

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*)

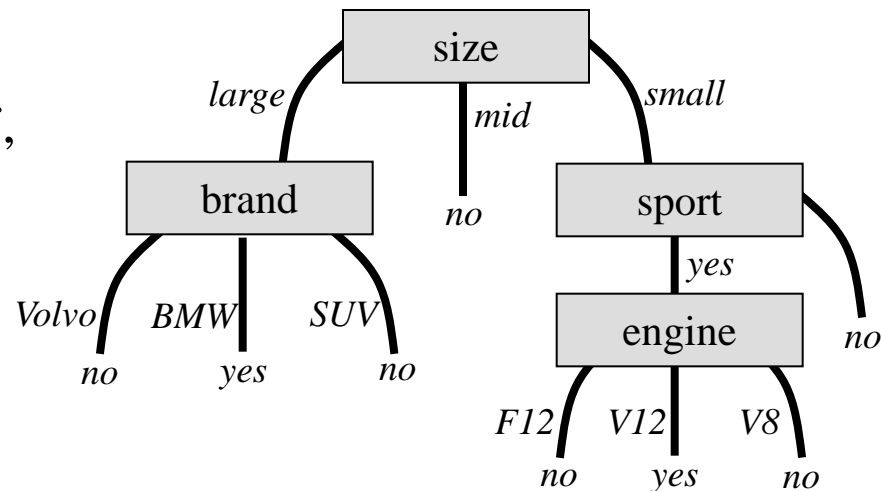
Decision Trees & CBC Intro – Lecture Overview

- 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 \leftrightarrow classification OR test of a specific attribute of the instance
 - tree branch \leftrightarrow possible value for the attribute in question

- Concept: *Good Car*
 $\langle \text{size} = \text{small}, \text{brand} = \text{Ferrari}, \text{model} = \text{Enzo}, \text{sport} = \text{yes}, \text{engine} = \text{V12}, \text{colour} = \text{red} \rangle$



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 \leftrightarrow disjunction (\vee) of sub-trees
 - sub-tree \leftrightarrow conjunction (\wedge) of constraints on the attribute values

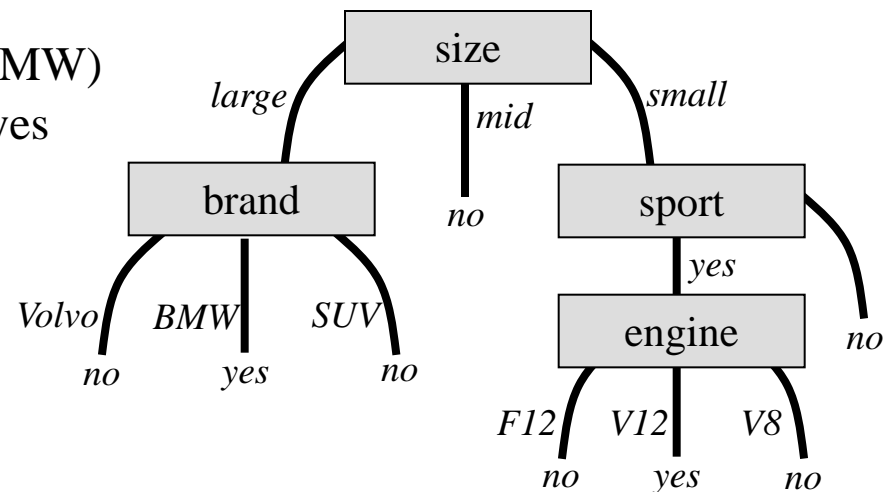
- Rule: *Good Car*

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 $\uparrow \rightarrow E(S) \uparrow$):

$$E(S) \equiv -p^{\oplus} \log_2 p^{\oplus} - p^{\otimes} \log_2 p^{\otimes},$$

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

$$\leftrightarrow p^{\oplus} = 1, p^{\otimes} = 0, E(S) = -1 \cdot 0 - 0 \cdot \log_2 p^{\otimes} = 0$$

(Note: $E(S) = 1$ if S contains equal amount of positive and negative examples

$$\leftrightarrow p^{\oplus} = 1/2, p^{\otimes} = 1/2, E(S) = -1/2 \cdot (-1) - 1/2 \cdot (-1) = 1$$

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

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

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 $\uparrow \rightarrow E(S)\uparrow$):
$$E(S) \equiv -p^{\oplus} \log_2 p^{\oplus} - p^{\otimes} \log_2 p^{\otimes},$$
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
 $\leftrightarrow p^{\oplus} = 1, p^{\otimes} = 0, E(S) = -1 \cdot 0 - 0 \cdot \log_2 p^{\otimes} = 0$
(Note: $E(S) = 1$ if S contains equal amount of positive and negative examples
 $\leftrightarrow p^{\oplus} = 1/2, p^{\otimes} = 1/2, E(S) = -1/2 \cdot (-1) - 1/2 \cdot (-1) = 1$
In the case that the target attribute can take n values:
$$E(S) \equiv -\sum_i p_i \log_2 p_i, i = [1..n]$$
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
$$IG(S, A) = E(S) - \sum_{v \in \text{values}(A)} (|S_v| / |S|) E(S_v)$$
where $\text{values}(A)$ are all possible values for attribute A , $S_v \subseteq S$ 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:

$$IG(S, A) = E(S) - \sum_{v \in \text{values}(A)} (|S_v| / |S|) E(S_v), E(S_v) \equiv \sum_v -p_v \log_2 p_v, v = [1..n].$$

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)

	<i>PlayTennis(d)</i>	<i>outlook</i>	<i>temperature</i>	<i>humidity</i>	<i>wind</i>
1	0	sunny	hot	high	weak
2	0	sunny	hot	high	strong
...
13	1	overcast	hot	normal	weak
14	0	rain	mild	high	strong

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

ID3 Algorithm – Example

	<i>PlayTennis(d)</i>	<i>outlook</i>	<i>temperature</i>	<i>humidity</i>	<i>wind</i>
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:

$$IG(S, A) = E(S) - \sum_{v \in \text{values}(A)} (|S_v| / |S|) E(S_v), E(S_v) \equiv \sum_v -p_v \log_2 p_v, v = [1..n].$$

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)

<i>PlayTennis(d)</i>	<i>outlook</i>	<i>temperature</i>	<i>humidity</i>	<i>wind</i>
...

