Classifier Systems

Introduction

- Learning
  - "A computer program is said to *learn* from experience ***E*** with respect to some class of tasks ***T*** and performan ce measure ***P***, if its performance at tasks in ***T***, as mea sured by ***P***, improves with experience ***E***"

Machine Learning & Classification

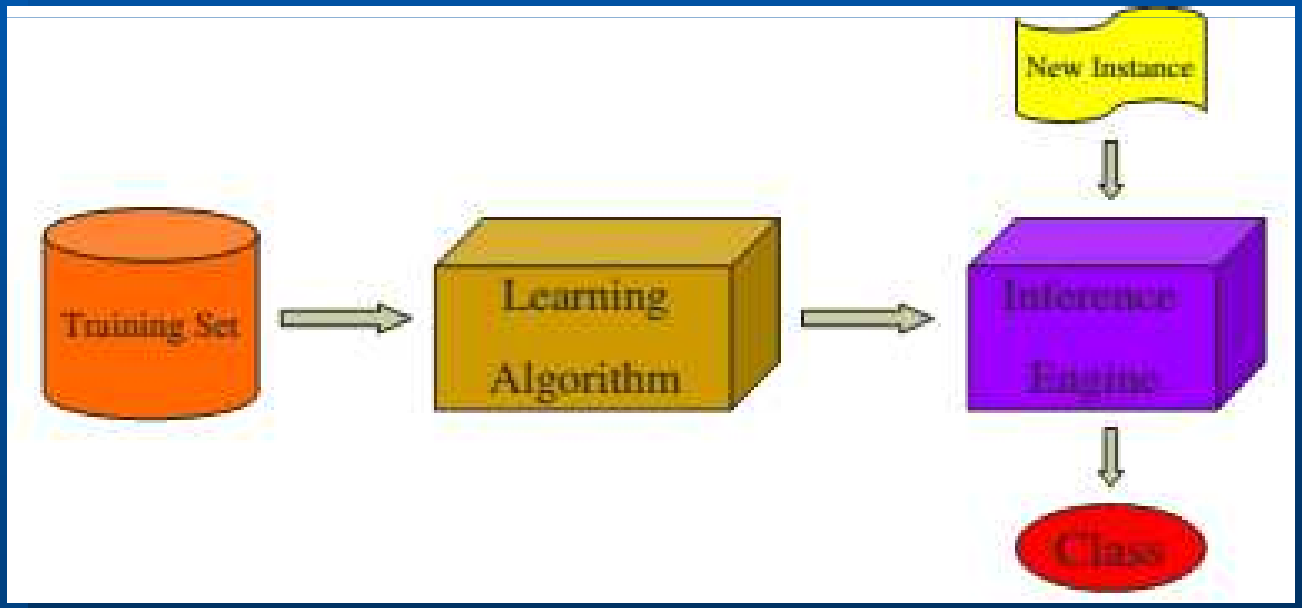
- A checkers learning problem
  - Task T: playing checkers
  - Performance measure P: percent of games won against opponents
  - Training experience E: playing practice games against itself
- A handwriting recognition learning problem
  - Task T: recognizing and classifying handwritten words using images
  - Performance measure P: percent of words correctly identified
  - Training experience E: a database of handwritten words with given classifications

Machine Learning & Classification

- A robot driving learning problem
  - Task T: driving on public four-lane highways using vision sensors
  - Performance measure P: average distance traveled before an error (as judged by human overseer)
  - Training experience E: a sequence of images and steering commands recorded while observing a human driver

Machine Learning & Classification

- Classification task: Learning how to label corr ectly new instances from a domain based on a set of previously labeled instances

