Evolving Classification of the Behaviour of Computer Users

Jose Antonio Iglesias

CAOS Research Group.
Carlos III University of Madrid (Spain)
jiglesia@inf.uc3m.es

Plamen Angelov

Comunication System Department.
Lancaster Univeresity (UK)
p.angelov@lancaster.ac.uk

InfoLab21

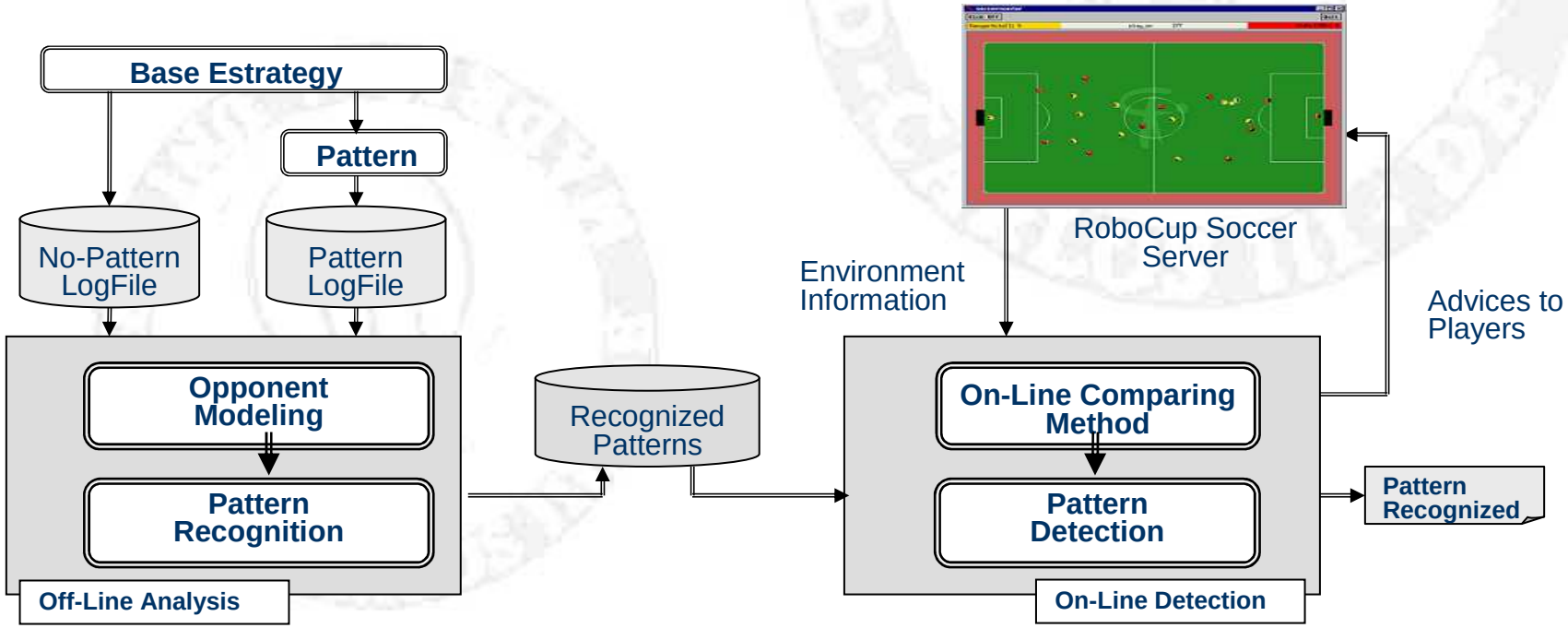
Universidad Carlos III de Madrid

Outline

- From RoboCup …
- … to UNIX User Classification
- Creating a Profile Model
- Evolving classification
- Results
- Conclusions

From RoboCup…

Opponent behavior Modelling / Classification (Environment: soccer simulation domain)



From RoboCup…

Behavior Classification

Behavior as ***sequence*** **of actions**

Pass1to2, Dribble2, Pass2to10, Goal10,

Pass5to6, Pass6to3, Foul5, MissedShot3…

… to UNIX Users Commands

Behavior Classification

Behavior as ***sequence*** **of commands**

SS, postnews, enscript, fg, enscript, cd, *ES*, *SS*, rlogin…

SS, date, ls, emacs, rm, ls, date ...

