Incorporating Fuzzy Rules in LCS: Fuzzy-XCS and Fuzzy-UCS

Jorge Casillas
Dept. Computer Science and Artificial Intelligence
University of Granada
http://decsai.ugr.es/~casillas

Granada, May 15, 2008
Outline

1. **Fuzzy-XCS (for reinforcement learning)**
   1. Introduction
   2. Proposal description
   3. Experiments
   4. Conclusion

2. **Fuzzy-UCS (for supervised learning)**
   1. Introduction
   2. Proposal description
   3. Experiments
   4. Conclusion
Fuzzy-XCS: A Michigan-style Genetic Fuzzy System for Reinforcement Learning

Jorge Casillas
Dept. Computer Science and Artificial Intelligence
University of Granada
http://decsai.ugr.es/~casillas

In collaboration with
Brian Carse and Larry Bull
University of the West of England, Bristol, UK
Introduction

- Reinforcement learning with fuzzy rule-based systems
Introduction

  
  - Machine learning paradigm that uses genetic algorithms and other low-level induction operations to keep a population of classifiers which allow an Animat to optimally respond within an environment.

  - **Classifier**: contains a rule composed by the condition (state) and action parts, and some parameters for the learning process.

  - **Animat**: Animal simulated within a constrained environment, with a limited set of detectors and effectors.

- Michigan-style classifier systems are the most commonly used (1 individual = 1 rule) for reinforcement learning.
Introduction

• Why it is interesting Michigan-style LCS?
  ■ Designed to do an “on-line” learning
  ■ The system is build at the same time that stimulus are received
  ■ This allow the system to adapt itself to variable environment

• Thanks to these characteristics, Michigan-style LCS is ideal for:
  ■ Robot control
  ■ Behavior learning
  ■ Self-governing systems
  ■ Data mining
  ■ …
Introduction

- Traditionally, Michigan-style LCS was based on strength
  - Each classifier accumulates strength during its interaction with the environment through rewards and punishments
  - Strength represents an estimation of the reward that the classifier will receive from the environment
  - Informally speaking, strength is a kind of average of the received reward

- This parameter is used for two objectives:
  - Resolve conflicts among classifiers that are activated simultaneously during the learning process
  - Assess the fitness degree of each classifier for the evolutionary algorithm
Introduction

- In 1995, S.W. Wilson proposes an alternative (XCS) based on accuracy to compute the fitness
  - Those classifiers with a high capability to predict the received payoff are rewarded
  - That is, only those classifiers that obtain a mean reward more or less constant survive
  - Informally speaking, accuracy is a kind of standard deviation of the received reward
  - This shows some advantages: overgeneralization is avoided, classifiers with an optimal generalization are obtained, and the complete cover map is learnt

- Currently, XCS-based LCS is the subject of extensive ongoing research
Introduction

- **Michigan-style genetic fuzzy system**: it is a LCS composed by fuzzy rules

- There are not many proposals of fuzzy LCS. A list almost exhaustive is the following:
  1. [M. Valenzuela-Rendón, 1991]
  2. [A. Parodi, P. Bonelli, 1993]
  3. [T. Furuhashi, K. Nakaoka, Y. Uchikawa, 1994]
  4. [K. Nakaoka, T. Furuhashi, Y. Uchikawa, 1994]
  5. [A. Bonarini, 1996]
  6. [M. Valenzuela-Rendón, 1998]
  7. [J. R. Velasco, 1998]
  8. [H. Ishibuchi, T. Nakashima, T. Murata, 1999]
  9. [A. Bonarini, V. Trianni, 2001]
  11. [M. C. Su, et al., 2005]
  12. [C.-F. Juang, 2005]

All except [10] are based on strength. In [10], the output is discrete and generality is not considered.
Introduction

- An accuracy-based Fuzzy Classifier System has the following difficulties:
  - Since several rules fire in parallel and the output is due to a combination of them, credit assignment is much more difficult.
  - The payoff a fuzzy rule receives depends on the input vector, an active fuzzy rule will receive different payoffs for different inputs.
  - Measuring the accuracy of a rule's predicted payoff is difficult since a fuzzy rule will fire with many different other fuzzy rules at different time-steps, giving very different payoffs.
Proposal