$$IG(D, Outlook) = E(D) - 5/14 E(D_{\text{sunny}}) - 4/14 E(D_{\text{overcast}}) - 5/14 E(D_{\text{rain}}) \\ = 0.940 - 0.357 \cdot 0.971 - 0 - 0.357 \cdot 0.971 = 0.246$$

$$IG(D, Temperature) = E(D) - 4/14 E(D_{\text{hot}}) - 6/14 E(D_{\text{mild}}) - 4/14 E(D_{\text{cool}}) \\ = 0.940 - 0.286 \cdot 1 - 0.429 \cdot 0.918 - 0.286 \cdot 0.811 = 0.029$$

$$IG(D, Humidity) = E(D) - 7/14 E(D_{\text{high}}) - 7/14 E(D_{\text{normal}}) = 0.940 - 1/2 \cdot 0.985 - 1/2 \cdot 0.591 = 0.151$$

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

ID3 Algorithm – Example

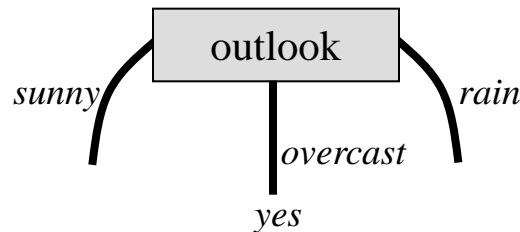
1. For each attribute A of the training examples in set S calculate:

$$IG(S, A) = E(S) - \sum_{v \in \text{values}(A)} (|S_v| / |S|) E(S_v), \quad E(S_v) \equiv \sum_v -p_v \log_2 p_v, \quad v = [1..n].$$

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)

<i>PlayTennis(d)</i>	<i>outlook</i>	<i>temperature</i>	<i>humidity</i>	<i>wind</i>
...



ID3 Algorithm – Example

	<i>PlayTennis(d)</i>	<i>outlook</i>	<i>temperature</i>	<i>humidity</i>	<i>wind</i>
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

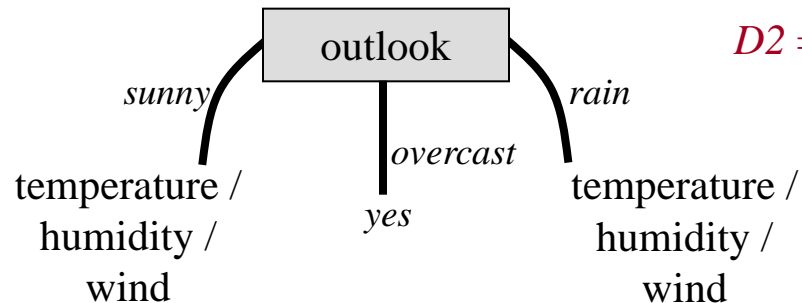
1. For each attribute A of the training examples in set S calculate:

$$IG(S, A) = E(S) - \sum_{v \in \text{values}(A)} (|S_v| / |S|) E(S_v), E(S_v) \equiv \sum_v -p_v \log_2 p_v, v = [1..n].$$
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)

<i>PlayTennis(d)</i>	<i>outlook</i>	<i>temperature</i>	<i>humidity</i>	<i>wind</i>
...

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




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

ID3 Algorithm – Example

	<i>PlayTennis(d)</i>	<i>outlook</i>	<i>temperature</i>	<i>humidity</i>	<i>wind</i>	
1	0	sunny	hot	high	weak	} <i>D1</i>
2	0	sunny	hot	high	strong	
8	0	sunny	mild	high	weak	
9	1	sunny	cool	normal	weak	
11	1	sunny	mild	normal	strong	
4	1	rain	mild	high	weak	} <i>D2</i>
5	1	rain	cool	normal	weak	
6	0	rain	cool	normal	strong	
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

	<i>PlayTennis(d)</i>	<i>outlook</i>	<i>temperature</i>	<i>humidity</i>	<i>wind</i>
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
...



$$E(D1) = \text{abs} \left(-\frac{2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5} \right) = 0.971$$

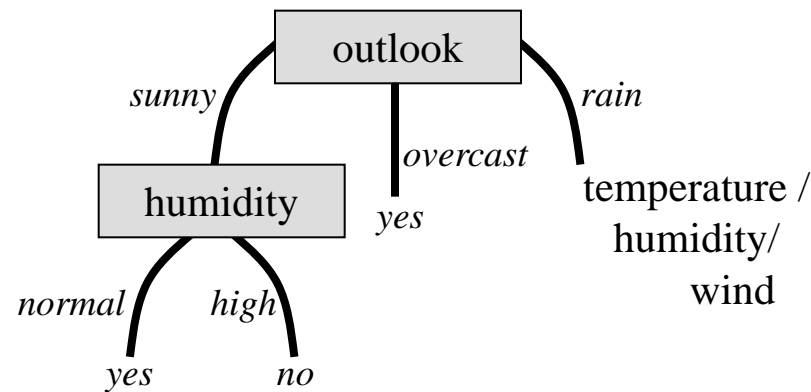

$$\begin{aligned} IG(D1, \text{Temperature}) &= E(D1) - \frac{2}{5} E(D1_{\text{hot}}) - \frac{2}{5} E(D1_{\text{mild}}) - \frac{1}{5} E(D1_{\text{cool}}) \\ &= 0.971 - 0.4 \cdot 0 - 0.4 \cdot 1 - 0.4 \cdot 0 = 0.571 \end{aligned}$$

$$\begin{aligned} IG(D1, \text{Humidity}) &= E(D1) - \frac{3}{5} E(D1_{\text{high}}) - \frac{2}{5} E(D1_{\text{normal}}) \\ &= 0.971 - 0.6 \cdot 0 - 0.4 \cdot 0 = 0.971 \end{aligned}$$

$$\begin{aligned} IG(D1, \text{Wind}) &= E(D1) - \frac{3}{5} E(D1_{\text{weak}}) - \frac{2}{5} E(D1_{\text{strong}}) \\ &= 0.971 - 0.6 \cdot 0.918 - 0.4 \cdot 1 = 0.02 \end{aligned}$$


ID3 Algorithm – Example

	<i>PlayTennis(d)</i>	<i>outlook</i>	<i>temperature</i>	<i>humidity</i>	<i>wind</i>
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
...



ID3 Algorithm – Example

	<i>PlayTennis(d)</i>	<i>outlook</i>	<i>temperature</i>	<i>humidity</i>	<i>wind</i>
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
...



