AN ADAPTIVE MULTIOBJECTIVE EVOLUTIONARY APPROACH TO OPTIMIZE ARTMAP NEURAL NETWORKS

Date: *November 23rd , 2010*

Michael Georgiopoulos
Department of EECS, University of Central Florida

MACHINE LEARNING LAB

Outline

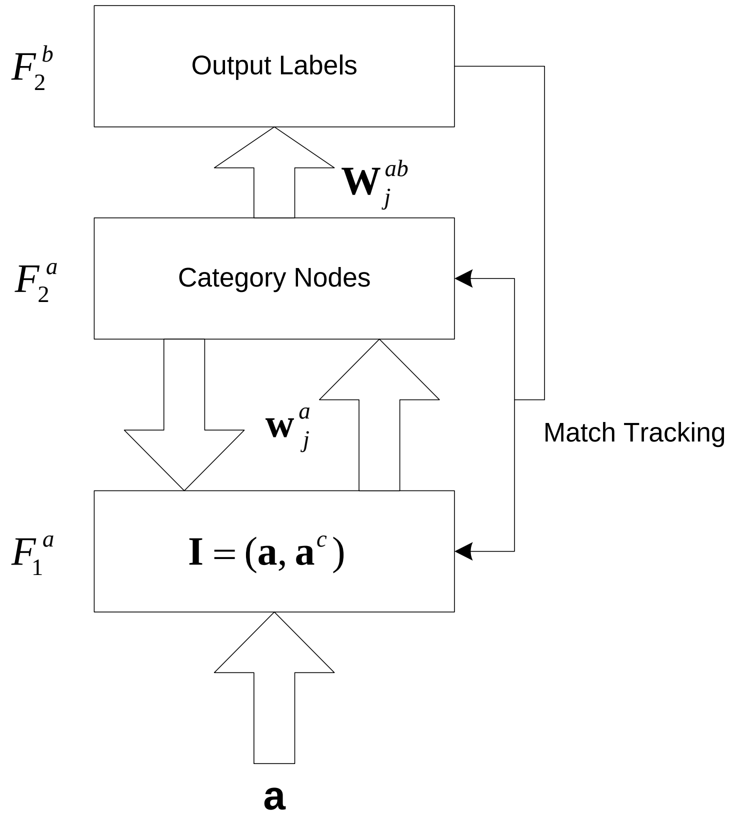
- ART Architectures
- Motivation for Genetic ART
- Multi-Objective Genetic ART (MO-GART)
- Fitness Function and Selection
- Confidence Factor
- Genetic Operators (Pruning, Mutation, Cross-Over)
- Experiments
- Results and Comparisons
- Summary

ART Neural Networks

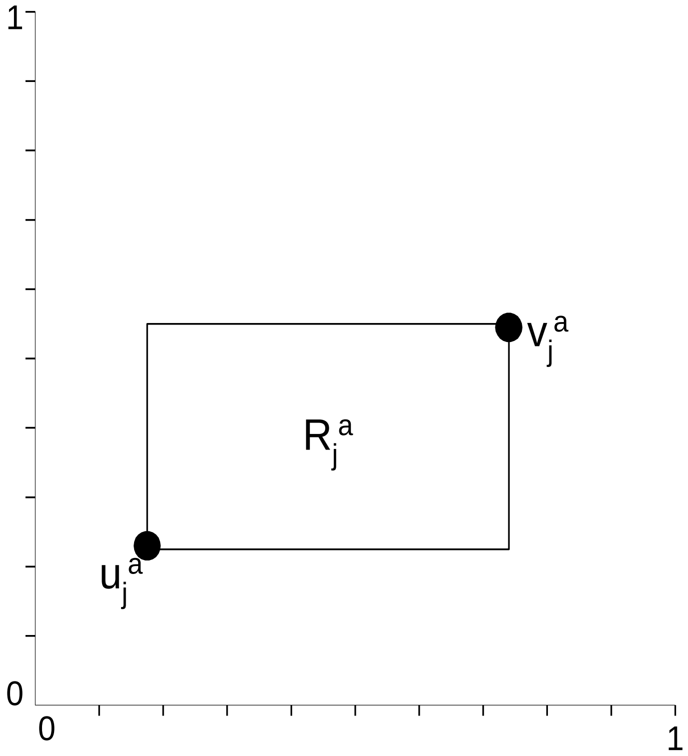
- The Adaptive Resonance Theory (ART) was developed by (Grossberg, 1976).
- Fuzzy ARTMAP introduced in 1992 (Carpenter et. al., 1992).
- A number of variations were introduced:
  - Gaussian ARTMAP (Williamson, 1996)
  - Ellipsoidal ARTMAP (Anagnostopoulos, 2001)
- Advantages:
  - Able to handle complex classification problems
  - Converge quickly
  - Able to recognize novelty
  - Answers can be explained with relative ease

Fuzzy ARTMAP

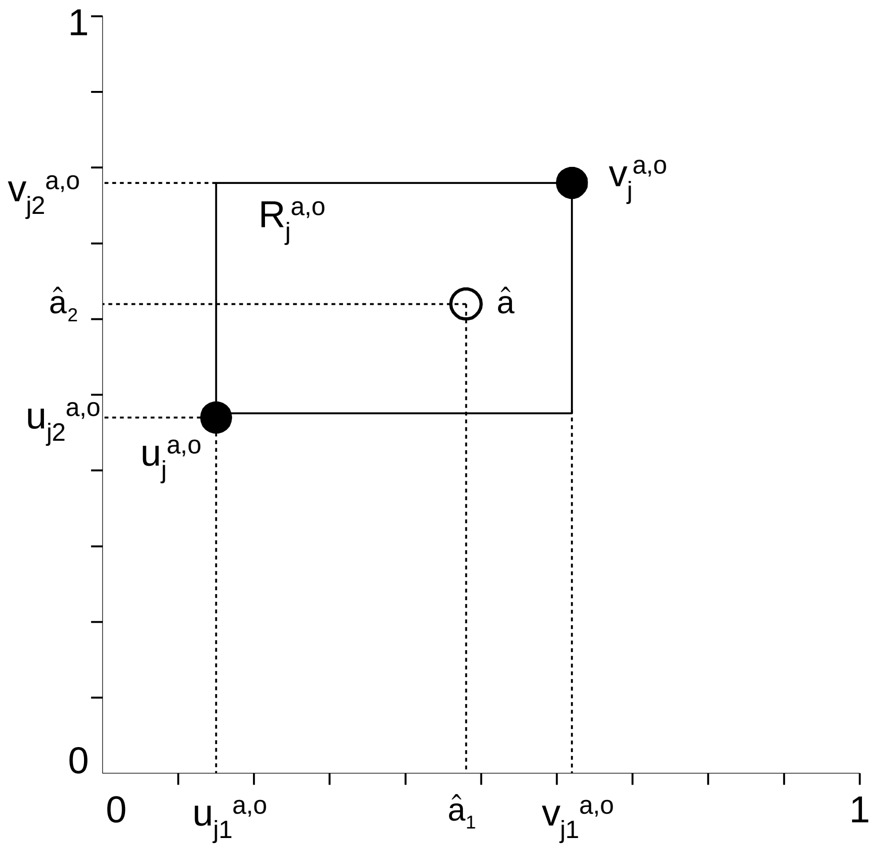
- Input patterns are compressed to form regions or categories in the input space
- Learning or training is accomplished using examples
- Each category is mapped to a class label
- GAM (Williamson, 1996) and EAM (Anagnostopoulos, 2001) have similar architectures, but the category structure differs



