1

Introduction

The main differentiator of the new generation of autonomous systems that is emerging in the twenty-first century is the *adaptivity* of their intelligence. They are not simply automatic (usually remote) control devices, not only adaptive control systems in the narrow sense of systems with *tunable* parameters as in the last decades of the past century, but they are rather systems with a certain level of evolving intelligence. While conventional adaptive techniques (Astroem and Wittenmark, 1989) are suitable to represent objects with slowly changing parameters, they can hardly handle complex (usually, nonlinear, nonstationary) systems with multiple operating modes or abruptly changing characteristics since it takes a long time after every drastic change in the system to update model parameters. The *evolving* systems paradigm (Angelov, 2002) is based on the concept of evolving (expanding or shrinking) system *structure* that is capable of adapting to the changes in the environment and internal changes of the system itself that cannot solely be represented by parameter tuning/ adjustment.

Evolving intelligent systems (eIS) the concept of which was pioneered recently (Angelov, 2002; Kasabov, 2002; Angelov and Kasabov, 2005; Kasabov and Filev, 2006, Jager, 2006), develop their structure, their functionality, and their internal knowledge representation through *autonomous learning* from data *streams* generated by the (possibly unknown) environment and from the system self-monitoring. They often (but not necessarily) use as a framework of implementation fuzzy rule-based (FRB) and neurofuzzy (NF) or neural-network (NN) based systems and machine learning as a tool for training. Alternative frameworks (such as conventional multimodel systems, decision trees, probabilistic, e.g. Markov, mixture Gaussian models, etc.) can also be explored as viable frameworks of eIS and *autonomous learning* systems.

It should be noted that the physical embodiments of such systems can range from micro-systems-on chip (Everett and Angelov, 2005), motes of a wireless sensor network (Andreu, Angelov and Dutta Baruah, 2011), mobile robots (Zhou and

Autonomous Learning Systems: From Data Streams to Knowledge in Real-time, First Edition. Plamen Angelov.

Angelov, 2007; Liu and Meng, 2004; Kanakakis, Valavanis and Tsourveloudis, 2004) to unmanned airborne vehicles (Valavanis, 2006) and computer-controlled industrial processes (Filev, Larson and Ma, 2000; Macias-Hernandez *et al.*, 2007).

The potential of these systems for industry was acknowledged by leading researchers with a solid industrial background such as Dr. A. Kordon, R&D Leader, Dow Chemical, TX, USA who said in 2006 "*Evolving Intelligent Systems have a high potential for implementation in industry*" (http://news.lancs.ac.uk/Web/News/Pages/930389757F5B0BF4802571FB003CB1A2.aspx); Dr. D. Filev, Senior technical Staff at Ford R&D, Dearborn, MI, USA who also in 2006 said "*Embedded soft computing applications are the natural implementation area of evolving systems as one of the main tools for design of real time intelligent systems*" (same web reference as above).

The problem of automatic design of computationally intelligent systems for modelling, classification, time-series prediction, regression, clustering from data has been successfully addressed during the previous century by a range of techniques such as by gradient-based techniques (as in the neurofuzzy approach ANFIS (Jang, 1993)), by genetic/evolutionary algorithms (Fogarty and Munro, 1996; Angelov and Buswell, 2003), by using partitioning by clustering (Babuska, 1998), learning by least squares (LS) techniques and so on. But, these approaches were assuming all data to be known in advance (a *batch* or *offline* mode of learning) and were not directly applicable to data *streams*.

At the same time, the twenty-first century is confronting us with a range of new challenges that require completely new approaches. As John Naisbitt famously said "today we are drowning in information but starved for knowledge" (Naisbitt, 1988). We are in the midst of an information revolution witnessing an exponential growth of the quantity and the rate of appearance of new information by; Internet users, consumers, finance industry, sensors in advanced industrial processes, autonomous systems, space and aircraft, and so on.

It is reported that every year more than 1 Exabyte ($=10^{18}$ bytes) of data are generated worldwide, most of it in digital form (http://news.bbc.co.uk/2/hi/technology/4079417.stm). Toshiba recently coined the phrase '*digital obesity*' to illustrate the ever-growing amount of data that are generated, transmitted and consumed by the users today. In this ocean of data the useful information and knowledge very often is difficult to extract in a clear and comprehensive, human-intelligible form. The availability of convenient-to-use and efficient methods, algorithms, techniques, and tools that can assist in extracting knowledge from the data (Martin, 2005) is a pressing demand at individual and corporate level, especially if this can be done online, in real time.

The new challenges that cannot be successfully addressed by the existing techniques, especially in their complexity and interconnection, can be summarised as follows:

i. to cope with huge amounts of data;
ii. to process streaming data online and in real time;
iii. to adapt to the changing environment and data pattern autonomously;

iv. to be computationally efficient (that means, to use recursive, one-pass, noniterative approaches);
v. to preserve the interpretability and transparency in a dynamic sense.