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

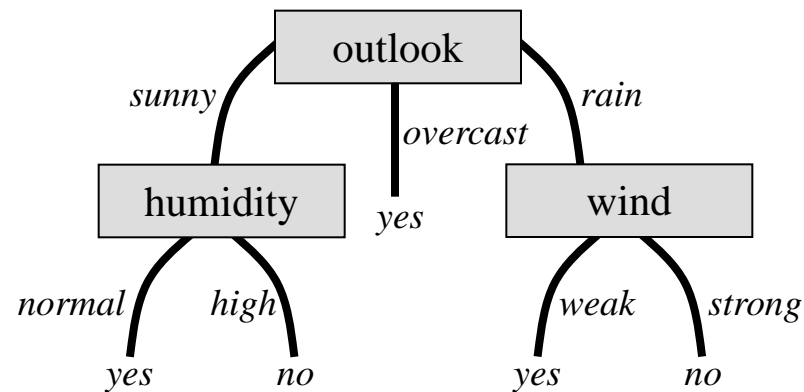
$$\begin{aligned} IG(D2, \text{Temperature}) &= E(D2) - 0/5 E(D2_{\text{hot}}) - 3/5 E(D2_{\text{mild}}) - 2/5 E(D2_{\text{cool}}) \\ &= 0.971 - 0 - 0.6 \cdot 0.918 - 0.4 \cdot 1 = 0.02 \end{aligned}$$

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

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

ID3 Algorithm – Example

	<i>PlayTennis(d)</i>	<i>outlook</i>	<i>temperature</i>	<i>humidity</i>	<i>wind</i>
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
...



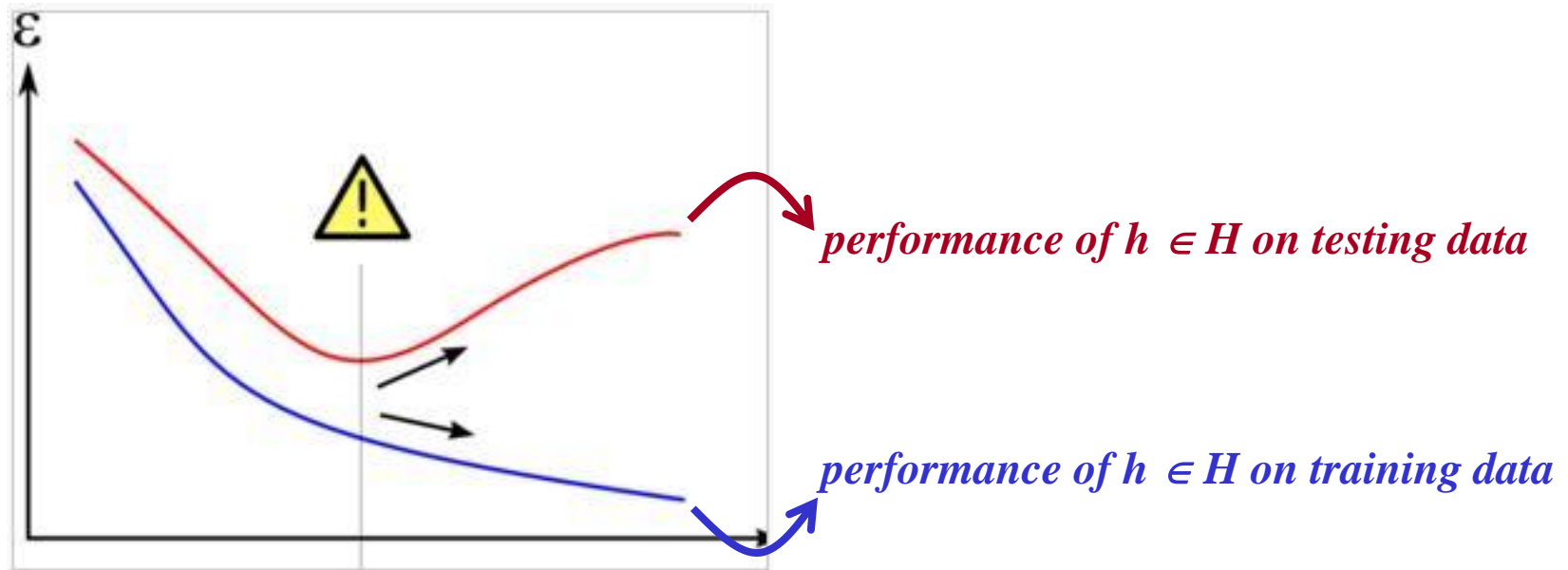
ID3 Algorithm – Advantages & Disadvantages

- 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 , $h \in H$ overfits the training data if $\exists h' \in H$ such that h has smaller error over the training examples, but h' has smaller error than h over the entire distribution of instances.