- Fuzzy-XCS: an accuracy-based fuzzy classifier system

- This kind of system would have some important advantages:
  - Compared with LCS:
    - Fuzzy rules are a natural framework for real-valued input and output environments. This is very usual in robotics, for instance
  - Compared with the rest of fuzzy LCS:
    - The accuracy-based scheme allows to obtain rules with optimal generalization. This generalization generates compact rule sets, properly face the course of dimensionality, quick inference, better interpretability, …
    - Overgeneralized classifiers are avoided
    - It can maintain both consistently correct and consistently incorrect classifiers which allows learning of a complete covering map
Proposal
Continuous output

Discrete output

Continuous output
Proposal
Optimal generality

If there is a region with equal or similar reward, it is better to obtain a single rule that describes the complete area.

If $X_1$ is $A$ and $X_2$ is $D \Rightarrow Y$ is $F$
If $X_1$ is $A$ and $X_2$ is $E \Rightarrow Y$ is $F$
If $X_1$ is $B$ and $X_2$ is $D \Rightarrow Y$ is $F$
If $X_1$ is $B$ and $X_2$ is $E \Rightarrow Y$ is $F$

Rule with optimal generality
If $X_1$ is $\{A,B\}$ and $X_2$ is $\{D,E\} \Rightarrow Y$ is $F$

Over-general rule
If $X_1$ is $\{A,B,C\}$ and $X_2$ is $\{D,E\} \Rightarrow Y$ is $F$

Graphical representation of the reward function

If there is a region with equal or similar reward, it is better to obtain a single rule that describes the complete area.
Proposal
Complete covering map

Regions with low reward tend to disappear in favor of regions with high reward

If the **complete covering map** is generated, the system also obtains rules in areas with low reward
(3.15, 1.8)

**Environment**

**Detectors**

- Population
  - Matching

**Environment**

- Environment
  - Effectors
  - Reward

**Action Set [A]**

- (S,M)⇒L: 43, 0.01, 99, 2
- (L,M)⇒S: 32, 0.13, 9, 15
- (S,L)⇒M: 14, 0.05, 52, 10
- (L,L)⇒L: 27, 0.24, 3, 1
- (SM,S)⇒M: 18, 0.02, 92, 5
- (*,M)⇒L: 24, 0.17, 15, 23

**Parameter updates:** error (ε), prediction (p), fitness (F), and exp

**Candidate Subsets [CS]**

- (S,M)⇒L: 43, 0.01, 99, 2
- (SM,S)⇒M: 18, 0.02, 92, 5
- (*,M)⇒M: 24, 0.17, 15, 23

**Match Set [M]**

- (S,M)⇒L: 43, 0.01, 99, 2
- (SM,S)⇒M: 18, 0.02, 92, 5
- (*,M)⇒M: 24, 0.17, 15, 23

**Match Set [M]**

- (S,M)⇒L: 43, 0.01, 99, 2
- (SM,S)⇒M: 18, 0.02, 92, 5
- (*,M)⇒M: 24, 0.17, 15, 23

**Environment**

- 8.6

**Effectors**

- Fuzzy inference

**EA = selection + crossover + mutation**

- Apply EA? yes

**Previous Action Set**

- Exploration? yes

**Credit distribution**

**Parameter updates:** error (ε), prediction (p), fitness (F), and exp

**Apply EA?** yes
Experiments Results in a Laboratory Problem

- First experiments on a laboratory problem
- We have developed a laboratory problem to play a similar role as the multiplexer problem for discrete-valued classifier systems
- The problem is built by generating a data set from a fuzzy system previously fixed
- The objective is to obtain the set of rules that best approximate the data with the highest degree of generalization, i.e., a rule base as accurate and compact as possible
- The reward depends inversely on the difference between the inferred and the desired output in a non-linear way

\[
f(x, y) = \frac{1 - \rho}{1 + \frac{y - F(x)}{\rho}}, \quad \rho = 0.3
\]
Experiments Results in a Laboratory Problem

