Context-Dependent Predictions and Cognitive Arm Control with XCSF

TECHNISCHE UNIVERSITÄT DARMSTADT

Learning Classifier Systems (LCS/XCSF)

Laurentius Florentin Gruber

Seminar aus Künstlicher Intelligenz
WS 2015/16
Professor Johannes Fürnkranz

Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

0) Abstract View

1) Introduction

2) XCS/XCSF: A Short Overview

3) Separated Inputs

4) Cognitive Arm Control

5) Summary and Conclusion

6) Final Discussion

Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

0) Abstract View

1) Introduction

2) XCS/XCSF: A Short Overview

3) Separated Inputs

4) Cognitive Arm Control

5) Summary and Conclusion

6) Final Discussion

0) Abstract View

TECHNISCHE UNIVERSITÄT DARMSTADT

John Holland has always envisioned learning classifier systems as cognitive systems.

- **most work on LCSs has focused:**
  - classification
  - datamining
  - function approximation

- **Goal of this Paper:**
  - Show that XCSF classifier system can be very suitable modified to control a robot system with redundant degrees of freedom (e.g.: robot arm).

- **Inspiration:**
  - Recent research suggest that sensorimotor codes are nearly everywhere in the brain and an essential ingredient for cognition in general.

- **for That:**
  - The XCSF system is modified to learn classifiers that encode piecewise linear sensorimotor structures, which are conditioned on prediction-relevant contextual input.

0) Abstract View
What's comming?:

TECHNISCHE UNIVERSITÄT DARMSTADT

- **robot arm problem:**
  - Show that XCSF partitions the (contextual) posture space of the arm in such a way that accurate hand movements can be predicted given particular motor commands.

  - Show that the inversion of the sensorimotor predictive structures enables accurate goal-directed closed-loop control of arm reaching movements.

- **performance of the modified XCSF system:**
  - Specific on a set of artificial functions.

- **Conclusion:**
  - XCSF is a useful tool to evolve problem space partitions that are maximally effective for the encoding of sensorimotor dependencies.

- **final discussion:**
  - Relation of taken approach to actual brain structures and cognitive psychology theories of learning and behavior.

Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

0) Abstract View

1) Introduction

2) XCS/XCSF: A Short Overview

3) Separated Inputs

4) Cognitive Arm Control

5) Summary and Conclusion

6) Final Discussion

1) Introduction

Difference LCS and XCSF:

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Michigan-style LCSs:**

  - evolve a set of rules (called population of classifiers)

  - every rule consists of:

    - a condition part (is evolved online)
    - an action part
    - a prediction part (is usually developed by some form of gradient-based approximation)

- **XCSF classifier system:**

  - conditions represent hyper-rectangles in a real-valued input space

    - Enhancements have shown that they cold also be changed to (e.g. hyperelllipsoidal conditions)

  - predictions are computed linearly from the condition input and an offset value

    - Enhancements have shown that they do not necessarily need to be linear (e.g. polynomial, or even predictions formed by a multilayer perceptron)

1) Introduction

TECHNISCHE UNIVERSITÄT DARMSTADT

- **XCSF research had focused on improvements of the:**
  - piece-wise approximations formed by the predictions
  - condition structures
  - evolutionary mechanisms

  - It's also possible to form predictions from variables that are different from, but related to, the input processed by the condition.

  - In this case, a classifier condition can be seen as identifying the context in which a certain (linear) prediction applies.

- **John Holland had introduced LCSs as cognitive systems.**

  - solving a food-water maintenance task (by CS1)

  - LCSs have mainly been applied to: classification, datamining problems, prediction and control problems (e.g. stock-market predictions, industrial plant control,...)

1) Introduction

TECHNISCHE UNIVERSITÄT DARMSTADT

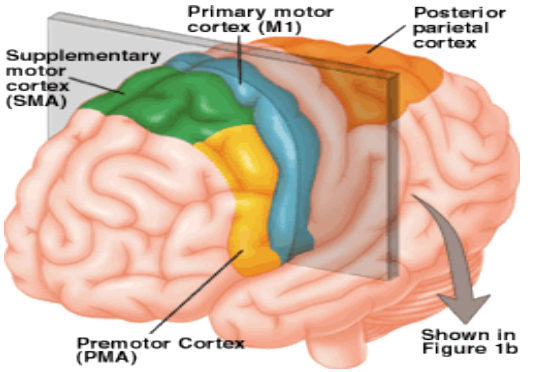
- **Robot control applications:**

  - remain space and focus on immediate control problems (s.a. light following)

  - XCSF can also evolve classifiers in an arm posture space for the prediction of motor activity-dependent hand position changes of a simulated robot arm. (as we will see)

  - A simple inversion of the this XCSF can be used to control the robot arm.

1) Introduction



- **Simulated neural approaches for directional, cortical arm control:**
  - **premotor/motor cortical areas of the brain** represent the body by means of population encodings, where each neuron covers a certain subspace of the respective body part in its receptive field.
  - **XCSF can evolve a similar representation**, in which each classifier condition specifies a certain subspace and the classifier population covers the complete problem space.
  - **XCSF evolves its population encoding** (the set of classifiers) in order to optimize a prediction task (prediction of hand position changes dependent on contextual arm posture and predictive motor activity information sources).
  - **The sensory system shows patterns** that can have only developed to improve motor control capabilities, and thus, that sensory representations develop for motor control purposes.
  - **The inversion of the learned forward predictions** enable directional arm control.