To address these new challenges efficient approaches are needed that deal with data streams (Domingos and Hulten, 2001), not just with batch sets of data (Fayyad *et al.*, 1996), detect, react and take advantage of concept *shift* and *drift* in the data streams (Lughofer and Angelov, 2011). Efficient collaborative and interactive schemes are also needed for a range of applications in process industries (for self-calibrating, self-maintaining intelligent sensors of the new generation), in autonomous systems and robotics (for systems that have self-awareness, replanning and knowledge summarisation capabilities), in multimedia and biomedical applications, to name a few.

Autonomous learning (AL) is understood in this book in the context of both system structure and system parameters. This means that the overall process of design, development, redesign/update, adaptation, use and reuse of such systems is autonomous, including the stages of the system design that traditionally assume heavy human involvement and are normally done *offline* (the system being designed not in *real time* in which the process that is using this system runs in). Therefore, our understanding of AL and our concept of eIS is intricately related to the concepts of *online* and *real-time* system structure and parameter design and exploitation and to data *streams* rather than to data *sets*. This is the main differentiator in comparison with the traditional disciplines.

1.1 Autonomous Systems

Autonomous systems are often seen as the physical embodiments of machine intelligence. The concept of autonomous systems (AS) is not new and is closely related to AI and cybernetics, but became more popular during the last decade or so mainly due to the interest (and funding) from the defence and aerospace industries. AS are significantly different from simple automatic control systems, ACS (Astroem and Wittenmark, 1989). In fact, each AS has at its lower level (Layer 1) an ACS, usually, for motion control, for control of the sensors and actuators, and so on. An AS, however, has also important upper layers in its architecture (Figure 1.1) that concern perceptions-behaviours (layer 2 that also corresponds to structure identification in AL systems) and the representation of the environment (usually in a form of rule base, states, or a map, but not necessarily limiting to these forms of representation) in the model and self-monitoring functions (layer 3 that is linked to the prediction).

AS can be seen as a fusion of computationally enabled sensor platforms (machines/devices) that possess the algorithms (respectively, the software) needed to empower the systems with evolving intelligence that is manifested through interaction with the outside environment and self-monitoring.

Examples include, but are not limited to unmanned airborne systems, UAS, unmanned ground-based vehicles, UGVs (Figure 1.3), and so on.

JWST237-c01 JWST237-Angelov October 8, 2012 14:59 Trim: 244mm X 168mm Printer Name: Markono

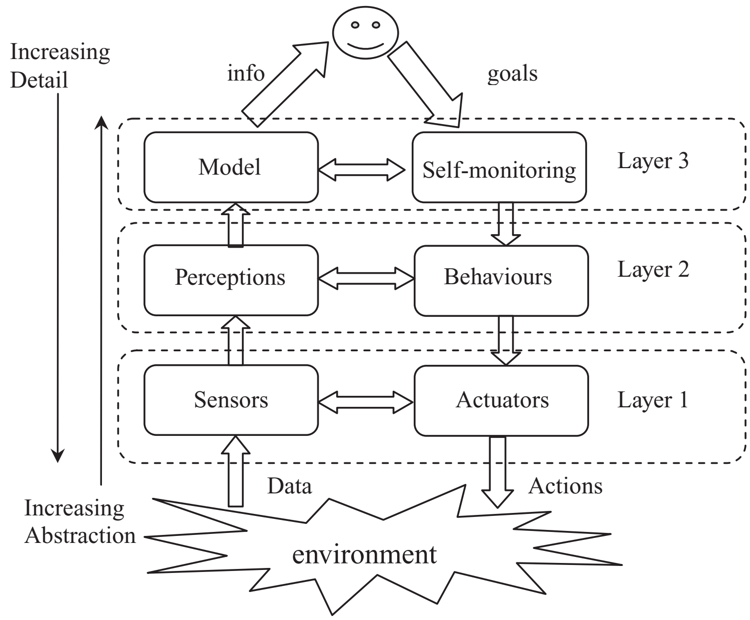

Figure 1.1 A three-layer structure of an autonomous system. (layer 1 – low-level direct control, including teleoperation; layer 2 – a more abstract, behavioural autonomy, specific tasks; layer 3, often called deliberate autonomy – the upper abstract layer of modelling the environment and self-monitoring)

1.2 The Role of Machine Learning in Autonomous Systems

The core functionality of an AS depends on the ability to be aware of the environment (through data streams generated by the sensors) and to make decisions accordingly. Obviously, such decisions cannot be made on the basis of a preprogrammed logic because this will assume a full knowledge of all the environments in which the system will operate and will not be flexible enough. Therefore, core elements of any AS are self-monitoring and self-adaptation. Autonomous learning and extracting new knowledge as well as updating the existing knowledge base are vitally important for such type of systems.