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

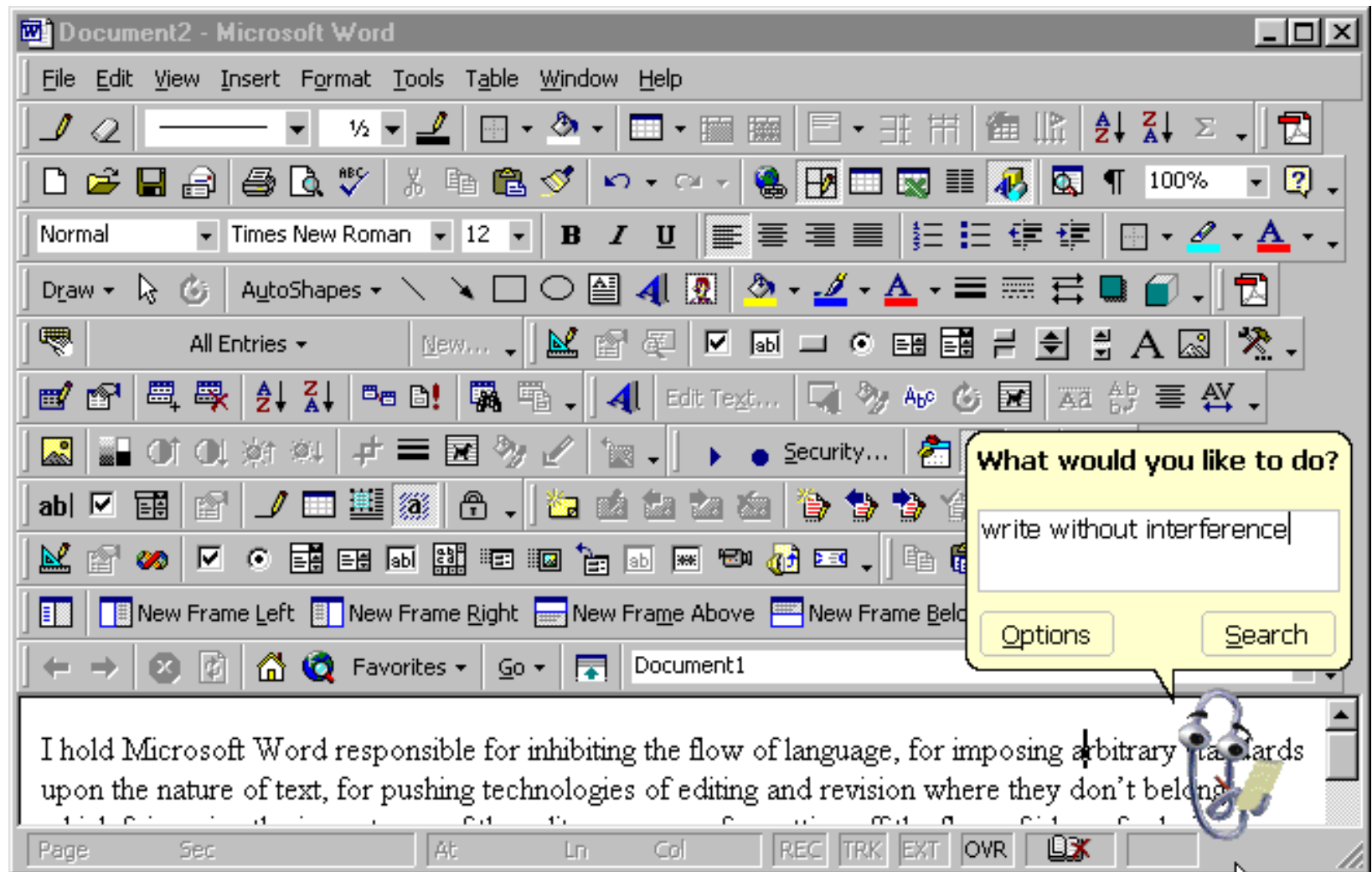
5.1 hours per week wasted trying to use computers. More time is wasted in front of computers than on highways. The **frustration** and anxiety of users is growing, and the number of nonusers is still high. Low-cost hardware, software, and networking will bring in many new users, but **interface** and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



 BEN SHNEIDERMAN



Current Human-Computer Interfaces



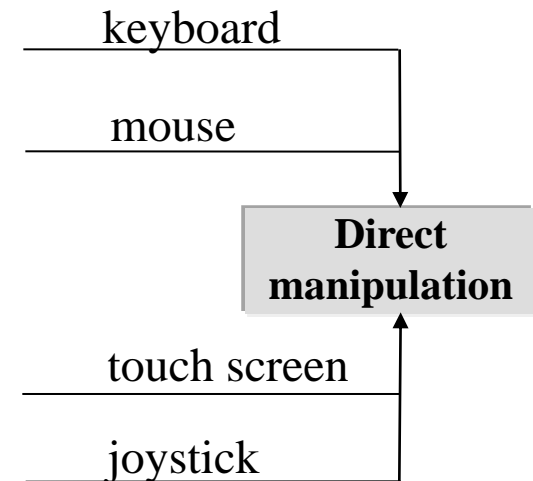
Current Human-Computer Interfaces

Human-Human Interaction:



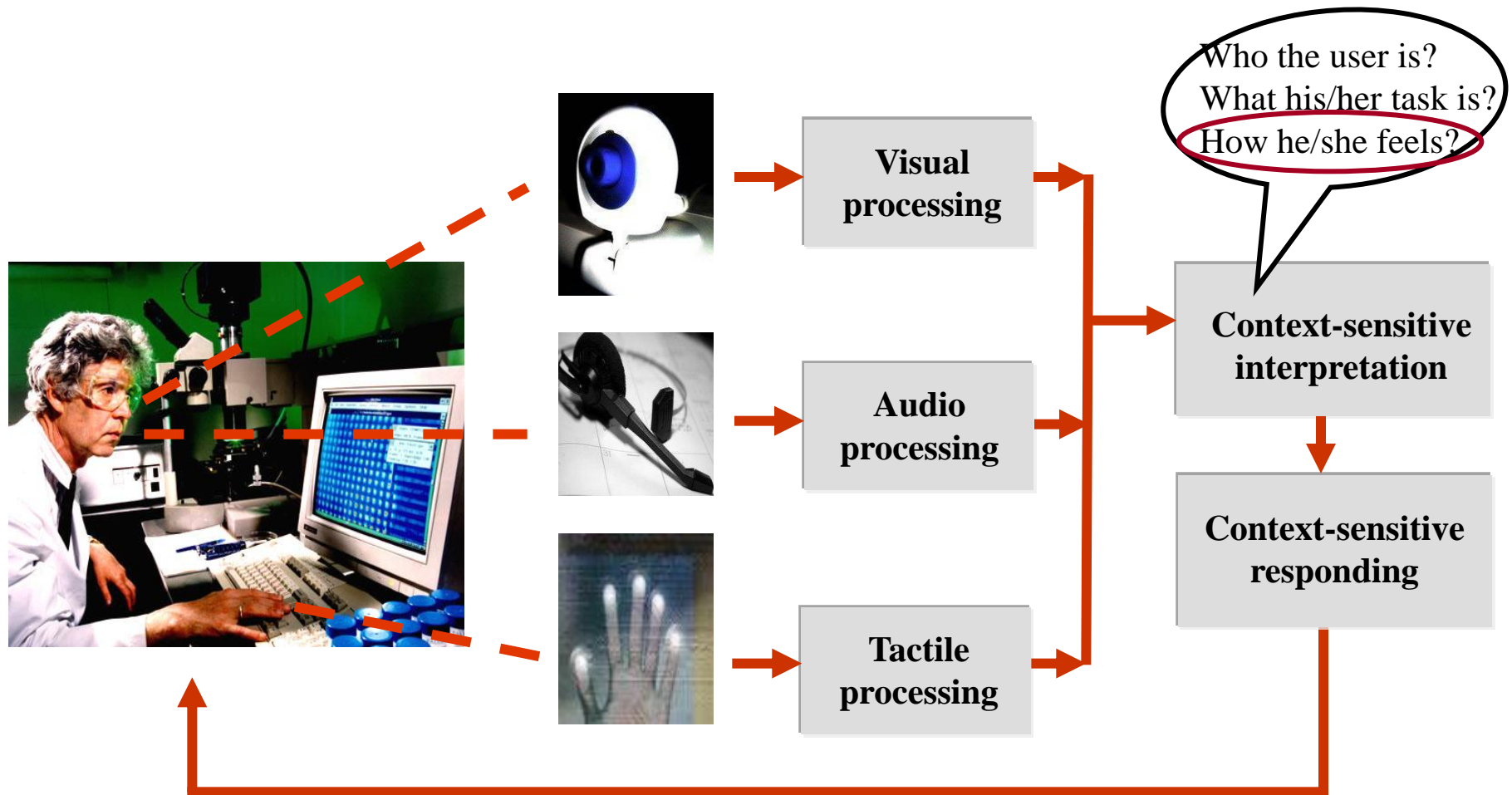
Simultaneous employment
of sight, sound and touch