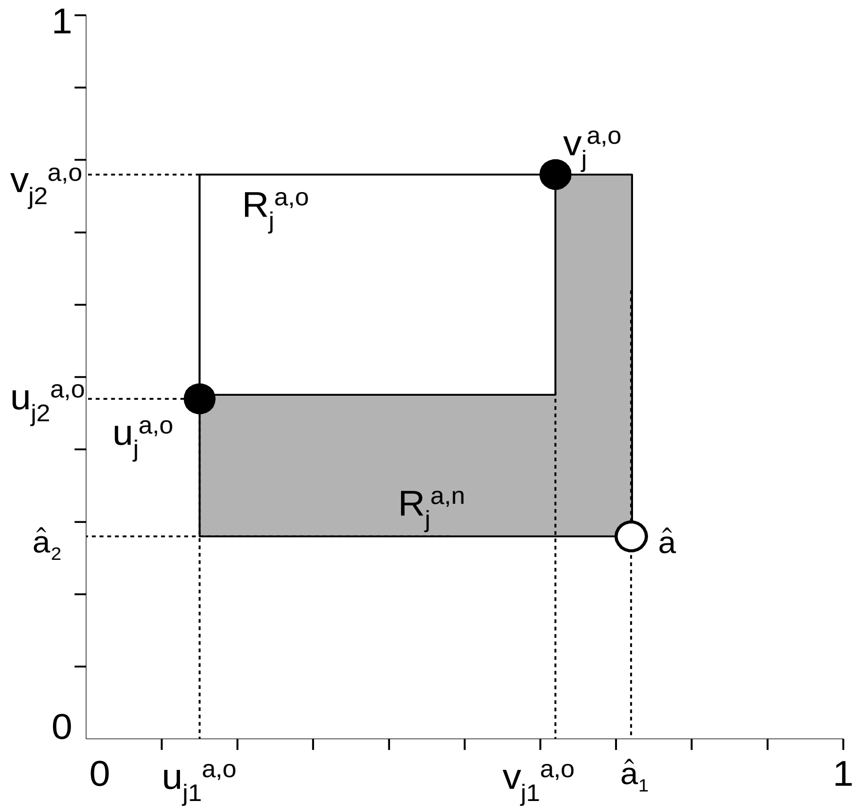
Learning in Fuzzy ARTMAP



Learning in Fuzzy ARTMAP



Learning in Fuzzy ARTMAP



Fuzzy ARTMAP Equations

- Category Choice Function

$$T_j^a(I) = \frac{M_a - dis\,(I, R_j^a) - s(R_j^a)}{\beta_a + M_a - s(R_j^a)}$$

- Category Match Function

$$s(R_j^{a,n}) \leq M_a(1 - \rho_a)$$

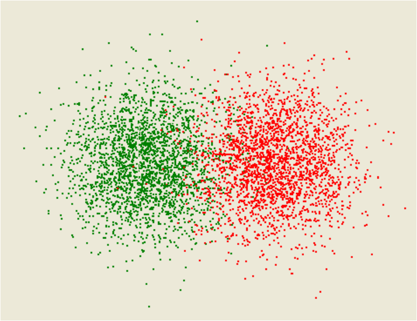
- Weight Update Function

$$s(R_j^{a,n}) = s(R_j^a) + dis\,(I, R_j^a)$$

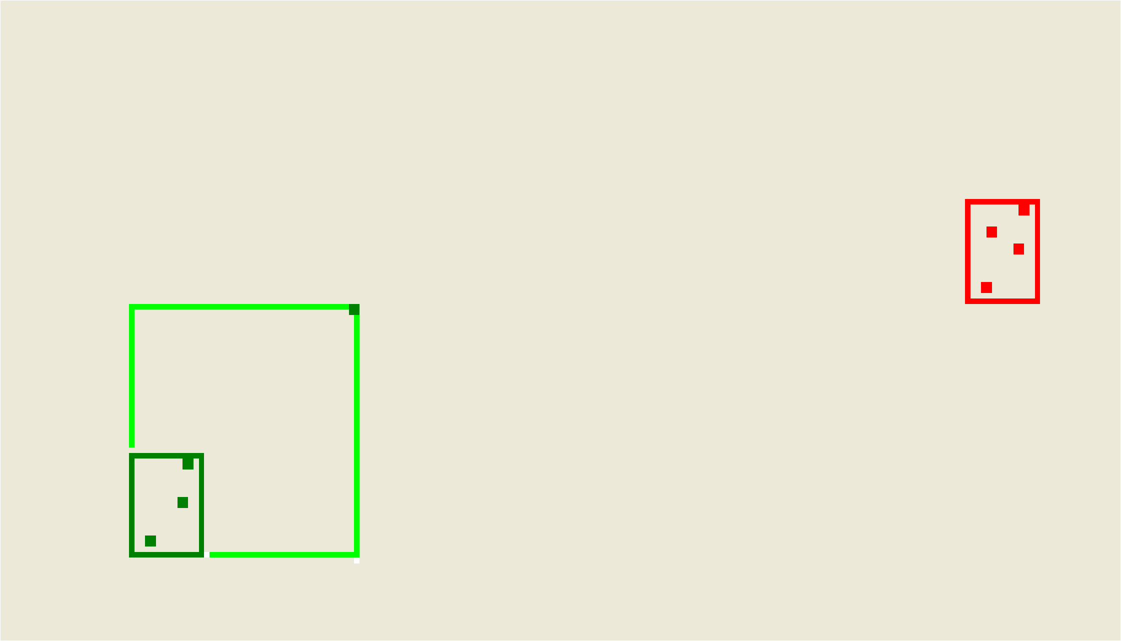
Fuzzy ARTMAP Parameters

- **Choice Parameter:** Determines the value of the bottom-up inputs at the input category representation layer.
- **Vigilance Parameter:** Determines whether coarse of fine clusters are going to be formed in the input category representation layer.
- **Order of Input pattern Presentation:** Determines the Order according to which the input training data are going to be presented to Fuzzy ARTMAP