The dependence between autonomy and learning is a two-way process – on the one hand, autonomous systems require learning in order to be aware of, explore, and adapt to the dynamic environment; on the other hand, learning algorithms require autonomy to make them independent of human involvement. The lack or low level of autonomy in most of the currently existing algorithms leads to the need to develop new generations of AL systems (ALS) to play an important role in the design and maintenance of autonomous systems (e.g. UAV, UGV, intelligent/soft sensors, etc.). A system (however well may it have been designed) that is not empowered by an autonomous machine learning capability will fail helplessly in a situation that was

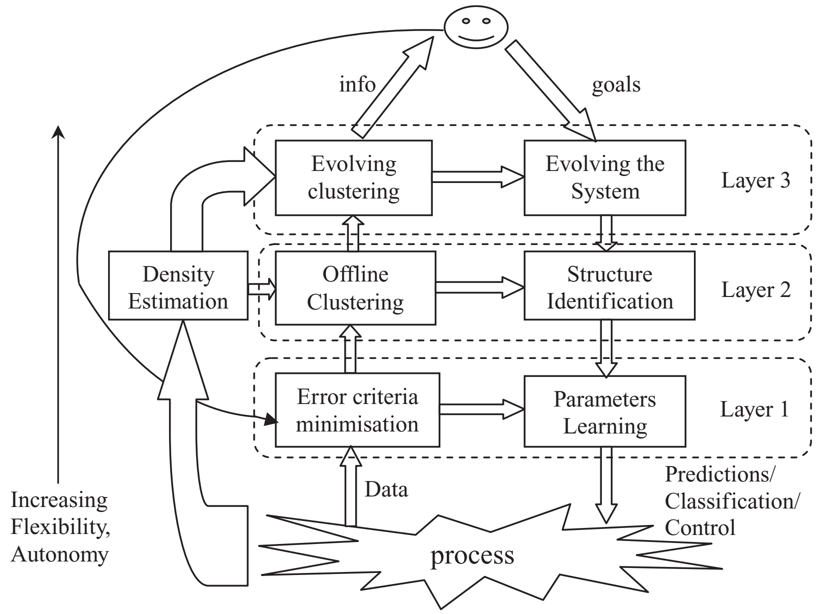

Figure 1.2 A graphical representation of an autonomous machine learning system (layer 1 – 'traditional' (parameters only learning) approach; layer 2 – offline learning of system structure using clustering and data density (it should be noted that other methods instead of density-based clustering can be used at this layer); layer 3 represents the evolving system structure – the upper layer of autonomous learning that often also includes self-monitoring (not represented for simplicity)

not predicted at the design stage or a situation that is described by parameters widely out of the range of parameters considered during the design stage.

A system that has learning capabilities and an evolving model of the real world will try to adapt and create new rules, to drop rules that are outdated and irrelevant to the new situation and will at least have a higher chance to succeed. In reality, most of the complex environments are unpredictable, nonlinear and nonstationary. An

Figure 1.3 Autonomous UGVs (laboratory-scale mobile robot Pioneer3DX) in a convoy formation outside Infolab21, Lancaster University campus

autonomous system must have the ability to learn quickly (from a single or few data samples) and to extract knowledge from the data streams collected by the sensors in real time, to rank the previously existing knowledge and to compare the relevance of the new knowledge to the previous knowledge leading to an update of the world model. The role of specific types of machine learning that are particularly suitable for online, real-time update of a real world model with evolving (growing or shrinking) nature is vitally important for the development of truly autonomous systems.

1.3 System Identification – an Abstract Model of the Real World

An autonomous system must have a model of the world (the environment that surrounds the AS and its internal functioning). Usually, this model is in the context of the goal that the AS must perform. Development of such models is governed by the system identification (Ljung, 1987), which is a topic usually considered in relation to control theory. Systems are often considered to be described by a set of (differential) equations. Alternative representations, for example statistical Bayesian, Markov models, decision trees, and so on. are also viable world models (see Chapter 2). An alternative that is particularly suitable to represent intelligent systems and knowledge is the fuzzy rule-based form of representation (to be discussed in Chapter 4). Whichever framework is used, however, the identification is usually considered in terms of:

a. the structure (with heavy human involvement, usually offline, at the design stage); and
b. parameters (often automatically, online, during the process of exploitation).

In what follows the concept for each of the two key aspects of identification problem will be briefly described.

1.3.1 System Structure Identification

The structure of the world model or the system is usually considered to be suggested by the human expert. It may take the form of:

a. a set of differential equations;
b. transfer function (time or frequency domain);
c. a set of fuzzy rules;
d. a neural network;
e. a stochastic model (e.g. Markov states model), and so on.

