POLITECNICO DI MILANO

Learning Classifier Systems
A Gentle Introduction

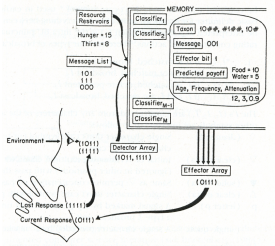
Outline

- Introduction
  - Why? When?
  - What areas? What Applications?
- Learning Classifier Systems
  - What Learning?
  - Classifiers?
  - How do they work?
  - What decisions?
  - General principles
  - Better classifiers
  - Theory?
- Survey of applications

Why?

Why? What was the goal?

a real system with an unknown underlying dynamics



apply a classifier system online to generate a behavior matched the real system

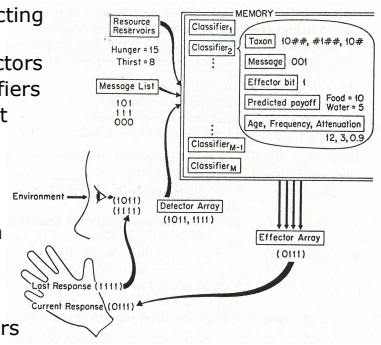
if C_1 :buy ⇒30
if C_2 :sell ⇒-2
…

evolved rules provide a plausible, human readable, model of the unknown system

Holland's Vision, Cognitive System One

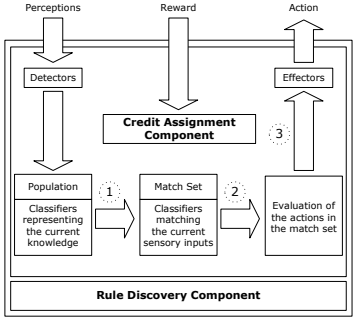
To state, in concrete technical form, a model of a complete mind and its several aspects

- A cognitive system interacting with an environment
- Binary detectors and effectors
- Knowledge = set of classifiers
- Condition-action rules that recognize a situation and propose an action
- Payoff reservoir for the system's needs
- Payoff distributed through an epochal algorithm
- Internal memory as message list
- Genetic search of classifiers



Holland's Learning Classifier Systems

- Explicit representation of the incoming reward
- Good classifiers are the ones that predict high rewards
- Credit Assignment using Bucket Brigade
- Rule Discovery through a genetic algorithm applied to the entire rule base (on the whole solution)
- Description was vast It did not work right off! Very limited success



- David E. Goldberg: Computer-aided gas pipeline operation using genetic algorithms and rule learning, PhD thesis. University of Michigan. Ann Arbor, MI.

Learning System LS-1 & Pittsburgh Classifier Systems

Holland models learning as ongoing adaptation process

De Jong instead views learning as optimization
Genetic algorithms applied to a population of rule sets

1. t := 0
2. Initialize the population P(t)
3. Evaluate the rules sets in P(t)
4. While the termination condition is not satisfied
5. Begin
6. Select the rule sets in P(t) and generate Ps(t)
7. Recombine and mutate the rule sets in Ps(t)
8. P(t+1) := Ps(t)
9. t := t+1
10. Evaluate the rules sets in P(t)
11. End

Pittsburgh Classifier System

No apportionment of credit
Offline evaluation of rule sets

When?

As time goes by…

Reinforcement Learning & Machine Learning

1970's — Genetic algorithms and CS-1
Research flourishes
Success is limited

1980's — Evolving rules as optimization
Research follows Holland's vision
Success is still limited

Stewart Wilson creates XCS

…pplications
…ts on classification
…he interest fades away

2000's — Classifier systems finally work
Large development of models, facetwise theory, and applications

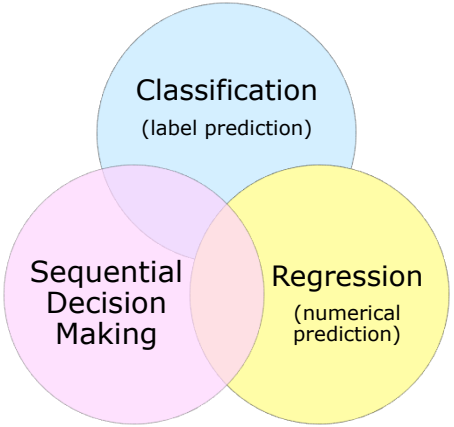
Stewart W. Wilson & The XCS Classifier System

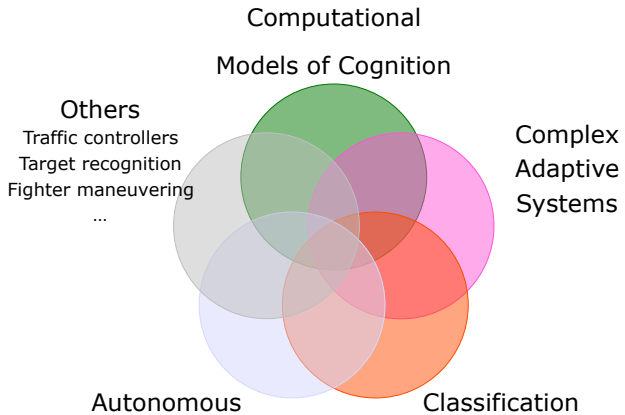
1. Simplify the model
2. Go for accurate predictions not high payoffs
3. Apply the genetic algorithm to subproblems not to the whole problem
4. Focus on classifier systems as reinforcement learning with rule-based generalization
5. Use reinforcement learning (Q-learning) to distribute reward

