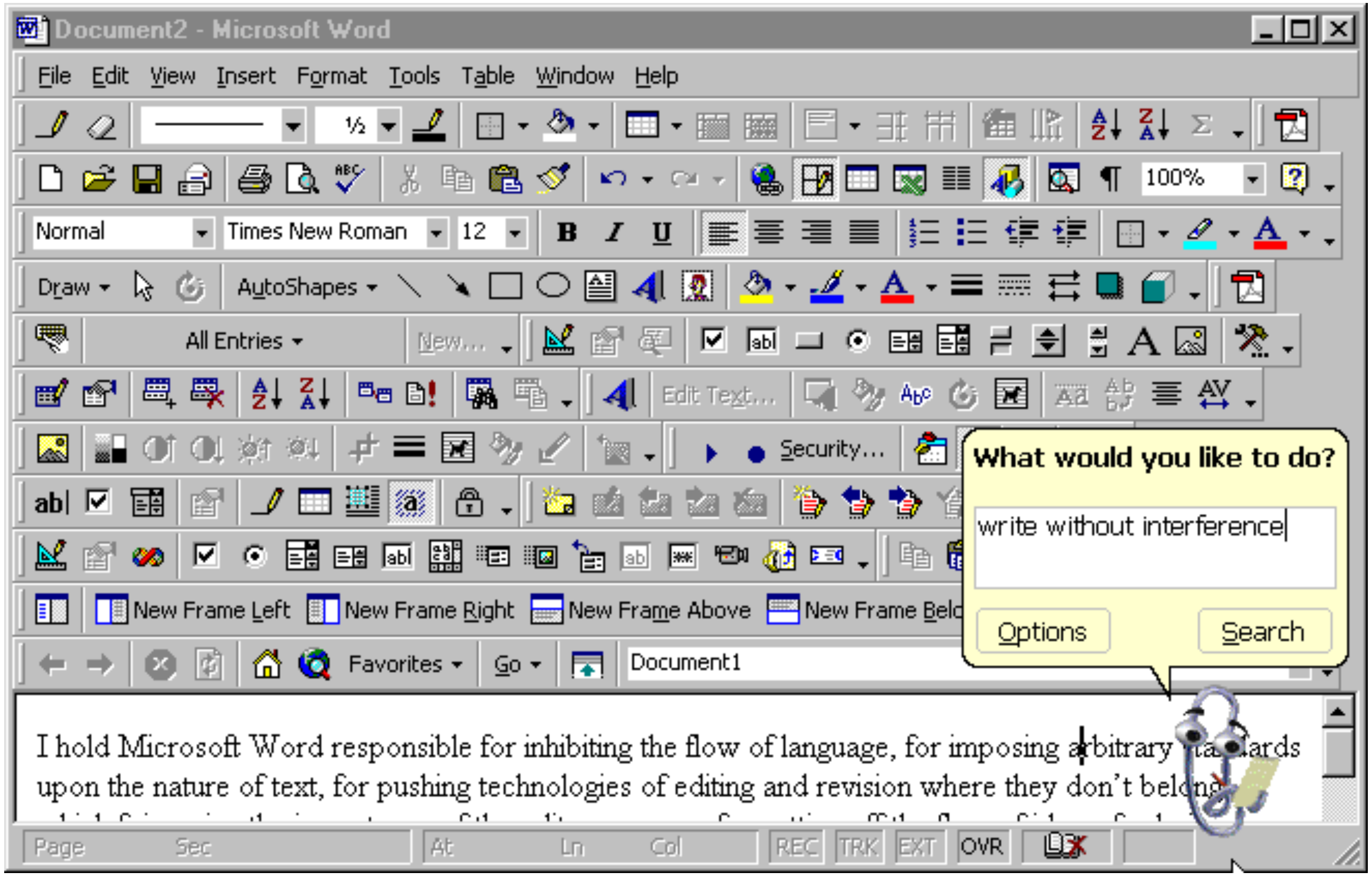
5.1 hours per week wasted trying to use computers. More time is wasted in front of computers than on highways. The frustration and anxiety of users is growing, and the number of nonusers is still high. Low-cost hardware, software, and networking will bring in many new users, but interface and information design breakthroughs are necessary to achieve higher levels of success.

