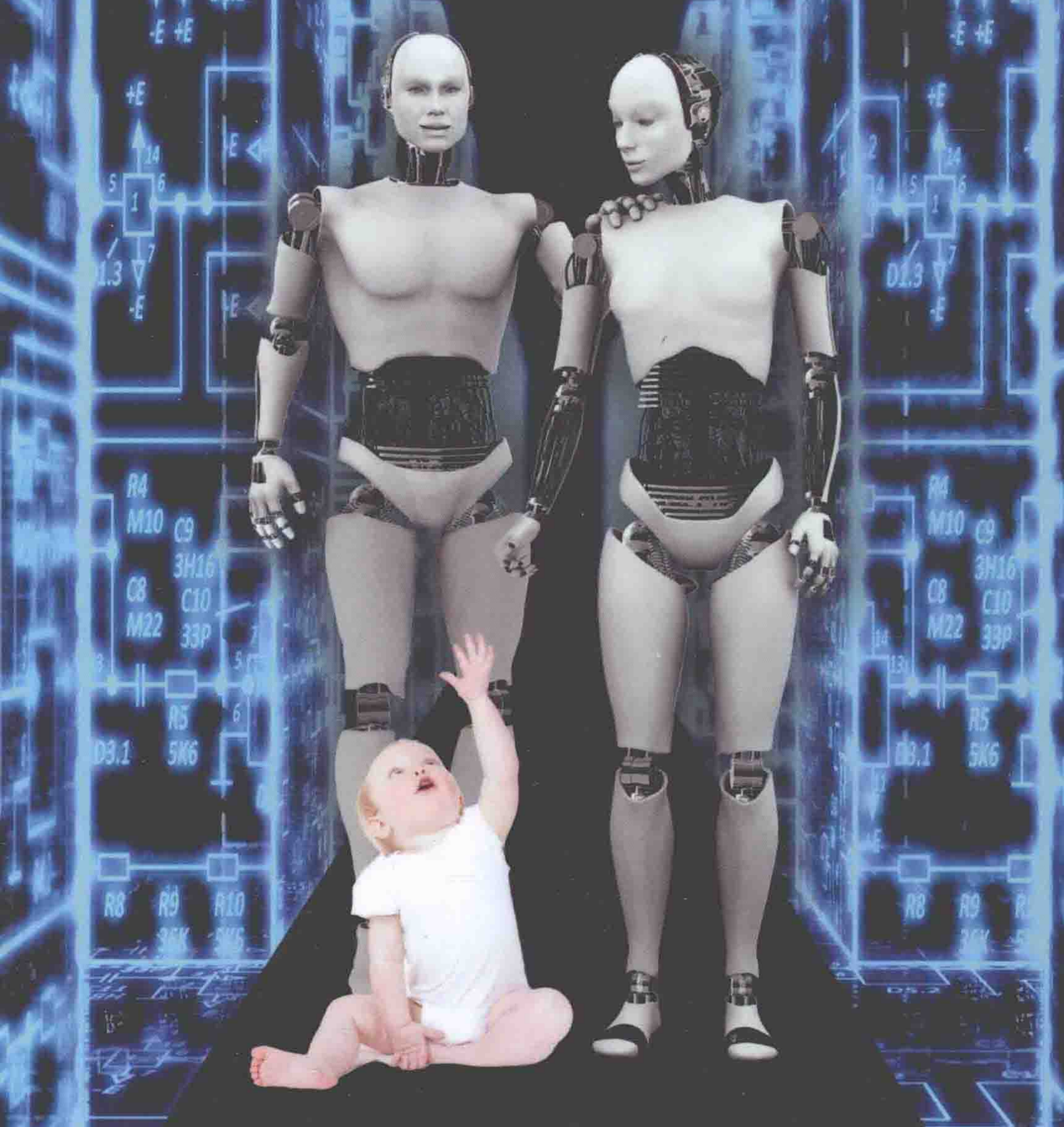
CONTEMPORARY ARTIFICIAL INTELLIGENCE



RICHARD E. NEAPOLITAN
XIA JIANG

CRC Press
Taylor & Francis Group
A CHAPMAN & HALL BOOK

CONTEMPORARY ARTIFICIAL INTELLIGENCE

RICHARD E. NEAPOLITAN
XIA JIANG

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business
A CHAPMAN & HALL BOOK

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

Printed in the United States of America on acid-free paper
Version Date: 20120612

International Standard Book Number: 978-1-4398-4469-4 (Hardback)

Library of Congress Cataloging-in-Publication Data

Neapolitan, Richard E.
Contemporary artificial intelligence / Richard E. Neapolitan, Xia Jiang.
p. cm.
Includes bibliographical references and index.
ISBN 978-1-4398-4469-4 (hardback)
1. Artificial intelligence. I. Jiang, Xia, 1967- II. Title.