Most successfull model developed so far

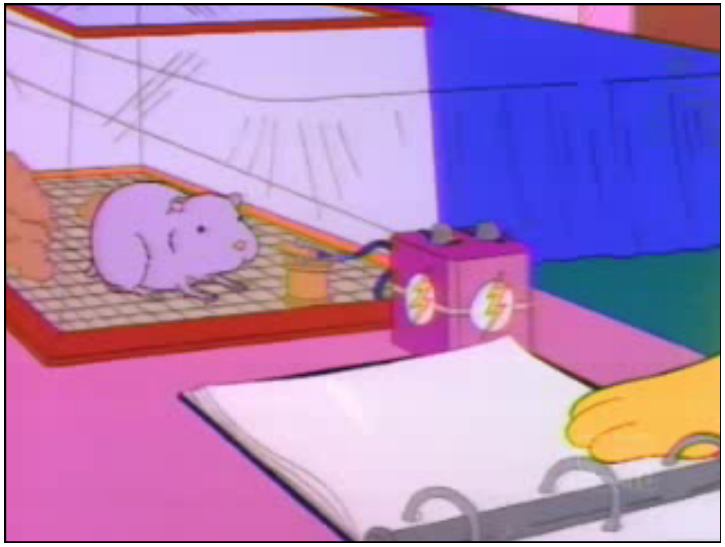
- Wilson, S.W.: Classifier Fitness Based on Accuracy. Evolutionary Computation 3(2), 149-175 (1995).

What area?
What applications?

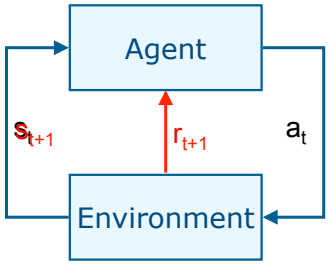




Learning Classifier Systems?



Learning Classifier Systems as Reinforcement Learning Methods



- The goal: maximize the amount of reward received
- How much future reward when a_t is performed in s_t?
- What is the expected payoff for s_t and a_t?
- Need to compute a value function, $Q(s_t,a_t) \rightarrow$ payoff

Define the inputs, the actions, and how the reward is determined

Define the expected payoff

Compute a value function $Q(s_t,a_t)$ mapping state-action pairs into expected payoffs

How does reinforcement learning work? Then, Q-learning is an option.

- At the beginning, $Q(\cdot,\cdot)$ is initialized with random values
- At time t,

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \beta\left(r_{t+1} + \gamma \max_{a \in A} Q(s_{t+1},a) - Q(s_t,a_t)\right)$$

- Parameters,
  - Discount factor γ
  - The learning rate β
  - The action selection strategy

previous value

new estimate

incoming reward

new estimate

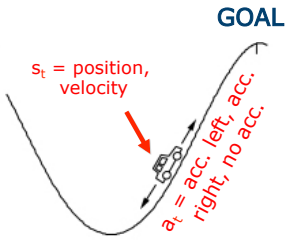
Reinforcement learning assumes that $Q(s_t,a_t)$ is represented as a table

But the real world is complex, the number of possible inputs can be huge!

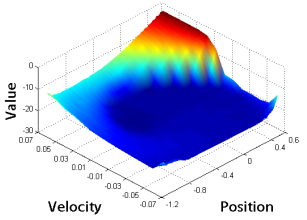
We cannot afford an exact $Q(s_t,a_t)$

Example: The Mountain Car

r_t = 0 when goal is reached, -1 otherwise



Value Function $Q(s_t,a_t)$



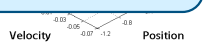
Task: drive an underpowered car up a steep mountain road

What are the issues?

Learning an unknown payoff function
while also trying to approximate it

Approximator works on intermediate estimates
While also providing information for the learning

Convergence is not guaranteed



They solve reinforcement learning problems

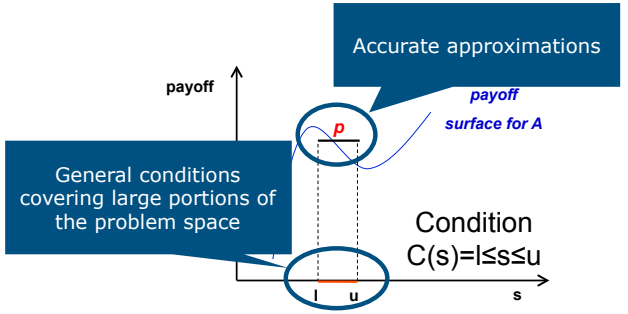
Represent the payoff function $Q(s_t, a_t)$ as
a population of rules, the classifiers

Classifiers are evolved while
$Q(s_t, a_t)$ is learned online

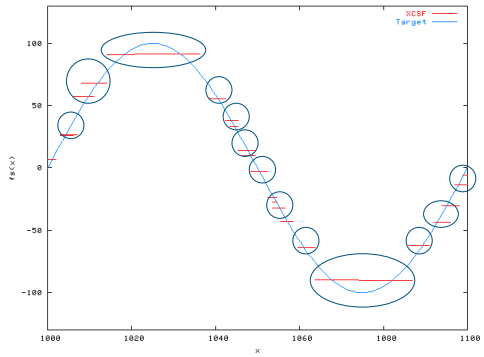
Classifiers?