Outline

- From RoboCup …
- … to UNIX User Classification
- **Creating a Profile Model**
- Evolving classification
- Results
- Conclusions

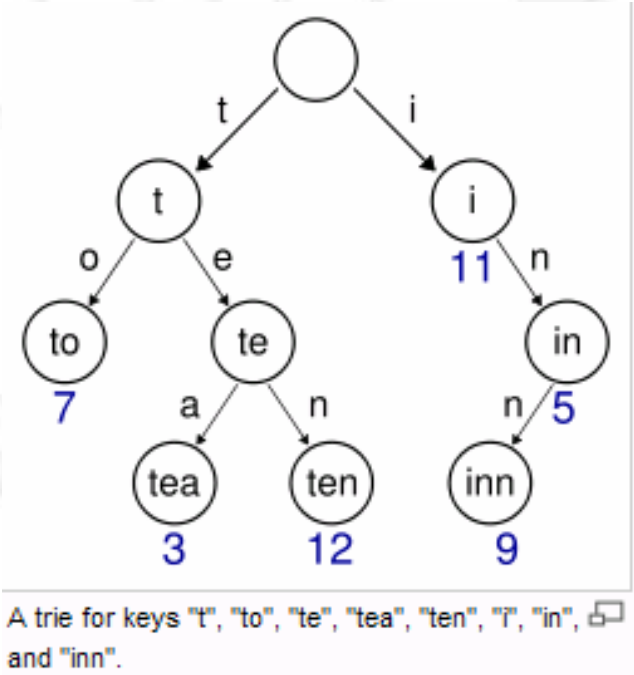
Sequence of commands → Profile Model

Behavior as ***sequence*** **of commands**

Important: sequentiality

Trie *(re**trie**val)* data structure:

Special search tree used for storing elements and its prefixes.



A trie for keys "t", "to", "te", "tea", "ten", "i", "in", and "inn".

Every node:

- **represents a command**
- **stores useful information (times appeared,…)**

Sequence of commands → Profile Model

An example trie

Sequence
pwd
vi
pwd
vi
pwd
ls

Sequence to insert initially in the trie:

{ls → date → ls → dat → cat}

An example trie

pwd
vi
pwd
vi
pwd
ls

Sequence

Sequence to insert initially in the trie:

{ls → date → ls → date → cat}

Sub-sequence length: 3

{ls → date → ls → date → cat}

Sub-sequences to insert in the trie:

{ls → date → ls} and {date → ls → date } and {ls → date → cat}

Sequence of commands → Profile Model

An example trie

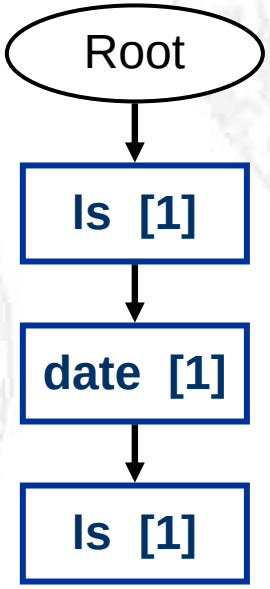
Sub-sequences to insert in the trie:

{ls → date → ls} and {date → ls → date} and {ls → date → cat}

An example trie

Sub-sequences to insert in the trie:

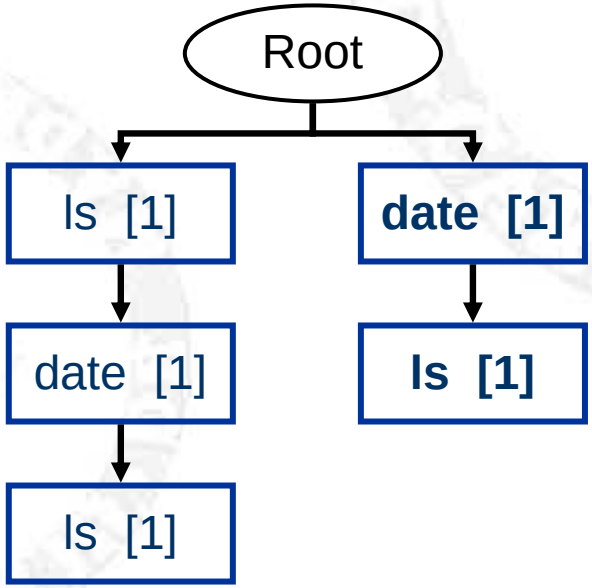
{ls → date → ls} and {date → ls → date} and {ls → date → cat}



An example trie

Sub-sequences to insert in the trie:

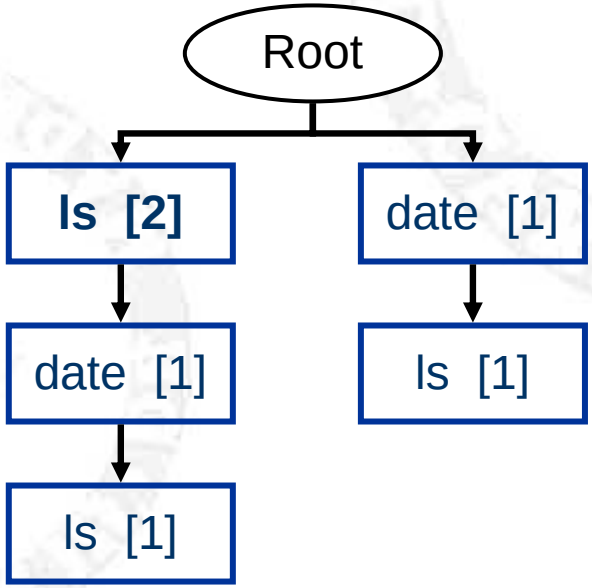
{ls → date → ls} and {date → ls → date} and {ls → date → cat}