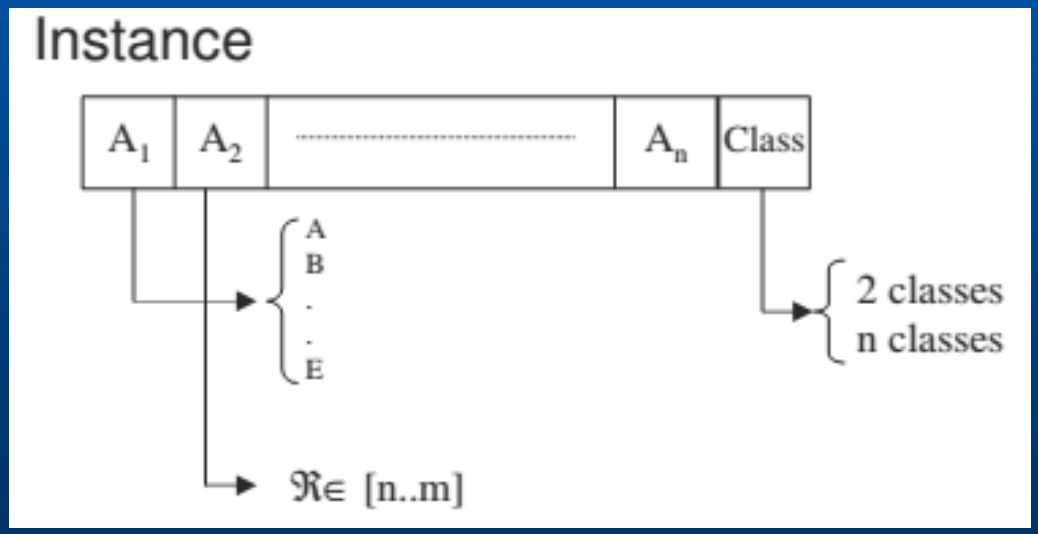


Machine Learning & Classification

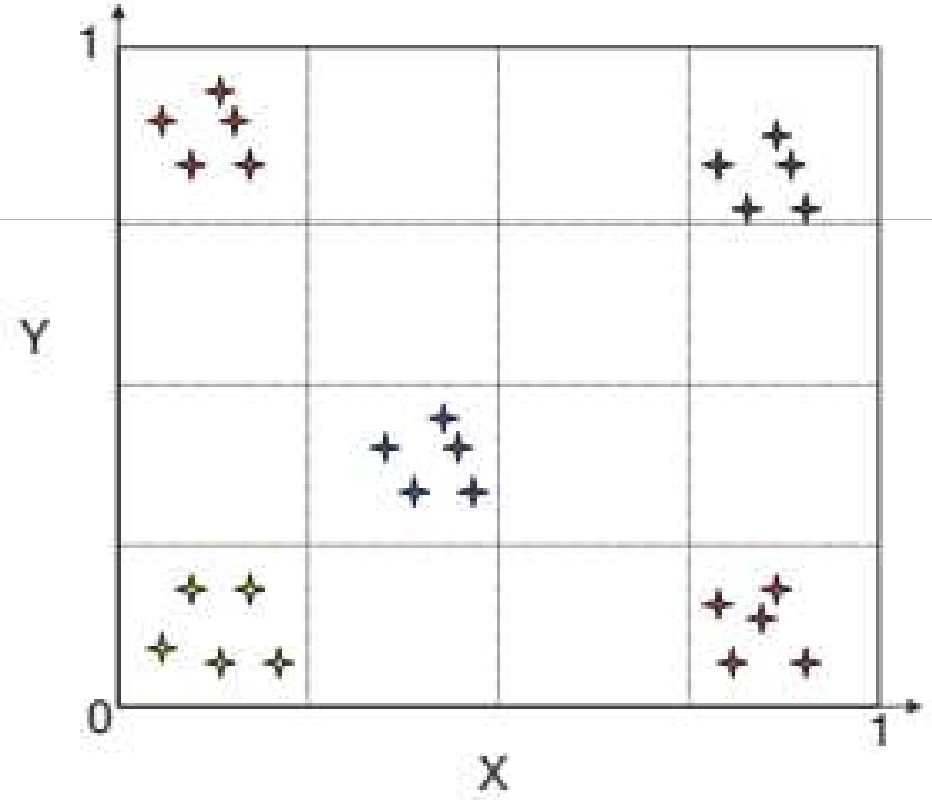
- Goal of classification is to learn how to predict the class of an instance from its attributes
- Instance: individual, independent example of the domain that has to be learned
- Instances have regular structure:
  - Fixed number of attributes: features that characterize an instance
  - A class: a label belonging to a finite and discrete domain

Machine Learning & Classification

- Attributes can be of diverse type
  - Nominal: discrete and finite variable
  - Integer, Continuous



Machine Learning & Classification



Instance: (X,Y||Colour)