- **The resulting modified XCSF is a self-developing, cognitive arm control system.**

Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

0) Abstract View

1) Introduction

2) XCS/XCSF: A Short Overview

3) Separated Inputs

4) Cognitive Arm Control

5) Summary and Conclusion

6) Final Discussion

2) XCS/XCSF: A Short Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

- **XCS:**
  - Originally designed as an online generalizing reinforcement learning system that approximates the Q-value function of a Markov decision problem.

    - approximate Q-value with a compact and highly general representation

  - is also able to approximate real-value functions and can be applied to datamining problems.

  - evolves rules to partition the experienced problem space in a way, that accurate, general classifiers emerge
    - together cover the whole problem space
    - and yield accurate predictions

2) XCS/XCSF: A Short Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

- **XCSF:**
  - The classifier structure consist only of:

    - condition
    - prediction

  - The condition part determines the function domain subspace, in which the prediction applies.
  - The prediction is then determined from the inner product of the problem input vector and the classifier prediction weight vector plus a (possibly scaled) offset weight.
  - Classifier fitness is derived from the inverse of the estimated mean error of the classifier prediction relative to the errors of overlapping classifiers.
  - XCSF is an online learning, piecewise-linear function approximation system, in which classifier condition structures are evolved to improve the accuracies of their corresponding local linear approximations.

2) XCS/XCSF: A Short Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

- **here used modified XCSF:**
  - **Classifier parameters** are updated in online interaction with the problem at hand, iteratively processing problem instances and corresponding actual function values.
  - **Gradient techniques** are used to approximate prediction, error, and fitness values.
  - **niche-based steady-state genetic algorithm** (dependent on the fitness values): selects highly accurate classifiers and deletes low-fitness classifiers in overcrowded niches.

  => Together, the resulting evolutionary pressures evolve generalized, piecewise-linear approximations of the target function.

  - **rotating hyper-ellipsoidal condition structures:** update the predictions with the recursive least squares technique, and apply set-proportionate tournament selection for offspring selection.
    > these mechanisms improve XCSF's learning speed, accuracy, robustness
  - **compaction mechanism**: switches to closest classifier matching and turns off mutation and crossover
    > to investigate the possibility of generating even more general function approximation

Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

0) Abstract View

1) Introduction

2) XCS/XCSF: A Short Overview

3) Separated Inputs

4) Cognitive Arm Control

5) Summary and Conclusion

6) Final Discussion

3) Separated Inputs

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Functions to test XCSF's learning capabilities, robustness, structural capabilities:**

$$f_{1,m}^{n,2}(x_1,..,x_n,y_1,y_2) = a*(b*x_1+y_1+j*y_2) \quad (1)$$

$$f_{2,m}^{n,2}(x_1,..,x_n,y_1,y_2) = a*(b*\sum_i^n x_i+y_1+j*y_2) \quad (2)$$

$$f_{3,m}^{n,2}(x_1,..,x_n,y_1,y_2) = a*(\sin(b*x_j)+y_1+j*y_2) \quad (3)$$

- (a, b): scalar function modifiers
- (n): number of condition inputs dimensions
- (m): output dimensions (with $1 \leq j \leq m$)
- (2): prediction input dimensions
- (x_i): condition inputs
- (y_i): prediction inputs

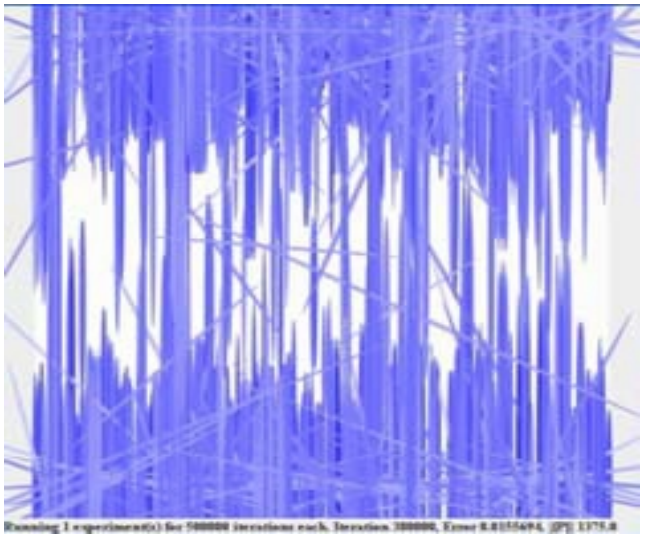
3) Separated Inputs
f_1

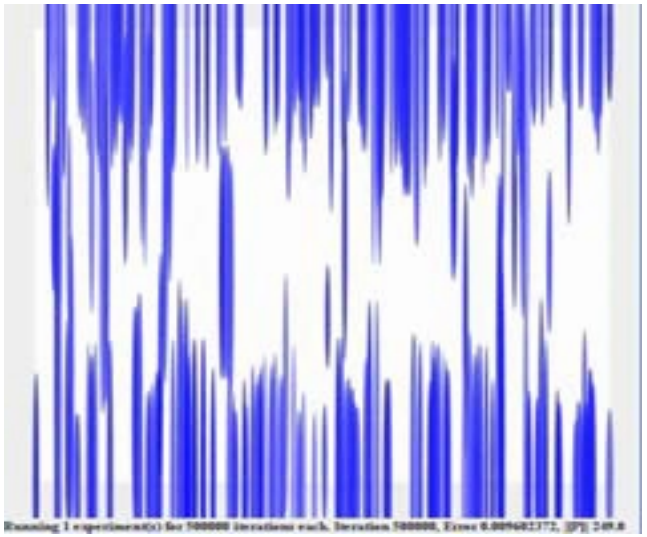
TECHNISCHE UNIVERSITÄT DARMSTADT

- **Classifiers only focus on partitioning the first context dimension in problem f_1**

  - n = 2

  - the classifier conditions (20% of actual size) distributed over the condition-input space
    before (380k learning iterations) and after compaction (500k learning iterations)





  - For each separate output dimension, the function changes only linearly by a larger factor j.