An example trie

Sub-sequences to insert in the trie:

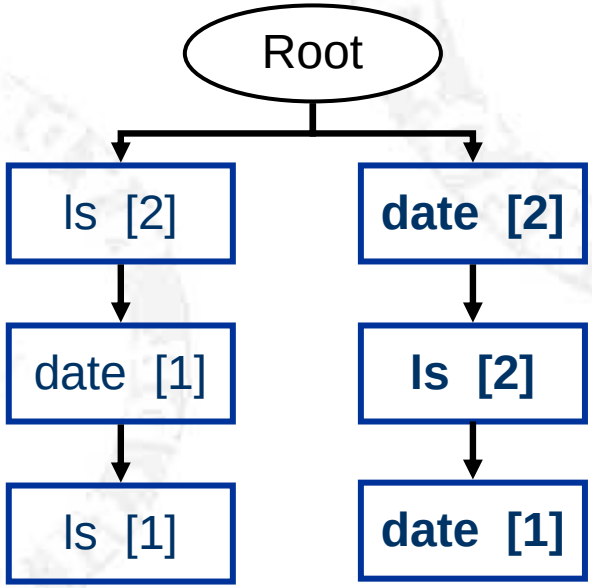
{ls → date → ls} and {date → ls → date} and {ls → date → cat}



An example trie

Sub-sequences to insert in the trie:

{ls → date → ls} and {date → ls → date} and {ls → date → cat}



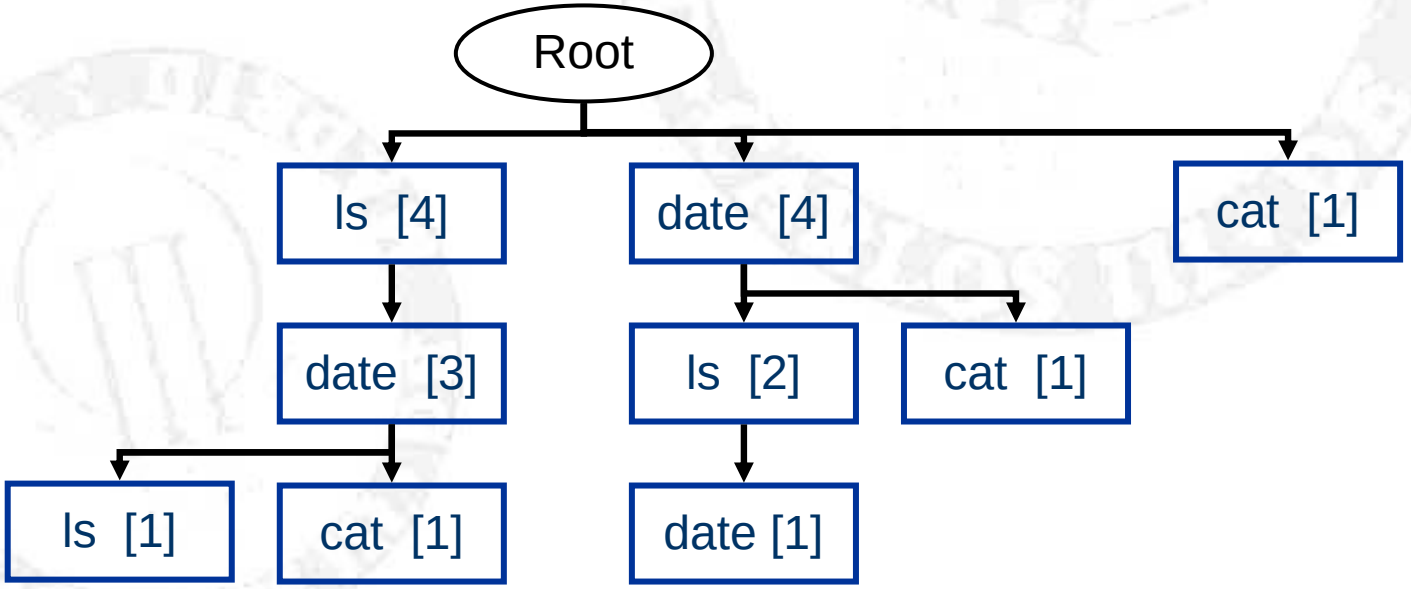
Sequence of commands → Profile Model

An example trie

Sub-sequences to insert in the trie:

{ls → date → ls} and {date → ls → date} and {ls → date → cat}

pwd
vi
pwd
vi
pwd
ls



Evaluating Dependences

Evaluate the relation/dependence between an element and its prefix

Two approaches:

- Frequency-based method.
- ➢ Statistical dependence method.