In this book, without limiting the concepts, the last three forms will be considered as examples. The main reason is their suitability to represent human intelligible knowledge in a granulated form.

System structure in the case of differential equations may comprise of the number and type of the differential (or difference) equations, the number of inputs and outputs. In the case of the transfer function it may include the order and type (e.g. 'all poles' or 'all zeros'). For neural networks (NN) the structure may define layers, feedback or feedforward, memory, number of inputs, outputs, and other elements that are optional. For the case of fuzzy systems the structure includes the following:

a. number of fuzzy rules;
b. number of inputs (features) and outputs;
c. type of the membership functions and their position in the data/feature space (this is not necessary for the specific type of fuzzy rule-based models considered in Section 4.3);
d. type of antecedents (scalar or vector);
e. type of the consequents (e.g. Zadeh–Mamdani (Zadeh, 1975; Mamdani and Assilian, 1975) or Takagi–Sugeno (Takagi and Sugeno, 1985));
f. type of connectives used (AND, OR, NOT);
g. type of inference (centre of gravity, 'winner takes all', etc.).

These will be further detailed and described in the next chapter.

Structure identification is an open research problem that still does not have a satisfactory and universally accepted answer. Structure identification can be seen as a nonlinear optimisation problem (Angelov, Lughofer and Klement, 2005) that aims to select the best structure in terms of minimum error in prediction/classification/control. Usually, it is not solved directly, but the structure is assumed to be provided. In some works the authors apply genetic algorithms, GA (Michalewicz, 1996), genetic programming, GP (Koza, 1992) and other numerical techniques for (partially) solving it. In this book a systematic approach will be used that is based on density increment that relates to the data density and distribution in the data space also taking into account the time element (*shift* of the data density). A fully theoretical solution is difficult, if possible at all.

Figure 1.4 illustrates in a very simplistic form the difference between the proposed and the traditional approach with respect to the role of the system structure identification – in ALS it is part of the automated process, while traditionally it is outside of the loop of automation.

There are different ways to devise automatically the structure of the model, including data space (regular) partitioning (Carse, Fogarty and Munro, 1996), clustering (offline or online and evolving) (Chiu, 1994; Babuska, 1998; Angelov, 2004a), based on data density (Angelov, 2002), based on the error (Leng, McGuinty and Prasad, 2005), and so on. The principle behind most of them is the old Latin proverb '*divide et impera*' which means 'divide and conquer' and leads to decomposition of a complex problem into (possibly overlapping and interdependent) subproblems or subspaces of the data space. The key questions that arise are:

- How to divide the problem or data space objectively (based on data density or the error are two obvious options); note that the traditional criteria for cluster quality, for example (Akaike, 1970) and so on, are designed to separate clusters well while

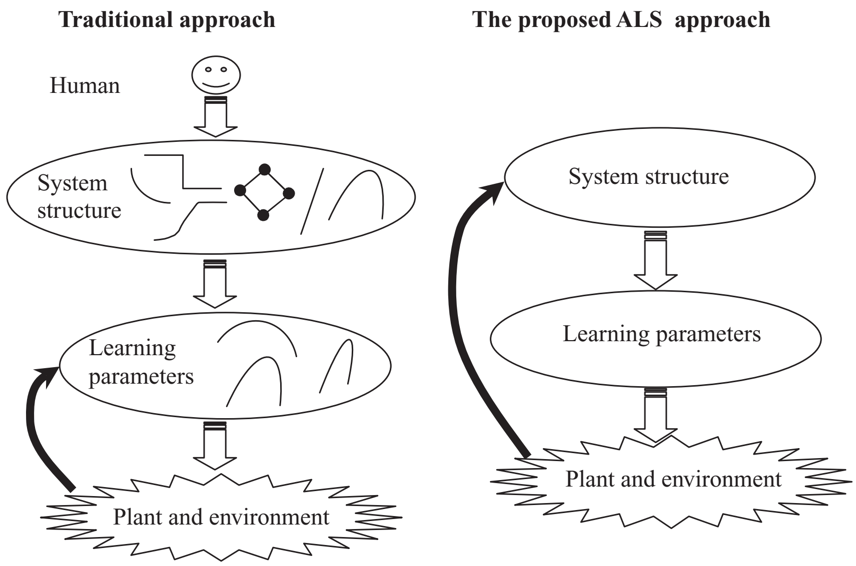

Figure 1.4 The traditional versus the proposed approach

for the purpose of model structure identification the overlap must be tolerated to avoid abrupt transitions between local models and gaps between them.