- **Problem:**
  - 2 inputs and 1 output
  - 5 linguistic terms for each variable (triangular-shape fuzzy sets)
  - 5 fuzzy rules of different generality degree
  - 576 examples uniformly distributed in the input space (24 x 24)
  - The output value for each input is the result of the inference with the fixed fuzzy system
Experiments Results in a Laboratory Problem

Fuzzy-XCS

![Graph showing the results of Fuzzy-XCS experiments. The graph plots average error and relative numerosity over trials.]
Experiments Results in a Laboratory Problem

Valenzuela-Rendón

Bonarini
# Experiments Results in a Laboratory Problem

<table>
<thead>
<tr>
<th></th>
<th>Fuzzy-XCS</th>
<th>Valenzuela-Rendón</th>
<th>Bonarini</th>
<th>Pittsburgh GFS</th>
</tr>
</thead>
<tbody>
<tr>
<td>R₁</td>
<td>0.7</td>
<td>0.0</td>
<td>0.0</td>
<td>0.1</td>
</tr>
<tr>
<td>R₂</td>
<td>0.9</td>
<td>0.0</td>
<td>0.0</td>
<td>0.2</td>
</tr>
<tr>
<td>R₃</td>
<td>0.8</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>R₄</td>
<td>0.7</td>
<td>0.0</td>
<td>0.0</td>
<td>0.2</td>
</tr>
<tr>
<td>R₅</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.1</td>
</tr>
<tr>
<td>no. suboptimal rules</td>
<td>0.1</td>
<td>0.0</td>
<td>4.8</td>
<td>7.6</td>
</tr>
<tr>
<td>no. non-suboptimal rules</td>
<td>1.3</td>
<td>1.0</td>
<td>20.2</td>
<td>2.0</td>
</tr>
<tr>
<td>MSE</td>
<td>0.001614</td>
<td>0.144631</td>
<td>0.052516</td>
<td>0.001892</td>
</tr>
<tr>
<td>no. analyzed examples</td>
<td>25,000</td>
<td>60,000</td>
<td>60,000</td>
<td>4,212,864</td>
</tr>
</tbody>
</table>
Experiments Results in the Mobile Robot Problem

- Real-world problem
- **On-line learning** of the wall-following behavior for a mobile robot (Nomad 200 model)

- **Inputs**: right relative distance, distance coefficient, orientation, and linear velocity
- Variables are computed **exclusively from sensorial information** of the robot, which is more realistic

- **Reward**:

\[
R(DD, VL, \theta_{pared}) = 1 - \left( \alpha_1 \frac{|DD - 1|}{3} + \alpha_2 |VL - 1| + \alpha_3 \frac{\theta_{pared}}{45} \right)
\]
Experiments Results in the Mobile Robot Problem
4. Resultados Experimentales

4.2. Problema de Simulación de Robot Realista
Conclusion

- A fuzzy classifier system for real-valued output that properly generates the complete covering map in reinforcement problems is proposed.

- It is the first algorithm with such characteristics (at least as far as we known).

- Future work involves investigating the behavior of the proposal in multi-step and real-world problems.
Fuzzy-UCS: A Michigan-style Learning Fuzzy-Classifier System for Supervised Learning

Jorge Casillas
Dept. Computer Science and Artificial Intelligence
University of Granada, Spain
http://decsai.ugr.es/~casillas

In collaboration with
Albert Orriols-Puig and Ester Bernadó-Mansilla
Ramon Llull University, Barcelona, Spain
Motivation

- **Michigan-style LCSs for supervised learning**
  Eg. XCS and UCS
  - Evolve online highly accurate models
  - Competitive to the most-used machine learning techniques
    - [Bernadó et al, 2002; Wilson, 2002; Bacardit & Butz, 2004; Butz, 2006; Orriols & Bernadó, 2007]

- **Main weakness: Interpretability of the rule sets**
  - Continuous attributes represented with intervals: \([l_i, u_i]\). Semantic-free variables
  - Number of rules or classifiers
    - Tackled with *reduction schemes*
      - [Wilson, 2002; Fu & Davis, 2002; Dixon et al., 2003]
Fuzzy-UCS’s Aim