Q335.N425 2012
006.3--dc23 2012017897

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

and the CRC Press Web site at
http://www.crcpress.com

CONTEMPORARY ARTIFICIAL INTELLIGENCE

Preface

Over the years my view of an artificial intelligence (AI) course has changed significantly. I used to view it as a course that should discuss our efforts to develop an artificial entity that can learn and make decisions in a complex, changing environment, affect that environment, and communicate its knowledge and choices to humans; that is, an entity that can think. I would therefore, cover the weak AI methods that failed to scale up. However, as strong methods that solved challenging problems in limited domains became more predominant, my course increasingly concerned these methods. I would cover backward chaining, forward chaining, planning, inference in Bayesian networks, normative decision analysis, evolutionary computation, decision tree learning, Bayesian network learning, supervised and unsupervised learning, and reinforcement learning. I would show useful applications of these methods. These techniques have come to be as important to a computer science student's repertoire as techniques such as divide-and-conquer, greedy methods, branch-and-bound, etc. Yet a student would not see them unless the student took an AI course. So my AI course evolved into a course that undergraduate students would take either concurrently or following an analysis of algorithms course, and would cover what I viewed as important problem-solving strategies that have emerged from the field of AI. I feel such a course should be a standard component of every computer science curriculum just like data structures and analysis of algorithms.

No text satisfied my needs for the course I taught for two reasons:

1. AI is a vast field that has included the development of many and varied techniques in the past 50 years. Current texts tried to include most of what has been going on rather than simply providing useful methods and algorithms.

2. No current text was accessible to students at a mainstream university like Northeastern Illinois University. I had this same problem with my analysis of algorithms course, and was the reason I wrote *Foundations of Algorithms.*

So I taught the course using my own Bayesian network texts and class notes. I finally decided to turn these notes into this textbook so professors at other mainstream universities could provide a similar course. This text is not meant to be an encyclopedia or history of AI, but rather is meant to be a text that can be covered in one semester; and, in the amount of time one semester allows, provide the student with what I consider the most useful techniques that have emerged from AI. These techniques include the following:

1. Logic-based methods

2. Probability-based methods

3. Evolutionary computation and methods based on swarm intelligence

4. Language understanding

The text clearly reflects my own bias. I have not discussed neural networks, fuzzy logic, support vector machines, and many other endeavors in AI. I also do not include searching because most searching techniques appear in data structures and algorithms texts. Almost half the text is about probabilistic methods. This is perhaps partially due to the fact that I know these methods best because they are my own area of research, but also due to the fact that I view them as most important (that is why they are my area of research).

I have written the material in the order I teach it. So I recommend simply covering the chapters from first to last. If there is not time to cover the entire book, I recommend leaving out Sections 8.3 through 8.7 where I have explored more advanced topics in decision analysis, and Section 12.3 which concerns causal learning. Sections marked with a ★ contain material that is inherently more difficult than the other material in the text. However, they do cover important topics and should not be skipped if the students have the sophistication to grasp them.

I thank Dawn Holmes and Kevin Korb for reading the manuscript and providing useful comments. I also thank Prentice Hall for allowing me to include excerpts from my text *Learning Bayesian Networks*, and Morgan Kaufmann for allowing me to include excerpts from my texts *Probabilistic Methods for Financial and Marketing Informatics* and *Probabilistic Methods for Bioinformatics*. Finally, I thank Relu I. Jianu for designing an attractive and thought-provoking cover.