ART Learning

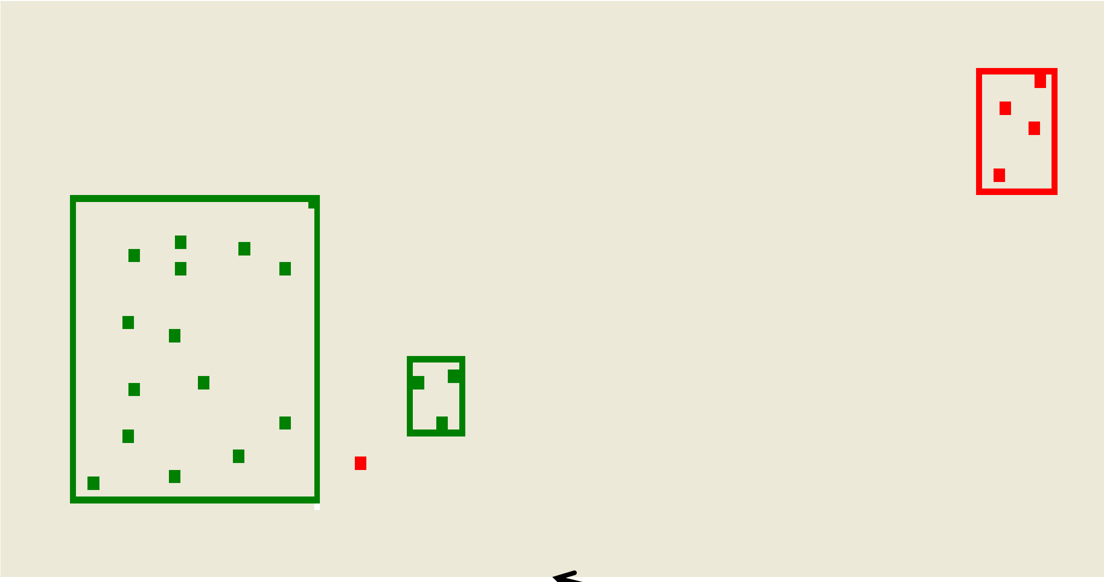


Training Patterns



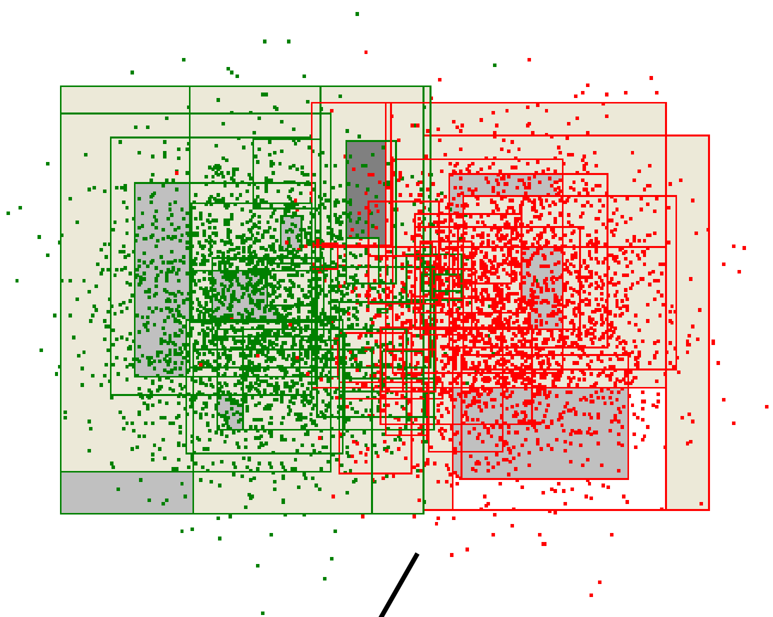
Box Creation in ART

Noisy/Overlapping Data

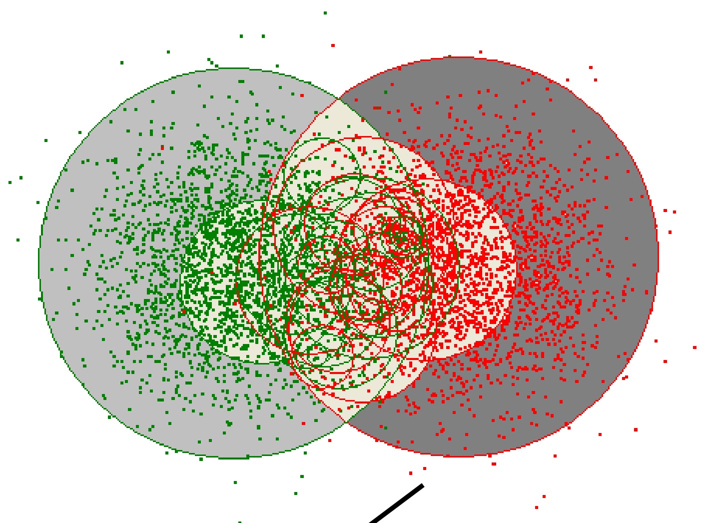


Noise / Overlapping patterns cause creation of unnecessary categories

Effect of Noisy/Overlapping Data



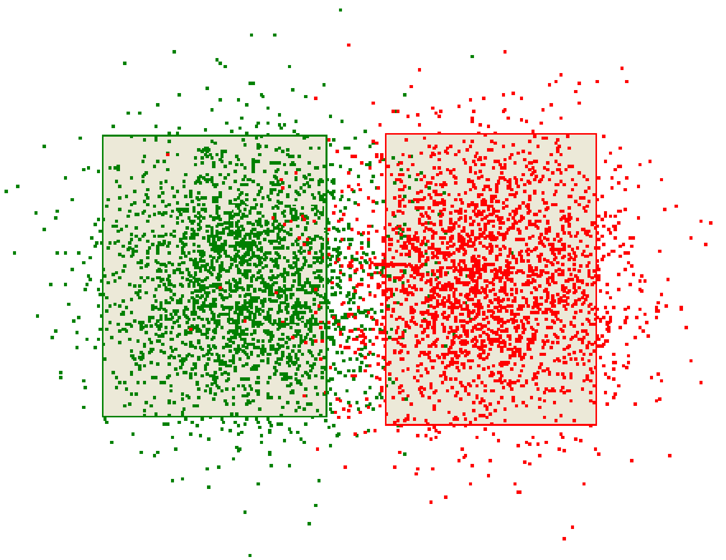
57 Categories formed



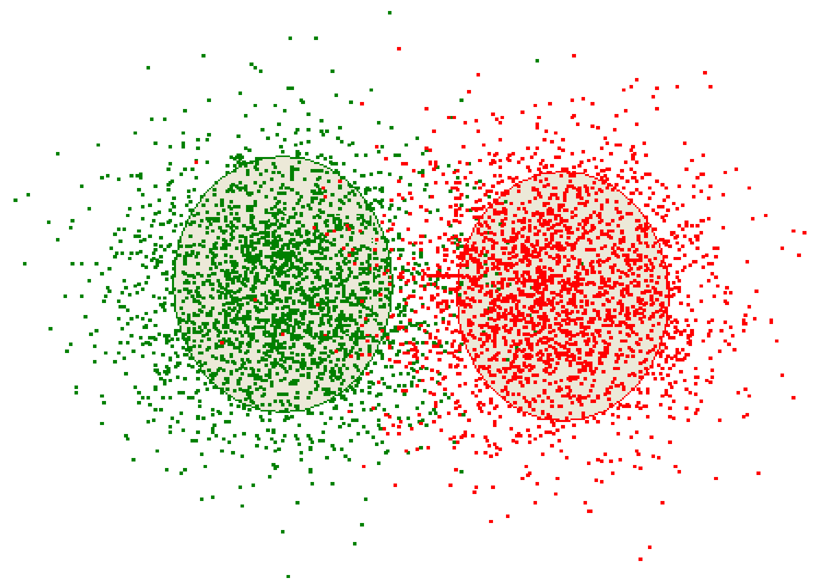
39 Categories formed

Ideal ART Classifier

Fuzzy ARTMAP



Ellipsoidal ARTMAP



Motivation for Work

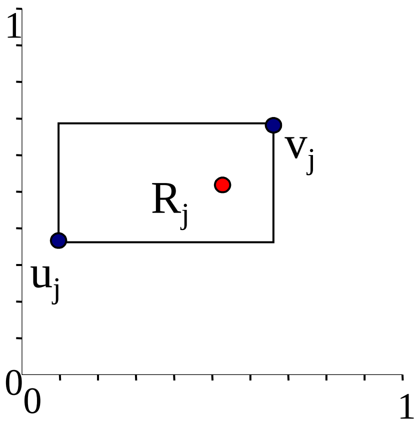
- There are two objectives for this research:
  - Design an ART classifier that has a small size and is of good generalization, thus addressing the category proliferation problem
  - Design an ART classifier system that does not require the user to experiment with network parameters