- Shall a data sample that is an outlier (which differs significantly from the existing local models or clusters) be ignored or it may be a start of a new local model (regime of operation); this problem is much more acute in online and real-time implementations when the decision must be taken based on the current data samples and no or little history.
- The optimality of the structure is, generally, a nonlinear problem, and therefore, its solution is, in principle, possible only numerically and offline; a possible pragmatic solution is to optimise the parameters subject to a structure that is selected automatically, but the optimality is then conditioned on the assumptions (as in other existing approaches).
- The dilemma between plasticity and stability – how often the structure can and should change – if it changes too often the system will lose its robustness; if it changes very rarely it will lose its sensitivity.
- Ideally, an automated algorithm for model structure identification should not depend on user- or problem-specific thresholds and parameters.

1.3.2 Parameter Identification

The problem of parameter identification is a much more established one (Ljung, 1987). The aim is to determine the optimal values of parameters of the model/system in terms of minimisation of the error of prediction/classification/control. If we use a fuzzy rule-based model as a framework that include parameters of the consequents of the fuzzy rules and parameters of the membership functions of the antecedent part of the rules (to be described and discussed in Chapter 4). If we use the particular type of fuzzy rule-based model introduced recently by (Angelov and Yager, 2012), (see Section 4.3) then the antecedent part is nonparametric.

The problem of parameter identification is also an optimisation one, but often can be considered as a linear or quadratic optimisation that guarantees uniqueness of the solutions subject to certain constraints. This is the basis of the widely used recursive least squares (RLS) method (Ljung, 1987). For general, nonlinear cases, they also use numerical procedures, such as error back-propagation, EBP (Werbos, 1990), other gradient-based techniques (e.g. conjugate gradients approach), and so on.

1.3.3 *Novelty Detection, Outliers and the Link to Structure Innovation*

The topic of novelty (respectively, anomaly) detection is pivotal for fault detection (Filev and Tseng, 2006) and video-analytics (Elgammal *et al.*, 2002; Cheung and Kamath, 2004). It has its roots in statistical analysis (Hastie, Tibshirani and Friedman, 2001) and analysis of the probability density distribution. The rationale is that novelties (respectively, anomalies, outliers) significantly differ and their probability density is significantly lower. Therefore, the test for a data sample to be considered as an outlier/anomalous is to have a low density.

The problem of system structure identification, especially in real time, is closely related to the outliers and anomaly detection, because an outlier at a given moment in time may be a start of a new regime of operation or new local model. In such case, the structure innovation will lead to increase of the density locally (around the new focal point). In this book we argue that the data density (local and global) can be used as an indicator for automatic system structure innovation and identification. *A drop of the global density indicates an innovation*; *an increase of local density indicates a consolidation* of a new regime of operation/new local behaviour.

1.4 Online versus Offline Identification

Autonomous systems have to be able to process and extract knowledge from streaming data in a so-called *online* mode. This means that the data stream is being processed sample-by-sample (here sample also means data item/instant) in a serial fashion, that is, in the same order as the data item was fed to the ALS without having the entire data stream/set available from the start. Imagine, a video stream – online processing (Figure 1.5, right) means processing it frame by frame, not storing (buffering) the whole video and then processing it *offline* (Ramezani *et al.*, 2008).

Systems that operate in offline mode may be good in scenarios that are very close or similar to the ones that they are specifically designed and tuned for. They need, however, to be redesigned or at least retrained/recalibrated each time when the environment or the system itself changes (e.g. in industry, the quality of raw materials, such as crude oil entering a refinery varies; catalysers are being removed or added to

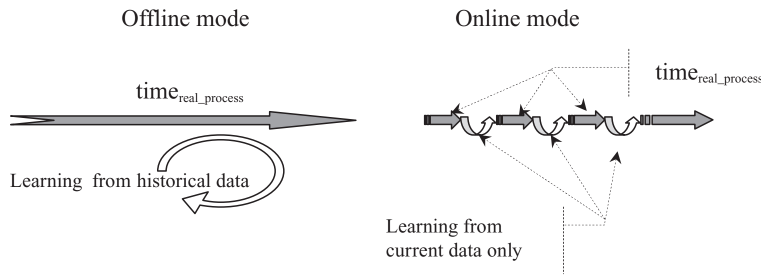

Figure 1.5 Online and offline modes of operation of a system

the polymerisation tank; hackers change their behaviour when they attack a computer system, UGV may enter an unknown zone, faults may develop in the subsystems of an AS, etc.). Offline systems (Figure 1.5, left) work with a historical 'snapshot' of the data *stream* and require all the previous data, which implies a much higher memory and computational requirements. In contrast to that, online systems work on a per sample basis and only require the current data sample plus a small amount of aggregated information; they do not require all the history (all previously seen data samples) of the data stream.

