

Learning with Robots

CSCI 4155/6505 2013

Thomas P. Trappenberg

Dalhousie University

Acknowledgements

These lecture notes have been inspired by several great sources, which I recommend as further readings. In particular, Andrew Ng from Stanford University has several lecture notes on Machine Learning (CS229) and Artificial Intelligence: Principles and Techniques (CS221). His lecture notes and video links to his lectures are available on his web site (<http://robotics.stanford.edu/~ang>) and he is also teaching online courses. Excellent books on the theory of machine learning are *Introduction to Machine Learning* by Ethem Alpaydin, 2nd edition, MIT Press 2010, and *Pattern Recognition and Machine Learning* by Christopher Bishop, Springer 2006. A wonderful book on Robotics with a focus of Bayesian models is *Probabilistic Robotics* by S. Thrun, W. Burgard, and D. Fox, MIT Press 2005, and the standard book on RL is *Reinforcement Learning: An Introduction* by Richard Sutton and Andrew Barto, MIT press, 1998. The popular AI, *Artificial Intelligence: A Modern Approach* by Stuart Russell and Peter Norvig, 2nd edition, Prentice Hall, 2003, does also include some chapters on Machine Learning and Robotics.

Several people have contributed considerably to the development of examples in this book. In particular, Leah Brown and Ian Graven have created early examples of Lego Mindstorm implementations in Matlab and André Reis de Geus and Jerome Verney helped considerably with some updates and Python migration of earlier versions. Chris Maxwell also helped with some implementation issues. Some recent additions and useful changes have been provided by Adrien Desies and Dominique Monnet. I also would like to thank Paul Hollensen who has solved many issues over the years, and Patrick Connor for many useful comments.

Preface

Only a few years ago it seemed like science fiction that cars would drive safely through San Francisco by themselves, that computers could search photographs for content, or that game consoles could ‘see’ your posture. The progress has been so rapid in recent years that large companies like Google and Microsoft are not only taking notice but are actively advancing and employing this field. Engineers have worked on these dreams for decades, so what has made this enormous progress possible?

Traditional science and engineering, so successful in flying men to the moon and building ever taller buildings, has long recognized that dynamic environments like city traffic or human interactions are challenging for traditional methods. In a traditional way of designing solutions, smart engineers use the laws of physics and mathematics to predict future states so that the machine can execute manipulations as desired. But the problem with dynamic and complex environments is that future states are difficult to predict for several reasons. For example, sensors might be too inaccurate or information processors or might be too slow to be able to predict environment sufficiently despite measurements. Another problem is the sheer number of possible states that we might have to consider.

Two major ingredients have been contributing to the recent success. The first is the acknowledgment that the world is uncertainty. Making this the premise from the outset has drastic consequences. For example, rather than following only the most likely explanations for a given situation, keeping an open mind and considering also other possible explanations has proven to be essential in systems that have to work in a real world environment rather than only in very controlled lab environments. The language of describing uncertainty, that of probability theory, has proven to be elegant, and probability formalisms tremendously simplify arguing in such worlds. This book is an introduction to the probabilistic formulation of machine learning.

The second ingredient for the recent breakthroughs is building system that adapt to unforeseen events. In other words, we must build learning machines since the traditional method of encoding appropriate responses to all future situations is impossible. Like humans, machines should not be static entities that only blindly follow orders which might be outdated by the time real situations are encountered. Although learning machines have been studied for at least half a century, often inspired by human capabilities, the field has matured considerably in recent years through more rigorous formulations of early learning machines and the realization of the importance of predicting previously unseen events rather than only memorizing previous events. Machine learning is now a well established discipline within artificial intelligence.

The power of the probabilistic approach to machine learning can best be acknowledged and experienced in the real world. We therefore chose to demonstrate and illuminate the theoretical constructs with robotics environments. The Lego system was chosen as it is not only one of the cheapest systems around, but the limitations of the system compared to more elaborate robots provide provide a nice challenge for smart

solutions over gloomy hardware. We also choose a high-level programming language so that we can concentrate on algorithmic programming and minimize system-level hacking.