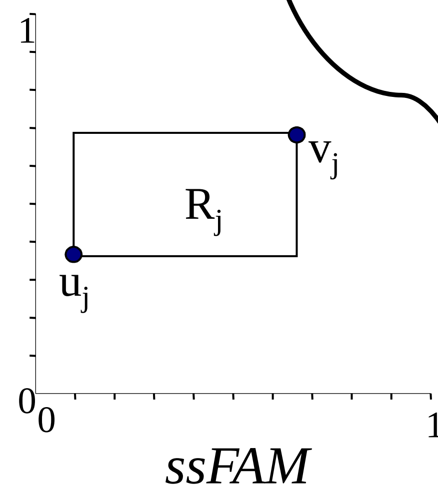
Category Proliferation : Solutions

- Eliminate match tracking mechanism such as in PROBART (Marriott et. al., 1995), micro-ARTMAP introduced by (Gomez-Sanchez et. al., 2000) and safe micro-ARTMAP (Gomez-Sanchez et. al. 2001)
- Cross-validation: Stop learning when over-training is observed on a validation set (Koufakou et. al., 2001)

- Semi-supervised learning: Allow categories to encode patterns that are not mapped to the same label (Anagnostopoulos et. al., 2003)
- GART: Single objective generic optimization of ART architectures (Al-Daraiseh, et al., 2006)

Functionality of ssFAM

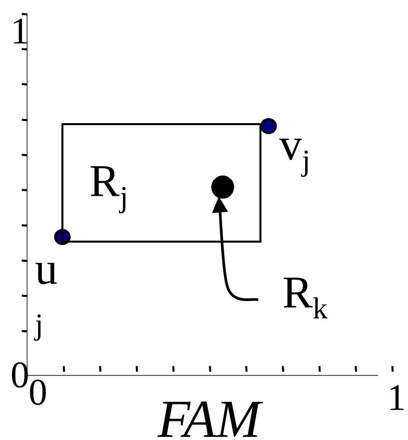




Red Pattern will be absorbed by the shown category, if the category passes prediction test

ssFAM has one more parameter

Allowed Prediction Error



Red Pattern will create a point category in FAM.

