# An Introduction to Machine Learning

Miroslav Kubat

# An Introduction to Machine Learning

Second Edition

Springer

Miroslav Kubat
Department of Electrical and Computer Engineering
University of Miami
Coral Gables, FL, USA

ISBN 978-3-319-63912-3 ISBN 978-3-319-63913-0 (eBook)
DOI 10.1007/978-3-319-63913-0

Library of Congress Control Number: 2017949183



Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

*To my wife, Verunka.*

# Contents

# Introduction

Machine learning has come of age. And just in case you might think this is a mere platitude, let me clarify.

The dream that machines would one day be able to learn is as old as computers themselves, perhaps older still. For a long time, however, it remained just that: a dream. True, Rosenblatt's perceptron did trigger a wave of activity, but in retrospect, the excitement has to be deemed short-lived. As for the attempts that followed, these fared even worse; barely noticed, often ignored, they never made a breakthrough—no software companies, no major follow-up research, and not much support from funding agencies. Machine learning remained an underdog, condemned to live in the shadow of more successful disciplines. The grand ambition lay dormant.

And then it all changed.

A group of visionaries pointed out a weak spot in the knowledge-based systems that were all the rage in the 1970s' artificial intelligence: where was the "knowledge" to come from? The prevailing wisdom of the day insisted that it should take the form of *if-then* rules put together by the joint effort of engineers and field experts. Practical experience, though, was unconvincing. Experts found it difficult to communicate what they knew to engineers. Engineers, in turn, were at a loss as to what questions to ask and what to make of the answers. A few widely publicized success stories notwithstanding, most attempts to create a knowledge base of, say, tens of thousands of such rules proved frustrating.

The proposition made by the visionaries was both simple and audacious. If it is so hard to tell a machine exactly how to go about a certain problem, why not provide the instruction indirectly, conveying the necessary skills by way of examples from which the computer will—yes—*learn*!

Of course, this only makes sense if we can rely on the existence of algorithms to do the learning. This was the main difficulty. As it turned out, neither Rosenblatt's perceptron nor the techniques developed after it were very useful. But the absence of the requisite machine-learning techniques was not an obstacle; rather, it was a challenge that inspired quite a few brilliant minds. The idea of endowing computers with learning skills opened new horizons and created a large amount of excitement. The world was beginning to take notice.

The bombshell exploded in 1983. *Machine Learning: The AI Approach*[1] was a thick volume of research papers which proposed the most diverse ways of addressing the great mystery. Under their influence, a new scientific discipline was born—virtually overnight. Three years later, a follow-up book appeared and then another. A soon-to-become-prestigious scientific journal was founded. Annual conferences of great repute were launched. And dozens, perhaps hundreds, of doctoral dissertations, were submitted and successfully defended.

In this early stage, the question was not only *how* to learn but also *what* to learn and *why*. In retrospect, those were wonderful times, so creative that they deserve to be remembered with nostalgia. It is only to be regretted that so many great thoughts later came to be abandoned. Practical needs of realistic applications got the upper hand, pointing to the most promising avenues for further efforts. After a period of enchantment, concrete research strands crystallized: induction of the *if-then* rules for knowledge-based systems; induction of classifiers, programs capable of improving their skills based on experience; automatic fine-tuning of Prolog programs; and some others. So many were the directions that some leading personalities felt it necessary to try to steer further development by writing monographs, some successful, others less so.

An important watershed was Tom Mitchell's legendary textbook.[2] This summarized the state of the art of the field in a format appropriate for doctoral students and scientists alike. One by one, universities started offering graduate courses that were usually built around this book. Meanwhile, the research methodology became more systematic, too. A rich repository of machine-leaning test beds was created, making it possible to compare the performance or learning algorithms. Statistical methods of evaluation became widespread. Public domain versions of most popular programs were made available. The number of scientists dealing with this discipline grew to thousands, perhaps even more.

Now, we have reached the stage where a great many universities are offering machine learning as an undergraduate class. This is quite a new situation. As a rule, these classes call for a different kind of textbook. Apart from mastering the baseline techniques, future engineers need to develop a good grasp of the strengths and weaknesses of alternative approaches; they should be aware of the peculiarities and idiosyncrasies of different paradigms. Above all, they must understand the circumstances under which some techniques succeed and others fail. Only then will they be able to make the right choices when addressing concrete applications. A textbook that is to provide all of the above should contain less mathematics, but a lot of practical advice.

These then are the considerations that have dictated the size, structure, and style of a teaching text meant to provide the material for a one-semester introductory course.

[1] Edited by R. Michalski, J. Carbonell, and T. Mitchell.

[2] T. Mitchell, *Machine Learning*, McGraw-Hill (1997).

The first problem is the choice of material. At a time when high-tech companies are establishing machine-learning groups, universities have to provide the students with such knowledge, skills, and understanding that are relevant to the current needs of the industry. For this reason, preference has been given to Bayesian classifiers, nearest-neighbor classifiers, linear and polynomial classifiers, decision trees, the fundamentals of the neural networks, and the principle of the boosting algorithms. Significant space has been devoted to certain typical aspects of concrete engineering applications. When applied to really difficult tasks, the baseline techniques are known to behave not exactly the same way they do in the toy domains employed by the instructor. One has to know what to expect.

The book consists of 17 chapters, each covering one major topic. The chapters are divided into sections, each devoted to one critical problem. The student is advised to proceed to the next section only after having answered the set of 2–4 "control questions" at the end of the previous section. These questions are here to help the student decide whether he or she has mastered the given material. If not, it is necessary to return to the previous text.

As they say, only practice makes perfect. This is why at the end of each chapter are exercises to encourage the necessary practicing. Deeper insight into the diverse aspects of the material will then be gained by going through the thought experiments that follow. These are more difficult, but it is only through hard work that an engineer develops the right kind of understanding. The acquired knowledge is then further solidified by suggested computer projects. Programming is important, too. Nowadays, everybody is used to downloading the requisite software from the web. This shortcut, however, is not recommended to the student of this book. It is only by being forced to flesh out all the details of a computer program that you learn to appreciate all the subtle points of the machine-learning techniques presented here.