1: If (X<0.25 and Y>0.75) or
(X>0.75 and Y<0.25) then → +

2: If (X>0.75 and Y>0.75) then → +

3: If (X<0.25 and Y<0.25) then → +

4: If (X ∈ [0,25,0,50] and
Y ∈ [0.25,0.50]) then → +

4': Everything else then → +

Learning Paradigms and General Aspects of Learning

- Different Forms of Learning:
  - Learning agent receives feedback with respect to its actions (e.g. from a teacher)
  - **Supervised Learning**: feedback is received with respect to all possible actions of the agent
  - **Unsupervised Learning:** Learning when there is no hint at all about the correct action

Learning Paradigms and General Aspects of Learning

- **Inductive Learning**
  - It is a form of supervised learning that centers on learning a function based on sets of training examples.
  - Popular inductive learning techniques include decision trees, neural networks, nearest neighbor approaches, discriminant analysis and regression.
- The **performance of an inductive learning system** is usually evaluated using **n-fold cross-validation.**

Classifier Systems

- According to Goldberg, a classifier system is "a machine learning system that learns syntactically simple string rules to guide its performance in an arbitrary environment".
- First implemented of a system called CS1 by Holland/ Reitman(1978).
- Learning Classifier Systems (LCS) are one of the major families of techniques that apply evolutionary computation to machine learning tasks
- Machine learning: How to construct programs that automatically learn from experience
- LCS are almost as ancient as GAs, Holland made one of the first proposals

Classifier Systems

- Example of classifier rules:

  00##:0000

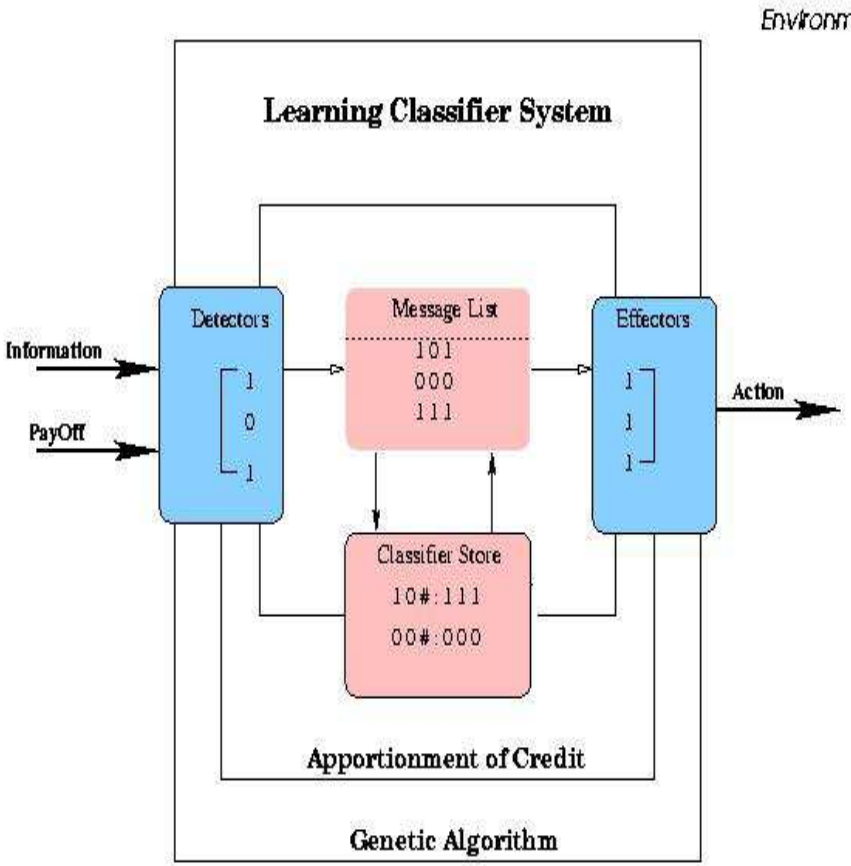
  00#0:1100

  11##:1000

  ##00:0001

- Fitness of a classifier is defined by its surrounding environments that pays payoff to classifiers and extract fees from classifiers.