Choosing Network Parameters

- There are some guidelines of how to choose the ART network parameters, such as choice parameter and vigilance parameter
- Unfortunately there are no good guidelines of how to choose the order of training pattern presentation
- Here comes ***Genetic ART***

Genetically Engineered ART

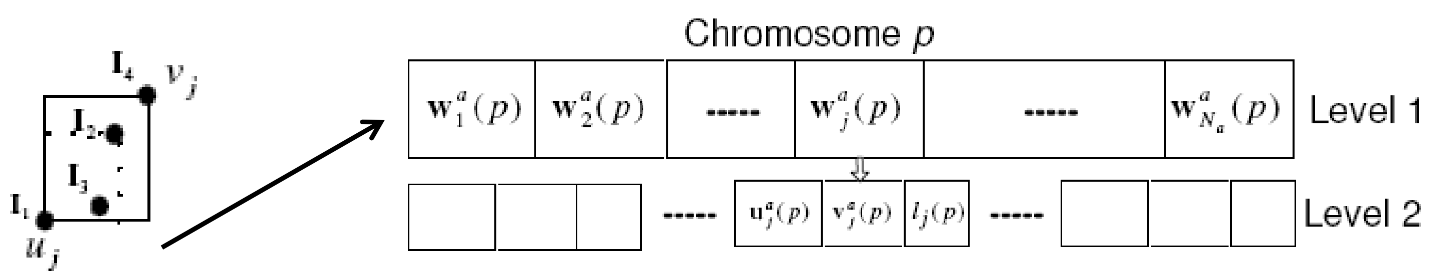
- Chromosomes encode categories belonging to an ART NN
- A population of ART NN is initially trained, and then evolved for a number of generations
- GA is used to evolve the structure and weights of ART NN's
  - Minimize complexity
  - Maximize accuracy

Advantages of Genetic ART

- Competitive results in terms of accuracy and size, compared to other ART architectures (to be seen)
- Genetic optimization of ARTMAP NN may achieve performance that might not be attainable by original ARTMAP training rules
  - Genetic operators allow mixing NNs (crossover), reducing the size (deletion), and altering categories (mutation)
- Genetic optimization provides opportunity for automated model selection
  - Avoiding NN parameter tweaking (to be seen)
  - Minimizing interaction with human decision maker (to be seen)

Genetically Optimized ART

- GA is used to evolve ART NN's
  - Topology: Number of categories
  - Weights: Category size and location
- Two objectives:
  - Minimize error rate
  - Minimize size (number of categories)
- Chromosomes encode categories belonging to a network



MACHINE LEARNING LAB

From GART to MO-GART

- Adaptive genetic algorithm
  - Better utilizes the information gained from the testing of solutions during the genetic search
  - Improves effectiveness of genetic operators
  - Improves efficiency of the algorithm
  - Eliminates pre-specification of GA parameters

- Controlling the number of validation patterns used in the evolution
  - Utilizes the ability of genetic algorithm to operate in noisy environments
  - Improves the convergence speed

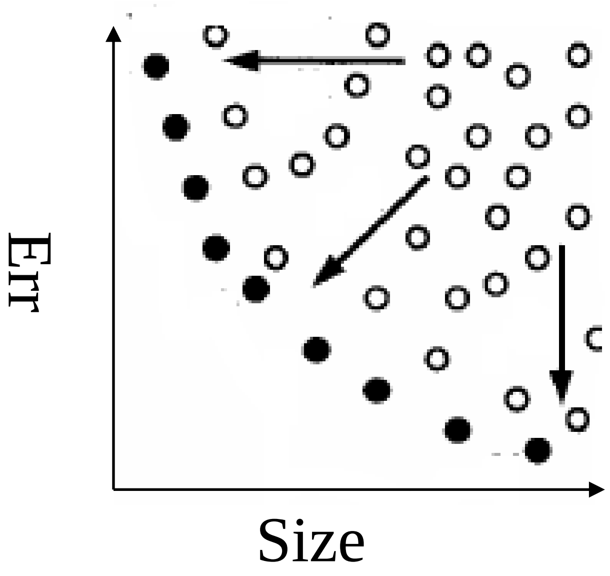
- Multi-objective evolution
  - Better way to address a two-objective optimization problem
  - Finds better solutions
  - Utilizes the fact that GAs are population based, and can thus return multiple solutions in one run

MO-GART

$P(0) \leftarrow$ Generate-Initial-Population();
$A(0) \leftarrow$ Initialize-Empty-Archive();
for $t \leftarrow 1$ **to** Gen_{max} **do**
    Evaluation();
    Update-Archive($P(t)$, $A(t)$);
    if *stopping criteria met* **then** exit for;
    $P'(t) \leftarrow$ Selection($P(t)$, $A(t)$);
    $P(t) \leftarrow$ Reproduction($P'(t)$);
end
return $A(t)$;

MO-GART: Multi-objective

- It is often desirable to find all tradeoff solutions as they provide alternative solutions to the problem
- Availability of what is achievable allows the decision maker to choose appropriate compromise solutions to the problem
- GAs are population based search algorithms, and therefore can be used to find the solutions on the tradeoff surface in a single run



MO-GART: Adaptive Evolution

- Deterministic
  - Without feedback about the performance or quality of solution achieved
  - According to a schedule
- Adaptive
  - Based on feedback
  - Constructs a relationship between feedback signal and parameter value
- Self-Adaptive
  - GA parameters are encoded and evolved as part of the problem

MO-GART: Adaptive Evolution

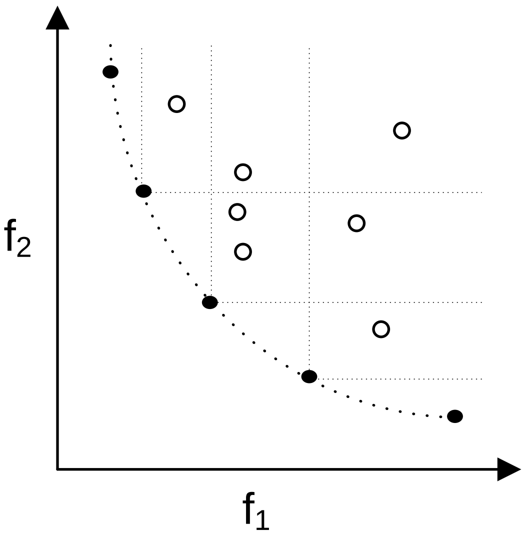
- Population level
  - Adaptation of global parameters that are applied to all individuals
- Individual level
  - For each individual separately
  - E.g., mutation rate for every individual
- Component level
  - Mutation rate for a component within each individual