~Ben Shneiderman





Current Human-Computer Interfaces



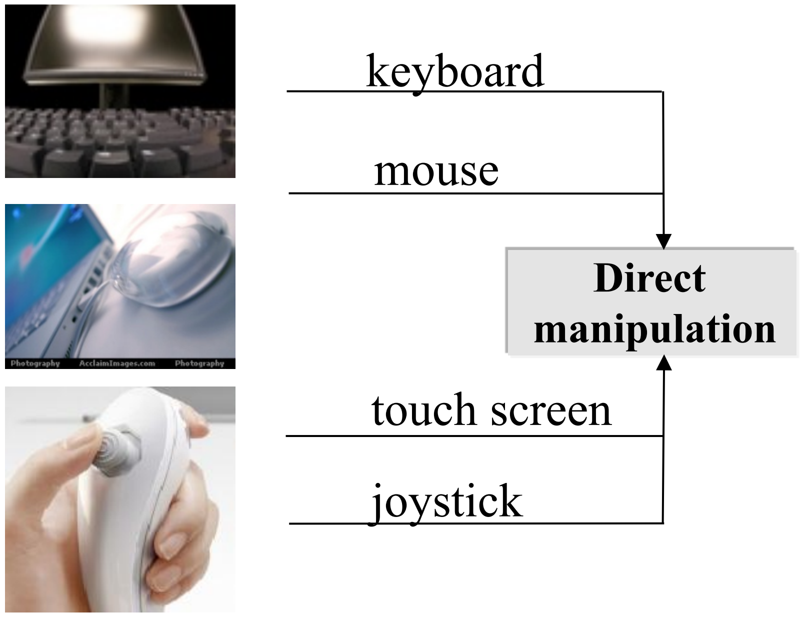
I hold Microsoft Word responsible for inhibiting the flow of language, for imposing arbitrary standards upon the nature of text, for pushing technologies of editing and revision where they don't belong

Current Human-Computer Interfaces

Human-Human Interaction:

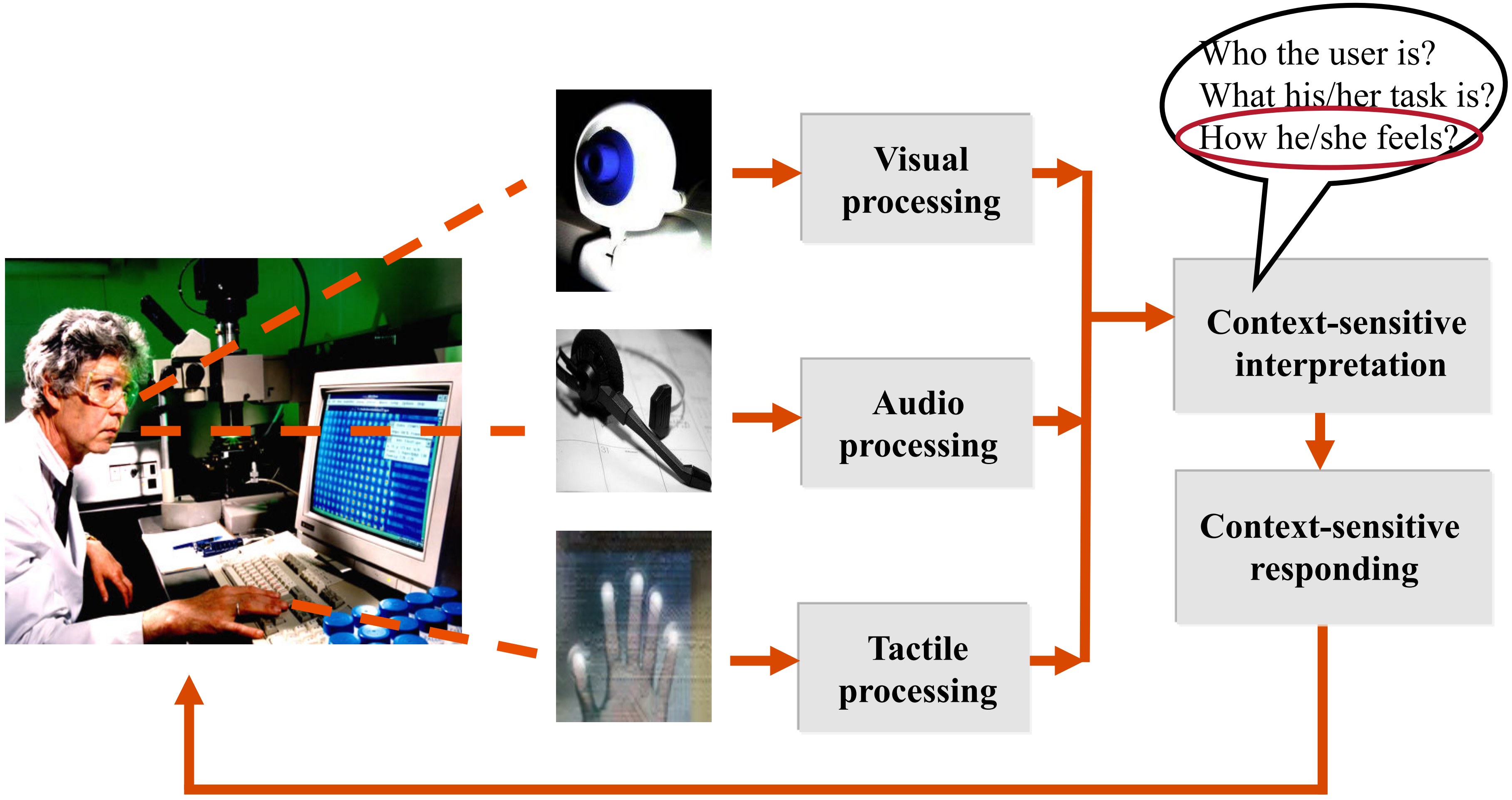
Simultaneous employment of sight, sound and touch

Human-Computer Interaction:

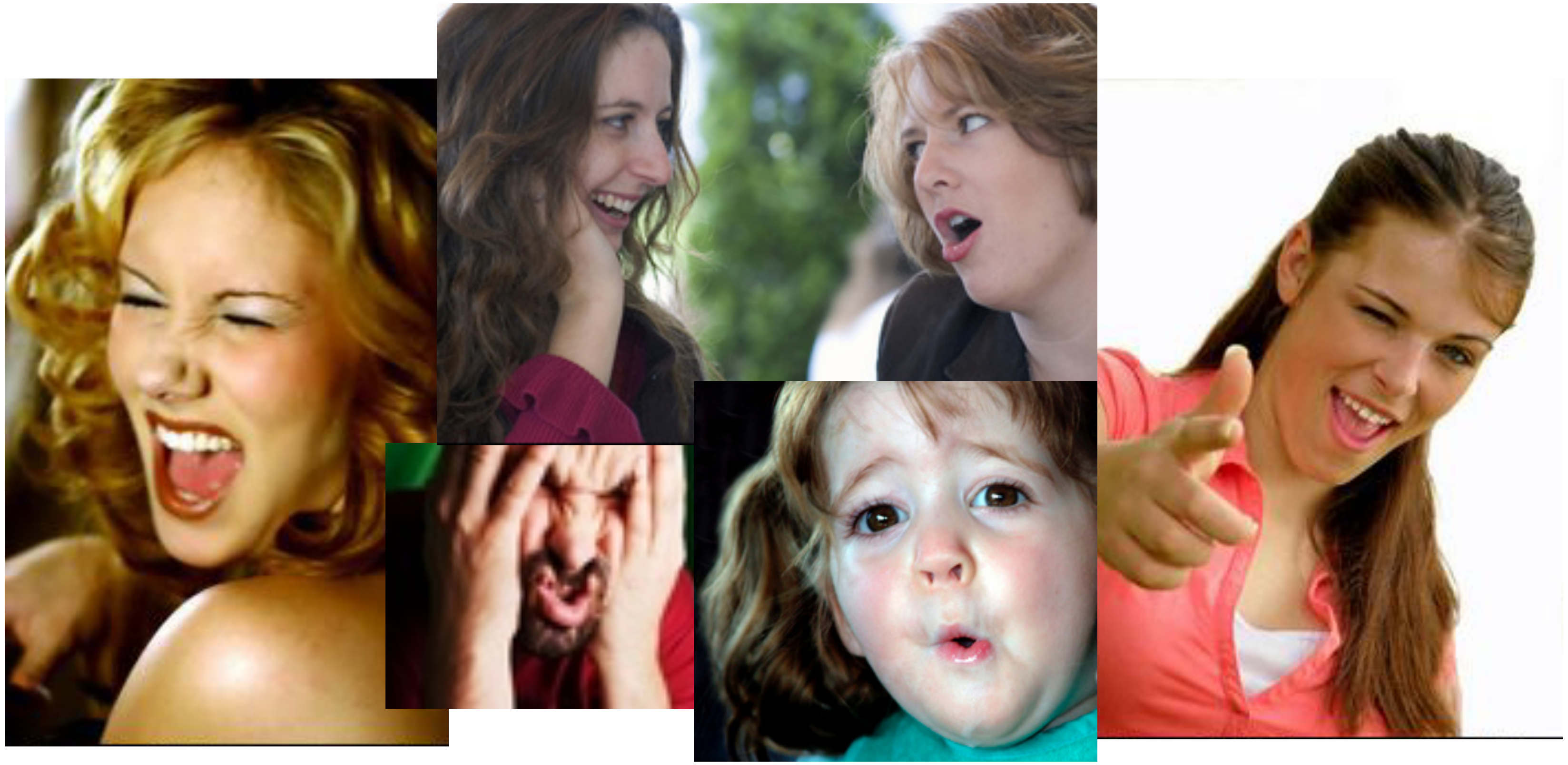


Current HCI-designs are single-modal and context-insensitive

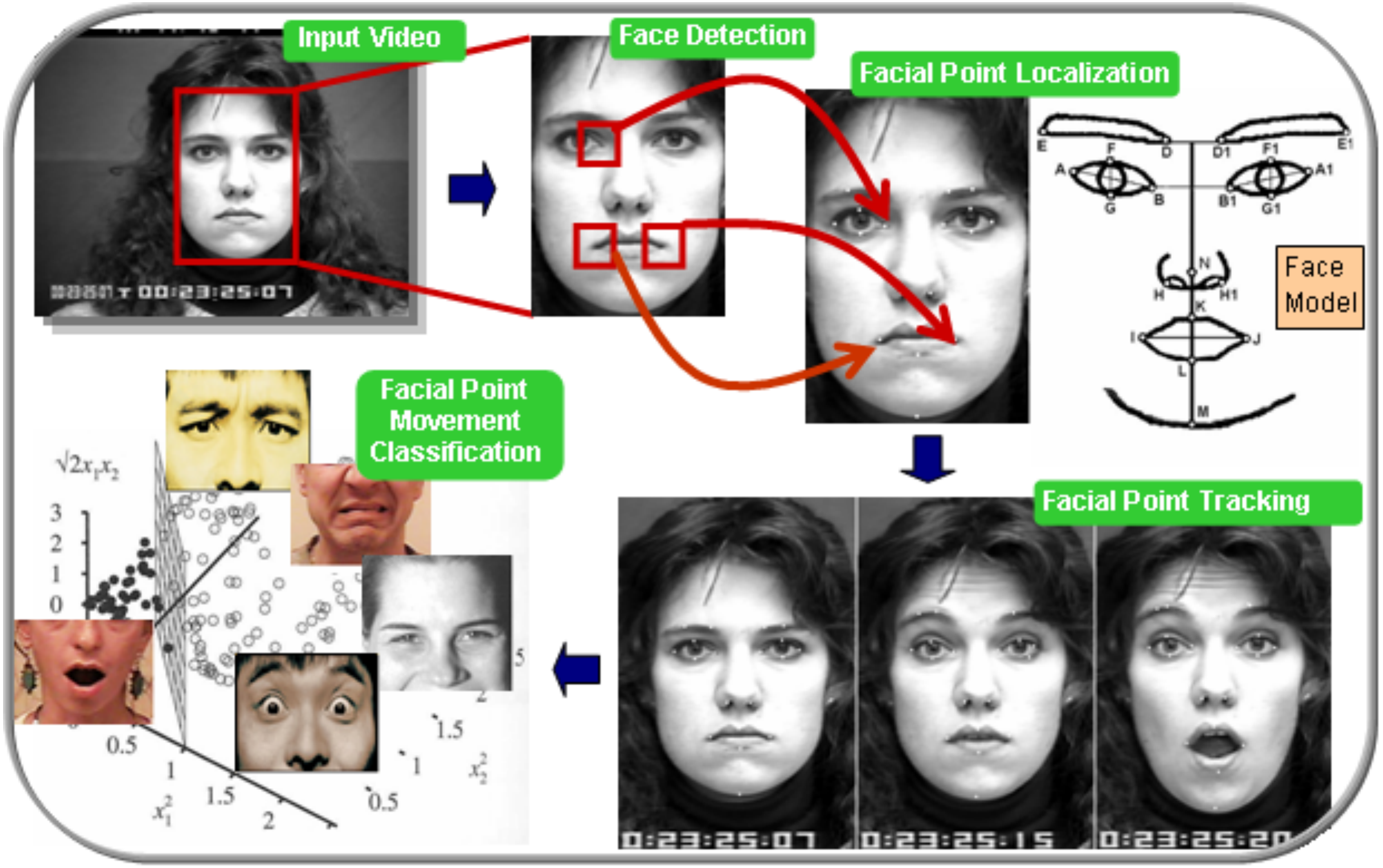
Future Human-Computer Interfaces



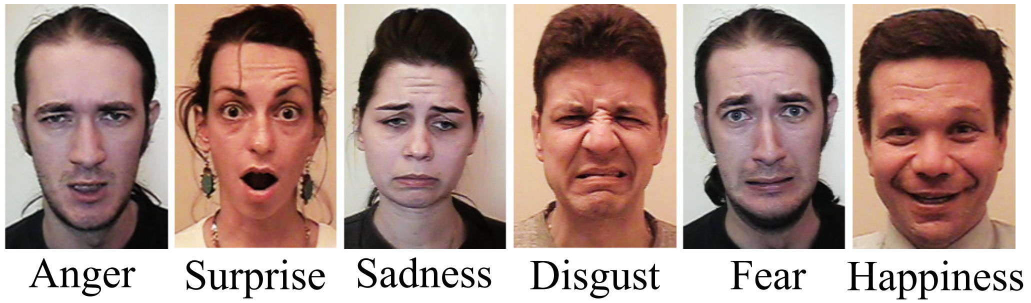
Face for Interfaces

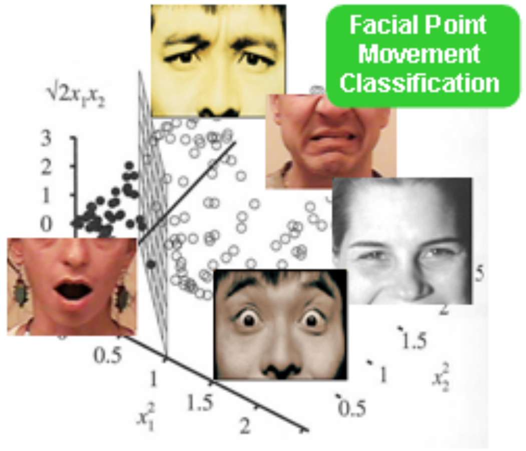


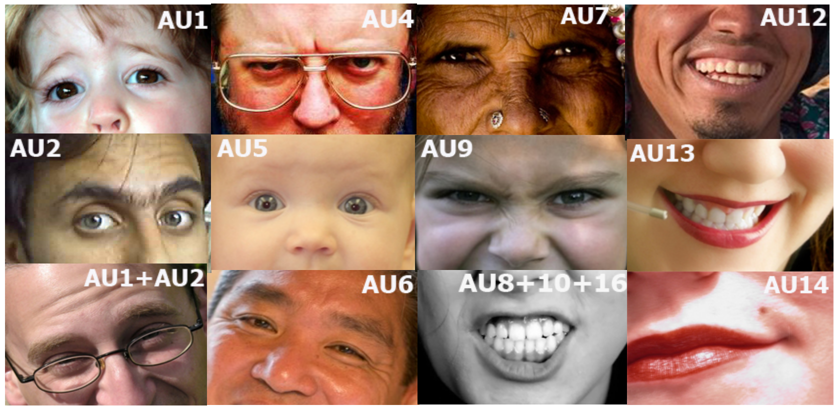
Automatic Facial Expression Analysis



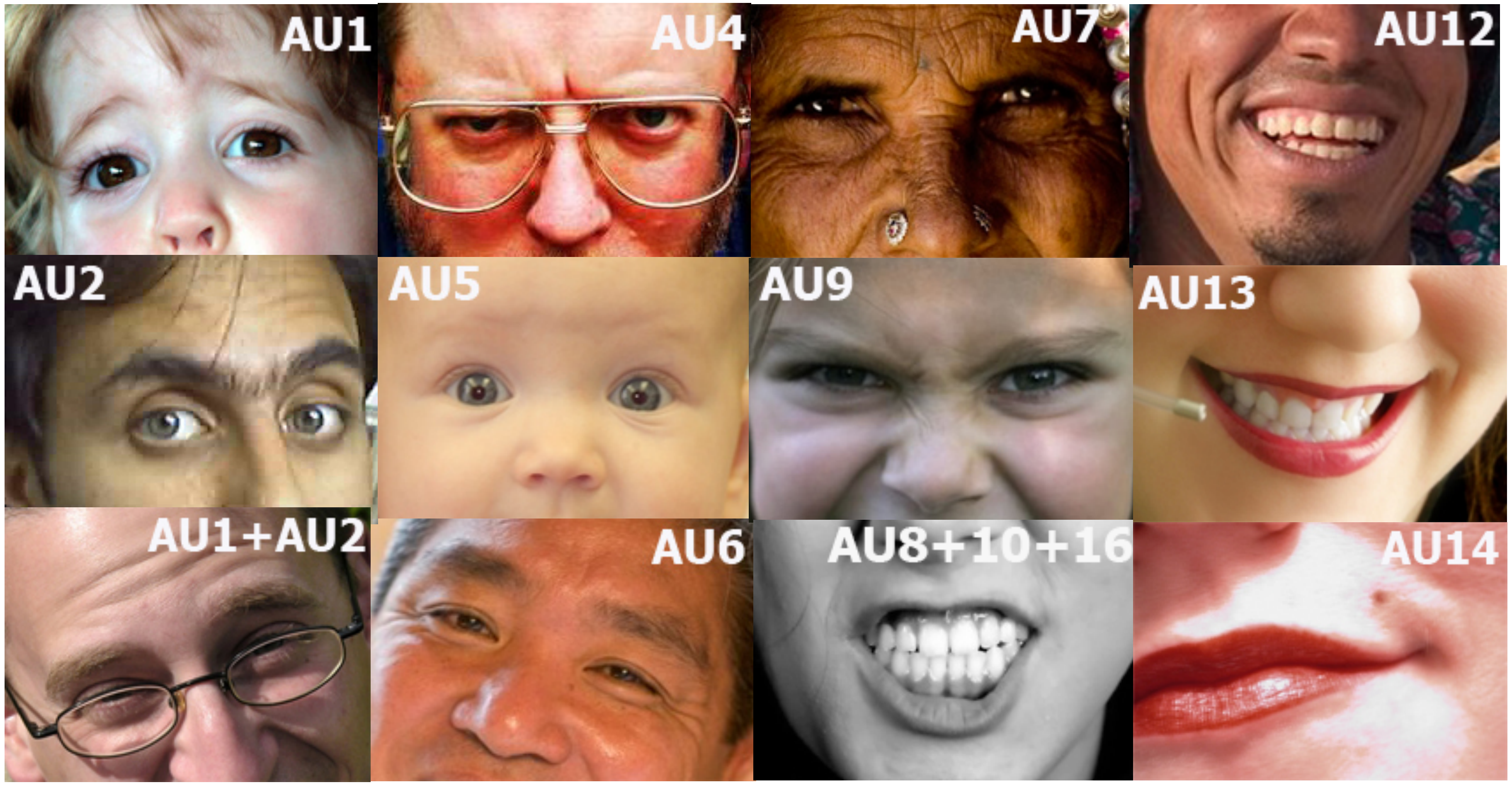
Automatic Facial Expression Analysis



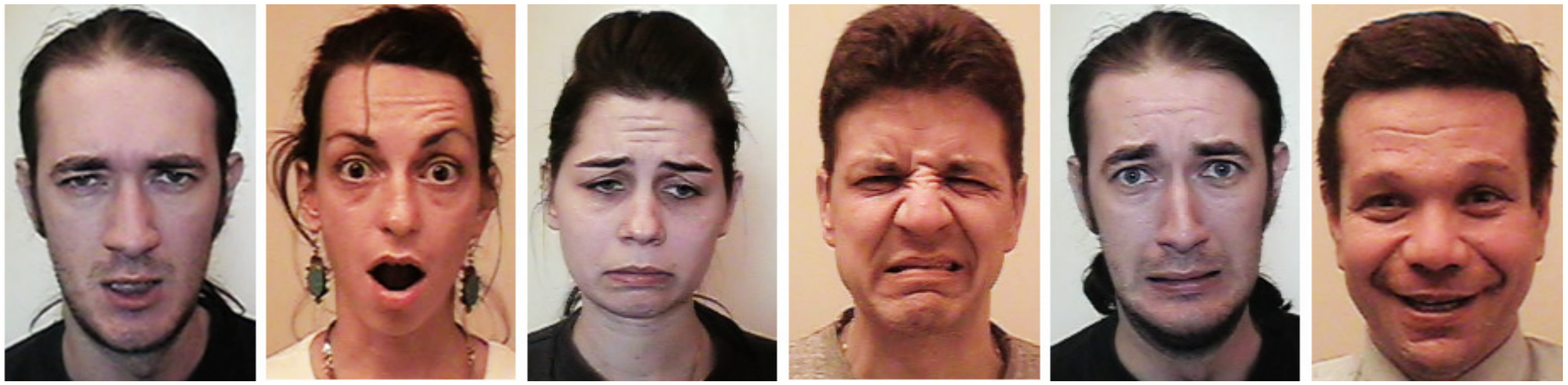