Classifier System



- Learn simple string rules in an arbitrary environment
- A *classifier* is a simple string rule
- Components
  - Rule and Message System
  - Apportionment of credit system
  - Genetic Algorithm

Knowledge representations

- All the initial approaches were rule-based
- In recent years several knowledge representations have been used in the LCS
- For example:
  - Decision Trees
  - Neural Network

Generating better rules

- Use a Genetic Algorithm (GA) to generate new rules
- A classifier's strength (S) is used as its fitness
- Similar to the simple genetic algorithm
  - Entire population is not replaced at the next generation
- *Crowding* to maintain diversity
- Mutation over a *ternary* alphabet `{1, 0, #}`
- Selection is performed using roulette-wheel selection

Paradigms of LCS

- The Michigan approach
  - Holland & Reitman, 78
- The Pittsburgh approach
  - Smith, 80
- The Iterative Rule Learning approach
  - Venturini, 93

The Pittsburgh approach

- This approach is the closest one to the standard concept of GA
- Each individual is a complete solution to the classification problem
- Traditionally this means that each individual is a variable-length set of rules
- GABIL [De Jong & Spears, 93] is a well known representative of this approach

The Pittsburgh approach

- More than one rule could be used to classify a given instance
- Match process: deciding which rule is used in these cases
- An usual approach is that individuals are interpret ed as a decision list [Rivest, 87]: an ordered rule set

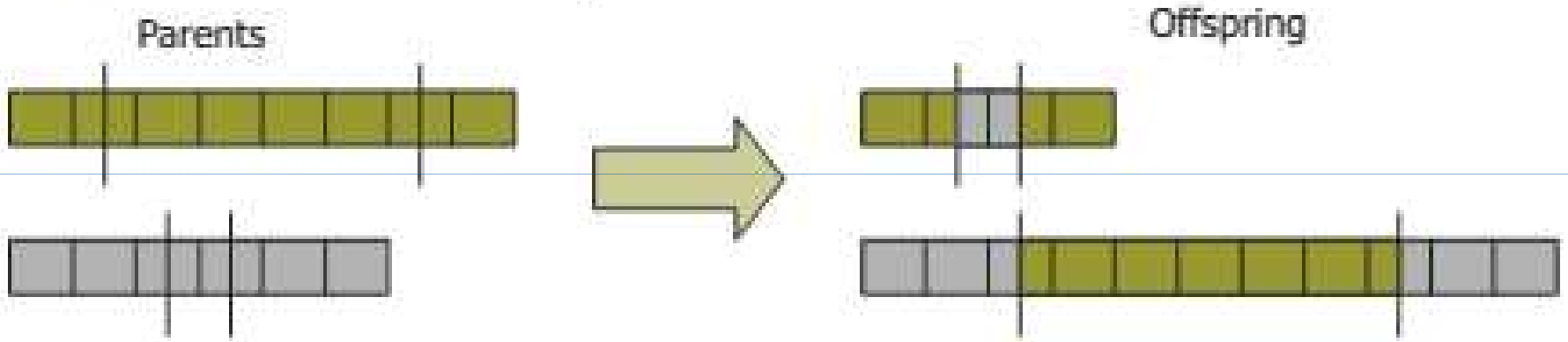
Instance 1 matches rules 2, 3 and 7 → Rule 2 will be used
Instance 2 matches rules 1 and 8 → Rule 1 will be used
Instance 3 matches rule 8 → Rule 8 will be used
Instance 4 matches no rules → Instance 4 will not be classified

Pittburgh-style Systems

- Pittsburgh approach systems usually have to cope with variable length chromosomes.
- Popular Pittsburgh-style systems include:
  - Smith's LS-1-system (learns symbolic rule-sets)
  - Janikov's GIL system
  - Giordana&Saita's REGAL (learns symbolic concept descriptions)
  - DELVAUX (learns numerical Bayesian rule-sets)

The Pittsburgh approach

- Crossover operator



- Mutation operator: classic GA mutation of bit inversion

Paradigms of LCS

- In the other two approaches each individual is a rul e
- What happens usually in the evolutionary process of a GA?
  - All individuals converge towards a single solution
- Our solution is a set of rules. Therefore we need some mechanism to guarantee that we generate all of them.
- Each approach uses a different method for that