The online mode is often related to the real-time operation. It is important to stress that there is a subtle difference in the sense that an algorithm can be online (in terms of not storing the whole data sequence and processing data items one-by-one) and yet it might work slowly enough to be real time (if the real-life process is very fast while the computer processing unit, CPU is not that fast). In such cases there will be a delay in the output (prediction, class label, control action) produced by the model/system with respect to the real-world response. At the same time, a system may operate in a *real-time* manner and yet be offline if the sampling rate is extremely low. For example, in some biomedical problems the sampling (frequency of visits to the doctor and taking of measurements) can be as low as once every week or even month. In such cases, the system can learn from the whole previous history, process all previous data samples iteratively. Since these are extreme cases, in this book the focus will be on the ability of autonomous systems to learn online and in real time. Moreover, we will be primarily interested in so-called *recursive* algorithms that assume no iterations over past data, no storage/buffering of previous data and in a so-called *one-pass* processing mode.

1.5 Adaptive and Evolving Systems

As was already said, an AS should adapt to the environment. The theory of adaptive systems (Astroem and Wittenmark, 1989) is now a well-established part of control theory and digital signal processing (Haykin, 2002). It usually is restricted to systems with linear structures only and, more importantly; it does not consider the problem

JWST237-c01 JWST237-Angelov October 8, 2012 14:59 Trim: 244mm X 168mm Printer Name: Markono

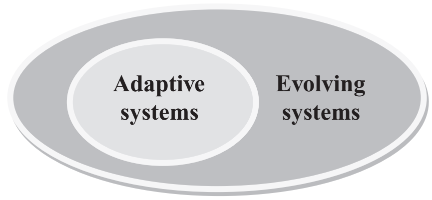

Figure 1.6 Evolving systems as a superset of adaptive systems

of system structure adaptation. An adaptive system is considered a system with a *fixed*, known structure that allows its parameters to vary/be adjusted. In this respect the concept of evolving systems (Angelov, 2002) as a system with evolving structure differs significantly. It is true, however, that evolving systems are also adaptive, but the subject of the adaptation are both system parameters as with the adaptive (in a narrow sense) systems as well as its structure. In this context, evolving systems can be seen as a superset of adaptive systems, Figure 1.6.

The area of evolving systems (as described above) that was conceived around the turn of the century (Angelov and Buswell, 2001; Angelov, 2002; Kasabov and Song, 2002) is still under intensive development and 'fermentation'. It is closely related to (albeit developing independently from) the works on self-organising systems (Lin, Lin and Shen, 2001; Juang and Lin, 1999) and growing neural networks (Fritzke, 1994). In the late 1990s and until 2001–2002 the term 'evolving' was also used in a different context – in terms of evolutionary (this will be clarified in the next section). Since 2002 and especially since 2006 when the IEEE started supporting regular annual conferences and other events (the last one, the 2012 IEEE Conference on Evolving and Adaptive Intelligent Systems, being in May 2012 in Madrid) it is used for dynamically evolving in terms of system structure systems. In 2010, the publishing company Springer started a new journal on Evolving Systems (http://www.springer.com/physics/complexity/journal/12530) and the number of papers and citations is growing exponentially.

The research area of evolving systems is central to the very notion of autonomous systems and autonomous learning and this will be made clearer and detailed in the rest of the book.

1.6 Evolving or Evolutionary Systems

In computational intelligence research evolutionary algorithms (EA), including such specific examples as genetic algorithms, GA (Goldberg, 1989; Michalewicz, 1996), genetic programming, GP (Koza, 1992), artificial immune system (Kephart, 1994), and so on. are computational algorithms that borrow heavily from the natural evolution. They often use a 'directed' random search for solving loosely formulated optimisation problems. They mimic a specific aspect of the natural evolution that is related to the population-based genetic evolution that is driven by such mechanisms as *mutation*,

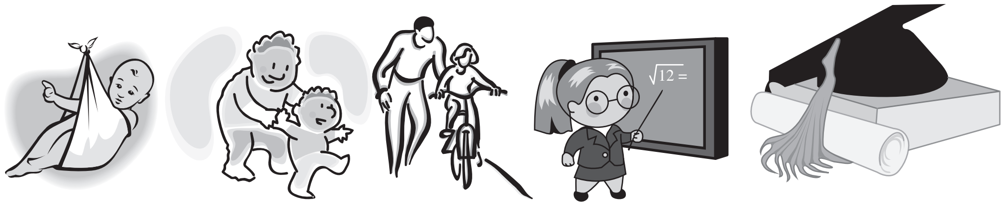

Figure 1.7 Human beings are a good example of an ALS that evolves by learning (new rules) from experience through their sensors using their brain

chromosomal *crossover, reproduction, selection.* The natural evolution also has the aspect of individual self-development, especially for the case of human beings (Figure 1.7). Starting as small babies we do not have any idea about the surrounding world, but we start to collect data streams through our sensors and soon we start to create rules using and evolving our brains. We start to recognise what is *good* and what is *bad*, what is *dangerous* and what is *safe*, and so on. With time, our rule base grows; some rules stop being used or become irrelevant or need some adaptation and adjustment throughout our whole life. Some rules we are taught, some we infer ourselves.