Human-Computer Interaction:

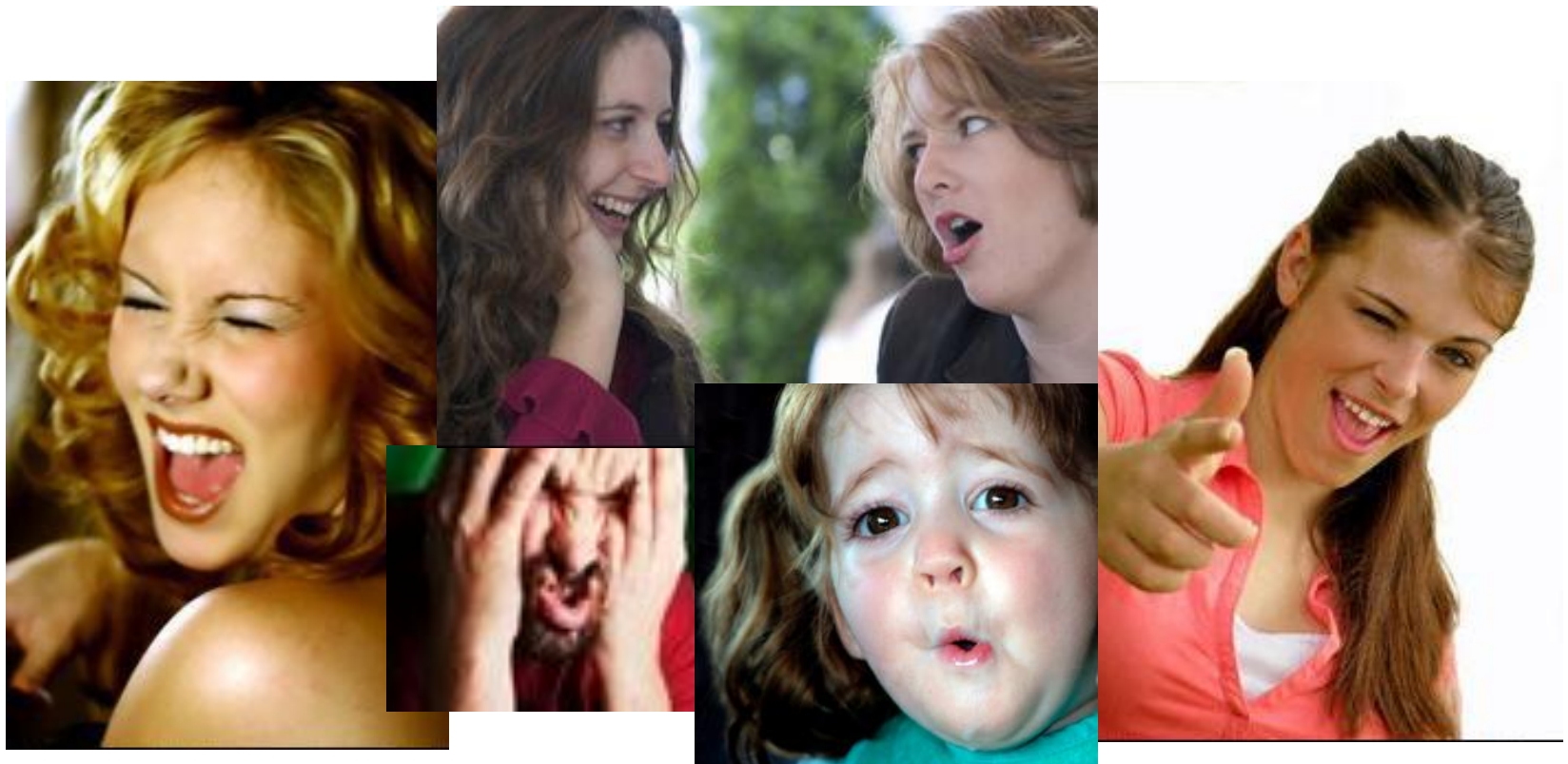


Current HCI-designs are single-
modal and context-insensitive

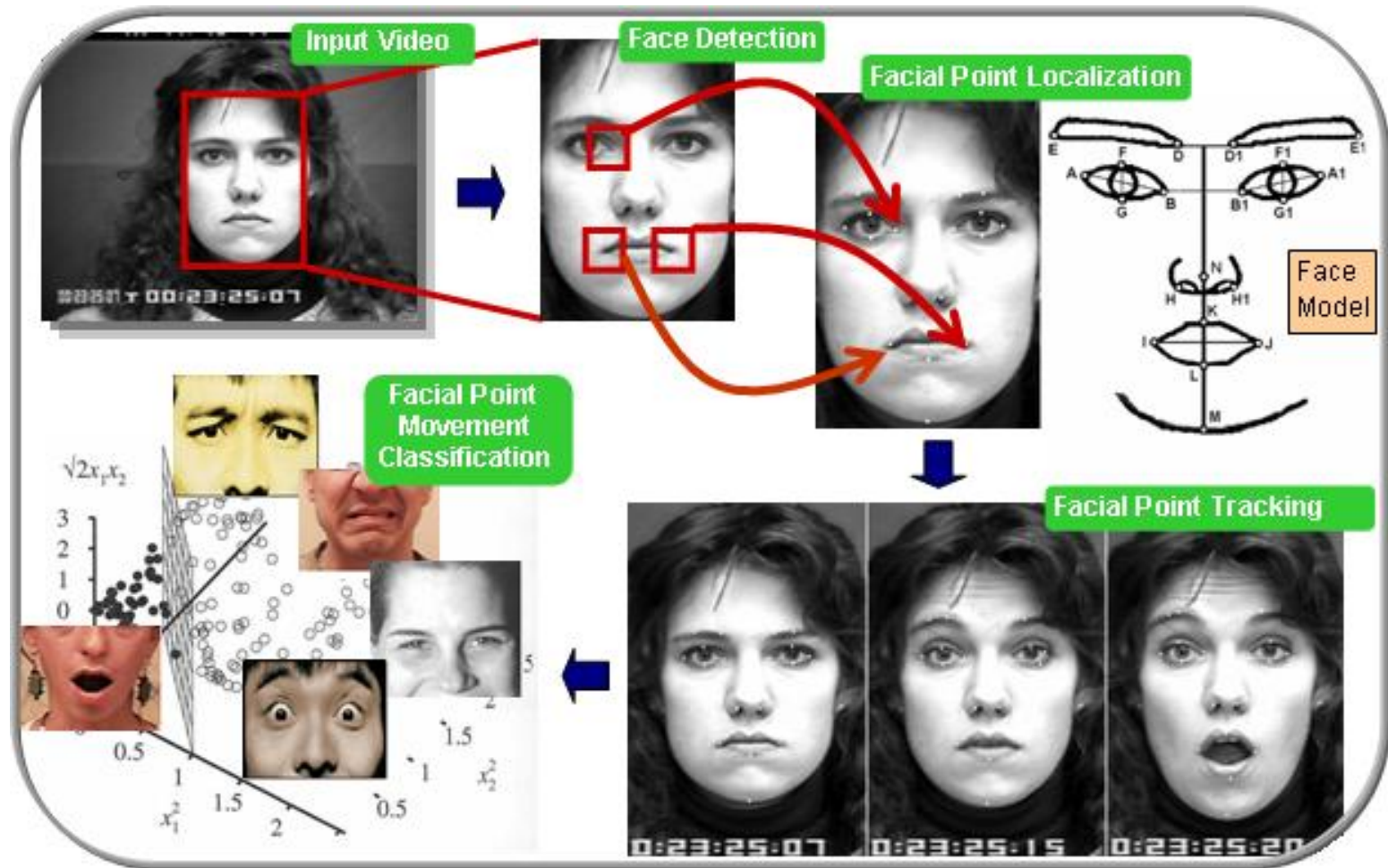
Future Human-Computer Interfaces



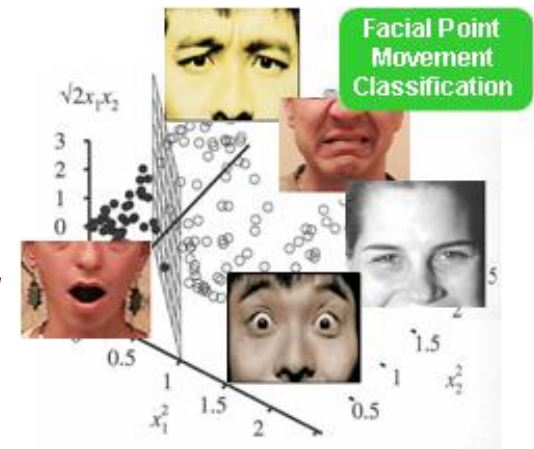
Face for Interfaces



Automatic Facial Expression Analysis



Automatic Facial Expression Analysis



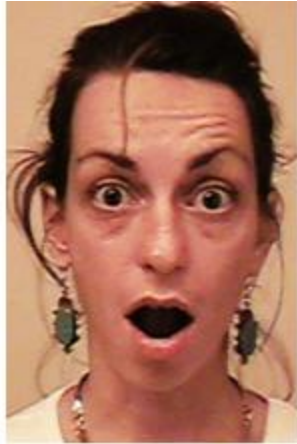
Facial Muscle Actions (Action Units - AUs)



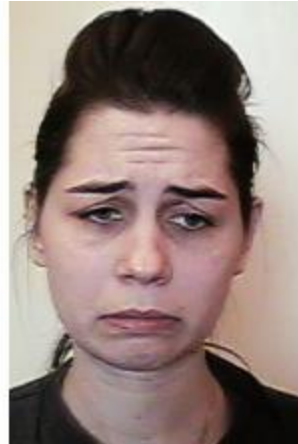
CBC – Emotion Recognition



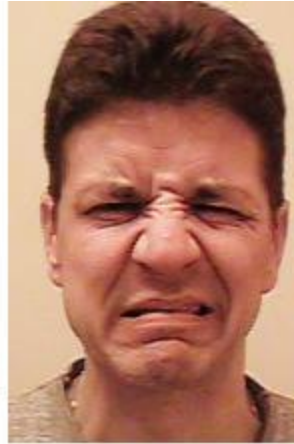
Anger



Surprise



Sadness



Disgust



Fear



Happiness

- 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 \leftrightarrow 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}

$V: AUs \rightarrow \text{basic-emotions}$

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

learning algorithms:

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

evaluating developed systems:

- t-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

Group Forming

