

Machine Learning within an
Unbalanced Distributed Environment
Research Notes: Model Development

Hans J. Pruesser

Centre for Computational Intelligence,
DeMontfort University, Leicester, UK

hansp@dmu.ac.uk

Autumn/Winter 2006

Contents

1	Introduction	3
1.1	Research Aims Revisited	3
1.2	Thesis Working Title	3
2	Mobile Intelligence	4
2.1	Introduction	4
2.2	Hybrid Fuzzy System	4
2.2.1	Why Fuzzy Logic?	4
2.2.2	Learning Fuzzy Controllers	5
2.2.3	Genetic Algorithm vs. Artificial Neural Network	5
2.3	Real Time Learning	7
2.3.1	Introduction	7
2.3.2	Global vs. Local Optimizing Learners	7
2.3.3	The ECOS Paradigm	8
2.3.4	ECOS Implementations	9
2.4	Anticipating Systems	9
3	Backend Intelligence (Server)	9
4	Integrating Mobile and Backend	10
4.1	Two intelligences on two nodes (multiagent approach)	10
4.2	A single intelligence on two nodes (morphing agent approach)	11
5	Further Notes	12

1 Introduction

Up to now, research investigated rather general parts of this project. The aim was to get an overview of project-related issues and their state-of-the-art ("as-is analysis"). Topics under investigation so far have been intelligent (patient) monitoring, embedded/mobile/energy aware computing and modern approaches for Artificial Intelligence (AI) and Machine Learning (ML). One major requirement of the intelligence on the mobile monitoring node that has been identified, is the ability learn about the normal behavior of the monitored patient and to be capable of doing that on-line and nearly in real-time (real-time learning, take behavior based actions). Although it would be possible to prepare the monitoring node with an already offline-trained AI system, this training can only be performed with general data - representative training data of each single patient is simply not available - every person is individually different. A novel approach of a real-time learning monitoring intelligence working within this unbalanced, distributed environment has to be found. Recalling the outcomes of the fundamental research activities done so far allows to start drawing a (theoretical) model of a system that addresses these research aims.

Annotation: The term "model development" used in this context just means that the focus of investigation has changed from doing a general view on the state-of-the-art of related topics (as-is analysis), the next steps are focusing on the specific research aims of this project (start concrete work towards the research aims). This does clearly NOT mean that development in terms of programming code is started, it is meant on a more theoretical level at this point of time.

1.1 Research Aims Revisited

//todo: Summarize the research aims once again - point out that the research aim is NOT to enhance a certain type of hybrid AI system; the research aim is much more to enable a distributed decision taking process within an unbalanced, distributed, gsm-connected environment where AI is utilized. *For sure the AI approaches have to be adapted and enhanced but only to achieve higher research goals; the adaption/enhancement itself is not a primary objective.*

1.2 Thesis Working Title

It has been irritating for the last months, that every document produced within the context of this research was entitled with a different thesis working title. This simply is caused by the fact, that it is hard to find the right title at

the early stages of such a complex investigation. It is quite easier to describe circumstances in a report than to compress the most important things into the title of a report. As the fundamental investigations on related topics progressed, the working title got clearer and more specific to this research.

For the remaining time of this research - which clearly is going to be the main part - I would suggest to agree on a final working title until the end of this project, the written thesis. This should allow people to clearly identify this research and all related documents, notes, talks, etc. and additionally reduces time spent on thinking about a good working title for this project. I think I found a title that points out the most important issues that are addressed within this research:

“Online Machine Learning and Control in an Unbalanced Distributed System”

Comments are welcome ;-)

2 Mobile Intelligence

2.1 Introduction

The intelligent, mobile patient monitoring system should allow patients to leave stationary care in hospitals or nursing homes and enjoy life as normal as possible at home. The task of such patient monitoring systems (“telecare systems”) is to permanently monitor the patients health state and raise an alarm in case of emergency (or upcoming emergency). Current intelligent telecare systems are stationary, not portable. This restricts the free moving space of the patients to a single building. Upcoming mobile devices address this problem but due to the limited resources (cpu, memory, battery power) of the hardware they do not have the capability of intelligent (AI-based) patient monitoring, in most cases the patient has to raise the alarm manually by pushing an alarm button. The major aim of this research is to build a true mobile, intelligent patient monitoring system which combines the intelligence of current stationary systems with the free moving space of current non-intelligent mobile systems. For more information see the research notes for intelligent monitoring: [12].