  - inputs are only dependent on x_1, so that classifiers evolve that ignore additional input dimensions.

  - The plot confirms that classifier conditions partition the function-relevant input dimension x_1 and generalize over x_2.

3) Separated Inputs
XCSF performance:

TECHNISCHE UNIVERSITÄT DARMSTADT

- $f_{1,m}^{n,2}(x_1, .., x_n, y_1, y_2) = a * (b * x_1 + y_1 + j * y_2)$

- the condition structure is equally effective for all output dimensions in f1 & the output dimensions are approximated in parallel in each classifier

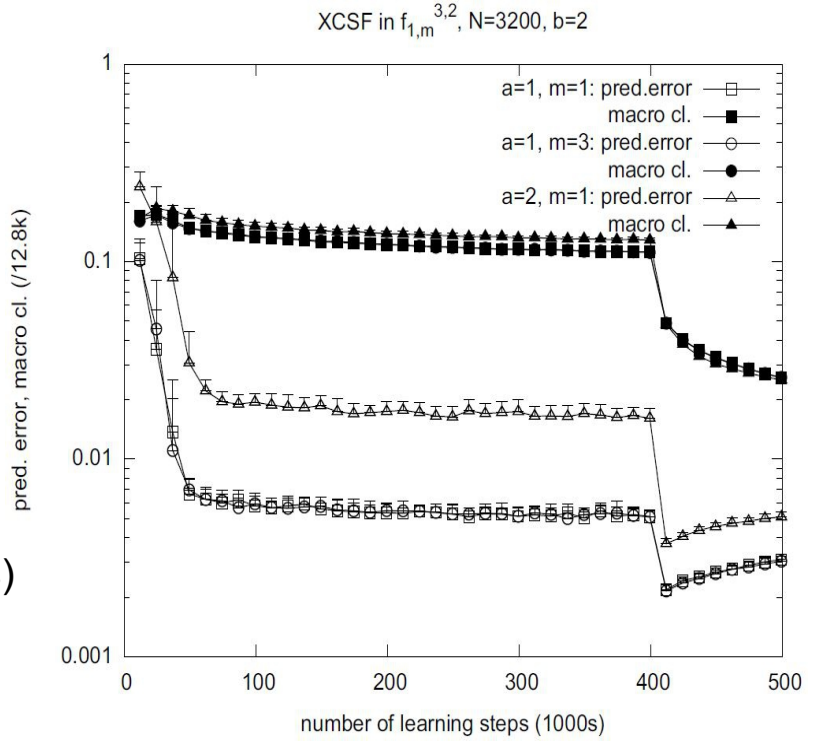
  => The number of output-dimensions affects not the learning performance.

- **The graphic:** the number of input dimensions doesn't affect performance but, a larger scaling factor does, because the scaling doubles the importance of the context dimension x_1.

  - with n =3

  - compaction algorithm (after 400k learning iterations):

  => decrease the population size (number of distinct classifiers) and the error;
  selectively deleting inaccurate offspring
  and less accurate overlapping classifiers.



3) Separated Inputs
XCSF performance:

TECHNISCHE UNIVERSITÄT DARMSTADT

- $f_{2,m}^{n,2}(x_1, .., x_n, y_1, y_2) = a * (b * \sum_i^n x_i + y_1 + j * y_2)$

  - requires the structure of all n context dimensions

  - each output dimension is equally dependent on the context input dimensions

  - Due to the linear combination of the context dimension $\sum_i^n x_i$, the context space needs to be structured obliquely in the three dimensions.

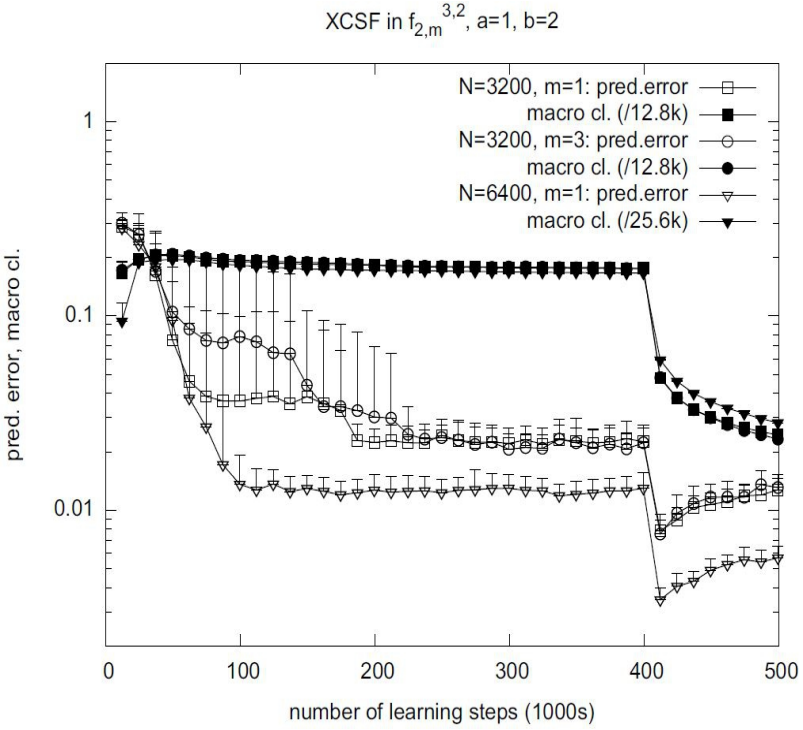
- **The graphic**:

  - with n = 3

    - The number of output dimensions does hardly affect learning accuracy and XCSF evolves increasingly accurate space partitions.

    - With a larger population size of N = 6400, an error of less than 0.01 is reached.

    - Compaction (after 400k l. it.) improves performance and decreases populations size of all cases.

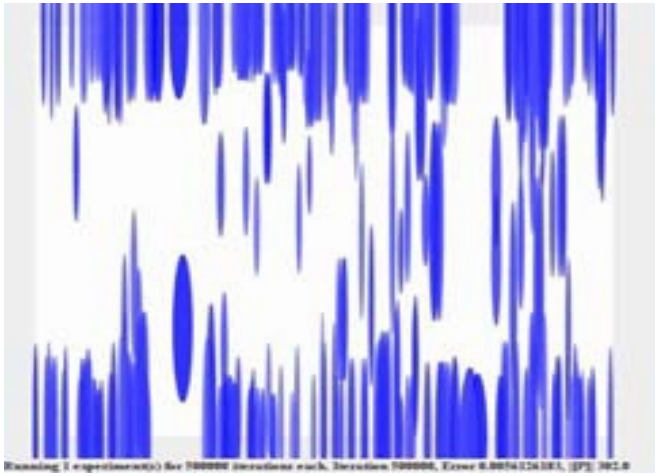


3) Separated Inputs
f_3

TECHNISCHE UNIVERSITÄT DARMSTADT

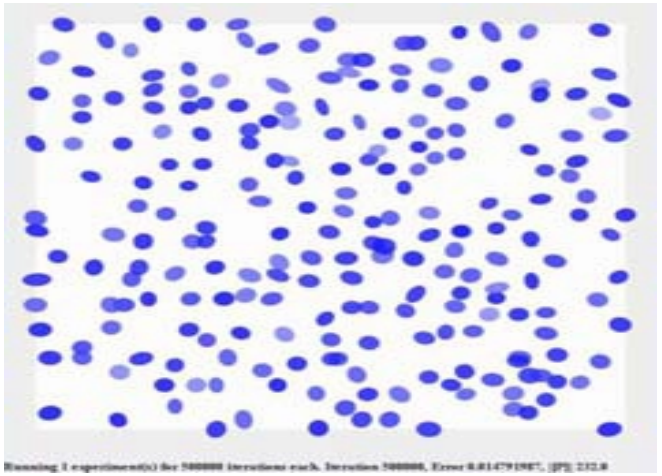
- **Typical final populations of XCSF running in Function f3 after compaction:**

(n=2, **m=1**, a=1, b=2π)



- the partitioning reflects the sine wave with larger derivatives

(n=2, **m=2**, a=1, b=1)



- the classifier conditions distribute themselves somewhat equally over the condition input space, reflecting a checkerboard quality of the problem.

- (checkerboard problem: where each input dimension should be structured finer, the larger the absolute derivative of the sine function in this dimension.)

3) Separated Inputs
XCSF performance:

TECHNISCHE UNIVERSITÄT DARMSTADT

- $f_{3,m}^{n,2}(x_1, .., x_n, y_1, y_2) = a * (\sin(b * x_j) + y_1 + j * y_2)$

  - f3 (m=1) is essentially identical to f1, except for the context is still processed by a sine function.

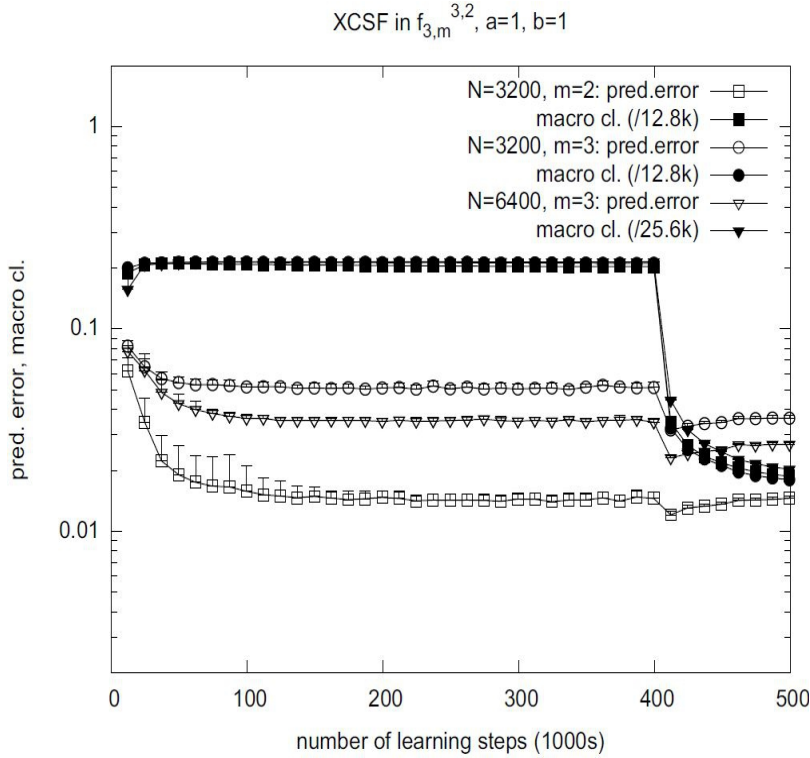
  - f3(m>1) each output dimension depends on a another context dimension.