What is a classifier?

IF condition C is true for input s
THEN the payoff of action A is p

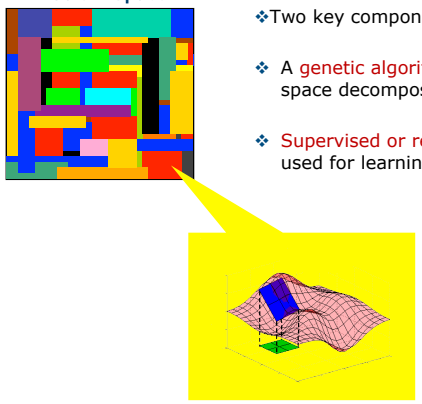


What types of solutions?



How do they work?

Problem Space

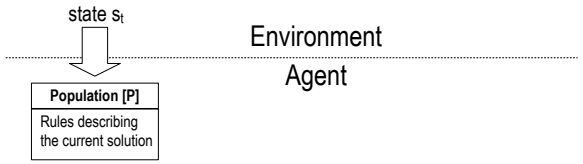


- Two key components
- A genetic algorithm works on problem space decomposition (condition-action)
- Supervised or reinforcement learning is used for learning local prediction models

How do learning classifier systems work? The main performance cycle

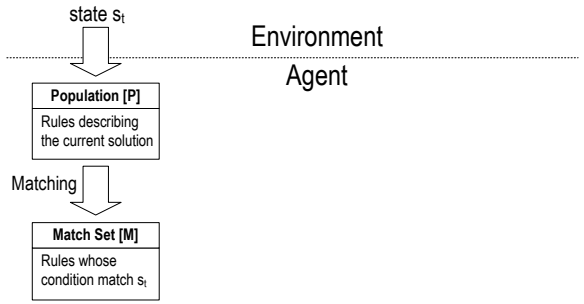
How do learning classifier systems work? The main performance cycle

29



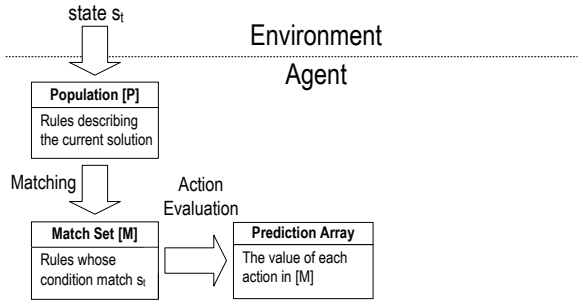
How do learning classifier systems work? The main performance cycle

30



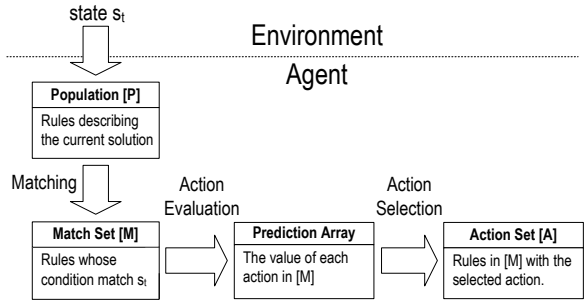
How do learning classifier systems work? The main performance cycle

31

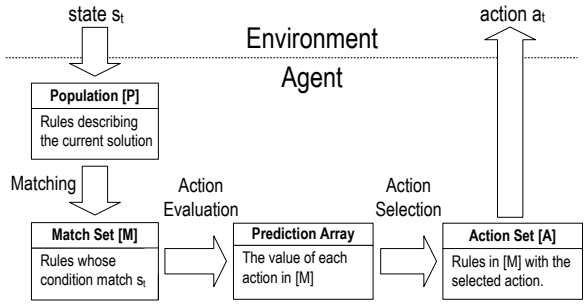


How do learning classifier systems work? The main performance cycle

32



How do learning classifier systems work? The main performance cycle



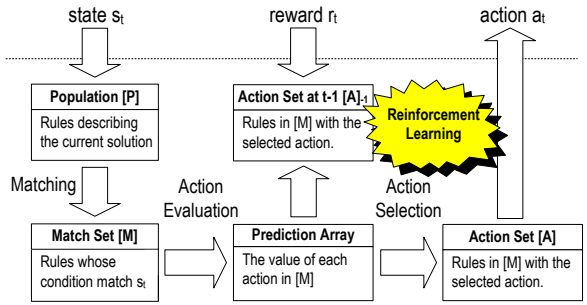
How do learning classifier systems work? The main performance cycle

The classifiers predict an expected payoff

The incoming reward is used to update the rules which helped in getting the reward

Any reinforcement learning algorithm can be used to estimate the classifier prediction

How do learning classifier systems work? The main performance cycle



How do learning classifier systems work? The reinforcement component

- For each classifier **C** in $[A]_{-1}$ the prediction **p** is updated as follows:
  - $P \leftarrow r + \gamma \max_{a \in A} \text{PredictionArray}(a)$
  - $p \leftarrow p + \beta(P - p)$
- Compare this with Q-learning
  - A rule "corresponds" to Q-table
  - p "corresponds" to the value Q(s,a)
  - P "corresponds" to "$r+\gamma \max_a Q(s_t,a)$"

In principle, any search method may be used

I prefer genetic algorithms
because they are representation independent

A genetic algorithm select, recombines,
mutate existing classifiers to search for better ones

The goal is to approximate a target value function
with as few classifiers as possible

We wish to have an accurate approximation

One possible approach is to define fitness
as a function of the classifier prediction accuracy

The genetic algorithm can take care of this

General classifiers apply more often,
thus they are reproduced more

But since fitness is based on classifiers accuracy
only accurate classifiers are likely to be reproduced

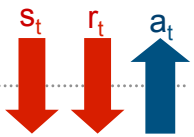
The genetic algorithm evolves
maximally general maximally accurate classifiers

What decisions?