Facial Muscle Actions (Action Units - AUs)

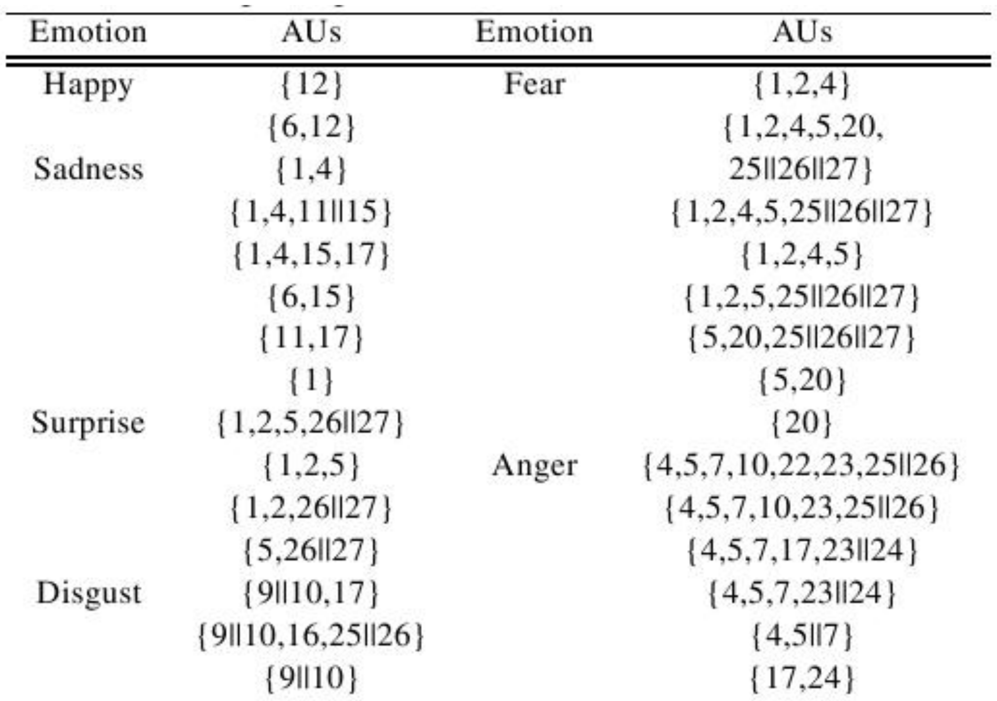


CBC – Emotion Recognition



- Prototypic facial expressions of the **six basic emotions** were introduced by Charles Darwin (1872) and elaborated by Ekman
- These prototypic facial expressions can be described in terms of AUs (e.g., surprise ↔ AU1 + AU2 + AU5 + AU26 / AU27)

CBC – Emotion Recognition



Emotion	AUs	Emotion	AUs
Happy	{12}	Fear	{1,2,4}
	{6,12}		{1,2,4,5,20,
Sadness	{1,4}		25||26||27}
	{1,4,11||15}		{1,2,4,5,25||26||27}