    => Thus, since each classifier predicts all output dimensions , the problem becomes similar to a real-value checkerboard problem.

- **The graphic:**

  - with n=3

    - the additional output dimension results in a significantly harder problem.



Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

0) Abstract View

1) Introduction

2) XCS/XCSF: A Short Overview

3) Separated Inputs

4) Cognitive Arm Control

5) Summary and Conclusion

6) Final Discussion

4) Cognitive Arm Control

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Various research directions suggest that infants learn internal forward models of body movement during development.**
- **This enable the inverse goal-directed control of the body.**

=> The learning process during development:

- is self-supervised
- solves the challenge of control structures for a highly dynamic and redundant plant.

4) Cognitive Arm Control
Setup

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Mimicking a neural network approach:**

- XCSF is trained on a three-degreed of freedom arm control problem with context input that encodes the current arm posture (tree angles).

  - The simulated arm joints (limb lengths: 1.4, 1.4, 0.8) can rotate freely between:
    60-150° (shoulder); 0-150° (elbow); 10-150° (wrist)

    - In training randomly the arm is moved between -0.18° and 0.18°
    - Error threshold ϵ_0 = 0.0001

  - the prediction input encodes current motor activity (change in angles)

  - XCSF is trained on prediction of the resulting change in hand location (difference in arm end-point coordinates before and after movement)

    - evaluates the predictive capabilities

    - evaluates the resulting inverse control capabilities

4) Cognitive Arm Control
Performance in the Arm Prediction Task

TECHNISCHE UNIVERSITÄT DARMSTADT

- Prediction accuracy continuously improves.
- Compaction decreases population size over 90% while affecting performance only marginally.

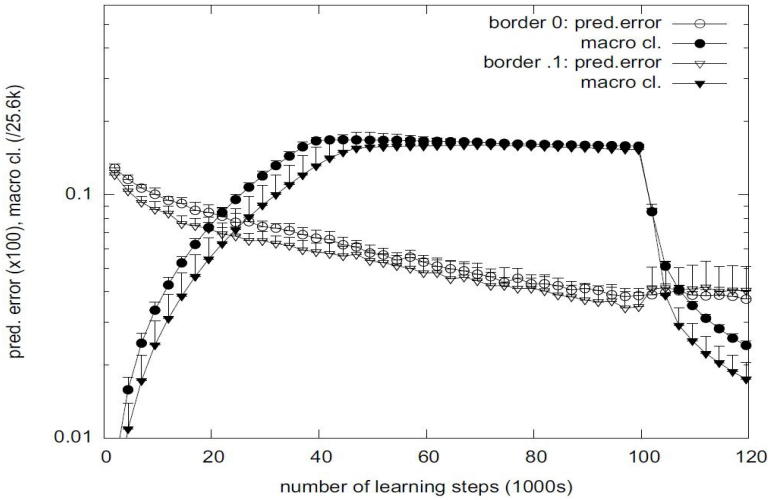
  - is the case after both 120k and 500k l. it.

- **Performance with border effects:**
  - Full problem space (border 0)

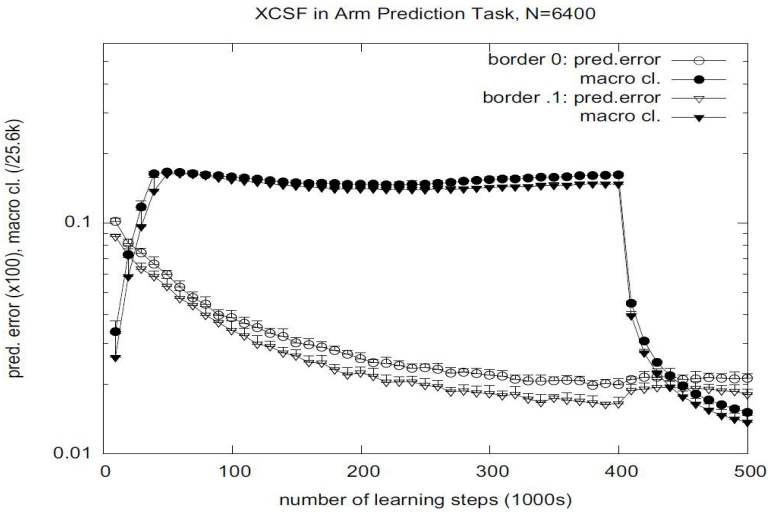
    - yields accurate predictions

  - Inner problem space
    (surrounding unsampled region covering the out 10% of the length of each dimension)