2.2 Hybrid Fuzzy System

2.2.1 Why Fuzzy Logic?

Fuzzy logic is a way to enable our boolean-logic based computers to deal with vagueness and impreciseness as humans can do. Fuzzy logic converts crisp values

(“true” or “false”: if temperature > 30 then) into imprecise words (“if temperature is *warm* then ...”) and therefore is also called “computing with words” or “soft computing”. The strengths of fuzzy systems lie in their ability to deal with complex, indeterministic, vague and imprecise systems and the ability to easily incorporate human expert knowledge. As we have all these attributes present in the scenario of this research project, a fuzzy logic based controller seems to be the most valuable approach addressing the distributed decision taking problem. One major drawback of standard fuzzy systems is their lack of the ability to learn, So called hybrid fuzzy systems (fuzzy systems combined with machine learning algorithms) are addressing this issue. Hence, a hybrid approach of a learning fuzzy controller has been identified as being the best possible solution for the mobile side intelligence (see[11]).

2.2.2 Learning Fuzzy Controllers

As standard fuzzy systems do not have the ability to learn (i.e. to adapt their rule-base over time autonomously), the fuzzy controller has to be merged with a machine learning algorithm. Depending on the specific AI approach the fuzzy controller is being merged with, the inherent attributes of this approach are transferred to the new arising hybrid system. For instance, building a hybrid fuzzy controller based on genetic algorithms (GA) inherits the GA’s applicability for optimization tasks, merging Artificial Neural Networks (ANN) with fuzzy controllers enables the arising hybrid fuzzy controllers to learn complex coherence (see [11] for a description of AI approaches and their properties and intended fields of application).

2.2.3 Genetic Algorithm vs. Artificial Neural Network

The use of genetic algorithms (GA) is a popular approach in the field of machine learning currently. GA’s main purpose is optimization by countless iterations [reference RD_AI_analysis], and thus does not seem to be applicable for our restricted, real-time learning intelligence on the mobile node. There are efforts to develop incremental real-time learning GAs, as presented by [1]. The authors are building a system of distributed, embedded, autonomous agents acting and learning in real time, depending on a "behavior based" fuzzy logic controller to cope with the complexity and non-deterministic attributes of an intelligent buildings system environment. The main parts of the system are a hierarchical fuzzy controller and a GA based optimiser, combined within a single "associative experience engine". To improve the learning GA’s performance, the authors narrowed the search space for the GA optimizing algorithm mainly through 2 enhancements:

- if the learning algorithm is started it does not start search from random (as usual in GAs do) but starts searching from an already known last best result
- furthermore the search space is narrowed by incorporating preset rules from human experts knowledge, restricting the search algorithm from areas of the solution space where solutions are unlikely to be found

Simulation/test results showed, that the developed hierarchical genetic fuzzy controlled Intelligent Building system performs well and achieves comparable results against offline methods like Mendel-Wang and ANFIS by having all the advantages of on-line adaptive systems over these offline-learners [1].

It is a question whether the way the authors made the GA performant enough for on-line learning makes sense: GA's are known for their abilities and good results when searching for an optimized solution in a rather random search through the search space inspired by natural evolution (random mutation and pairing). This random search for an optimum in a huge solution space is performed by nearly countless iterations. From my point of view this is a strong approach for searching optimized results in a very indeterministic system which is hard to get a grip on for human experts and developers, where pre-defined rules from human experts are hard to formulate.

In a case where the solution space can be narrowed, where pre-defined rules from human expert knowledge can be incorporated, the popular approaches for learning algorithms are decision trees and connectionist based systems (neural networks), not GAs. I think that using an ANN for the authors project would achieve similar or even better results than the "cutted down" GA they use. GA's should not be used just for the sake of used GAs (because they are very popular currently), they should be used because of their attributes fitting best to the requirements of a problem [list/reference NEGevitstky]. From my point of view when learning from interaction with the user a GA is not the best solution.

I would suggest that the learning capabilities of neural networks and the ability to deal with impreciseness and vague terms of fuzzy controllers seem to be able to address the requirements of this project. Recent studies have shown, that hybrid neuro-fuzzy learners even outperform classical Artificial Neural Network approaches [2]. These hybrid kind of controllers are also called "connectionist based systems" in more general terms.

Having a closer look at connectionist based fuzzy controllers shows that there are many (more or less) popular approaches, e.g. ANFIS, ANNBIFIS, DENFIS, FALCON, GARIC, NEFCLASS, NEFPFOX.SANFIS, FuNN, etc [9].