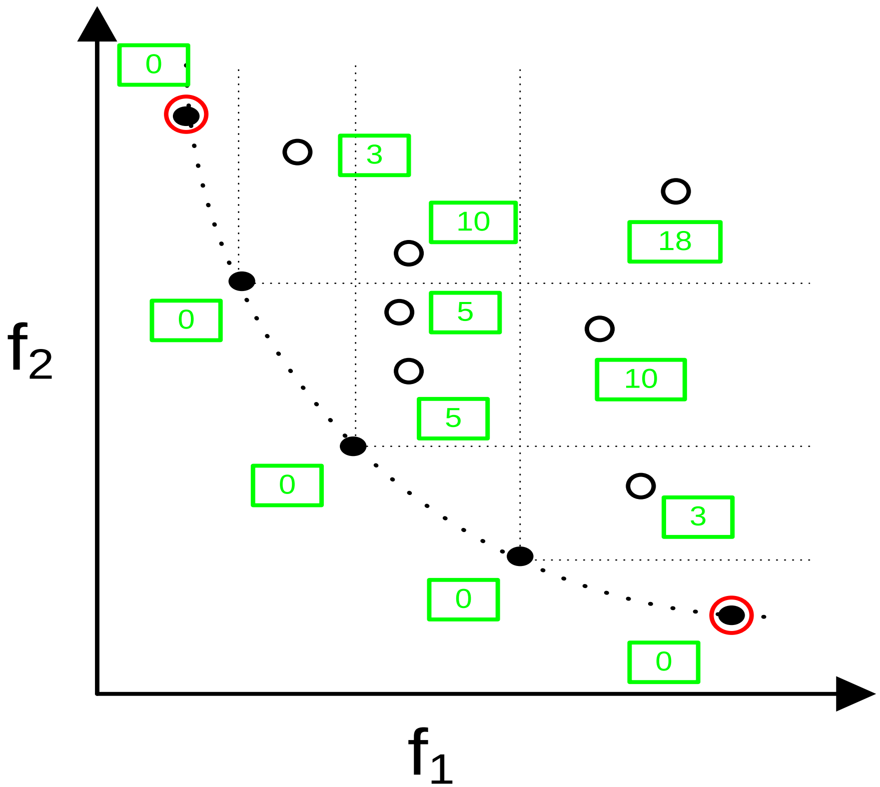
Fitness Assignment/Selection

- The fitness function is defined as the sum of the raw fitness (*R(x)*) and another term that penalizes solutions that are crowded by other solutions (Zietzler, 2001; SPEA2)
- *R(x)* is equal to the sum of the strengths of all its dominators
- The strength of an individual is equal to number of solutions it dominates

$$Fit(x) = R(x) + \frac{1}{2 + dis_k}$$

Fitness Assignment/Selection





Credit Assignment

- For each network, and for each the category confidence factor is calculated as follows:

Confidence of Category

$$CF_j^k(p) = 0.5 \cdot A_j^k(p) + 0.5 \cdot S_j^k(p)$$

- where,

Accuracy of Category

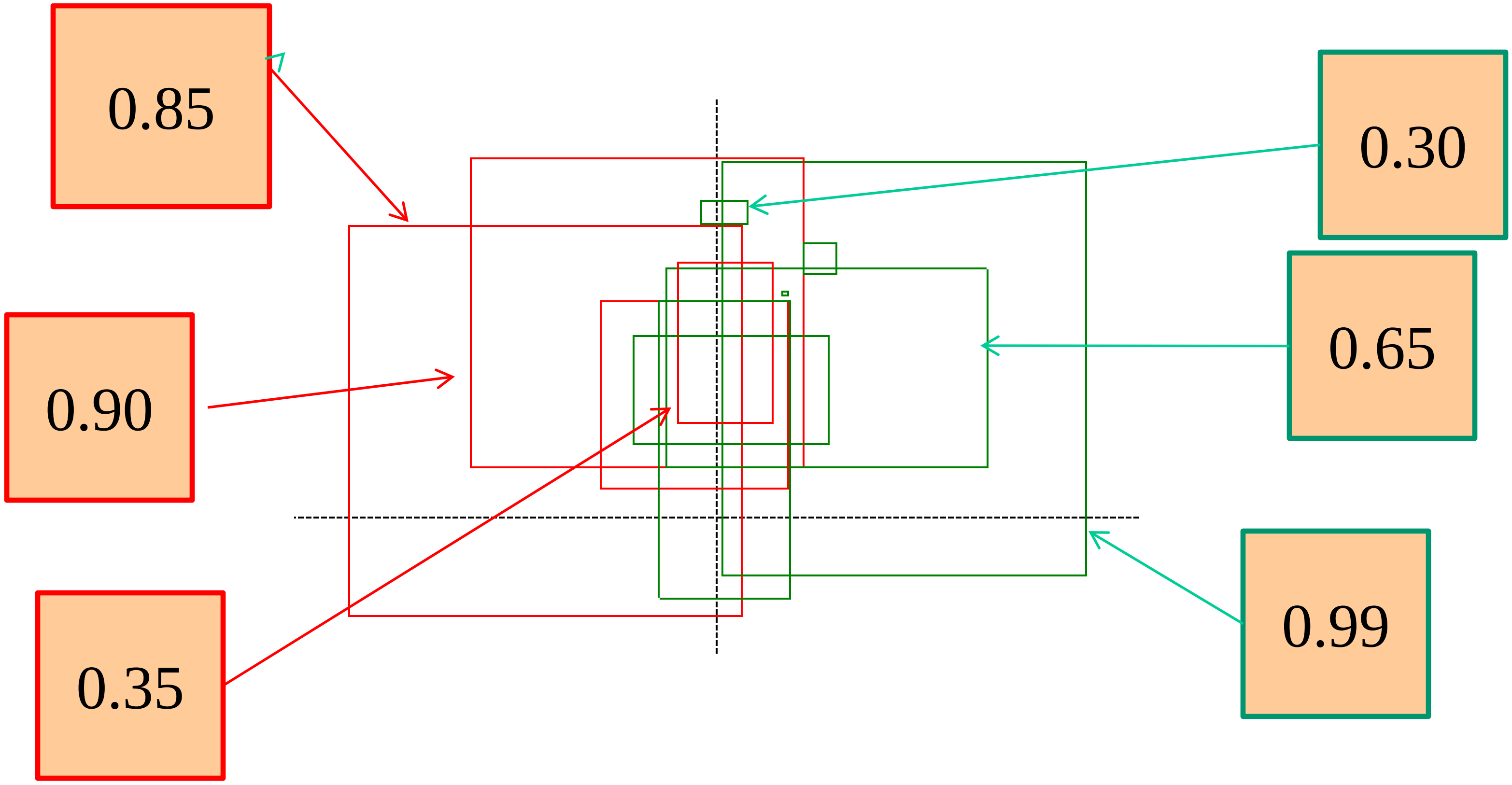
$$A_j^k(p) = \frac{P_j^k(p)/C_j^k(p)}{\max_j P_j^k(p)/C_j^k(p)}$$