- Students will be divided in groups of 4 students
- Simply fill in the **excel form** with the following information (for each group member):
 - Student login, email, First Name, Last Name, Course, CID
- You can find the excel form on <http://ibug.doc.ic.ac.uk/courses>, Section: Group Forming
<http://ibug.doc.ic.ac.uk/media/uploads/documents/ml-cbc-groupform.xls>
- Email the **excel form** by **Wednesday 16th October**
machinelearningtas@gmail.com

Group Forming

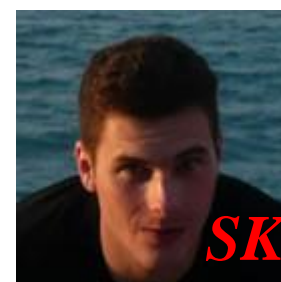
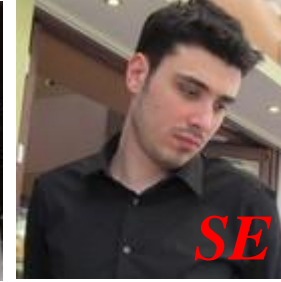
- 4 members per group, this is a hard limit!
 - In other words, groups of 5, 6, 7, etc are not allowed!!
- If you cannot form a team with 4 members then just email us the above information and we will assign you to a team.
 - Case 1 member: We will add you to a group of 2-3 members
 - Case 2 members: 1-2 more members will be added to your group
 - Case 3 members: 1 member may be added to your group
- We sometimes allow groups of 3
- Once the groups are formed you cannot change groups