	{1,4,15,17}		{1,2,4,5}
	{6,15}		{1,2,5,25||26||27}
	{11,17}		{5,20,25||26||27}
	{1}		{5,20}
Surprise	{1,2,5,26||27}		{20}
	{1,2,5}	Anger	{4,5,7,10,22,23,25||26}
	{1,2,26||27}		{4,5,7,10,23,25||26}
	{5,26||27}		{4,5,7,17,23||24}
Disgust	{9||10,17}		{4,5,7,23||24}
	{9||10,16,25||26}		{4,5||7}
	{9||10}		{17,24}

V: *AUs* → *basic-emotions*

V': $\langle a_1, \ldots, a_{45} \rangle \rightarrow [1..6]$

learning algorithms:

- decision trees (ID3)
- Neural Networks
- Case-based Reasoning

evaluating developed systems:

- t-test
- ANOVA test

Decision Trees & CBC Intro – Lecture Overview

- Problem Representation using a Decision Tree
- ID3 algorithm
- The problem of overfitting
- Research on affective computing, natural HCI, and ambient intelligence
- Facial expressions and Emotions

➢ Overview of the CBC

➢ Group forming

CBC - Goal

- Hands-on experience in implementing and testing basic machine learning techniques

- Work with other team members

➢ Both your group and individual effort/performance are graded!