- **Accuracy-based** Michigan-style LFCS
- **Supervised learning** scheme
- Derived from **UCS** [Bernadó & Garrell, 2003]
  - Introduction of a linguistic fuzzy representation
  - Modification of all operators that deal with linguistic rules
- **We expect:**
  - Achieve similar performance than UCS
  - Higher interpretability since we would deal with linguistic rules
  - Lower number of fuzzy rules in the final population
Description of Fuzzy-UCS

Instance: (3.15, 1.8) → Class: 1

Population

<table>
<thead>
<tr>
<th>cond.</th>
<th>class</th>
<th>F. num</th>
</tr>
</thead>
<tbody>
<tr>
<td>(M,M)</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>(S,M)</td>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>(M,S)</td>
<td>0</td>
<td>.8</td>
</tr>
<tr>
<td>(S,L)</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>(ML,M)</td>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>(*.M)</td>
<td>1</td>
<td>.9</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Matching

Stream of instances

Attribute 1

<table>
<thead>
<tr>
<th>μ_{(3.15)}&gt;0</th>
<th>μ_{L(3.15)}&gt;0</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>L</td>
</tr>
</tbody>
</table>

Attribute 2

<table>
<thead>
<tr>
<th>μ_{(1.8)}&gt;0</th>
<th>μ_{R(1.8)}&gt;0</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>M</td>
</tr>
</tbody>
</table>

(ML, SM) → (ML, M)

Match Set [M]

<table>
<thead>
<tr>
<th></th>
<th>(M,M)</th>
<th>0</th>
<th>1</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>(M,S)</td>
<td>0</td>
<td>.8</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>(ML,M)</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>(*.M)</td>
<td>1</td>
<td>.9</td>
<td>23</td>
<td></td>
</tr>
</tbody>
</table>

Correct Set [C]

<table>
<thead>
<tr>
<th></th>
<th>(ML,M)</th>
<th>1</th>
<th>1</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>(*.M)</td>
<td>1</td>
<td>.9</td>
<td>23</td>
<td></td>
</tr>
</tbody>
</table>

Reasoning = Infer class

Parameter’s Update

Apply EA?

GA = selection + crossover + mutation
Description of Fuzzy-UCS

- **Rule representation**
  - Linguistic fuzzy rules
  - E.g.:
    
    \[
    \text{IF } x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } A_2 \ldots \text{ and } x_n \text{ is } A_n \text{ THEN class}_1
    \]

  - All variables share the same semantics
  - **Example**: \( A_i = \{\text{small, medium, large}\} \)

    \[
    \text{IF } x_1 \text{ is small and } x_2 \text{ is medium or large THEN class with weight}
    \]

  - **Codification**:
    
    \[
    \text{IF } [100 | 011] \text{ THEN class}_1
    \]
Description of Fuzzy-UCS

- Each classifier has the following parameters:

1. Weight per class \( w_j \):
   - Soundness with which the rule predicts the class \( j \).
     \[
     \forall j : w_j^k = \frac{cm_{j+1}^k}{exp_{t+1}^k}
     \]
   - The class value is dynamic and corresponds to the class \( j \) with higher \( w_j \)

2. Fitness:
   - Quality of the rule
     \[
     F_{t+1}^k = w_{\text{max}_{t+1}}^k - \sum_{j|j \neq \text{max}} w_j^k
     \]

3. Other parameters:
   - numerosity
   - correct set size
   - experience
Description of Fuzzy-UCS

- **Class inference of a test example \( e \)**
  - **Weighted average inference**
    - All experienced rules vote for the class they advocate:
      \[
      v_k = \mu_{A^k}(e) \cdot F^k
      \]
    - The votes for each class \( j \) are added:
      \[
      \forall j : \text{vote}_j = \sum_{k \mid c^k = j} v_k
      \]
      - The most-voted class is returned as output
  - **Action winner inference**
    - Select the experienced rule \( k \) that maximizes \( \mu_{A^k}(e) \cdot F^k \)
    - Choose the class of such a rule as output
Description of Fuzzy-UCS