The Michigan approach

- Each individual (classifier) is a single rule
- The whole population cooperates to solve the classification problem
- A reinforcement learning system is used to identify the good rules
- A GA is used to explore the search space for more r ules
- XCS [Wilson, 95] is the most well-known Michigan LCS

The Michigan approach

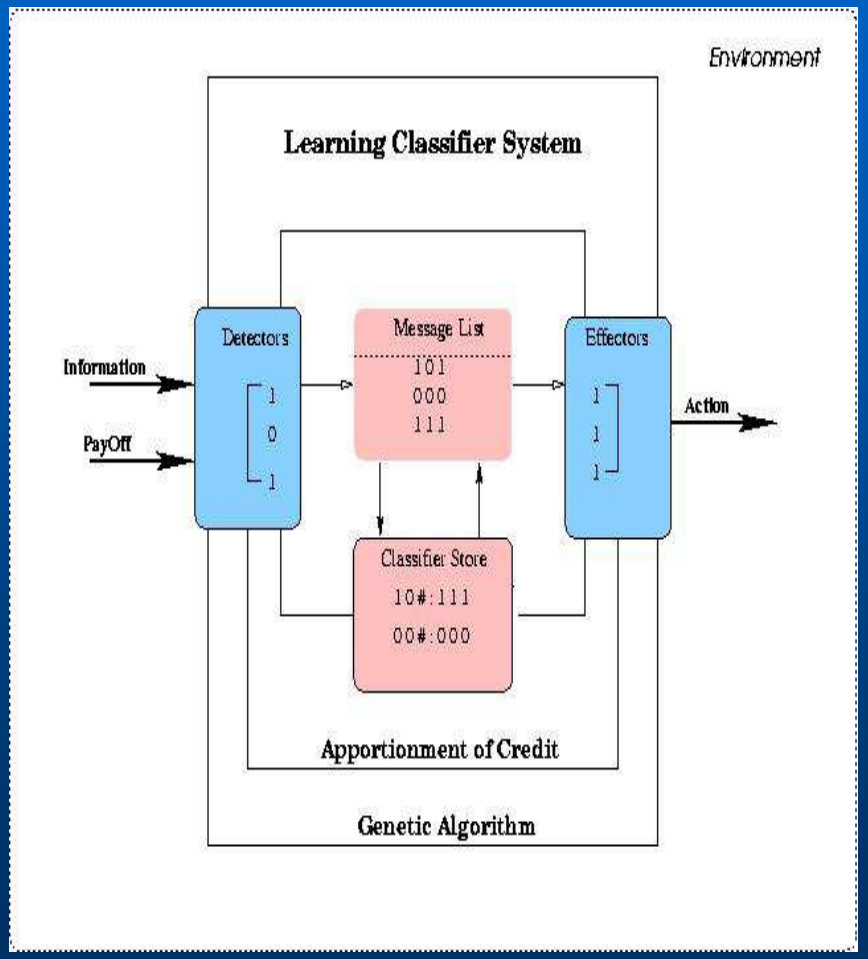
- What is Reinforcement Learning?
  - "a way of programming agents by reward and punishment without needing to specify how the task is to be achieved".
  - Rules will be evaluated example by example, receiving a positive/negative reward
  - Rule fitness will be updated incrementally with this reward
  - After enough trials, good rules should have high fitness

The Iterative Rule Learning approach

- Each individual is a single rule
- Individuals compete as in a standard GA
  - A single GA run generates one rule
- Instances already covered by previous rules are removed from the training set of the next iteration
- The GA is run iteratively to learn all rules that solve the problem
- Also known as separate-and-conquer

Summary

- A *classifier* is a simple string rule
- Classifier System
  - rule-message system,
  - apportionment of credit mechanism
  - GA
- Advantages of CS
  - rules are simple
  - use fixed length representation
  - parallel activation



Summary

- The Learning Classifier Systems means EC techniques applied to Machine Learning
- Description of the three main paradigms
  - Pittsburgh
  - Michigan
  - Iterative rule learning

Summary

- Description of several knowledge representations
  - Rule based
    - Nominal attributes
    - Continuous attributes
  - Decision trees

Summary

- Applications to real-world domains
  - Medical
  - Industrial
  - Military
- Recent trends
  - Explore better
  - Model the problem better
  - Understand better