    - is even easier to approximate



(a) Compaction after 100k learning steps



(b) Compaction after 400k learning steps

4) Cognitive Arm Control
Arm Control Task

TECHNISCHE UNIVERSITÄT DARMSTADT

- Given a current arm posture, the current match set is determined, which is then used to determine the inverse motor command.:
  - for each classifier in the set, the inverse of the forward prediction is calculated
  - there are only two linear equations, which usually are used to predict hand displacements $(\Delta x, \Delta y)$ given three angular motor commands
  - ✖ But to determine the thee motor commands to achieve a desired hand displacement, an additional constraint would be necessary.
  - ➢ turns are taken and a different motor command is set to zero in each movement command calculation
  - ➢ The overall motor command in one calculation is determined by the average movement command suggested by all micro-classifieres in a match set.
  - ➢ Before executing the command, the movement vector is normalized to a total angular movement of 1.8°
- 78 start-posture are tested ([[shoulder, elbow, wrist] ^ [flexed, centred, bend]-1]*[3 goal locations])

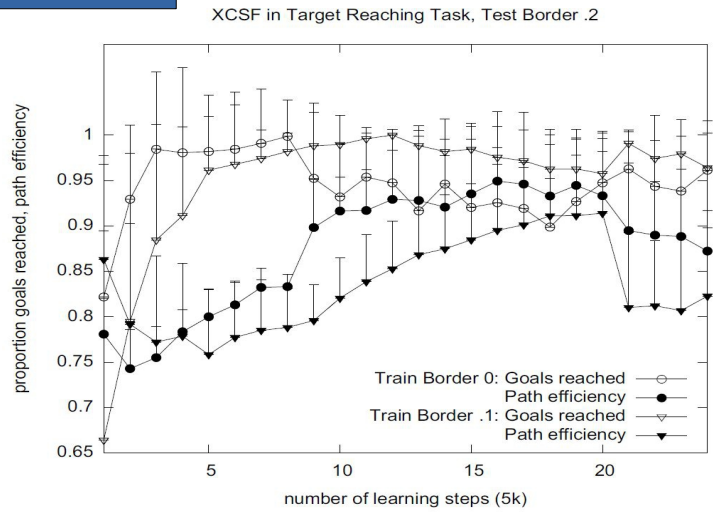
4) Cognitive Arm Control
Arm Control Task

TECHNISCHE UNIVERSITÄT DARMSTADT

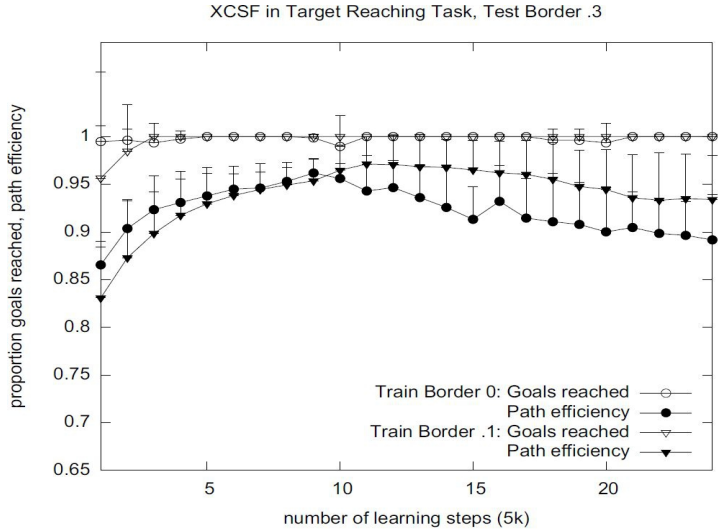
- The success rate increases during learning
- Path efficiency ([Euclidean distance hand to goal]/[actual distance the hand moved to the goal])

  - improves but plateaus after about 60k l. it.

- ✖ After compaction (100k it.) performance decrease
  - XCSF over-specializes its population for the generation of accurate forward predictions so that representation becomes over-generalized for inverse computation
- Especially the inner area of the arm is controlled reliably and effectively.
- ✖ Some of the generated trajectories end in a circling behavior around the goal location.
  - May be due to the simple inverse computation approach, where iteratively each angular movement was set to zero.



(a) Wider area test



(b) Narrower area test

4) Cognitive Arm Control
alternative, non-deterministic approach

TECHNISCHE UNIVERSITÄT DARMSTADT

- Rao and Ballard showed that a neural network can iteratively approximate a linear function accurately-enough to develop emergent visual cortical features (s.a. edge detection)

  - The forward linear equation is transposed determining an approximate inverse.