- Rule set reduction
  - Reduction based on weighted average (wavg)
    - Remove all the rules that a) are not experienced enough or b) have zero or negative fitness. Weighted average inference is used.
  - Reduction based on action winner (awin)
    - Only the rules that maximizes the vote $v_j$ for at least one example are kept. Action winner inference is used.
  - Reduction based on the most numerous and fitted rules (nfit)
    - For each training example, the rule that maximizes the vote weighted by its numerosity (number of copies) is kept:
      $$F^k \cdot \mu_{A^k}(e) \cdot \text{num}^k$$
    - Weighted average inference is used.
Experimental Methodology

- Compare Fuzzy-UCS to **14 methods** (from authors’ codes as well as KEEL and Weka software):
  - 6 fuzzy GBMLs (GP, GAP, SAP, AdaBoost, LogitBoost, MaxLogitBoost)
  - 2 state-of-the-art interval-rule-based GBMLs (GAssist, UCS)
  - 6 classical classifiers (C4.5, IBk, Part, NaiveBayes, SMOp3, SMOrbf)

- **21 public data sets** (examples with missing values are kept)

- **10-fold cross-validation**, averages over 10 runs per partition

- 5 linguistic terms in fuzzy approaches

- **Non-parametric statistical tests** [Demšar, 2006]: Friedman, Bonferroni-Dunn, and Wilcoxon
## Fuzzy-UCS vs. Non Fuzzy Approaches

### Performance (test success rate)

<table>
<thead>
<tr>
<th>Pos.</th>
<th>Algorithm</th>
<th>Ranking</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SMOp3</td>
<td>4.28</td>
</tr>
<tr>
<td>2</td>
<td>UCS</td>
<td>4.65</td>
</tr>
<tr>
<td>3</td>
<td>Fuzzy-UCS (wavg)</td>
<td>4.68</td>
</tr>
<tr>
<td>4</td>
<td>IB5</td>
<td>5.28</td>
</tr>
<tr>
<td>5.5</td>
<td>C4.5</td>
<td>5.95</td>
</tr>
<tr>
<td>5.5</td>
<td>Part</td>
<td>5.95</td>
</tr>
<tr>
<td>7</td>
<td>GAssist</td>
<td>6.25</td>
</tr>
<tr>
<td>8</td>
<td>Fuzzy-UCS (awin)</td>
<td>6.50</td>
</tr>
<tr>
<td>9</td>
<td>Naïve Bayes</td>
<td>6.63</td>
</tr>
<tr>
<td>10</td>
<td>Fuzzy-UCS (nfit)</td>
<td>7.40</td>
</tr>
<tr>
<td>11</td>
<td>SMOrbf</td>
<td>8.95</td>
</tr>
<tr>
<td>12</td>
<td>ZeroR</td>
<td>11.50</td>
</tr>
</tbody>
</table>

### Wilcoxon signed-ranks test

![Diagram showing Wilcoxon signed-ranks test](attachment:image.png)
## Fuzzy-UCS vs. Non Fuzzy Approaches