- and,

Selectivity of Category

$$S_j^k(p) = \frac{C_j^k(p)}{\max_j C_j^k(p)}$$

CF Example Values



MO-GART Pruning

- Adaptive, probabilistic pruning
- Based on the confidence factor for each category
- Probability of elimination is inversely proportional to a category's CF:

$$PDel_j^k(p) = (1 - CF_j^k(p))$$

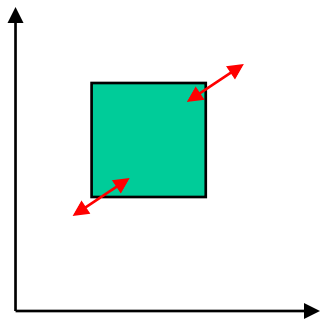
- The rate of pruning is automatically adjusted; does not need user to specify as parameter

MO-GART Mutation

- Automatically adjusted mutation severity:

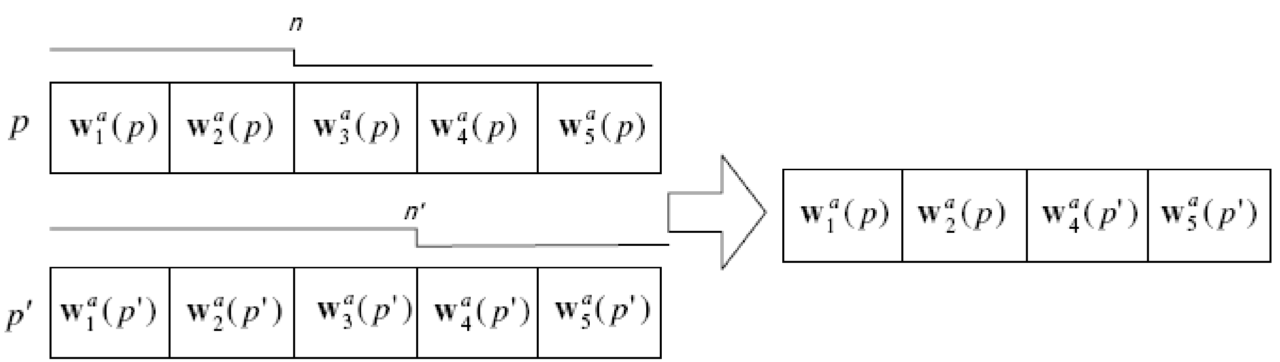
$$SF_j^k(p) = 0.05 \cdot (1 - CF_j^k(p))$$

- Therefore the mutation severity is automatically adapted based on the performance of the category



MO-GART Cross-Over

- Combine selected parents to form the new chromosomes – one point crossover



MO-GART (once more)

$P(0) \leftarrow$ Generate-Initial-Population();
$A(0) \leftarrow$ Initialize-Empty-Archive();
for $t \leftarrow 1$ **to** Gen_{max} **do**
    Evaluation();
    Update-Archive($P(t)$, $A(t)$);
    if *stopping criteria met* **then** exit for;
    $P'(t) \leftarrow$ Selection($P(t)$, $A(t)$);
    $P(t) \leftarrow$ Reproduction($P'(t)$);
end
return $A(t)$;

Experiments

- We experimented with a number of datasets
- For each dataset we had a training, a validation and a test set
- We used the training set to design the model, the validation set to choose the network parameters, and the test set to report the network's performance
- We experimented with MO-GART, GART, ssART, SVM, and CART
- Experiments were fair: Used same datasets, and had the code implemented for all algorithms

Measures of Comparison

- Accuracy of the best performing network (highest PCC)
- Size of the best performing network
- Time to produce the best performing network
- Metric C

$$C(A,B)=\frac{|\, b\in B : \exists\, a\in A,\, a\succ b \,|}{|\, B \,|}$$

  - *C(A, B)* close to 1…most members of *B* are dominated by a member of *A*
  - *C(A, B)* close to 0 … very few members of *B* are dominated by a member of *A*

Datasets

Dataset	Tra	Val	Tes	Attr	Class	Major
Ci/Sq	2000	5000	3000	2	2	50%
G4C-25	500	5000	5000	2	4	25%
G6-15	504	5004	5004	2	6	16.7%
Iris	500	4800	4800	2	2	50%
Page	500	2486	2487	10	5	89.8%
Pdigits	4494	3000	3498	16	10	10%
Sat	2000	2436	2000	36	6	34.2%
Seg	800	810	700	19	7	14.2%
Wave	1000	2000	2000	21	3	33.3%
Abalone	501	1838	1838	7	3	33.3%
Odigits	1823	2000	1797	64	10	10%

MO-GART vs ssFAM

Dataset	MO-GFAM PCC/Size	ssFAM PCC/Size
1Ci/Sq (2000)	97.97 /31	98.10/78
G4C-25 (500)	76.00/4	74.22/4
G6C-15 (504)	84.59/6	82.49/9
Iris (500)	95.19/2	94.56/2
Page (500)	96.45/5	94.77/6
Pendigits (4494)	98.27/271	97.14/66
Sat (2000)	89.12/175	84.20/51
Segmentation (800)	95.43/25	94.14/32
Waveform (1000)	86.30/3	75.65/16
Abalone (501)	**66.50/5**	**56.89/34**
Optidigits (1823)	98.05/272	87.20/52
Average PCC	89.44	85.40