It is interesting to note that the rules we acquire, update or stop using are not precise, for example 'IF *we lift a bag weighing over 63.241 kg* THEN *we will get a broken back*', but they are rather ***fuzzy***, for example. 'IF *we lift* ***heavy*** *loads* THEN *we may get a broken back*' or 'IF *it is cold* THEN we *take a coat*', and so on.

In essence, we self-develop. In this book we propose a systematic approach that allows building autonomous systems with such capabilities – to self-develop, to learn from the interaction with the environment and through exploration.

The Oxford Dictionary (Hornby, 1974, p. 358) gives the following definition of **genetic** – "a branch of biology dealing with the heredity, the ways, in which characteristics are passed on from parents to off-springs". The definition of **evolving** it gives (p. 294) is "unfolding; developing; being developed, naturally and gradually". In brief, despite some similarity in the names, EA differ significantly from the more recently introduced concept of evolving systems. While genetic/evolutionary is related to **populations** of individuals and parents-to-offspring heredity, evolving is applicable to **individual** system **self-development** (known in humans as autonomous mental development, Figure 1.7). 'Evolving' relates more to learning from experience, gradual change, knowledge generation from routine operation, rules extraction from the data. Such capabilities are vital for autonomous systems and, therefore, we will expand this idea in the book.

If we consider a fuzzy rule-based system as a framework, an *evolving* FRB system will learn new rules from new data *gradually* preserving majority of the rules learned already (Angelov, 2002). This is very similar to the way that individual people learn, see Figure 1.7. In a similar way to humans, an evolving fuzzy system (EFS) can be initiated by an initial rule base (in a supervised manner the way we learn from parents and teachers) or can start learning 'from scratch', autonomously.

1.7 Supervised versus Unsupervised Learning

The very notion of autonomous systems is closely related to the unsupervised learning and reinforcement learning (Sutton and Barto, 1999). However, semisupervised learning also has an important part to play because pragmatically no autonomous system is assumed to be reproductive and out of users' (human's) control in terms of monitoring – remember the famous Azimov's laws of robotics (Azimov, 1950). In other words, the level of autonomy of systems that are of practical interests for industry, including defence and security is 4 or maximum 5a according to Table 1.1. Examples of systems with lower level of autonomy (1–3) are decision support systems, DSS (McDonald, Xydeas and Angelov, 2008).

In this book, we are interested in autonomy of the knowledge extraction from data streams (autonomous learning) that (the same as the overall scheme of an AS, Figure 1.2 – see the smiley face at the very top of the figure) does not fully exclude the human user, but reduces his/her role to bare provision of goals and monitoring plus the option to abort the operation on safety grounds (autonomy level 5a, see Table 1.1). Provision of goals itself can be a source of definition of criteria for optimisation and learning objectives. Most often the latter are related to minimisation of the prediction error, maximisation of the classification rate, and so on.

The autonomous learning (AL) that can enable AS to adapt and evolve should acquire more than a simple input–output mapping that is typical for traditional (machine learning, fuzzy systems, neural networks, etc.) model learning techniques.

Table 1.1 Autonomy levels adapted from (Hill, Crazer, and Wilkinson, 2007)

Level	Autonomy	Authority	Interaction
5b	Full	Machine **monitored** by human	Machine does everything autonomously
5a			Machine chooses action, performs it and informs human
4b	Action unless revoked	Machine **backed up** by human	Machine chooses action and performs it unless human disapproves
4a			Machine chooses action and performs it if human approves
3	Advise and, if authorised, act	Human **backed up** by machine	Machine suggests options and proposes one of them
2	Advice	Human **assisted by** a machine	Machine suggests options to human
1	Advise only if requested	As above **when requested**	Human asks machine to propose actions and human selects
0	None	Human	Whole task done by human except for actual operation

Instead, the emphasis in AL is on building and constantly monitoring the quality and updating the structure of the system. The extracted knowledge usually (but not necessarily) is in the form of human interpretable, fuzzy rules (Hopner and Klawonn, 2000). This learning is 'on the fly' starting from few or even a single data sample, if needed, adapting quickly, but also being able to accommodate previous knowledge (if it exists) and fuse it with the newly acquired knowledge.

The most effective scheme proved to be the combination of unsupervised learning for model structure identification and semisupervised learning for parameter adjustment where the supervision comes from the data stream but with a time delay and not necessarily after each time step. The key in this scheme that is very much like the scheme of adaptive filtering (Haykin, 2002) and adaptive control (Astroem and Wittenmark, 1989) is the timing. The data stream often provides both the input and output in terms of the AS but at the moment of prediction/classification/control action generation a value can be unavailable (thus, the need to be predicted) while at the next time instant (see Figure 1.5) these values (if available and measured) can serve to feed back the learning in a supervised form. In this way, the supervised learning can also be considered as an automatic process that is related more to the online form of operation.