Our approach: Statistical Value used: Chi-square value.

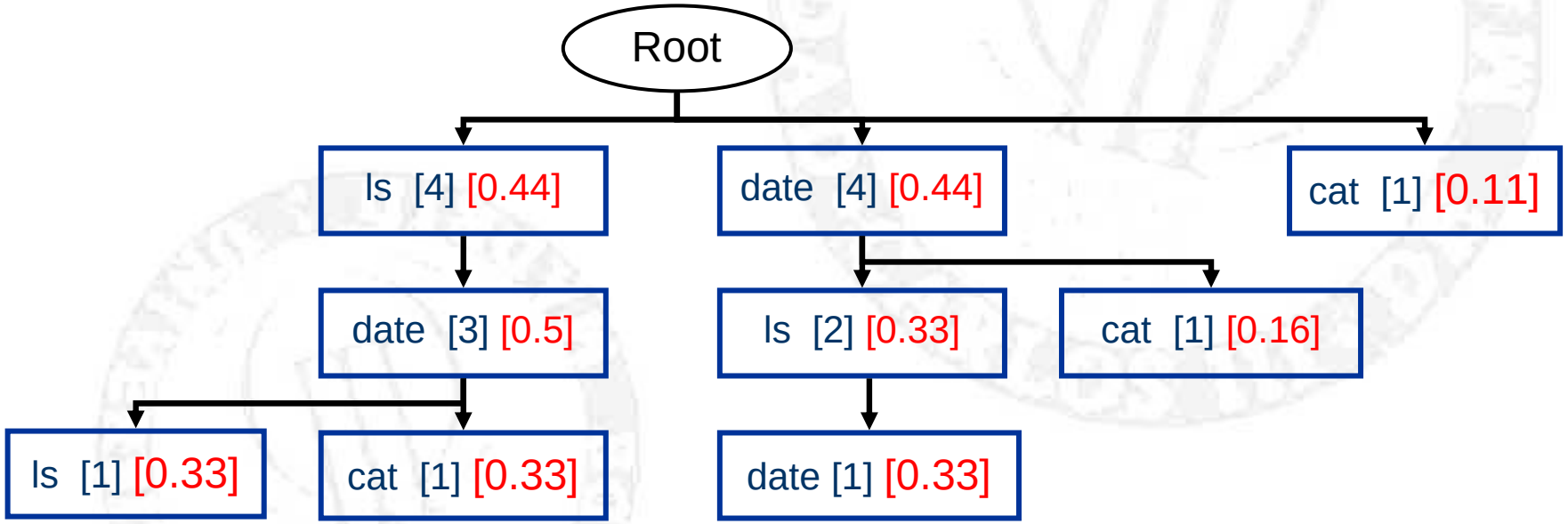
This value is stored in every node of the *trie*

Evaluating Dependences

Support or Relative Frequency (of a subseq) = $\dfrac{\text{Number of times the subsequence has been inserted into the tre.}}{\text{Total number of subsequences } \textbf{of equal size} \text{ inserted}}$