How to apply learning classifier systems

Environment

- Determine the inputs, the actions, and how reward is distributed
- Determine what is the expected payoff that must be maximized
- Decide an action selection strategy
- Set up the parameter γ

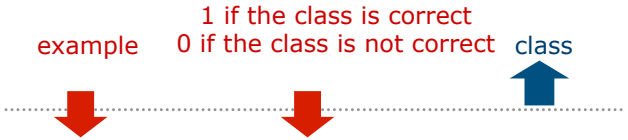


Learning Classifier System

- Select a representation for conditions, the recombination and the mutation operators
- Select a reinforcement learning algorithm
- Setup the parameters, mainly the population size, the parameters for the genetic algorithm, etc.
- Parameter β

Things can be extremely simple! For instance in supervised classification

Environment



- Select a representation for conditions and the recombination and mutation operators
- Setup the parameters, mainly the population size, the parameters for the genetic algorithm, etc.

Learning Classifier System

General principles

An Example...

Six Attributes

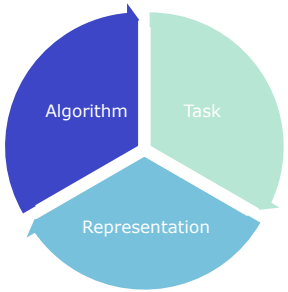
A hidden concept...

Several cases

A1	A2	A3	A4	A5	A6	CLASS
1	1	1	1	3	1	1
1	1	1	1	3	2	1
1	1	1	3	2	1	1
1	1	1	3	3	2	1
1	1	2	3	1	2	1
1	2	1	1	1	2	1
1	2	1	1	2	1	0
1	2	1	1	3	1	0
1	2	1	1	4	2	0
1	2	1	2	1	1	1
1	2	1	2	3	1	0
...	...	...	...	...	...	...

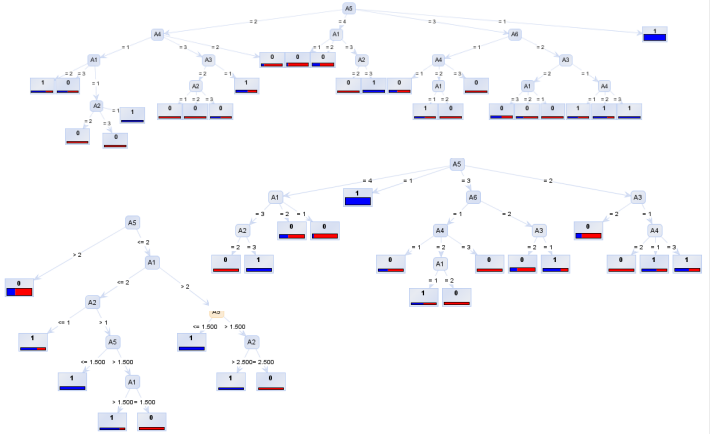
What is the concept?

Traditional Approach?



- Classification ⇨ Trees ⇨ C4.5? ID3? CHAID? …
- Classification ⇨ Rules ⇨ CN2? C4.5rules? …
- Prediction ⇨ Trees ⇨ CART? …

I Need to Classify, I Want Trees, What Algorithm? ID3? C4.5? CHAID?



I Need to Classify, I Want Rules! What Algorithm?

- OneRule

  if A5 = 3 then 0 (11/19)
  if A5 = 2 then 0 (11/20)
  if A5 = 4 then 0 (11/23)
  if A5 = 1 then 1 (29/0)

  correct: 91 out of 124 training examples.

- Rule Learner

  if A5 = 4 and A1 = 1 then 0 (1/13)
  if A5 = 1 then 1 (29/0)
  if A4 = 2 and A5 = 2 then 0 (1/6)
  if A1 = 1 and A2 = 2 then 0 (0/10)
  else 0 (27/29)

  correct: 87 out of 116 training examples.

- FOIL

  is_0(A1,A2,A3,A4,A5,A6) :- a1 ≠ a2 Æ a5 ≠ 1

Different task, different solution representation?
Completely different algorithm!