Tutorial Helpers

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

- Akshay Astana
- Stefanos Eleftheriadis
- Christos Georgakis
- Bihan Jiang
- Sebastian Kaltwang
- Ioannis Marras
- Brais Martinez
- Mihalís Nicolaou
- Symeon Nikitidis
- Ioannis Panagakis
- Athanasios Papaioannou
- Stavros Petridis

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



Communication

- Via the website: <http://ibug.doc.ic.ac.uk/courses/machine-learning-course-395/>
 - CBC Manual (Section Course Material)
 - Provided Matlab files, datasets (Section Data and Tools)
 - Tutorials
- Via email: machinelearningtas@gmail.com

ALWAYS put your group number in the subject line

CBC – Organisation

- Each group must hand in a report of ~3 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 – Report + Code marking

- The THs will mark your report and provide feedback
- The THs will test the implemented algorithms using a separate test set (not available to the students).
- We will inform you about the performance of your algorithm on our test set.
- We just want to check that your code runs and your classifier has been properly trained, i.e., it can generalise to unknown data
- Performance is not that important, the accuracy should be at least 60%

CBC – Interviews

- Each group will have an interview of ~20min with two THs after the completion of each assignment. ALL members must be present.
- Interviews will take place in pairs, i.e., first two members (~10min) and then the other two (~10)
- Questions will be similar
- We just want to check that you yourselves did all the work and you are familiar with the basic concepts
- All members should know everything that is included in the report

Lab Schedule

Assisted Labs (THs present to answer questions), starting on October 17th continuing until December 4th

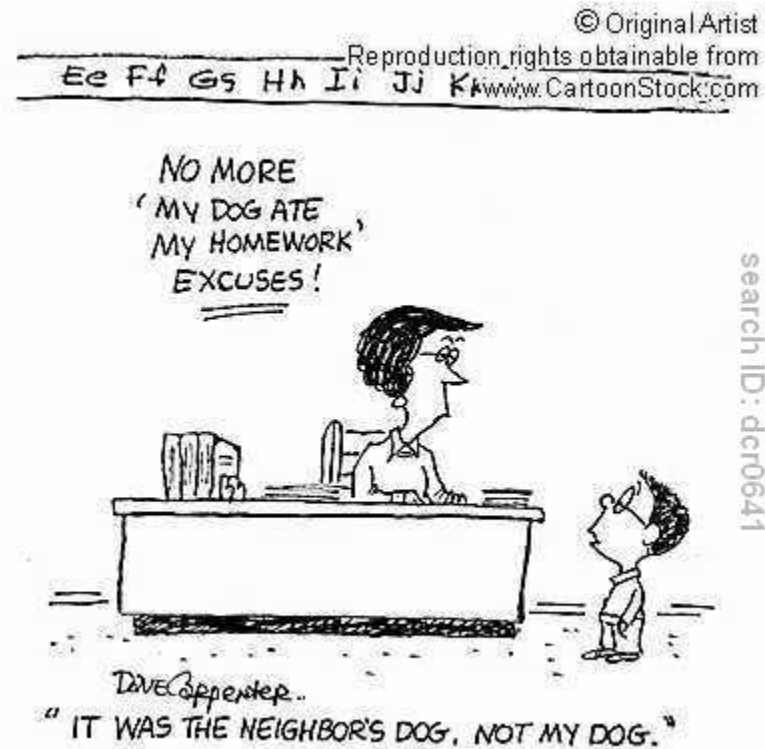
- Every Monday 12:00-13:00, lab 219
- Every Wednesday 9:00-11:00, 14:00 – 16:00, lab 219
- Every Thursday 14:00 – 15:00, lab 219

Deadlines

- Assignment 1: optional (no hand in required)
- Assignment 2: November 4th (Monday)
- Assignment 3: November 21st (Thursday)
- Assignment 4: December 4th (Wednesday)

Late Submissions

- *-20% up to 12h*
- *-40% up to 24h*
- *-75% up to 36h*
- *-100% >36h*



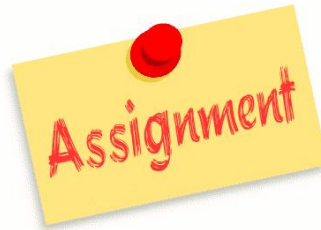
Interviews

- Week 7 (Nov 11 – 15) – Assignment 2
Monday 11/11, Wednesday 13/11, Thursday 14/11
 - Week 9 (Nov 25 – Nov 29) – Assignment 3
Monday 25/11, Wednesday 27/11, Thursday 28/11
 - Week 11 (Dec 9 – 13) – Assignments 4
Monday 9/12, Wednesday 11/12, Thursday 12/12
- You will receive your interviews timetable after the submission of each assignment.
 - If you are not available then let us know asap, the interview cannot take place without all the members being present.
 - We can only return the marked courseworks after the interviews take place.

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
75%

Code
15%

Report Quality
10%

$$\text{Group_grade} = 0.75 * \text{report_content} + 0.15 * \text{code} + 0.1 * \text{report_quality}$$

- ***Code***: Performance on unknown Test set (>60%) + Flowchart

CBC Grade



Group Grade

60%



Personal Grade

40%

Personal_grade = interview grade

*Interview grade [0, 1.1*Group Grade]*

*Assignment_grade = 0.6*group_grade + 0.4*personal grade*

Assignment Grading



Grade1



Grade2



Grade3

$CBC_grade = Average(Grade1, Grade2, Grade3)$

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 = $0.667 \times \text{exam_grade} + 0.333 \times \text{CBC_grade}$.***

CBC – Tools

- Training data and useful functions are provided via the course website in a separate .rar file.
- Implementation in MATLAB
 - MATLAB basics (matrices, vectors, functions, input/output)
(Assignments 2,4)
 - 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 2-4) 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).