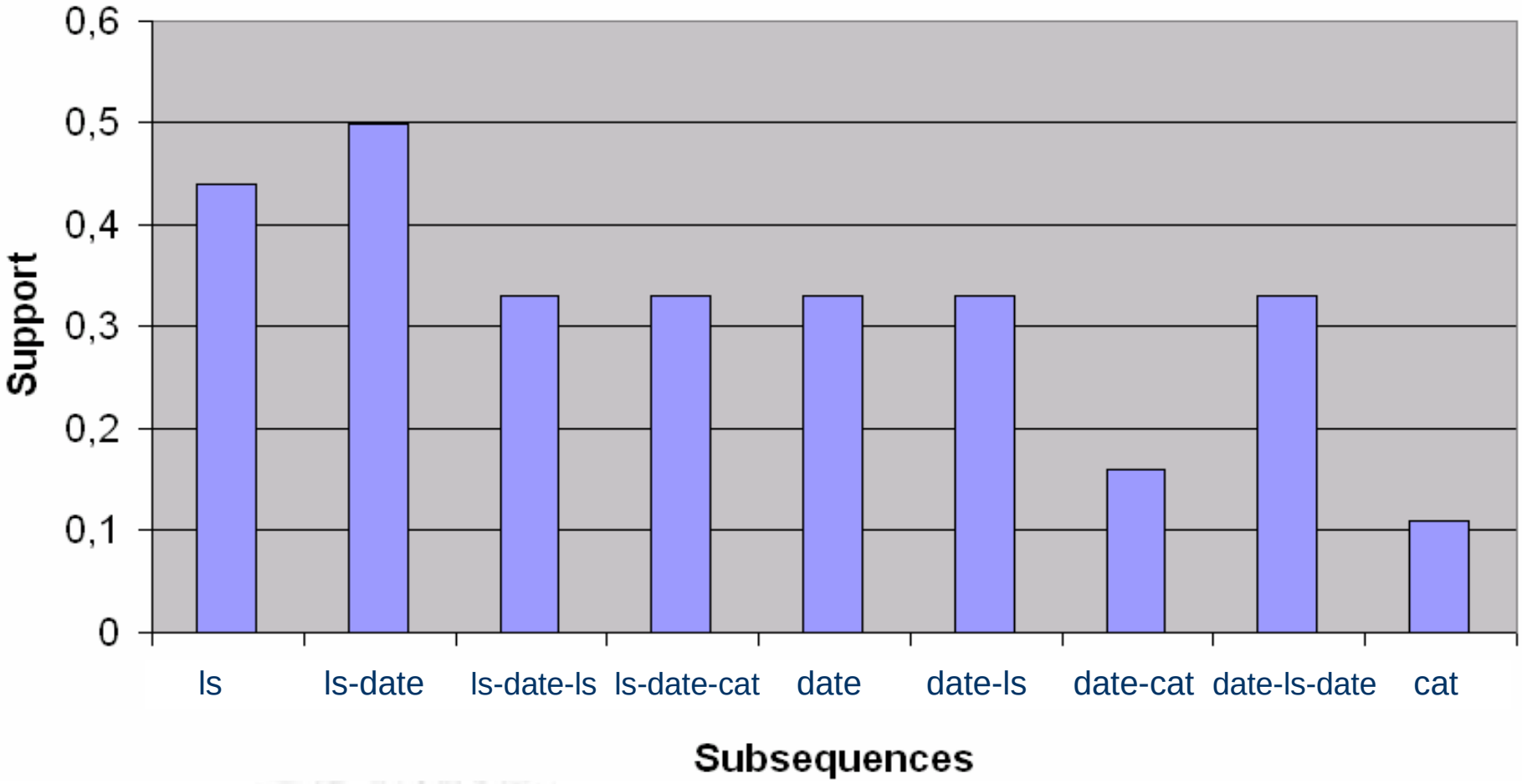
Sequence of commands → Profile Model

Evaluating Dependences



Sequence of commands → Profile Model

Behavior → Distribution of relevant events.



Outline

- From RoboCup …
- … to UNIX User Classification
- Creating a Profile Model
- Evolving classification
- Results
- Conclusions

Classification…

Profile a user is a difficult task for different aspects:

1. **Human behavior is usually erratic**
2. **Humans behave differently because of a change in their goals.**

Classification…

How to accurately profile a user while his/her behavior changes constantly ???

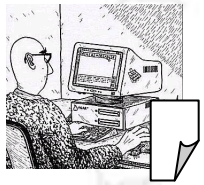
Using Evolving Fuzzy Systems !!