Accurate Estimates About Classifiers (Powerful RL)

Genetics-Based Generalization

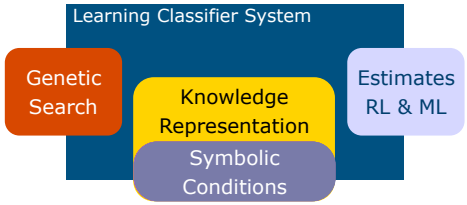
Classifier Representation

Learning Classifier Systems: One Principle Many Representations

Ternary rules
0, 1, #

if A1=1ÆA2=1⇨1
if A1=2ÆA2=2⇨1

if a_5<2 or
a_1=a_2, class=1



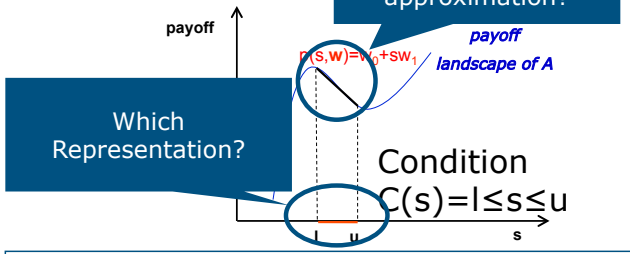
Same framework!
Just plug-in your favorite representation

Better classifiers

What is computed prediction?

Replace the prediction p by
a parametrized fu

Which type of approximation?



Which Representation?

Stewart W. Wilson: Classifiers that approximate functions. Natural Computing 1(2-3): 211-234 (2002)

XCSF: XCS with Computed Prediction

- Each classifier has a vector of parameters w
- Classifier prediction is computed as,

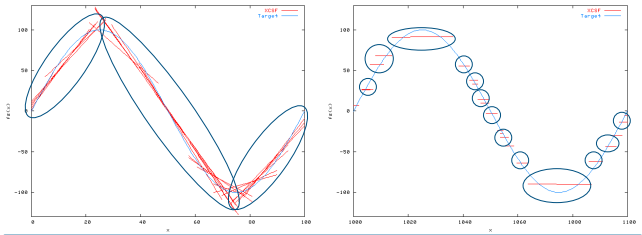
$$cl.p(\vec{x}) = cl.w_0 \times x_0 + \sum_{i>0} cl.w_i \times x_i$$

- Classifier weights are updated using Widrow-Hoff update,

$$\Delta w_i = \frac{\eta}{|x_i|^2}(P - cl.p(\vec{x}))x_i$$

$$cl.w_i \leftarrow cl.w_i + \Delta w_i$$

Same example with computed prediction



No need to change the framework

Just plug-in your favorite estimator

Linear, Polynomial, NNs, SVMs, tile-coding

Lanzi, PL, Extending XCSF beyond linear approximation. GECCO, 2005
Loiacono D. and Lanzi P.L.. Evolving neural networks for classifier prediction with XSCF. ECAI, 2006
Lanzi and Loiacono XCSF with neural prediction. IEEE CEC, 2006.
Loiacono, D., Marelli A., and Lanzi PL. Support vector regression for classifier prediction. GECCO, 2007.

What do we want?

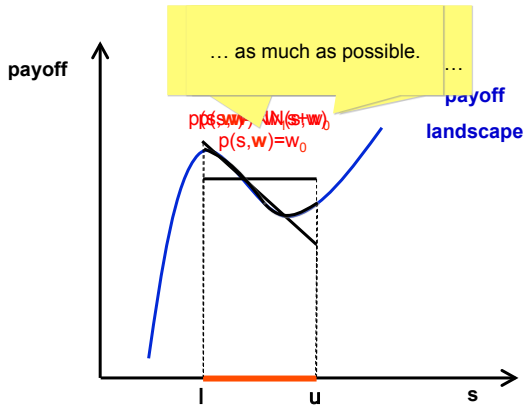
Fast learning

Learn something as soon as possible!

Accurate solutions

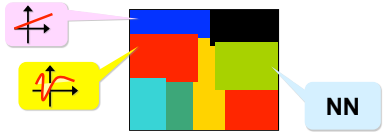
As the learning proceeds
the solution accuracy should improve

Is there another approach?



Ensemble Classifiers

None of prediction models is the "best"
Let evolution search for the "most suited"



Almost as fast as using best model
Model is adapted effectively in each subspace

What theory?

Learning Classifier Systems

Representation, Reinforcement Learning,
& Genetics-based Search

Unified theory is impractical

Develop facetwise models

Facetwise Models for a Theory of Evolution and Learning

- Prof. David E. Goldberg
  University of Illinois at Urbana Champaign
- David Goldberg & Kumara Sastry
  Genetic Algorithms: The Design of Innovation
  Springer-Verlag May 2008
- Facetwise approach for the analysis
  and the design of genetic algorithms

- In learning classifier systems
  - Separate learning from evolution
  - Simplify the problem by focusing
    only on relevant aspect
  - Derive facetwise models
- Applied to model several aspects of evolution

$S([A]) = S([P])/(2-S([P]))$

Since the genetic algorithm is applied to the action set,
there is a generalization pressure regulated by this equation.