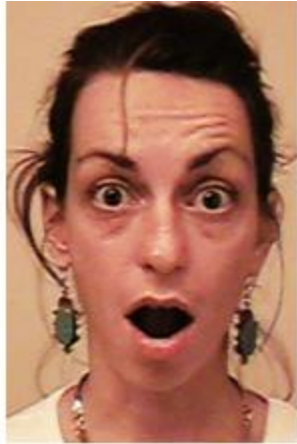
Facial Muscle Actions (Action Units - AUs)



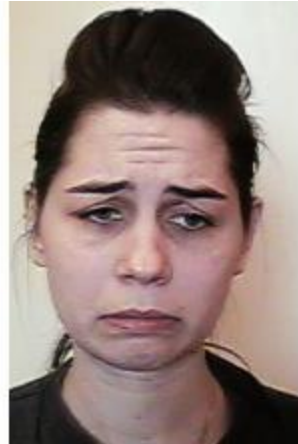
CBC – Emotion Recognition



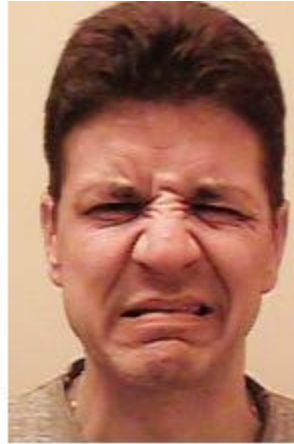
Anger



Surprise



Sadness



Disgust



Fear



Happiness

- 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 \leftrightarrow 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}

$V: AUs \rightarrow \text{basic-emotions}$

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

learning algorithms:

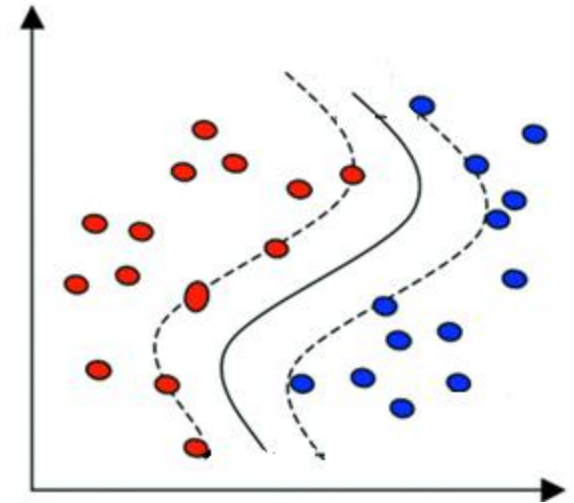
- decision trees (ID3)
- Neural Networks
- Case-based Reasoning

evaluating developed systems:

- t-test

Assignments 2-4 : Overview

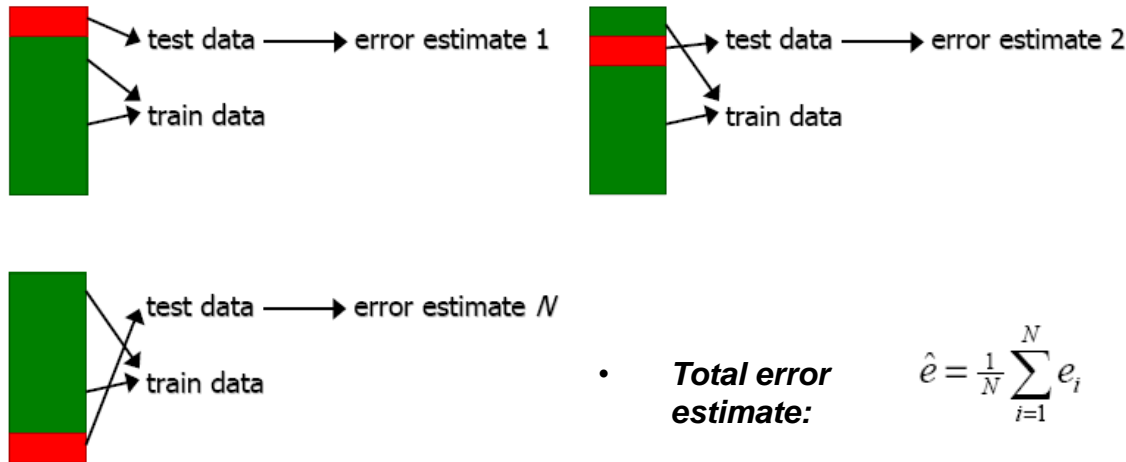
- Classification Problem
 - Inputs: x (*AU vectors*)
 - Desired Output: y (*Emotion label*)
- Use x and y to train your learning algorithms to discriminate between the 6 classes (emotions)
- Evaluate your algorithms using 10-fold cross validation
- Write a function $y^{pred} = \text{testLearner}(T, x)$, which takes your trained learners T and the features x and produces a vector of label predictions y^{pred}



Training – Validation – Test Sets

- Training Set: Used to train the classifiers
- Validation Set: Used to optimise the parameters of the classifiers
 - e.g. number of hidden neurons in neural networks
- Test Set: Used to measure the performance of the classifier

N-fold Cross validation



- Initial dataset is partitioned in N folds
- Training + **Validation** 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

- 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}
- Theoretical / Implementation questions

Assignment 3 : Artificial Neural Networks

- Use the Neural Networks toolbox (MATLAB built-in) to train your networks
- Evaluate your networks using 10-fold cross validation
- Write a function: $y^{\text{pred}} = \text{testANN}(N, x)$, which takes your trained networks N and produces a vector of label predictions y^{pred} .
- Theoretical / Implementation questions

Assignment 4 Part A : Case Based Reasoning

- Implement and train CBR system
- Evaluate your system using 10-fold cross validation
- Theoretical / Implementation questions

Assignment 4 Part B : T-test

- Evaluate if the performance of the algorithms implemented so far differ significantly.
- Use the results that were previously obtained from cross validation!
- Theoretical / Implementation questions

CBC – Tips

- Question from 3 years ago: “Can you give us extra work so we can get extra marks?”
 - No, the number of assignments is fixed, it is 3!
- Work together!!! Divide the work, but you should all know how you have implemented your algorithms, why you took some specific decision, etc...
- The algorithms are tested on Matlab so please do not send us images (jpg, png etc) of your implemented algorithms. Save your algorithm in .mat format.
- If you have any complaints (marking, THs etc) contact the course support leader
- All members should know everything that is included in the report

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

- Students will be divided in groups of 4 students
- Simply fill in the **excel form** with the following information (for each group member):
 - Student login, email, First Name, Last Name, Course, CID
- You can find the excel form on <http://ibug.doc.ic.ac.uk/courses>, Section: Group Forming
<http://ibug.doc.ic.ac.uk/media/uploads/documents/ml-cbc-groupform.xls>
- Email the **excel form** by **Tuesday 16th October**
machinelearningtas@gmail.com
- If you cannot form a team with 4 members then just email us the above information and we will assign you to a team.