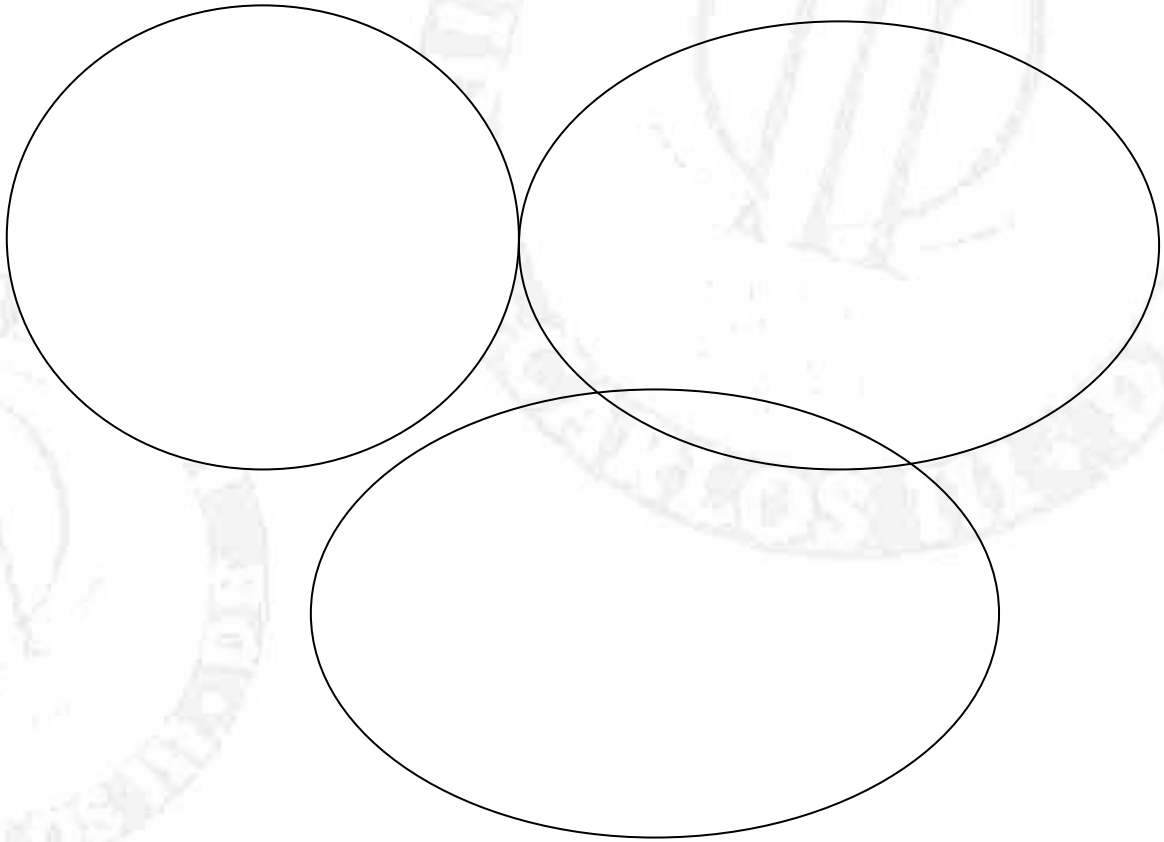
Evolving Classification…

Evolving fuzzy systems (EFS) can be defined as self-developing, self-learning fuzzy rule-based or neuro-fuzzy systems that have both their parameters but also **their structure self-adapting on-line.**

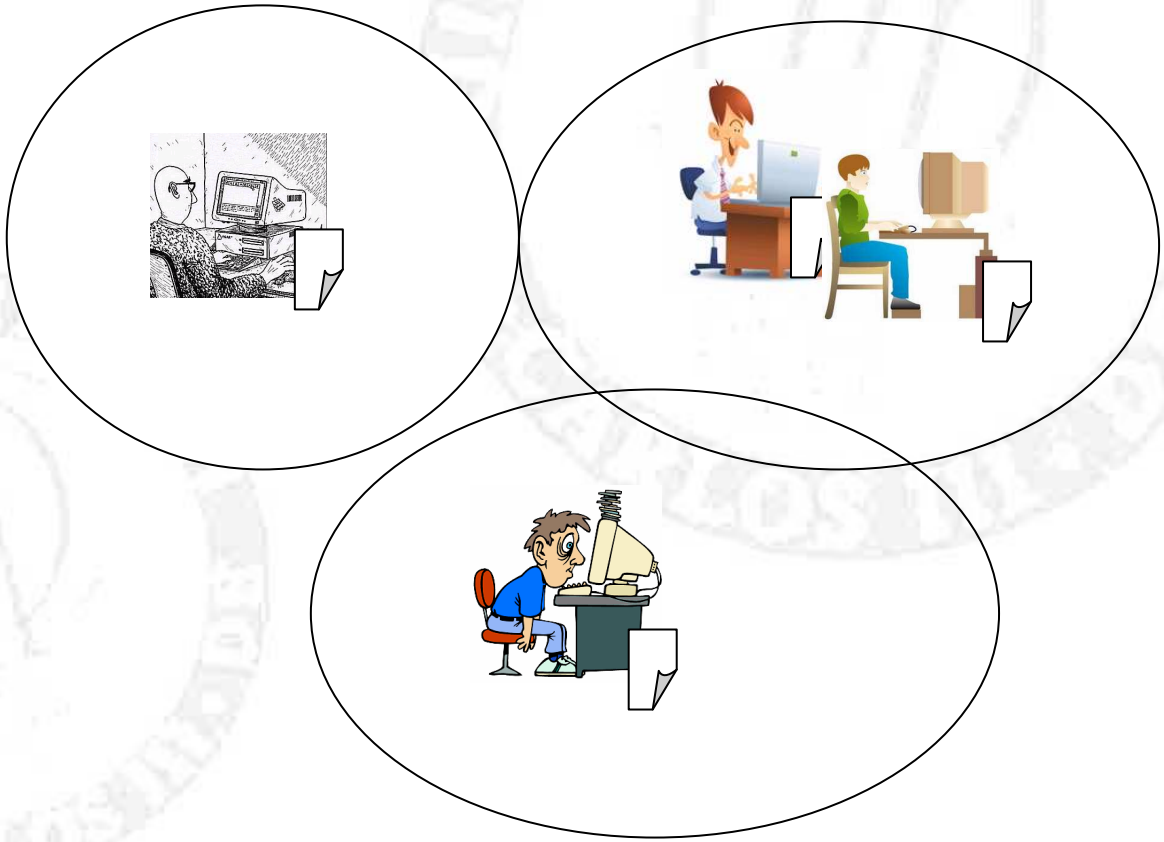
They are usually associated with streaming data and on-line (often real-time) modes of operation.