$N>-\log(1-\theta)/p$

Given the probability θ to maintain all the subsolutions
with occurrence probability p, then the population size N …

$O(L\ 2^{o+a})$

Time to converge for a problem of L bits order o
and with a problem classes

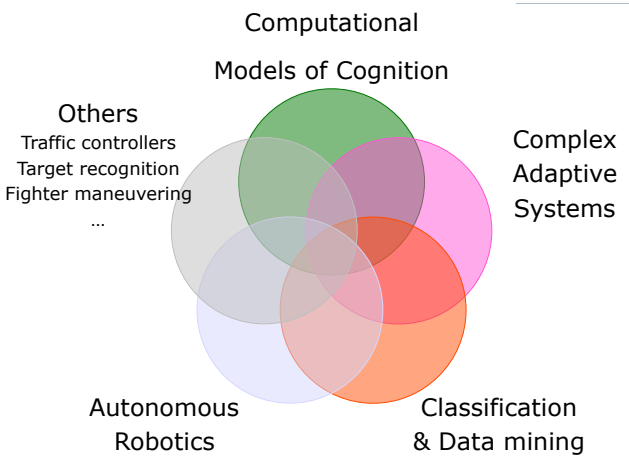
Martin V. Butz, Tim Kovacs, Pier Luca Lanzi, Stewart W. Wilson: Toward a theory of generalization and learning in XCS. IEEE Trans. Evolutionary Computation 8(1): 28-46 (2004)
Martin V. Butz, Kumara Sastry, David E. Goldberg: Strong, Stable, and Reliable Fitness Pressure in XCS due to Tournament Selection. Genetic Programming and Evolvable Machines 6(1): 53-77 (2005)
Martin V. Butz, David E. Goldberg, Pier Luca Lanzi: Bounding Learning Time in XCS. GECCO (2) 2004: 739-750

Advanced topics…

What the Advanced Topics?

- Models Tailored for Applications
  - Anticipatory Behavior (ACS)
  - Data Mining, classification and prediction (e.g., UCS)
  - Epidemiology (e.g., EpiCS, EpiXCS)
- Improved representations (GP, GEP, …)
- Improved genetic search (EDAs, ECGA, BOA, ….)
- Improved estimators
- Scalability
  - Matching
  - Distributed models

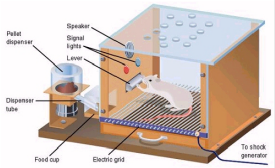
What Applications?



Modeling Cognition

What Applications? Computational Models of Cognition

- Learning classifier system model certain aspects of cognition
  - Human language learning
  - Perceptual category learning
  - Affect theory
  - Anticipatory and latent learning

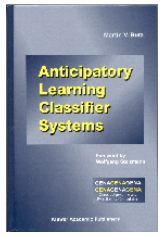


- Learning classifier systems provide good models for animals in experiments in which the subjects must learn internal models to perform as well as they do
- Martin V. Butz, University of Würzburg, Department of Cognitive Psychology III Cognitive Bodyspaces: Learning and Behavior (COBOSLAB)
- Wolfgang Stolzmann, Daimler Chrysler
- Rick R. Riolo, University of Michigan, Center for the Study of Complex Systems



References

- Butz, M.V.: Anticipatory Learning Classifier Systems, Genetic Algorithms and Evolutionary Computation, vol. 4. Springer-Verlag (2000)
- Riolo, R.L.: Lookahead Planning and Latent Learning in a Classifier System. In: J.A. Meyer, S.W. Wilson (eds.) From Animals to Animats 1. Proceedings of the First International Conferenceon Simulation of Adaptive Behavior (SAB90), pp. 316{326. A Bradford Book. MIT Press (1990)
- Stolzmann, W. and Butz, M.V. and Hoffman, J. and Goldberg, D.E.: First Cognitive Capabilities in the Anticipatory Classifier System. In: From Animals to Animats: Proceedings of the Sixth International Conference on Simulation of Adaptive Behavior. MIT Press (2000)



Computational Economics

What Applications? Computational Economics

- To models one single agent acting in the market (BW Arthur, JH Holland, B LeBaron)
- To model many interactive agents each one controlled by its own classifier system.

- Modeling the behavior of agents trading risk free bonds and risky assets
- Different trader types modeled by supplying different input information sets to a group of homogenous agents
- Later extended to a multi-LCS architecture applied to portfolio optimization
- Technology startup company founded in March 2005

References

- Sor Ying (Byron) Wong, Sonia Schulenburg: Portfolio allocation using XCS experts in technical analysis, market conditions and options market. GECCO (Companion) 2007: 2965-2972
- Sonia Schulenburg, Peter Ross: An Adaptive Agent Based Economic Model. Learning Classifier Systems 1999: 263-282
- BW Arthur, J.H. Holland, B. LeBaron, R. Palmer, and P. Tayler: "Asset Pricing Under Endogenous Expectations in an Artificial Stock Market," in The Economy as an Evolving Complex System II. Edited (with S. Durlauf and D. Lane), Addison-Wesley, 1997.
- BW Arthur, R. Palmer, J. Holland, B. LeBaron, and P. Taylor: "Artificial Economic Life: a Simple Model of a Stockmarket," Physica D, 75, 264-274, 1994

Data analysis

What Applications? Classification and Data Mining

- Bull, L. (ed) Applications of Learning Classifier Systems. Springer. (2004)
- Bull, L., Bernado Mansilla, E. & Holmes, J. (eds) Learning Classifier Systems in Data Mining. Springer. (2008)
- Nowadays, by far the most important application domain for LCSs
- Many models GA-Miner, REGAL, GALE GAssist
- Performance comparable to state of the art machine learning

Human Competitive Results 2007

X Llorà, R Reddy, B Matesic, R Bhargava: Towards Better than Human Capability in Diagnosing Prostate Cancer Using Infrared Spectroscopic Imaging

Hyper Heuristics

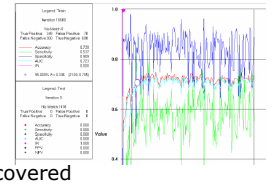
What Applications? Hyper-Heuristics

- Ross P., Marin-Blazquez J., Schulenburg S., and Hart E., Learning a Procedure that can Solve Hard Bin-packing Problems: A New GA-Based Approach to Hyper-Heuristics. In Proceedings of GECCO 2003
- Bin-packing and timetabling problems
- Pick a set of non-evolutionary heuristics
- Use classifier system to learn a solution process not a solution
- The classifier system learns a sequence of heuristics which should be applied to gradually transform the problem from its initial state to its final solved state.