➢ CBC = Computer-Based Coursework

Tutorial Helpers

- A Tutorial Helper (TH) will be assigned to each group

- Joan Alabort
- Bihan Jiang
- Ioannis Marras
- Brais Martinez
- Mihalis Nicolaou
- Antonis Oikonomopoulos
- Stavros Petridis
- Ognjen Rudovic
- Jie Shen

http://ibug.doc.ic.ac.uk/people

Communication

- Via the website: http://ibug.doc.ic.ac.uk/courses/machine-learning-course-395/
  - CBC Manual
  - Provided Matlab files, datasets

- Via email: machinelearningtas@gmail.com

ALWAYS put your group number in the subject line

CBC – Tools

- Training data and useful functions are provided via the course website in a separate .tar file.

- Implementation in MATLAB
  - MATLAB basics (matrices, vectors, functions, input/output) (Assignments 2,4,5)
  - ANN Toolbox (Assignment 3)

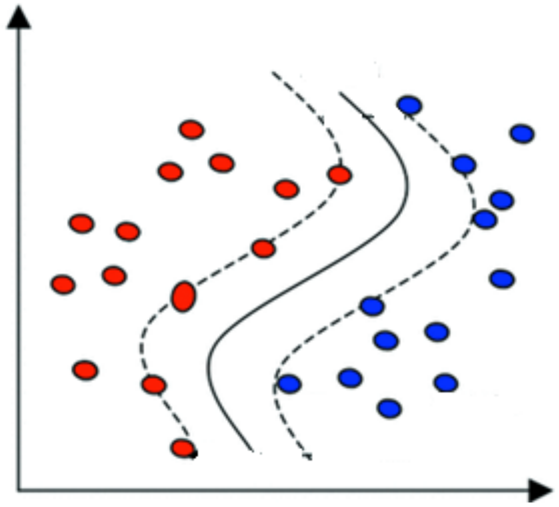
Students are strongly advised to use the MATLAB help files!

Assignment 1 : MATLAB Exercises

- Optional (no hand in required)
- A brief introduction to some basic concepts of MATLAB (that are needed in Assignments 2-5) without assessing students' acquisition, application and integration of this basic knowledge.
- The students, are strongly encouraged to go through all the material, experiment with various functions, and use the MATLAB help files extensively (accessible via the main MATLAB window).

Assignments 2-4 : Overview

- Classification Problem
  - Inputs: x *(AU vectors)*
  - Desired Output: y *(Emotion label)*

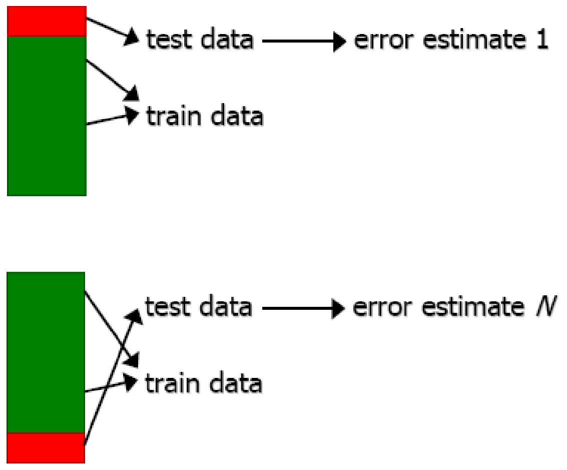


- Use x and y to train your learning algorithms to discriminate between the 6 classes (emotions)

- Evaluate your algorithms using 10-fold cross validation

- Write a function y^{pred} = *testLearner(T, x),* which takes your trained learners T and the features x and produces a vector of label predictions y^{pred}

N-fold Cross validation



- **Total error estimate:** $\hat{e} = \frac{1}{N}\sum_{i=1}^{N} e_i$

- Initial dataset is partitioned in N folds
- Training set: N - 1 folds, Test set: 1 fold
- This process is repeated N times → N error estimates
- Final error: Average of the N error estimates

Assignment 2 : Decision Trees

- Load the data using *loaddata.m* (only clean data will be used).
- Implement and train a decision tree learning algorithm
- Evaluate your trees using 10-fold cross validation
- Write a function $y^{pred} = testTrees(T, x)$, which takes your trained trees T and the features x and produces a vector of label predictions y^{pred}

Assignment 3 : Artificial Neural Networks

- Load the data using *loaddata.m* (only clean data will be used) and use the *ANNdata.m* function to transform the data into the required NN toolbox format.
- Use the Neural Networks toolbox (MATLAB built-in) to train your networks
- Evaluate your networks using 10-fold cross validation
- Write a function: $y^{pred} = testANN(N, x),$ which takes your trained networks N and produces a vector of label predictions y^{pred} .