MO-GART vs ssFAM

- The time required to produce the ssFAM solutions ended up being one to two orders of magnitude slower
- The *C (MO-GFAM, ss-FAM)* values are all larger than 0.5, and most of them close to 1 (meaning that many ss-FAM solutions are dominated by a MO-GFAM solution)
- The *C (ss-FAM, MO-GFAM)* values are all smaller than 0.5 and most of them close to zero (meaning that very few MO-GFAM solutions are dominated by an ss-FAM solution)
- The ss-FAM performances (PCC) exhibited high variability (more than 10%, a number of times)
- The MO-GFAM performances (PCC) exhibited low variability (less than 0.5% in most instances)

MO-GART vs ssEAM

Dataset	MO-GEAM PCC/Size	ssEAM PCC/Size
1Ci/Sq (2000)	97.76 /2	97.40/99
G4C-25 (500)	75.54/4	73.90/4
G6C-15 (504)	84.69/6	83.23/24
Iris (500)	95.24/2	94.65/2
Page (500)	96.40/5	94.44/24
Pendigits (4494)	98.90/331	96.60/179
Sat (2000)	88.34/198	85.50/141
Segmentation (800)	93.86/52	91.57/83
Waveform (1000)	86.35/5	79.80/12
Abalone (501)	66.40/6	57.42/5
Optidigits (1823)	**98.40/418**	**91.93/122**
Average PCC	89.44	86.04

MO-GART vs ssEAM

- The time required to produce the ssEAM solutions ended up being one to two orders of magnitude slower
- The *C (MO-GEAM, ss-EAM)* values are all larger than 0.5, and most of them close to 1
- The *C(ss-EAM, MO-GEAM)* values are all smaller than 0.5 and most of them close to zero
- The ss-EAM performances (PCC) exhibited high variability (more than 10%, at times)
- The MO-GEAM performances (PCC) exhibited low variability (less than 0.5% in most instances)

MO-GART vs ssGAM

Dataset	MO-GGAM PCC/Size	ssGAM PCC/Size
1Ci/Sq (2000)	99.80 /2	94.63/26
G4C-25 (500)	75.92/4	74.84/23
G6C-15 (504)	85.17/6	85.07/20
Iris (500)	94.90/2	95.21/7
Page (500)	96.38/5	94.52/7
Pendigits (4494)	98.10/88	97.43/87
Sat (2000)	88.75/106	87.00/81
Segmentation (800)	92.59/13	91.29/31
Waveform (1000)	87.15/4	85.35/11
Abalone (501)	67.30/5	57.19/30
Optidigits (1823)	97.15/161	92.21/55
Average PCC	89.38	86.79

MO-GART vs ssGAM

- The time required to produce the ssGAM solutions ended up being one to two orders of magnitude slower
- The *C (MO-GGAM, ss-GAM)* values are all larger than 0.5, and most of them close to 1
- The *C(ss-GAM, MO-GGAM)* values are all smaller than 0.5 and most of them close to zero
- The ss-GAM performances (PCC) exhibited high variability (more than 10% in a number of times)
- The MO-GGAM performances (PCC) exhibited low variability (less than 0.5% in most instances)

MO-GART vs SVM

Dataset	MO-GFAM PCC/Size	MO-GEAM PCC/Size	MO-GGAM PCC/Size	SVM PCC/Size
1Ci/Sq (2000)	97.97 /31	97.76 /2	99.80 /2	99.67/88
G4C-25 (500)	76.00/4	75.54/4	75.92/4	75.24/277
G6C-15 (504)	84.59/6	84.69/6	85.17/6	84.99/504
Iris (500)	95.19/2	95.24/2	94.90/2	95.04/79
Page (500)	96.45/5	96.40/5	96.38/5	95.30/150
Pendigits (4494)	98.27/271	98.90/331	98.10/88	99.54/929
Sat (2000)	89.12/175	88.34/198	88.75/106	90.25/1081
Seg (800)	**95.43/25**	**93.86/52**	**92.59/13**	**97.29/230**
Wav (1000)	86.30/3	86.35/5	87.15/4	87.45/574
Abalone (501)	**66.50/5**	**66.40/6**	**67.30/5**	**61.66/337**
Opti (1823)	**98.05/272**	**98.40/418**	**97.15/161**	**97.22/673**
Average PCC	89.44	89.44	89.38	89.41

MO-GART vs SVM

- The time required to produce the MO-GART solutions ended up being faster than the time required to produce the SVM solutions
- Overall, SVM performs better (PCC) than MO-GART but not statistically significantly better, except in one case
- The SVM performances (PCC) exhibited high variability (more than 10% in a number of times)
- The MO-GART performances (PCC) exhibited low variability (less than 0.5% in most instances)

MO-GART vs CART

Dataset	MO-GFAM PCC/Size	MO-GEAM PCC/Size	MO-GGAM PCC/Size	CART PCC/Size
1Ci/Sq (2000)	97.97 /31	97.76 /2	99.80 /2	97.57/28
G4C-25 (500)	76.00/4	75.54/4	75.92/4	73.50/4
G6C-15 (504)	84.59/6	84.69/6	85.17/6	80.42/6
Iris (500)	95.19/2	95.24/2	94.90/2	94.02/4
Page (500)	96.45/5	96.40/5	96.38/5	93.84/7
Pendigits (4494)	98.27/271	98.90/331	98.10/88	93.37/109
Sat (2000)	89.12/175	88.34/198	88.75/106	84.35/22
Seg (800)	95.43/25	93.86/52	92.59/13	93.43/17
Wave (1000)	86.30/3	86.35/5	87.15/4	75.20/14
Abalone (501)	**66.50/5**	**66.40/6**	**67.30/5**	**61.18/17**
Optidigits (1823)	**98.05/272**	**98.40/418**	**97.15/161**	**82.42/88**
Average PCC	89.44	89.44	89.38	84.48

MO-GART vs CART

- The time required to produce the MO-GART solutions ended up being orders of magnitude slower than the time require to produce the CART solutions
- Overall, MO-GART performs better (PCC) than CART, and in most instances statistically significantly better

Summary

- A new family of ART classifiers is introduced that
  - Has good generalization
  - Is of small size
  - Is efficient in terms of training time
  - Does not require tweaking of the network parameters
- Compared to previously introduced ART architectures and shown to be superior
- Shown to be competitive against other popular classifiers, such as SVM and CART