For example, if a system automatically models/infers/predicts the value of the outside temperature tomorrow or the exchange rate tomorrow based on some measurements and previous observations (history) then these predictions will be very useful until we get the real/true value the next day (so, in some 23–24 hours we can benefit from these predictions). The next day, we can use, however, the real/true values (if they are available because it may be available only sometimes, not necessarily every day). If and when the true values are available an autonomous learning system will be able to adjust and evolve without any direct human intervention.

1.8 Structure of the Book

The book is structured in three main parts preceded by this introductory chapter and closed by an Epilogue. This introductory chapter provided the motivation, background, a brief review of the previous and existing research work and publications in related areas as well as sets up some of the basic terminological definitions in the context of ALS.

The first part is dedicated to the systematic foundations of the methodology on which the ALS is based, including basics of probability theory (Chapter 2), pattern recognition and machine learning and especially clustering and classification (Chapter 3), the basics of fuzzy systems theory including neurofuzzy systems (Chapter 4).

Part II describes the methodology of autonomous learning systems. Chapter 5 introduces the *evolving* systems covering topics like data space partitioning, proximity measures, clustering, online input variable selection, monitoring the quality, utility and age of clusters, and so on. Chapter 6 describes the methodology for autonomous learning of the parameters of *evolving* systems stressing the difference between local and global learning methods. It also describes multi-input–multi-output (MIMO) systems, the inference mechanisms and methods for autonomous normalisation and standardisation of the data streams that the ALS processes online. In this chapter the fuzzily weighted recursive least squares (wRLS) method is described in the context of various possible learning modes. The issue of outliers and *drift* are discussed in the context of robustness.

In Chapter 7, the autonomous predictors, estimators, and filters are described. They are powerful tools for addressing time-series modelling and a range of other related problems of adaptive estimation and filtering. For example, the methodology behind one of the very interesting applications of ALS – autonomous sensors, AutoSense, is described in more detail in this chapter form the theoretical point of view and is revisited in Part III of the book from the application point of view.

Chapter 8 describes the autonomous classifiers using AutoClassify as an example that is based on evolving clustering and fuzzy rule-based systems.

Chapter 9 outlines autonomous learning controllers, AutoControl based on the concept of evolving fuzzy rule-base and the relatively old concept of indirect adaptive control.

Finally, Chapter 10 closes Part II with a discussion of collaborative ALS – a topic that has large potential for future development mainly in robotics, defence and related areas of security, surveillance, aerospace, and so on.

Finally, Part III is dedicated to various applications of the ALS with the clear understanding that the list of applications that the author and his students and collaborators have developed during the last decade is open for expansion. Indeed, a growing number of publications by other authors in the area of *evolving*, autonomous learning system, the regular IEEE conferences and events on this topic illustrate the huge potential for further growth. The adoption by leading industrial companies of these ideas demonstrates the potential which these pioneering concepts have for the Economy and the Society.

Chapter 11 describes the application aspects of AutoSense to a range of products (e.g. kerosene, gasoil, naphta) of a real large-scale oil refinery located in Santa Cruz de Tenerife, owned and run by CEPSA, Spain. One particular problem discussed in this chapter that has safety implications is the autonomous prediction of inflammability index (e.g. Pensky-Martens or Abel (Ishida and Iwama, 1984)) in real time. In the same chapter another range of application studies (courtesy of Dr. Arthur Kordon, The Dow Chemical, Texas, USA) are described. These include chemical compositions and propylene.

Chapter 12 is focused on the application issues of AutoClassify and AutoCluster in mobile robotics. The illustrative examples and video material are available at www.wiley.com/go/angelov. Both landmark identification and recognition and navigation and control subtasks were considered.

Chapter 13 describes applications of the recursive density estimation (RDE) approach to video surveillance applications (autonomous novelty detection and object tracking in video streams) that the author and his students introduced recently.

Chapter 14 provides a description of the application of the proposed ALS approach to model *evolving* user behaviour. This applies to users of computers, home appliances, the Internet, and so on. Most of the existing approaches ignore the aspect of dynamic evolution of the behaviours of the users and considers them as 'averaged' statistics very much in the sense of 'one size fits all' paradigm. The proposed ALS approach allows personalisation and learning specific users) in real time.

The book also provides a source of basic mathematical foundations used in the text, discusses the problems of real-life applications and will be very useful to be used with the software package available at www.entelsensys.com.

Additional teaching material (slides) that can be used for short courses or lectures can also be downloaded from the above website.