Medical data

What Applications? Epidemiologic Surveillance

- John H. Holmes
  Center for Clinical Epidemiology & Biostatistics
  Department of Biostatistics & Epidemiology
  University of Pennsylvania - School of Medicine
- Epidemiologic surveillance data need adaptivity to abrupt changes
- Readable rules are attractive
- Performance similar to state of the art machine learning



- But several important feature-outcome relationships missed by other methods were discovered
- Similar results were reported by Stewart Wilson for breast cancer data

References

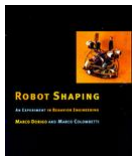
- John H. Holmes, Jennifer A. Sager: Rule Discovery in Epidemiologic Surveillance Data Using EpiXCS: An Evolutionary Computation Approach. AIME 2005: 444-452
- John H. Holmes, Dennis R. Durbin, Flaura K. Winston: A New Bootstrapping Method to Improve Classification Performance in Learning Classifier Systems. PPSN 2000: 745-754
- John H. Holmes, Dennis R. Durbin, Flaura K. Winston: The learning classifier system: an evolutionary computation approach to knowledge discovery in epidemiologic surveillance. Artificial Intelligence in Medicine 19(1): 53-74 (2000)

Autonomous Robotics

What Applications? Autonomous Robotics

- In the 1990's, a major testbed for learning classifier systems.

- Marco Dorigo and Marco Colombetti: Robot Shaping An Experiment in Behavior Engineering, 1997



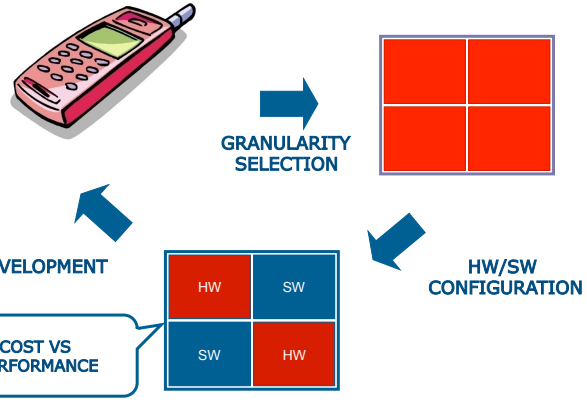
- They introduced the concept of robot shaping defined as the incremental training of an autonomous agent.
- Behavior engineering methodology named BAT: Behavior Analysis and Training.
- Recently, University of West England applied several learning classifier system models to several robotics problems

UWE BRISTOL University of the West of England

Design

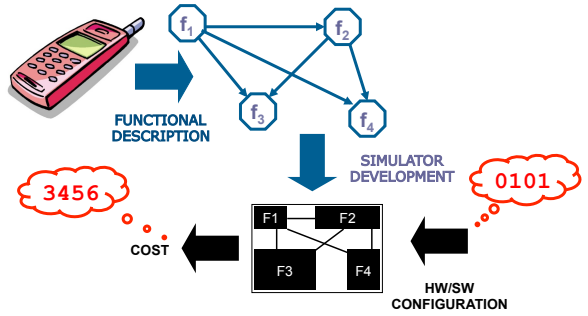
What Applications? Hardware/Software Codesign

81



What Applications? Hardware/Software Codesign

82



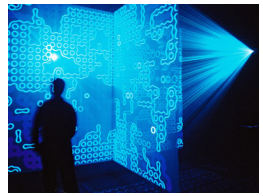
if 0##1 then change 4 to SW will increase cost

Artificial Ecosystems

What Applications? Modeling Artificial Ecosystems

84

- Jon McCormack, Monash University
- Eden: an interactive, self-generating, artificial ecosystem.
- World populated by collections of evolving virtual creatures
- Creatures move about the environment,
  - Make and listen to sounds,
  - Foraging for food,
  - Encountering predators,
  - Mating with each other.



- Creatures evolve to fit their landscape
- Eden has four seasons per year (15mins)
- Simple physics for rocks, biomass and sonic animals.

MONASH University Information Technology

Eden: An Evolutionary Sonic Ecosystem

References

- McCormack, J. Impossible Nature The Art of Jon McCormack Published by the Australian Centre for the Moving Image ISBN 1 920805 08 7, ISBN 1 920805 09 5 (DVD)
- J. McCormack: New Challenges for Evolutionary Music and Art, ACM SIGEVOlution Newsletter, Vol. 1(1), April 2006, pp. 5-11.
- McCormack, J. 2005, 'On the Evolution of Sonic Ecosystems' in Adamatzky, et al. (eds), Artificial Life Models in Software, Springer, Berlin.
- McCormack, J. 2003, 'Evolving Sonic Ecosystems', Kybernetes, 32(1/2), pp. 184-202.