Evolving Classification of UNIX Users





Evolving Classification of UNIX Users



Architecture of the Evolving Classifier

FRB Classifier:

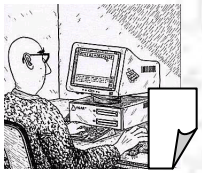
R^i = IF (x_1 is around x_1^{i*}) AND (x_2 is around x_2^{i*}) AND
… (x_n is around x_n^{i*})

THEN ($Label^i$)

$X = [x_1, x_2, \ldots x_n]$ denotes the **vector of features**.

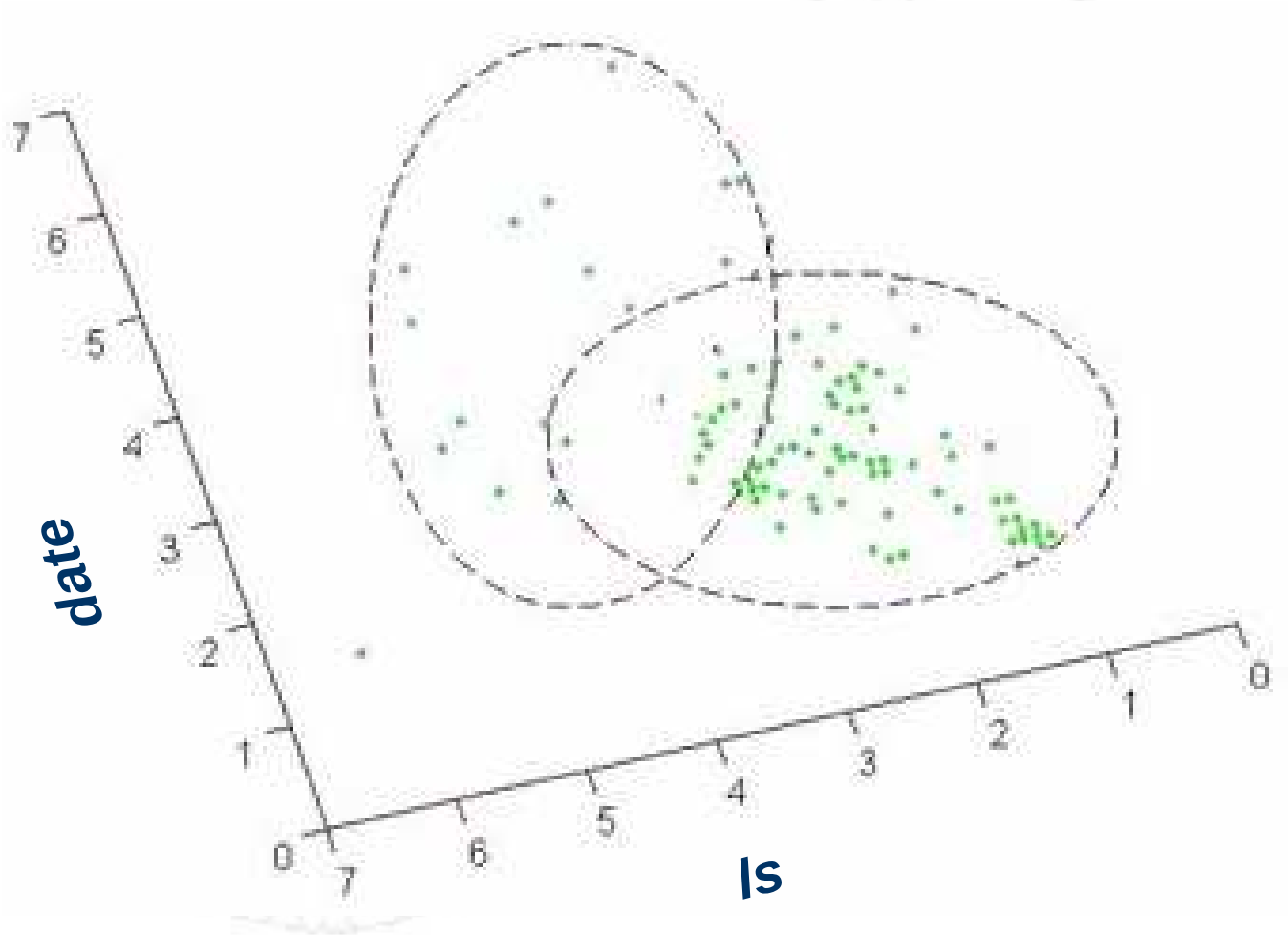
$$L = L^{i*};\ i^* = \underset{i=1}{\overset{N}{arg\ min}}(\textbf{\textit{DistCosine}})$$

Evolving Classification of UNIX Users



ls	date	ls-date	cat	vi	emacs	rm	mail	mv	asfd
0,5	0,3	0,75	0,2	0,1					
0,6	0,1	--	--	0,2	0,8	0,3			
0,3	--	--	--	0,5	--	--	0,9		
0,2	--	--	0,1	--	0.2	--	--	0.3	0.8

Architecture of the Evolving Classifier



Architecture of the Evolving Classifier

For each User…

1. **Recursive** Calculation of Potential of Data Vector (x) of a User.
2. Update Potentials of the F.P. with the new data (x)
3. If (Pot (x)) > Max Pot($F.P_i$))

   Insert New F.P.