2.3 Real Time Learning

2.3.1 Introduction

//todo: Notes related to[8]: two major areas of computer science are merging: artificial intelligence (AI) and real-world application domains requiring real-time responses. compare: "hard real time domains" (absolutely guaranteed response time is required) and vs. not-hard, "coincidentally real-time", i.e. just "fast enough" AI is applicable dynamic, uncertain and incompletely specified domains. AI in real-time environments has "bounded rationality" and "bounded reactivity" due to the environmental restrictions. example domain: medical intensive care unit monitoring intelligent real time control: control + small actions; alarming + diagnosis AI also accesses symbolic data like patient history, information about surgery, qualitative assessments of medical personnel.

In ICU context: need for guaranteed real-time reactions → that's not required in home-care/telecar because patients are not ICU'd, "coincidental real time" is enough

2.3.2 Global vs. Local Optimizing Learners

For the last years, many researchers and developers have been using neuro-fuzzy controllers within their projects because they provided working solutions for problems that could not be solved with traditional, "static" (fix programmed behavior, non-autonomous) algorithms. Although neuro-fuzzy controllers got more and more popular for AI tasks over the last years, several problems came up when being applied to real-world tasks such as online adaptation, real-time learning, life-long learning and performance problems. Kasabov mentions several investigations in [5] that proved most popular NN models like multi-layer perceptrons trained with backpropagation, radial basis functions, self organizing maps, fuzzy neural network models being not suitable for adaptive, on-line learning. Most of these approaches listed above are offline learners and only suitable for offline, multi-pass learning. The learning algorithm of those approaches is also called "globally optimized learning". Globally optimizing learners are trying to minimize the output error (this is the difference between the outcome of the system and the expected outcome) by taking all presented training examples into account. This means that best results are achieved the longer these systems are trained. Processing each single training data set is very time-consuming as the system has to take all presented datasets into account when a new one arrives. Some systems even require multi-pass learning (i.e. multiple iterations over the training data set). A common problem of such systems is "overfitting" (see also[7] for a good explanation of overfitting), i.e. it is too much trained which results in poor performance.

The opposite to globally optimizing learners are "locally optimizing learners". In contrary to the approach presented above, locally optimizing learners do not try to re-capitulate all training examples presented again when a new one arrives. As the name indicates, they do not try to reduce the learning error globally over all training data sets but they perform a "clustering" of all data sets presented and optimize the learning error per cluster on a local basis. These algorithms are computationally much more efficient and therefore applicable for on-line/real-time learning. They also avoid the problem of overfitting. The "ECOS" (evolving connectionist systems" principle ([4, 3]) describes the environment and requirements of such real-time learning, connectionist based systems:

2.3.3 The ECOS Paradigm

Kasabov starts developing his ECOS principle by having a look at well-known problems of existing systems and deriving seven major requirements for open connectionist based systems as they are:

- learn quickly from a large amount of data (fast, one-pass training)
- adapt in real time in an on-line mode
- open structure where new features can be introduced at later stages
- accomodate everything that is, was and will be in an incremental way
- be able to learn and improve through interaction with other COS
- adequately represent time and space in different scales
- should be able to analyse itself (self-analysis and rule-extraction)

The ECOS principle addresses all of those 7 requirements listed above. ECOS are multimodular, hierarchical open systems that evolve in time through interaction with the environment. The ECOS ist not a clear multilayer NN, in the beginning it is a rather un-organized population of neurons with only a few connections. initial knowledge could be inserted into this initial structure. Evolving over time, neurons and connections grow and the system gets more and more interconnected. A new node is created if its corresponding data sample is desired to be significantly different from the ones presented before. In addition there is a pruning algorithm included that removed neurons not actively used in the ECOS ("if case node is old and activation rate is low and density of neighbours is rather high and sum of incoming and outgoing weights is low THEN the probability of pruning this node is high" ...). ECOS systems can be trained in 2 ways:

actively by presenting training data sets and passively by the so called "eco" training mode: in this case stored training examples patterns stored as weights are used as eco training examples. Kasabov states that ECOS are up to 4-6 six orders of magnitude faster than multilayer perceptrons or fuzzy neural networks trained with backpropagation. One implementation of the ECOS principle is EFuNN (Evolving Fuzzy Neural Network) [5].

Many requirements that have been defined for this research project can be found in Kasabovs ECOS requirements. It seems to be a valuable approach to use an ECOS implementation like EFuNN (or DENFIS [6]) as the basis of our intelligent mobile monitoring node.

