Preface

In many practical situations, we have experts who are skilled in doing certain tasks: expert medical doctors are skilled in diagnosing and curing diseases, professional drivers are skilled in driving cars – in particular, driving them in difficult traffic and/or weather conditions, etc. It is desirable to incorporate the knowledge of these top experts in an automatic system that would help other users perform the corresponding tasks – and, ideally, perform these tasks automatically.

Experts are usually willing to share their knowledge, but the difficulty is that in many situations, experts describe their knowledge by using imprecise ("fuzzy") words from natural language, like "small". For example, an expert driver rarely describes his or her experience in precise terms like "if the car in front slows down by 10 km/h and it is at a distance of 10 meters, you should press the break for 0.6 seconds with a force of 2.7 Newtons"; most probably, the rule described by an expert driver is "if the car in front of you is close, and it suddenly slows down some, then you should break right away". In this rule, "close" and "some" are imprecise terms: while everyone would agree that, say 100 meters is not close while 5 meters is close, there will not be a precise threshold so that before this threshold the distance is close, and a 1 cm larger distance is not close.

To describe such imprecise (fuzzy) knowledge in computer-understandable precise terms, Professor Lotfi A. Zadeh invented, in the 1960s, a new approach called *fuzzy logic*. Zadeh's ideas led to *revolutionary* changes in many control situations: from the first successful control applications in the 1970s through the fuzzy control boom in the 1990s – when fuzzy-controlled washing machines, camcorders, elevators, trains were heavily promoted and advertised – to the current ubiquity of fuzzy controllers. Just like nowadays computers are ubiquitous – companies no longer brag about computer chips in their cars, since all the cars have such chips – similarly, fuzzy control is ubiquitous: for example, in many cars, automatic transmission systems use fuzzy control.

The existing fuzzy controllers are very successful, but they have a serious limitation: they do not learn. Once the original expert rules are implemented, these same rules are used again and again, even when it becomes clear that the rules need to be updated. We still need an expert to update these rules.

There are, of course, numerous intelligent systems which *can* learn, such as Artificial Neural Networks, but from the viewpoint of the user, these systems are "black boxes": we may trust them, but we cannot easily understand the recommendations. In contrast, fuzzy rules, by definition, are formulated in terms of understandable rules. If we could make fuzzy systems themselves learn, make them automatically update the rules – this would combine the clarity of fuzzy rules with the autonomous learning ability of neural networks. This would make learning fuzzy controllers even more efficient – and therefore, even more widely used. This would lead to a *second revolution* in intelligent control.

And this revolution is starting. This book, by Dr. Plamen Angelov, one of the world leading specialists in learning fuzzy systems, is the first book that summarizes the current techniques and successes of autonomously learning fuzzy (and other) systems – techniques mostly developed by Dr. Angelov himself, often in collaboration with other renowned fuzzy researchers (like Dr. Ronald Yager). Some of these techniques have previously appeared in technical journals and proceedings of international conferences, some appear here for the first time.

Ideas are many, it is difficult to describe them all in a short preface, so let us just give a few examples. The first example is an interesting AnYa algorithm invented by *An*gelov and *Ya*ger (Anya is also a Russian short form of Anna (Anne)). In fuzzy logic, each fuzzy term like "small" is described by a *membership function*, i.e., a function that assigns, to each possible value x, the degree $\mu(x)$ from the interval $[0,1]$ to which this value is small. The value $\mu(x)=1$ means that x is absolutely small, every expert would agree to this; $\mu(x)=0$ means that x is definitely not small, while values between 0 and 1 represent expert's uncertainty.

In the traditional fuzzy control algorithms, we select a finite-parametric family of membership functions – e.g., functions which are of triangular, trapezoid, or Gaussian shapes – and adjust parameters of these functions based on the expert opinions. As a result, sometimes, the resulting membership functions provide a rather crude and not very accurate description of the expert knowledge. To improve the situation, AnYa does not limit the shape of the membership function. Instead, it uses all the value $x_1, \ldots, x_n$ that the expert believe to be satisfying the property (like "small"), and defines the desired membership function by formalizing the statement "x is close to x_1 or x is close to x_2, ...". Now, all we need to do is describe what experts mean by "close" (and by "or"), and we will no only have a well-shaped membership function, we will also have a way to update its shape when new observations appear.

A similar idea can be implemented in probabilistic terms, when we use probability density functions (pdf) $\rho(x)$ instead of membership functions, but the authors show a clear computational advantage of their fuzzy approach: A pdf is normalized by the condition that the total probability is 1: $\int \rho(x)\,dx = 1$, so we need to go through a computationally intensive process of re-normalize all its values every time we update one value of $\rho(x)$. In contrast, a membership function is usually normalized by the condition that $\max\limits_{x} \mu(x) = 1$. Thus, if we change a value of the membership function, we only need to re-normalize other values when the changed value is $\mu(x) = 1$ – and this happens rarely.

Similar ideas are used to automatically decide how to adjust the rule's conclusions, when to subdivide the original rule into two sub-rules – that would provide a more subtle description of actions, when to dismiss the old data which is no longer representative of the system's inputs, etc.

Researchers and practitioners who have been using fuzzy techniques will definitely benefit from learning how to make fuzzy systems learn automatically (pun intended :-). But this book is not only for them. Readers who are not familiar with the current fuzzy techniques will also greatly benefit: the book starts with a nice introduction that explains, in popular terms, what is fuzzy, and why and how we can use fuzzy techniques. (Some math is needed – but math taught to engineers is quite enough.)

This book is not just for the academics, practitioners will surely benefit. In the last part of the book, numerous applications are described in detail, providing the reader with an understanding of how these new methods can be used in practical situations. It may be a good idea to glance through these exciting applications first, this will give the readers an excellent motivation to grind through all the formulas and algorithms in the main part of the book.

Applications include learning sensors for chemical and petro-chemical industries – industries where the chemical contents of the input (such as oil) changes all the time, and intelligent adjustments need to be constantly made. Another successful application example is mobile robotics, where the robot's ability to learn how to navigate in a new environment – and learn fast – is often crucial for the robot's mission. The new methods have also been applied to novelty detection and object tracking in video streams, to wireless sensor networks, and to many other challenging application areas.

The second revolution – of making intelligent control systems fast learners – has started. Its preliminary results are already exciting. This book will definitely help promote the ideas of this second revolution – and thus, further improve its methods and use these improved methods to solve numerous challenging problems of today.

Vladik Kreinovich
President-Elect
North American Fuzzy Information Processing Society (NAFIPS)