Richard E. Neapolitan
RE-Neapolitan@neiu.edu

About the Authors

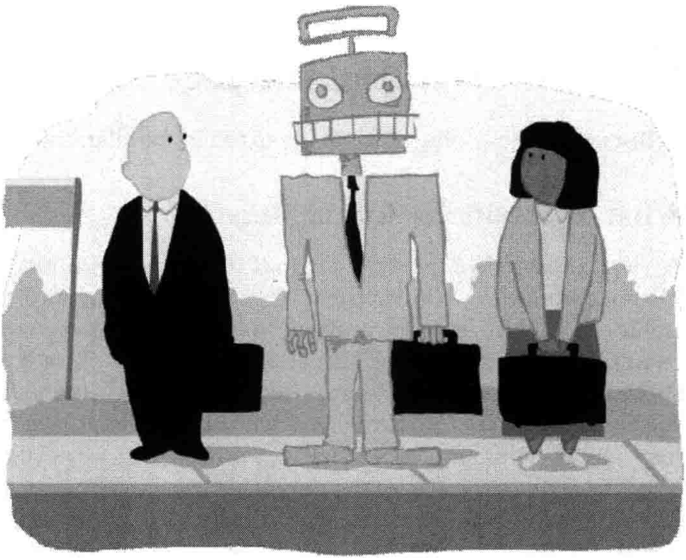
Richard E. Neapolitan is a professor of computer science at Northeastern Illinois University. His research interests include probability and statistics, expert systems, cognitive science, and applications of probabilistic modeling to fields such as medicine, biology, and finance. Dr. Neapolitan is a prolific author and has published in the most prestigious journals in the broad area of reasoning under uncertainty. He has previously written five books, including the seminal 1990 Bayesian network text *Probabilistic Reasoning in Expert Systems*; *Learning Bayesian Networks* (2004); *Foundations of Algorithms* (1996, 1998, 2003, 2010), which has been translated into three languages; *Probabilistic Methods for Financial and Marketing Informatics* (2007); and *Probabilistic Methods for Bioinformatics* (2009). His approach to textbook writing is innovative; his books have the reputation for making difficult concepts easy to understand while still remaining thought-provoking.

Xia Jiang is an assistant professor in the Department of Biomedical Informatics at the University of Pittsburgh. She has over 13 years of teaching and research experience in Bayesian network modeling, machine learning, and algorithm design. She is the co-author of the book *Probabilistic Methods for Financial and Marketing Informatics* (2007). Dr. Jiang is currently focusing on developing novel algorithms/systems that improve the computational efficiency of large data analysis, and network modeling of cancer genome data.

Contents

Chapter 1

Introduction to Artificial Intelligence



In 1990, I (Richard Neapolitan) was relatively new to the field of **Artificial Intelligence** (**AI**). At the *6th Conference on Uncertainty in Artificial Intelligence*, which was held at MIT, I met Eugene Charniak, who at the time was a well-known researcher in AI. During a conversation while strolling along the campus, I said, "I heard that the Japanese are applying fuzzy logic to AI." Gene responded "I don't believe the Japanese have been any more successful at AI than us." This comment substantiated that which I had already begun to realize, namely that AI seemed to be a dismal failure.

Dr. Charniak's comment was correct in 1990 and it is still correct today. If we consider AI to be the development of an artificial entity such as the Terminator in the movie by the same name or HAL in the classic sci-fi movie *Space Odyssey*, then we have not developed anything close. The Terminator and HAL are artificial entities that can learn and make decisions in a complex, changing environment, affect that environment, and communicate their knowledge and choices to humans. We have no such entities.

So why does the field of AI persist, and why was this book written? In their efforts to develop artificial intelligence, researchers looked at the behavior/reasoning of intelligent entities such as humans, and developed algorithms based on that behavior. These algorithms have been used to solve many interesting problems, including the development of systems that behave intelligently in limited domains. Such systems includes ones that can perform medical diagnosis, diagnose problems with software, make financial decisions, navigate a difficult terrain, monitor the possible failure of a space shuttle, recognize speech, understand text, plan a trip, track a target, learn an individual's preferences from the preferences of similar individuals, learn the causal relationships among genes, and learn which genes affect a phenotype. This book concerns these algorithms and applications. Before discussing the content of this book further, we provide a brief history of AI.

1.1 History of Artificial Intelligence

We start by discussing early efforts to define artificial intelligence.

1.1.1 What Is Artificial Intelligence?

Abandoning the philosophical question of what it means for an artificial entity to think or have intelligence, Alan Turing [1950] developed an empirical test of artificial intelligence, which is more appropriate to the computer scientist endeavoring to implement artificial intelligence on a computer. The **Turing test** is an operational test; that is, it provides a concrete way to determine whether the entity is intelligent. The test involves a human interrogator who is in one room, another human being in a second room, and an artificial entity in a third room. The interrogator is allowed to communicate with both the other human and the artificial entity only with a textual device such as a terminal. The interrogator is asked to distinguish the other human from the artificial entity based on answers to questions posed by the interrogator. If the interrogator cannot do this, the Turing test is passed and we say that the artificial entity is intelligent.

Note that the Turing test avoids physical interaction between the interrogator and the artificial entity; the assumption is that physical interaction is not necessary for intelligence. For example, HAL in the movie *Space Odyssey* is simply an entity with which the crew communicates, and HAL would pass the Turing test. If the interrogator is provided with visual information about the artificial entity so that the interrogator can test the entity's ability to perceive and navigate in the world, we call the test the **total Turing test**. The Terminator in the movie of the same name would pass this test.