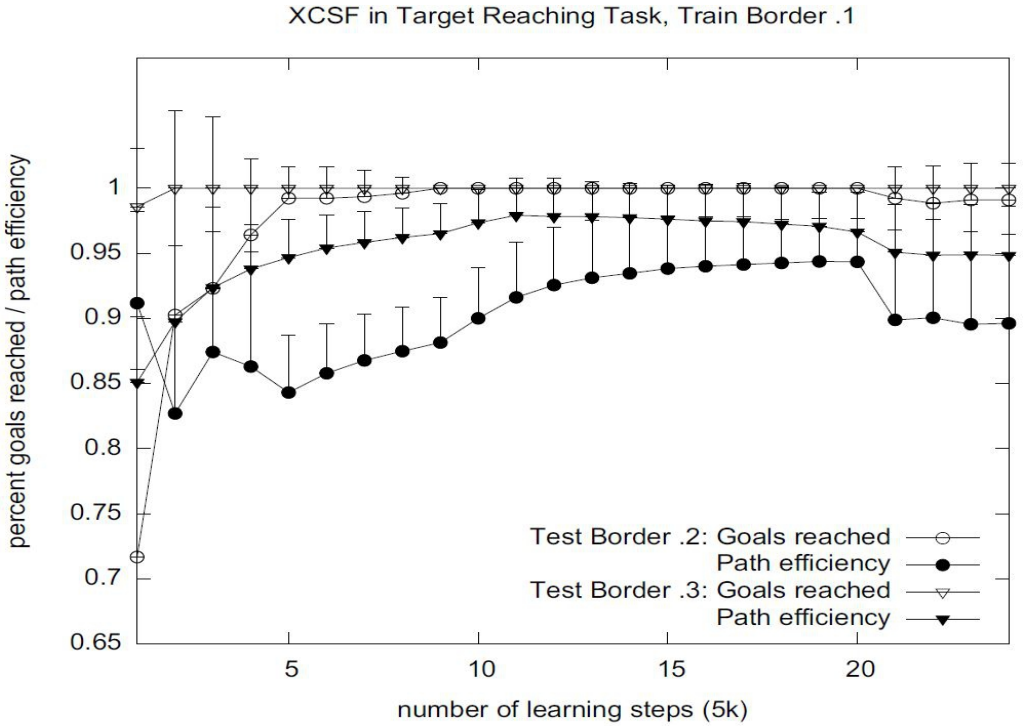
- The resulting activity is forwarded back to the input, resulting in an error between input activity and predicted activity.

- The difference is used to propagate the next inverse activity.

4) Cognitive Arm Control

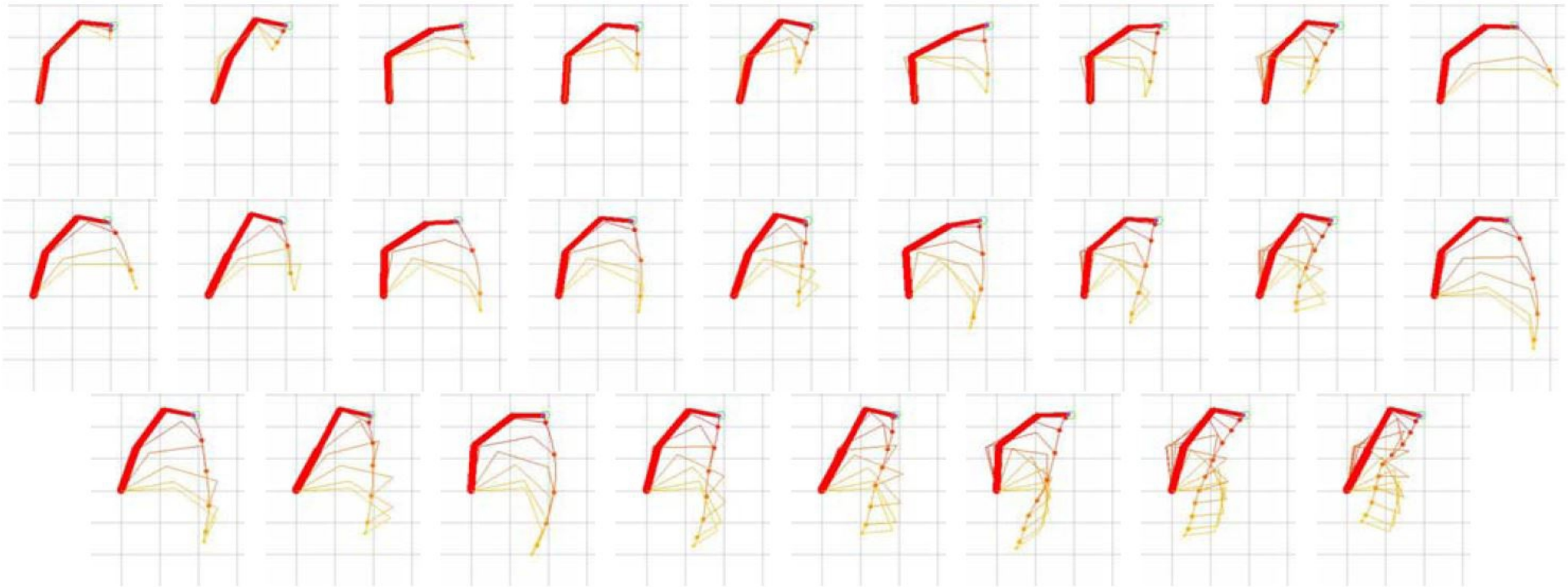
TECHNISCHE UNIVERSITÄT DARMSTADT

- The approximate inverse computation improves the success rate and the path efficiency (100% success before compaction)
- The over-fitted conditions for prediction yield a decreased inverse accuracy.
  - Path efficiency stalls at 95% and decreases after compaction



4) Cognitive Arm Control
resulting arm trajectories (outer goal)