Assignment 4 : Case Based Reasoning

- Load the data using *loaddata.m* (only clean data will be used)
- Implement and train CBR system
- Evaluate your system using 10-fold cross validation

Assignment 5 : T-test and ANOVA test

- Evaluate if the performance of the algorithms implemented so far differ significantly.
- Use the results that were previously obtained from cross validation!
- Both *clean* and *noisy* data will be used.

Group Forming

- Students will be divided in groups of 4 students

  (**NO MORE** than 5 students)

- Inform us about your team members by email by **Sunday 29th January via** machinelearningtas@gmail.com with the following information (for each group member):

  -Student login

  -Correspondence email

  -Full first Name

  -Family Name

  -Preferred name (if different from first name).

  -Degree, course/study taken, and the current year in that course.

- If you cannot form a team with 4 members then just email us the above information and we will assign you to a team.

CBC – Organisation

- Each group must hand in a report of 2 pages (excluding figures) per assignment, including discussion on implementation and answers to questions posed in the manual.
- ONE report per group
- Each group must hand in the code they implemented for each assignment.
- Hand in the code and the reports via **CATE**.

CBC – Assignment hand in

- Hand in via **CATE**
  - ➢ One group leader per group
  - ➢ Each and every group member individually has to confirm that s(he) is part of that particular group, for each and every assignment submission (under the pre-determined group leader) before each assignment submission deadline.

CBC – Organisation

- The THs will test the implemented algorithms using a separate test set (not available to the students).

- Each group will have an interview of 10-15min with two THs after the completion of each assignment. **ALL** members must be present.

- Groups **may be** invited for **a final interview** with the lecturer : March 23rd

Lab Schedule

Assisted Labs (THs present to answer questions), starting on 24th of January continuing until March 23rd

- Every Tuesday 11:00-13:00 - lab 202/206
- Every Thursday 12:00-13:00 - lab 202/206
- Every Friday 12:00-13:00 - lab 202/206

Deadlines

- Assignment 1: optional (no hand in required)
- Assignment 2: February 8th (Wednesday)
- Assignment 3: February 22nd (Wednesday)
- Assignment 4: March 6th (Tuesday)
- Assignment 5: March 14th (Wednesday)

Interviews

- Week 6 (Feb 13 – 17) – Assignment 2

  Tuesday 14/2, Thursday 16/2, Friday 17/2

- Week 8 (Feb 27 – Mar 2) – Assignment 3

  Tuesday 28/2, Thursday 1/3, Friday 2/3

- Week 11 (Mar 19 – 23) – Assignments 4, 5

  Tuesday 20/3, Thursday 22/3, Friday 23/3

➢ You will receive your interviews timetable on Tuesday Jan 31st.

CBC – Grading

- Grading will be done exclusively by the lecturer, taking into account the THs' recommendations.
- Every group member is expected to have sufficient contribution to the implementation of every assignment. Personal contribution will be evaluated during the interviews after each assignment.
- **Plagiarism is not allowed!**
  Involved groups will be instantly eliminated.

Assignment Grading

Report Content	***Implementation***	***Report Quality***
50%	***40%***	***10%***

*assignment_grade = 0.5*report_content + 0.4*implementation + 0.1*report_quality*

Assignment Grading

Grade1 ***Grade2*** ***Grade3*** ***Grade4***

four_assignment_ grade = Average(Grade1, Grade2, Grade3, Grade4)

CBC Grade

Group Grade

60%

Personal Grade

40%

*CBC_grade = 0.6*four_assignment_grade + 0.4*personal_grade*

Machine Learning Grade

CBC Grade

33.3%

Exam Grade

66.7%

- **CBC accounts for 33.3% of the final grade for the Machine Learning Course**. In other words, final grade = 0.667*exam_grade + 0.333*CBC_grade.

Decision Trees & CBC Intro – Lecture Overview

- Problem Representation using a Decision Tree
- ID3 algorithm
- The problem of overfitting
- Research on affective computing, natural HCI, and ambient intelligence
- Facial expressions and Emotions
- Overview of the CBC

➢ **Group forming**

Group Forming

- Inform us about your team members by email by **Sunday 29th January via** machinelearningtas@gmail.com with the following information (for each group member):

  -Student login

  -Correspondence email

  -Full first Name

  -Family Name

  -Preferred name (if different from first name).

  -Degree, course/study taken, and the current year in that course.

- If you cannot form a team with 4 members then just email us the above information and we will assign you to a team.

Course 395: Machine Learning – Lectures

- Lecture 1-2: Concept Learning (*M. Pantic*)
- Lecture 3-4: Decision Trees & CBC Intro (*M. Pantic*)
- ➢ Lecture 5-6: Artificial Neural Networks (*THs*)
- Lecture 7-8: Instance Based Learning (*M. Pantic*)
- Lecture 9-10: Genetic Algorithms (*M. Pantic*)
- Lecture 11-12: Evaluating Hypotheses (*THs*)
- Lecture 13-14: Guest Lectures on ML Applications
- Lecture 15-16: Inductive Logic Programming (*S. Muggleton*)
- Lecture 17-18: Inductive Logic Programming (*S. Muggleton*)