1. Classify x in a Cluster (using Cosine Distance)

   Add x as a new "point" in the Cluster.

1. ¿Necesary Remove any F.P.?

   Calculate μ for each F.P.

   If $\mu_i > e^{-1}$ → Remove F.P.

Outline

- From RoboCup …
- … to UNIX User Classification
- Creating a Profile Model
- Evolving classification
- Results
- Conclusions

Results - Experimental Design

168 Users, 4 different user groups:

1. Novice – programmers	**55 Users**
2. Experienced-programmers	**36 Users**
3. Computer-Scientists	**52 Users**
4. Non-Programmers	**25 Users**
	168 Users

Commands per user: from 600 to 3000

Results - Experimental Design

After processing the users commands:

Different commands per group:

1. **Novice – programmers (25614)**
2. **Experienced-programmers (43049)**
3. **Computer-Scientists (66490)**
4. **Non-Programmers (10572)**

Different commands in the 4 groups: 135317

Results-Using 10Fold Cross Validation

Number of F.P. per group:

FP Nov = 2
FP Exp = 2
FP Sci = 1
FP Non = 1

Number of F.P. using Cross Validation:

Novices:	3 / 2 / 7 / 2 / 2 / 2 / 3 / 4 / 2 / 2 /
Experiences:	2 / 3 / 3 / 2 / 2 / 2 / 2 / 2 / 2 / 2 /
Sciences:	2 / 2 / 1 / 2 / 1 / 1 / 1 / 1 / 3 / 3 /
Non:	2 / 2 / 1 / 2 / 2 / 2 / 2 / 2 / 2 / 2 /

Results-Using 10Fold Cross Validation

Evolving System… Number of commands subsequences: **135317**

	nov	exp	sci	non
nov	**55**	0	0	0
exp	0	**36**	0	0
sci	0	0	**52**	0
non	0	0	0	**25**

Classification Rate:

100%

C4.5	1NN	3NN	5NN	10NN
--	--	--	--	--

Results-Using 10Fold Cross Validation

Evolving System…

Similar number of F.P.

Number of commands subsequences: **45930**

Support > 0.005

	nov	exp	sci	non
nov	**55**	0	0	0
exp	1	**35**	0	0
sci	1	1	**50**	0
non	0	1	0	**24**

Classification Rate:

97,61%

C4.5	1NN	3NN	5NN	10NN
--	--	--	--	--

Results-Using 10Fold Cross Validation

Evolving System…

Number of commands subsequences: **26108**

Support > 0.01

	nov	exp	sci	non
nov	**54**	0	0	1
exp	0	**35**	0	1
sci	1	1	**50**	0
non	0	1	0	**24**

Classification Rate:

97,02%

C4.5	1NN	3NN	5NN	10NN
--	--	--	--	--

Results-Using 10Fold Cross Validation

Evolving System…

Number of commands subsequences: **6210**

Support > 0.05

	nov	exp	sci	non
nov	**50**	4	0	1
exp	4	**31**	0	1
sci	2	12	**36**	2
non	0	3	0	**22**

Classification Rate:

82,74%

C4.5	1NN	3NN	5NN	10NN
73,80%	54,16%	50,00%	36,90%	40,47%

Results-Using 10Fold Cross Validation

Evolving System…

Number of commands subsequences: **3531**

Support > 0.1

	nov	exp	sci	non
nov	**50**	3	0	2
exp	4	**36**	0	5
sci	6	5	**35**	6
non	4	5	0	**16**

Classification Rate:

81,54%

C4.5	1NN	3NN	5NN	10NN
73,80%	44,64%	53,57%	53,57%	52,97%

Results-Using 10Fold Cross Validation

Evolving System…

Number of commands subsequences: **714**

Support > 0.5

	nov	exp	sci	non
nov	**39**	12	1	3
exp	21	**12**	0	2
sci	41	1	**8**	2
non	15	2	1	**7**

Classification Rate:

39,28%

C4.5	1NN	3NN	5NN	10NN
52,97%	40,47%	30,35%	30,95%	30,95%

Results-Using 10Fold Cross Validation

Evolving System…

Number of commands subsequences: **6210**

Support > 0.05

	nov	exp	sci	non
nov	**50**	4	0	1
exp	4	**31**	0	1
sci	2	12	**36**	2
non	0	3	0	**22**

Classification Rate:

82,74%

C4.5	1NN	3NN	5NN	10NN
73,80%	54,16%	50,00%	36,90%	40,47%

Conclusions

The proposed Evolving System Classifier…

- ***Works* with large data stream of data.**
- **Good results with large data stream of data.**
- **Fast**
- **Stores data very reduced**
- **There is no thresholds**
- **Does not requiere pre-training**
- **Starts "from scratch"**
- **….**

Thanks!