Outline of this book

Chapter 1 provides some historical context of Artificial Intelligence, Machine Learning and Robotics. While the main scientific content of this course is Machine Learning, we will explore this fascinating area with the help of robotics. Robots provide us with data and uncertain environments that are typical of problems we are trying to solve. We also outline already the three major Machine Learning paradigms, Supervised, Unsupervised and Reinforcement learning so that we know what our targets are. Finally, we introduce some of the basic tools we will be using. This includes the mathematical construct of matrices, and the high-level programming environment called Matlab that we can use to implement the algorithms discussed in this book.

In **Chapter 2** we turn to the basic robotics tools that we will be using, the Lego NXT robot kit that we will interface with the Matlab programming environment. This chapter provides also more background and context to more traditional robotics issues. These discussion do not only provide us with more tools to be used later, but also allow us to practice some of the concepts with our system. Thus, the discussions include an introduction to basic computer vision, how to describe motions of robots, and introduces the concepts of feedback controllers.

In **Chapter 3** we will review basic probability theory since this is – as we already argued – a great language for describing uncertain sensors and environment. We will apply such concepts to motion and sensor models in order to characterize the behaviour of robots in a way that allows us to apply the concepts from probabilistic machine learning in later chapters.

Chapters 4-6 focus on Unsupervised Learning and introduces the ideas and tools that are central to most Machine Learning applications. Chapter 4 dives right down to the root of Supervised Learning in terms of (probability density) function regression. Chapter 5 dives into the Bayesian Learning world that will help us to solve an essential problems in mobile robotics, that of localization. Chapter 6 then explains popular learning machines such as Support Vector Machines and Deep Learning Networks.

In **Chapter 7** we discuss unsupervised learning. This area includes clustering and representational learning. While these concepts are less common in machine learning applications, these techniques can provide a cutting edge for solving real world problems.

Chapter 8 discusses some classical planing concepts including graph search and some more direct robotics algorithms. This chapter provides the backdrop of the discussions in the next chapter.

Chapter 9 is finally about reinforcement learning. This is an important learning paradigm which generalizes semi-supervised learning to temporal environments. It also provides a unifying language of adaptive and optimal control.

Finally, in **Chapter 10**, we discuss some non-traditional robotics that is inspired by neuroscience and general theories of the mind.

Contents

1	Background	1
1.1	Why Machines should learn	1
1.2	Classical Robotics	8
1.3	Vector and matrix notations	10
1.4	Scientific computing with Matlab	13
2	Sensing, acting and control	27
2.1	Basic computer Vision	27
2.2	Building and driving a basic Lego NXT robot	31
2.3	Pose and state space	41
2.4	Kinematics and motion models	44
2.5	Basic controllers	47
3	Probability theory and motion/sensor models	52
3.1	Random numbers and their probability density function	53
3.2	Density functions of multiple random numbers	59
3.3	Probabilistic sensor models	65
4	Probabilistic regression and maximum likelihood	68
4.1	Basic calculus and minimization	73
4.2	Minimization and gradient descent	76
4.3	Classification	81
4.4	Non-linear regression and the bias-variance tradeoff	86
5	Probabilistic reasoning and Bayes filtering	91
5.1	Multivariate generative models and probabilistic reasoning	91
5.2	Bayes Net toolbox	96
5.3	Temporal Bayesian networks: Markov Chains and Bayes filters	98
5.4	Simultaneous Localization and Mapping (SLAM)	102
6	General Learning Machines	103
6.1	The Perceptron	103
6.2	Multilayer perceptron (MLP)	106
6.3	Convolutional networks and deep learning	110

6.4	Support Vector Machine	111
6.5	SV-Regression and implementation	121
7	Unsupervised Learning	125
7.1	k-means clustering	125
7.2	Mixture of Gaussian and the EM algorithm	126
7.3	Dimensionality reduction	130
7.4	Representations and the restricted Boltzmann machine	131
7.5	Sparse representations	134
8	Navigation and Path planning	138
8.1	Making a plan by searching	138
8.2	Reactive paths: The tangent bug	139
9	Reinforcement Learning	140
9.1	Markov Decision Process	140
9.2	Temporal Difference learning	143
9.3	Function approximation and TD(λ)	145
10	Cognitive Robotics	148
10.1	Dynamic Neural Fields	148
10.2	Denosing and tracking	155
10.3	RatSLAM	157
10.4	Path integration with asymmetrical weight kernels	157
	References	162