2.3.4 ECOS Implementations

//todo

- EFuNN vs. DENFIS - meaningfulness of rules
- EFuNN's way of building clusters are representing a history of all incoming monitoring data, i.e. a compressed view on the sensor input history.
- CHECK: EFuNN was chosen somewhere in recent papers because of its readable rule extraction, DENFIS uses Takagi-Sugeno type inference perhaps this enhances performance but reduces feature extraction possibilities!
- kas01: p.42: "the clarity of funn rules though is much better than the takagi-sugeno rules used in anfis"
- compare kas02 - denfis: denfis uses takagi-sugeno fuzzy inference !!!

2.4 Anticipating Systems

//todo

- "anticipating" systems could help to foresee (anticipate) events that are going to happen (batter power, upcoming emergency,..) to be investigated at a later point [13]
- batt-status on MN as additional input parameter for the learning/fuzz controller? (e.g. "if <possible> upcoming emergency situation and battery rather <low> then ..."

3 Backend Intelligence (Server)

//todo

4 Integrating Mobile and Backend

It is clear that the mobile device cannot provide maximum performance due to its limited resources and it is clear too, that a device having those resources to do so is not applicable for acting as a portable, "mobile" device (see [10]). Hence, this research created the approach of an "assisted" intelligent mobile monitoring node. This means that the restricted mobile node is working autonomously and energy-efficient as long as possible and is only opening up the costly GSM/GPRS line in extraordinary cases (this is where the learning about the "normal" behaviour of the monitored patient comes into existence). Once the wireless communication line has been established, a much more powerful backend node (a backend server) can assist to analyse the input data and in decision taking (e.g. decide if it is really an emergency or not, etc).

In addition to the much more powerful hardware, the backend node has the possibilities to access historical, statistical and symbolic databases, etc. to be integrated into the decision taking process. This central database will not be locally available on a portable monitoring device, even if the mobile hardware will become powerful enough to perform more complex analysis than now within the near future. This "assisted" distributed decision taking process approach seems to be valuable within such unbalanced, distributed environments whereas we have energy-dependent and scarce resources on the one side and "over-the-air" (i.e. wireless) connected powerful resources and databases on the other side.

One research issue to be investigated is how the system on the mobile monitoring node should look like. It has to be able to learn about the individual patient to be able to better classify its health state and to work with resources in a responsible way. But it also has to take into account the energy-aware issues, the vague and imprecise rules for communication with the backend system combined with sensor input data (i.e. when to open up the wireless connection) and to integrate human experts knowledge (e.g. from medical personnel). As already described in the leading chapters of this document, a hybrid learning fuzzy logic controller seems to be the best solution for this issue. On the other hand we have to think about how the mobile and the backend node are integrated once the mobile side intelligent fuzzy controller is deployed: how do they work together, how does this distributed decision taking process look like?

There are mainly 2 approaches to choose:

4.1 Two intelligences on two nodes (multiagent approach)

This is the rather usual or "straightforward" approach: an intelligent algorithm is deployed on each node, i.e. the chosen hybrid fuzzy controller on the mobile

node and a more powerful algorithm on the backend system (e.g. a genetic algorithm based fuzzy optimizer). Both intelligences are working for their own in an autonomous way, although they have to cooperate in several cases. In the standard case the mobile intelligence keeps monitoring the patient as long everything is classified as normal. Meanwhile the backend intelligence could perform long-running tasks like GA based data mining and rule-optimizing. In case of an emergency the mobile intelligence opens up the connection to the backend system and streams all input data to the more powerful intelligence on the server. The server side algorithm now can analyse the data and take additional resources and it's meanwhile better optimized rule-base into account to take better decisions. Before the connection is closed, the backend algorithm could transfer the optimized rule sets to the mobile node to enhance its performance.

In this scenario we have two intelligent agents, mainly working autonomously for their own. They communicate and cooperate in a defined way and influence each other. This could be called a "multi agent" approach.

Advantages of this approach are that it can easily be developed as we have 2 isolated algorithms, defined rules of communication and cooperation and so each of the algorithms can be developed and enhanced for its own.

A major drawback of this approach seems to be the fact that the fixed rules of communication and cooperation are contraproductive to the distributed decision taking process. In fact at this stage of research the arising system seems to be vague, the rules when to open up the wireless line are imprecise and the whole scenario of cooperation of those two nodes cannot be completely specified within this undeterministic system.

A fuzzy approach for the mobile monitoring intelligence was chosen because of fuzzy logics strength when dealing with vague, imprecise and undeterministic systems. Why not just go a step further and expand the fuzzyness from the monitoring task to the whole distributed decision taking?