Chemical & Neuronal Networks

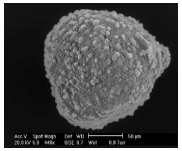
What Applications? Chemical and Neuronal Networks

- L. Bull, A. Budd, C. Stone, I. Uroukov, B. De Lacy Costello and A. Adamatzky University of the West of England

University of the West of England

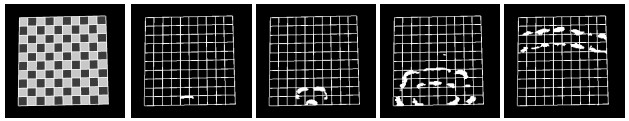
- Behaviour of non-linear media controlled automatically through evolutionary learning
- Unconventional computing realised by such an approach.

- Learning classifier systems
  - Control a light-sensitive sub-excitable Belousov-Zhabotinski reaction
  - Control the electrical stimulation of cultured neuronal networks



What Applications? Chemical and Neuronal Networks

- To control a light-sensitive sub-excitable BZ reaction, pulses of wave fragments are injected into the checkerboard grid resulting in rich spatio-temporal behaviour
- Learning classifier system can direct the fragments to an arbitrary position through control of the light intensity within each cell



- Learning Classifier Systems control the electrical stimulation of cultured neuronal networks such that they display elementary learning, respond to a given input signal in a pre-specified way
- Results indicate that the learned stimulation protocols identify seemingly fundamental properties of in vitro neuronal networks

References

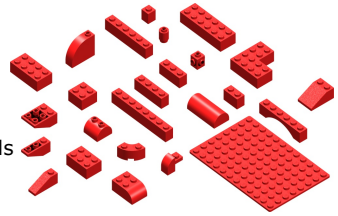
- Larry Bull, Adam Budd, Christopher Stone, Ivan Uroukov, Ben De Lacy Costello and Andrew Adamatzky: Towards Unconventional Computing through Simulated Evolution: Learning Classifier System Control of Non-Linear Media Artificial Life (to appear)
- Budd, A., Stone, C., Masere, J., Adamatzky, A., DeLacyCostello, B., Bull, L.: Towards machine learning control of chemical computers. In: A. Adamatzky, C. Teuscher (eds.) From Utopian to Genuine Unconventional Computers, pp. 17-36. Luniver Press
- Bull, L., Uroukov, I.S.: Initial results from the use of learning classier systems to control n vitro neuronal networks. In: Lipson [189], pp. 369-376

Conclusions

Conclusions

- Cognitive Modeling
- Complex Adaptive Systems
- Machine Learning
- Reinforcement Learning
- Metaheuristics
- …

- Many blocks to plug-in
- Several representations
- Several RL algorithms
- Several evolutionary methods
- …



Additional Information

- WWW
  - http://www.learning-classifier-systems.org
  - http://www.illigal.uiuc.edu/web/lcs-n-gbml/
  - Illinois Genetic Algorithm Laboratory
    http://www.illigal.org
  - UWE Learning Classifier Systems Group
    http://www.cems.uwe.ac.uk/lcsg/
  - A Learning Classifier Systems Bibliography
    http://www.cs.bris.ac.uk/~kovacs/lcs/search.html
- Mailing lists
  - lcs-and-gbml group @ Yahoo

Additional Information

- Software
  - http://www.illigal.org
    Martin Butz's ACS & XCS in C and Java
    Pier Luca Lanzi's C++ XCSLib (XCS and XCSF)
  - http://medal.cs.umsl.edu/files/XCSFJava1.1.zip
    Martin Butz's XCSF in Java
- Proceedings of the International Workshop on Learning Classifier Systems (Lanzi, Stolzmann, & Wilson, 2000, 2001, 2002; Kovacs, Llorà, & Takadama, 2003-2005; Bacardit, Bernadó-Mansilla, Butz, Kovacs, Llorà, Takadama, IWLCS2007)
- IWLCS here!

Books

- Bull, L. (Ed.). Applications of learning classifier systems. Berlin Heidelberg: Springer-Verlag.
- Butz, M. V. (2002). Anticipatory learning classifier systems. Kluwer Academic Publishers, Boston, MA.
- Butz, M. V. (2006). Rule-based evolutionary online learning systems: A principled approach to LCS analysis and design. Studies in Fuzziness and Soft Computing Series, Springer Verlag, Berlin Heidelberg, Germany.
- Bull, L. & Kovacs, T. (Eds.) (2005). Foundations of learning classifier systems. Berlin Heidelberg: Springer-Verlag.
- Lanzi, P. L., Stolzmann, W., & Wilson, S. W. (Eds.) (2000). Learning classifier systems: From foundations to applications (LNAI 1813). Berlin Heidelberg: Springer-Verlag.
- Drugowitsch, J., (2008) Design and Analysis of Learning Classifier Systems: A Probabilistic Approach, Springer-Verlag
- Goldberg, D. E. (1989). Genetic algorithms in search, optimization & machine learning. Addison-Wesley.
- Holland, J.H. (1975). Adaptation in natural and artificial systems. University of Michigan Press.
- Kovacs, T. (2004). Strength of accuracy: Credit assignment in learning classifier systems. Berlin Heidelberg: Springer-Verlag.