### Interpretability

#### Size of the learnt models

<table>
<thead>
<tr>
<th></th>
<th>C4.5</th>
<th>Part</th>
<th>GAssist</th>
<th>UCS</th>
<th>Fuzzy-UCS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>wavg</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>awin</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>nfit</td>
</tr>
<tr>
<td>ann</td>
<td>38</td>
<td>15</td>
<td>5</td>
<td>4494</td>
<td>2769</td>
</tr>
<tr>
<td>aut</td>
<td>44</td>
<td>21</td>
<td>7</td>
<td>4565</td>
<td>3872</td>
</tr>
<tr>
<td>bal</td>
<td>45</td>
<td>37</td>
<td>8</td>
<td>2177</td>
<td>1212</td>
</tr>
<tr>
<td>bpa</td>
<td>25</td>
<td>9</td>
<td>6</td>
<td>2961</td>
<td>1440</td>
</tr>
<tr>
<td>cmc</td>
<td>162</td>
<td>168</td>
<td>15</td>
<td>3634</td>
<td>1881</td>
</tr>
<tr>
<td>col</td>
<td>5</td>
<td>9</td>
<td>5</td>
<td>3486</td>
<td>4135</td>
</tr>
<tr>
<td>gts</td>
<td>24</td>
<td>15</td>
<td>5</td>
<td>3359</td>
<td>2799</td>
</tr>
<tr>
<td>h-c</td>
<td>29</td>
<td>21</td>
<td>6</td>
<td>2977</td>
<td>3574</td>
</tr>
<tr>
<td>h-s</td>
<td>17</td>
<td>18</td>
<td>5</td>
<td>3735</td>
<td>3415</td>
</tr>
<tr>
<td>irs</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>1039</td>
<td>480</td>
</tr>
<tr>
<td>pim</td>
<td>19</td>
<td>7</td>
<td>7</td>
<td>3605</td>
<td>2841</td>
</tr>
<tr>
<td>son</td>
<td>14</td>
<td>8</td>
<td>5</td>
<td>520</td>
<td>3042</td>
</tr>
<tr>
<td>tao</td>
<td>36</td>
<td>17</td>
<td>6</td>
<td>807</td>
<td>111</td>
</tr>
<tr>
<td>thy</td>
<td>8</td>
<td>4</td>
<td>4</td>
<td>1994</td>
<td>1283</td>
</tr>
<tr>
<td>veh</td>
<td>69</td>
<td>32</td>
<td>7</td>
<td>4941</td>
<td>3732</td>
</tr>
<tr>
<td>wbc</td>
<td>12</td>
<td>10</td>
<td>3</td>
<td>2334</td>
<td>3130</td>
</tr>
<tr>
<td>wdbc</td>
<td>11</td>
<td>7</td>
<td>4</td>
<td>5206</td>
<td>5412</td>
</tr>
<tr>
<td>wne</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>3685</td>
<td>3686</td>
</tr>
<tr>
<td>wpbc</td>
<td>12</td>
<td>7</td>
<td>4</td>
<td>5299</td>
<td>3772</td>
</tr>
<tr>
<td>zoo</td>
<td>11</td>
<td>8</td>
<td>6</td>
<td>1291</td>
<td>773</td>
</tr>
</tbody>
</table>

The table above compares the size of models learnt using C4.5, GA, and UCS approaches, with the additional size metrics for the Fuzzy-UCS method.
Fuzzy-UCS vs. Non Fuzzy Approaches

Interpretability

**SMO**

```
1.000  * <0.229 0.875 > * X]
0.298  * <0.708 0.437 > * X]
...```

**C4.5**

```
x <= -2.75
| x <= -3.25: red (308.0)
| x > -3.25
|   y <= 1.75: red (55.0)
|   y > 1.75
|   x <= -3: red (11.0/1.0)
|   x > -3
|   y <= 4.25: blue (6.0)
|   y > 4.25: red (4.0)
...```