4.2 A single intelligence on two nodes (morphing agent approach)

Instead of deploying multiple independent algorithms communicating based on programmed rules and parameters, a much more fuzzy approach seems to be promising for this project: the aim is to avoid having more than 1 algorithm because as soon as there are 1+x participants, communication rules and interfaces are required. These are only applicable in deterministic systems. As we have a very imprecise decision taking process, the goal is to avoid having those rigid interfaces between 1+x algorithms, so the aim is to have only a single algorithm

or a single "mind" on x nodes.

Having a single intelligence moving from one node to another eliminates the need of defining cooperation and communication rules and allows a more flexible and fuzzyfied distributed decision taking process. Picking up the previous example again: In the standard case the mobile intelligence keeps monitoring the patient as long as everything is classified as normal. In case of emergency the wireless connection is opened up and now NOT a server side algorithm takes control, but the mobile side algorithm somehow "morphs" from the restrited mobile device to the powerful backend device and resumes working in another, just much more powerful body. This step is essential: when morphing from the mobile into the backend "body", the intelligences "mind" has to be transferred into the new body. As soon as the intelligence (the agent) is respawned on the backend device it continues work as it would have been running on this hardware all the time. In addition the agent now can use all additionally available resources which were missing on the mobile device and take more precise decisions.

The "enabler" for this approach is the ECOS paradigm ([4]): ECOS are able to represent time and space (in a compressed way). This is done within their locally optimized clusters and therefore the "mind" of an ECOS can easily be extracted and injected elsewhere. This technique can be called "state serialization": we transfer the "mind" into a more powerful "body" to handle the problem....(the "clusters" in EFuNNs for example represent all input data presented in form of ECM - evolving clustering method - and what they've learned in form of neurons and weights; so somehow it must be possible to serialize and transfer this "mind" and respawn it on another machine with more power and more input data, etc etc...)

5 Further Notes

References

- [1] V.; Colley M.; Clarke G. Hagraas, H.; Callaghan. A hierarchical fuzzy-genetic multi-agent architecture for intelligent buildings online learning, adaption and control. *Elsevier Information Sciences*, (150):33–56, 2003.
- [2] C.M.; Rajendra Acharya U.; Sadasivan P.K. Kannathal, N.; Lim. Cardiac state diagnosis using adaptive neuro-fuzzy technique. *Medical Engineering & Physics*, 28:809–815, 2006.
- [3] M.; Lane T. Kasabov, N.; Middlemiss. A generic connectionist -based method for on-line feature selection and modelling with a case study of gene

expression. *Knowledge Engineering and Discovery Research Institute*, 2003?
School of Information Technology, Auckland University of Technology.

- [4] N. Kasabov. Ecos: Evolving connectionist systems and the eco learning paradigm. In *Proceedings of ICONIP'98, Kitakyushu*. Department of Information Science, University of Otago, October 1998.
- [5] N.K. Kasabov. On-line learning, reasoning, rule extraction and aggregation in locally optimized evolving fuzzy neural networks. *Elsevier Neurocomputing*, 41:2545, 2001.
- [6] Q. Kasabov, N.K.; Song. Denfis: dynamic evolving neural-fuzzy inference system and its application for time-series prediction. *IEEE Transactions on Fuzzy Systems*, 10(2):144–154, April 2002.
- [7] Tom M. Mitchell. *Machine Learning*. Number 0-07-115467-1. McGraw-Hill, international edition, 1997.
- [8] J.A.; Agrawala A.K. Musliner, D.J.; Hendler. The challenges of real-time ai. *U. Maryland Technical Report CS-TR-3290, UMIACS-TR-94-69*, 1994.
- [9] A. Nowicki, R.; Pokropinska. Neuro-fuzzy structures and their comparative analysis. In *Proceedings of the 4th intl. Symposium on Information and Communication Technologies WISICT'05*, 2005.
- [10] H.J. Pruessler. Research notes: Application of artificial intelligence within an unbalanced distributed system - project outline. Research Notes - Project Outline, April 2006.
- [11] H.J. Pruessler. Research notes: General ai applicability research. Research Notes - Machine Learning Approaches Evaluation, August 2006.
- [12] H.J. Pruessler. Research notes: (health) monitoring investigation - cooperative ai in an unbalanced distributed environment. Research Notes - Intelligent Health Monitoring Investigation, May 2006.
- [13] L.H. Tsoukalas. Neurofuzzy approaches to anticipation: A new paradigm for intelligent systems. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 28(4):573–582, August 1998.