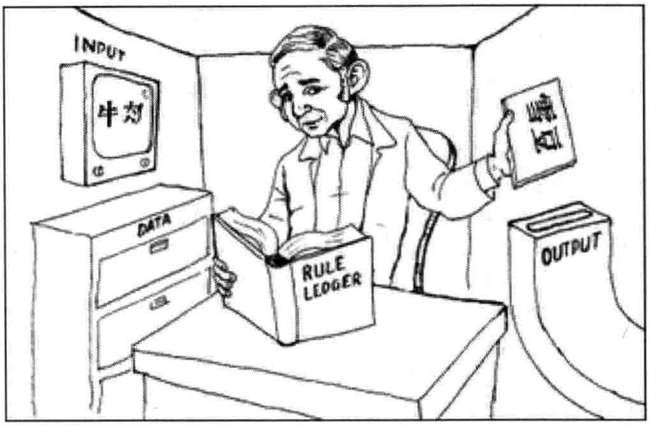


Figure 1.1 The Chinese room experiment.

Searle [1980] took exception to the Turing test with his **Chinese room** thought experiment. The experiment proceeds as follows. Suppose that we have successfully developed a computer program that appears to understand Chinese. That is, the program takes sentences written with Chinese characters as input, processes the characters, and outputs sentences written using Chinese characters. See Figure 1.1. If it is able to convince a Chinese interrogator that it is a human, then the Turing test would be passed.

Searle asks "Does the program literally *understand* Chinese, or is it only simulating the ability to understand Chinese?" To address this question, Searle proposes that he could sit in a closed room holding a book with an English version of the program, and adequate paper and pencils to carry out the instructions of the program by hand. The Chinese interrogator could then provide Chinese sentences through a slot in the door, and Searle could process them using the program's instructions and send Chinese sentences back through the same slot. Searle says that he has performed the exact same task as the computer that passed the Turing test. That is, each is following a program that simulates intelligent behavior. However, Searle notes that he does not speak Chinese. Therefore, because he does not understand Chinese, the reasonable conclusion is that the computer does not understand Chinese either. Searle argues that if the computer is not understanding the conversation, then it is not thinking, and therefore it does not have an intelligent mind.

Searle formulated the philosophical position known as **strong AI**, which is as follows:

> The appropriately programmed computer really is a mind, in the sense that computers given the right programs can be literally said to understand and have other cognitive states.
>
> — Searle, 1980

Based on his Chinese room experiment, Searle concludes that strong AI is not possible. He states that "I can have any formal program you like, but I still understand nothing." Searle's paper resulted in a great deal of controversy and discussion for some time to come (See, for example, [Harnad, 2001]).

The position that computers could appear and behave intelligently, but not necessarily understand, is called **weak AI**. The essence of the matter is whether a computer could actually have a mind (strong AI) or could only simulate a mind (weak AI). This distinction is of greater concern to the philosopher who is discussing the notion of consciousness [Chalmers, 1996]. Perhaps facetiously, a philosopher could even argue that emergentism might take place in the Chinese room experiment, and a mind might arise from Searle performing all his manipulations. Practically speaking, none of this is of concern to the computer scientist. If the program for all purposes behaves as if it is intelligent, computer scientists have achieved their goal.

1.1.2 Emergence of AI

Initial efforts at AI involved modeling the neurons in the brain. An artificial neuron is treated as a binary variable that is switched to either *on* or *off*. This notion was first proposed in [McCulloch and Pitts, 1943], and was furthered by Donald Hebb [1949] when he developed **Hebbian learning** for **neural networks**. In 1951, Marvin Minsky and Dean Edmonds built SNARC, the first neural network computer.

Following this accomplishment and Turing's development of the Turing test, researchers became increasingly interested in the study of neural networks and intelligent systems, resulting in John McCarthy[1] organizing a 2-month workshop involving interested researchers at Dartmouth University in 1956. He coined the term *Artificial Intelligence* at that workshop. Attendees included Minsky, Claude Shannon (the developer of information theory), and many others. AI emerged as a new discipline whose goal was to create computer systems that could learn, react, and make decisions in a complex, changing environment.

1.1.3 Cognitive Science and AI

Cognitive science is the discipline that studies the mind and its processes. It concerns how information is represented and processed by the mind. It is an interdisciplinary field spanning philosophy, psychology, artificial intelligence, neuroscience, linguistics, and anthropology, and emerged as its own discipline somewhat concurrently with AI. Cognitive science involves empirical studies of the mind, whereas AI concerns the development of an artificial mind. However, owing to their related endeavors, each field is able to borrow from the other.

[1] John McCarthy developed the LISP programming language for AI applications and is considered by many to be the father of AI.