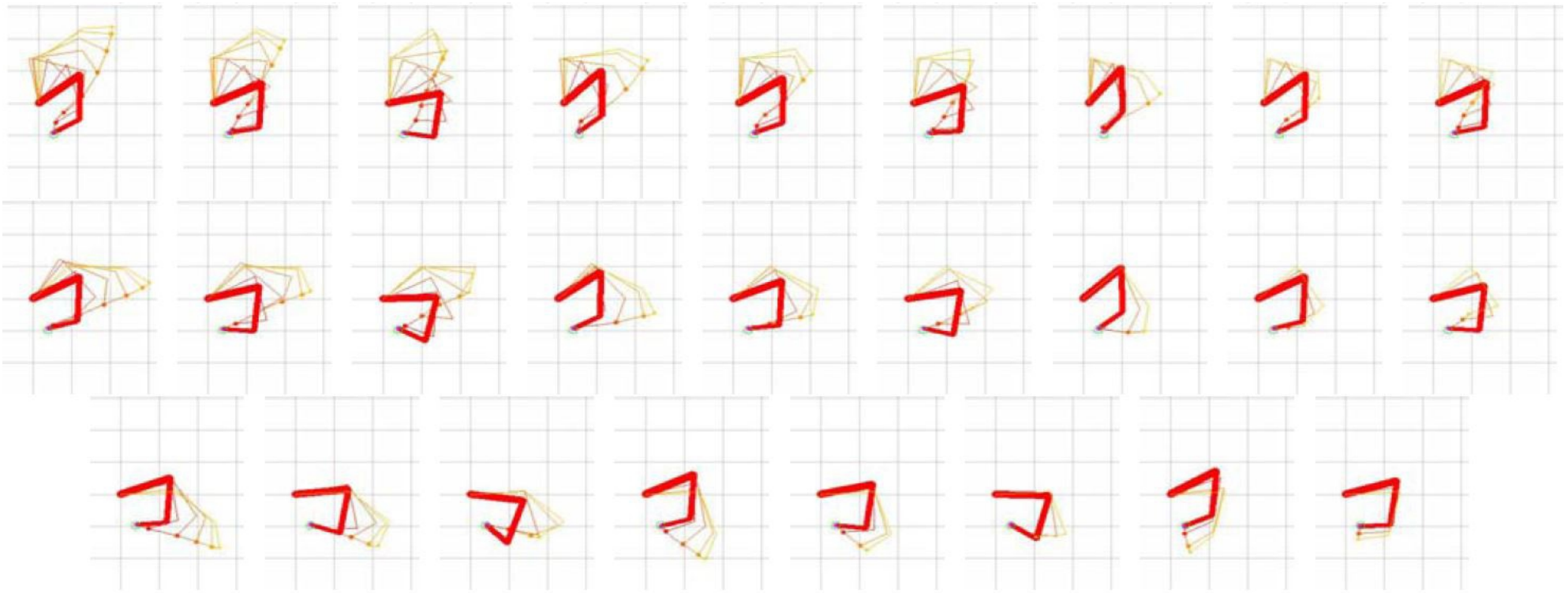
TECHNISCHE UNIVERSITÄT DARMSTADT



- Generated after 60k learning iterations; train border 0.1; test border 0.3; approximate inverse computation;
- Hand movement are rather direct and reach the goal location to 100%
- (final arm posture is drawn thicker)

4) Cognitive Arm Control
resulting arm trajectories (inner goal)

TECHNISCHE UNIVERSITÄT DARMSTADT



- Generated after 60k learning iterations; train border 0.1; test border 0.3; approximate inverse computation
- Hand movement are rather direct and reach the goal location to 100%
- (final arm posture is drawn thicker)

Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

0) Abstract View

1) Introduction

2) XCS/XCSF: A Short Overview

3) Separated Inputs

4) Cognitive Arm Control

5) Summary and Conclusion

6) Final Discussion

3) Summary and Conclusion

TECHNISCHE UNIVERSITÄT DARMSTADT

- **XCSF can be applied as a:**

  - forward predictive system approximating functions

  - context-dependent processing system

- **Modified XCSF:**

  - capable of processing context input separately from predictive input

  - distribute its classifier very effectively in such tasks

- **XCSF applied for the control of a three degree of freedom arm:**

  - generate accurate forward predictions

  - forward predictions can be inversely used to invoke directional arm control

- **Given a current desired direction to a goal, XCSF can invoke suitable motor commands that guide the arm to the goal on a approximately optimal path.**

Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

0) Abstract View

1) Introduction

2) XCS/XCSF: A Short Overview

3) Separated Inputs

4) Cognitive Arm Control

5) Summary and Conclusion

6) Final Discussion

6) Final Discussion

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Current computational models of human behavioural control** rely extensively on a representation of perception.

  - **This representation** requires perceptual input grouped into several more or less discrete units, which are of relevance for many aspects of behaviour control.

- On neural level, **parietal and motor cortical areas** encode sensory information in population codes, where each neuron represents a range of possible predictions.

- On a higher level, also the **integration of multiple skills** requires a partitioning of the behavioral context in order to enable stable inverse model learning and context-dependent control.

- The **partitioning of sensory spaces**, which underlies these models, is mostly not evolved but predefined or generated independent of the interaction between sensory input and motor activity.

- **Spatial coverages** are developed highly pro-actively, optimizing the spatial partitions for the individual behavioural sensorimotor demands.

6) Final Discussion

TECHNISCHE UNIVERSITÄT DARMSTADT

- XCSF is a self-organizing system, which clusters a problem space for the purpose of prediction.
- The proposed XCSF setup evolves a bodyspace partitioning for the prediction of accurate motor activity effects on sensory input, that is, for accurate sensorimotor learning.
- Thus, XCSF provides a platform that can evolve population encodings for the purpose of prediction and control.

- **XCSF can be applied as a complex cognitive control system**, which develops sensorimotor structures for scratch based on initially simple environment interactions.

Thank you for listening

TECHNISCHE UNIVERSITÄT DARMSTADT

Any Questions?