**Part**

```
if x <= -3.25 then red (308)
else if x > 2.75 then blue (347/1)
else if y <= 0 and x >= -1 then red (192/1)
...```
Fuzzy-UCS vs. Non Fuzzy Approaches
Interpretability

GAssist

if \( x > 2.72 \) and (\( y \) is \([0.92, 4.61]\) or \( y > 5.07 \)) then blue

else if (\( x \) is \([-0.54, 0.54]\) or \( x > 2.72 \)) and \( y \) is \([-4.28, -2.57]\) then blue

otherwise red

UCS

if \( x \) is \([-6.00, -0.81]\) and \( y \) is \([-6.00, 0.40]\) then red with acc = 1.00

if \( x \) is \([2.84, 6.00]\) and \( y \) is \([-5.26, 4.91]\) then blue with acc = 1.00

if \( x \) is \([-6.00, -0.87]\) and \( y \) is \([-6.00, 0.74]\) then red with acc = 1.00

Fuzzy-UCS

if \( x \) is XL then blue with \( F=1.00\)

if \( x \) is XS then red with \( F=1.00\)

if \( x \) is \{XS or S\} and \( y \) is \{XS or S\} then red with \( F=0.87\)

...
Experiments with a Large Data Set

- Fuzzy-UCS learns from a stream of examples.
- The learning can be stalled whenever required. The more learning iterations the system has performed, the more general and accurate the rules.
- The cost of the algorithm’s runtime increases linearly with the maximum population size, the number of variables per rule, and the number of learning iterations, but it does not depend directly on the number of examples.
- We exploit the benefits of online learning to mine large data sets.

<table>
<thead>
<tr>
<th></th>
<th>#Inst</th>
<th>#Fea</th>
<th>#Re</th>
<th>#No</th>
<th>#Cl</th>
<th>Disp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kdd'99</td>
<td>494,022</td>
<td>41</td>
<td>35</td>
<td>6</td>
<td>23</td>
<td>8.3 \cdot 10^{-5}</td>
</tr>
</tbody>
</table>
Experiments with a Large Data Set

<table>
<thead>
<tr>
<th>#Iter</th>
<th>wavg perf.</th>
<th>wavg #rules</th>
<th>awin perf.</th>
<th>awin #rules</th>
<th>nfit perf.</th>
<th>nfit #rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>500,000</td>
<td>99.32</td>
<td>1944</td>
<td>99.13</td>
<td>541</td>
<td>99.27</td>
<td>417</td>
</tr>
<tr>
<td>1,000,000</td>
<td>99.36</td>
<td>2089</td>
<td>99.07</td>
<td>492</td>
<td>99.25</td>
<td>369</td>
</tr>
<tr>
<td>1,500,000</td>
<td>99.37</td>
<td>2178</td>
<td>99.02</td>
<td>460</td>
<td>99.24</td>
<td>350</td>
</tr>
<tr>
<td>2,000,000</td>
<td>99.36</td>
<td>2257</td>
<td>99.00</td>
<td>428</td>
<td>99.19</td>
<td>323</td>
</tr>
</tbody>
</table>
Conclusion: Self-analysis

- SWOT analysis:
  - **Strengths**: main advantages of Fuzzy-UCS
  - **Weaknesses**: drawbacks of Fuzzy-UCS
  - **Opportunities**: further works on Fuzzy-UCS
  - **Threats**: optional approaches considered by other methods other than can compete with Fuzzy-UCS

<table>
<thead>
<tr>
<th></th>
<th>Positive</th>
<th>Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Internal</strong></td>
<td>Strengths</td>
<td>Weaknesses</td>
</tr>
<tr>
<td><strong>External</strong></td>
<td>Opportunities</td>
<td>Threats</td>
</tr>
</tbody>
</table>
## Conclusion: Self-analysis

<table>
<thead>
<tr>
<th>Strengths</th>
<th>Weaknesses</th>
</tr>
</thead>
<tbody>
<tr>
<td>- It shows a high performance regarding error rate; comparable with the state-of-the-art in classification</td>
<td>- It generates rule sets with size moderate or big (depending on the chosen configuration)</td>
</tr>
<tr>
<td>- It uses a highly legible knowledge representation based on linguistic fuzzy rules</td>
<td>- Though it can deal with real, integer or categorical features, it is recommended to be applied only in the two former cases</td>
</tr>
<tr>
<td>- It performs incremental, on-line learning</td>
<td></td>
</tr>
<tr>
<td>- It is capable of mining large data sets</td>
<td></td>
</tr>
</tbody>
</table>
## Conclusion: Self-analysis

<table>
<thead>
<tr>
<th>Opportunities</th>
<th>Threats</th>
</tr>
</thead>
<tbody>
<tr>
<td>• It is ready to be applied in data streams; further analysis on this problem will be made</td>
<td>• Few interval rules can be more easily interpreted than many linguistic fuzzy rules</td>
</tr>
<tr>
<td>• Because of the use of fuzzy logic, the algorithm could be adapted to deal with vague and uncertain data</td>
<td>• Other learning approaches combined with preprocessing can also deal with large data sets</td>
</tr>
<tr>
<td>• The proposed seminar system opens the door to further works on fuzzy knowledge representation, fuzzy inference engine, and evolutionary operators</td>
<td></td>
</tr>
</tbody>
</table>