Resources for Machine Learning

A supplement to the TRB 2019 workshop:

Big data without machine learning is just a lot of data: A guided tour to big data and machine learning

19 January 2019

ABJ70 – Committee on Artificial Intelligence and Advanced Computing Applications

David Reinke

Contents

1 Introduction

This brief guide to resources for machine learning to accompany the TRB 2019 workshop on machine learning and data put on by the TRB Committee on Artificial Intelligence and Advanced Computing Applications (ABJ70).

There are three main themes to this guide:

- What is machine learning and where can one find out more about it?
- What is the necessary background in mathematics, statistics, numerical analysis, and computing?
- What online educational resources are available?

The guide should be taken only as a starting point. We make no claim for completeness. The field of machine learning is growing at an exponential rate, as is the number of books and references on the subject. The references and other resources here are those that are capable of addressing the questions above and, in the case of some older references, that have stood the test of time as far as acceptance by the machine learning community goes.

1.1 What is machine learning?

Machine learning is a subfield of artificial intelligence whereby computers are programmed to extract patterns from raw data. We typically talk about **training** a computer on a set of data so that it can recognize patterns in the data. This field can be further subcategorized into the following areas:

- **Supervised learning** consists of trying to predict the value of an outcome based on one or more input measures. Put in more familiar terms, it is trying to predict the value of y given one or more x values. In order for this to be feasible, the data must be **labeled**: i.e., the data used to teach the computer must have an outcome variable y for each case.
- **Unsupervised learning** is applied when the data are unlabeled. In this case, the goal is to find some sort underlying pattern in the data by which they may be grouped or clustered. Because labeling data is typically expensive to do, most data are unlabeled. Hence, unsupervised learning is highly important.
- **Reinforcement learning** is a specialized case in which the only information available in the training process is "success" or "failure", for example, in playing chess or training an automated drone to fly. This field has been gaining in importance as new training techniques have been developed.

Supervised learning – which sometimes goes under the rubric **predictive analytics** – can be further subdivided into areas:

- **Classification.** In this case the *y* value takes on discrete values, or classes. A mode choice model is one example of a classification model in transportation.

- **Regression.** The *y* value can take on a continuous range of values, for example, travel time.

Machine learning can be regarded as a toolbox from which one can select an appropriate tool for a particular type of application. This is called the **no free lunch theorem** in optimization and machine learning, and can be loosely stated as follows:

> **There is no one method that works best over all possible problems**

1.2 What is in this resource guide

This guide provides a set of resources for AI that can provide a starting point for learning about machine learning. Because the field is expanding so rapidly, new resources are becoming available at an increasing rate. Hence, the emphasis has been on those resources that have had general acceptance in the field, although no claim is made for completeness.

The guide is divided into the following sections:

- **Machine learning** – references on machine learning in general and on specialized areas of machine learning

- **Background** – what is needed to learn about machine learning

- **Educational resources** – sources for online courses in machine learning and related fields

2 Machine learning

2.1 Machine learning – general

Key concepts in machine learning include:

- Feature selection
- Overfitting
- Model training and testing
- Specific machine learning methods, including Bayesian regression, regularization, neural networks, support vector machines

The texts by Hastie et al. [6], Bishop [1], and Murphy [17] are provide good coverage of the basic concepts of machine learning at a graduate school level; these are in ascending order of mathematical maturity required on the part of the reader. All three texts discuss machine learning principles, basic classification and regression techniques, and unsupervised

methods. Bishop's and Murphy's texts contain background material on probability and information theory.

At a more introductory level, the texts by James et al. [9] and Kelleher et al. [11] introduce machine learning at a more basic level. Kelleher et al. is interesting for providing different perspectives on viewing a machine learning problem. Both texts contain exercises in applications of machine learning methods.

Suthaharan's text [24] covers a number of machine learning methods and also includes extensive discussion on database methods for machine learning.

MacKay [13] gives an information-theoretic approach to machine learning; the focus is on neural networks and graphical models.

2.2 Specialized topics in machine learning

The number of specialized topics in machine learning is rapidly growing. Newer techniques are continually being developed. This section points out some references on specific topics that are covered in more detail.

Deep learning is a recent advance in machine learning that is based on neural networks. Deep learning solves many of the problems previously encountered with feature selection and overfitting. Goodfellow et al. [4] provides in-depth coverage of deep learning and includes coverage of the required mathematical and numerical analysis background.

Support vector machines (SVM) were developed as an alternative to neural networks for classification and regression. Most books on machine learning cover SVM. The book by Schölkopf, and Smola [22] is devoted entirely to the subject of SVM.

Bayesian networks are a type of probabilistic graphical model that uses Bayesian inference for probability computations. Bayesian networks aim to model conditional dependence, and therefore causation, by representing conditional dependence by edges in a directed graph that shows the relationships between variables. Modeling a complicated joint probability as a linked set of conditional probabilities will usually make a problem much easier to solve. Pearl [20] is one of the earliest, and still one of the best, references on Bayesian networks. Koller and Friedman [12] provides comprehensive coverage of probabilistic graphic models. Graphical models can include notions of **causality**; Pearl [19] provides an excellent introduction to causality and the language in which to couch causal arguments.

Reinforcement learning falls somewhere between supervised and unsupervised learning. As in supervised learning, the goal is to develop a predictive model. Unlike supervised learning, the only information available to the model training process is "success" or "failure". The recent text by Sutton and Barto [25] is a significant extension of the authors' previous book on the topic.

Boosting is a technique whereby a number of 'weak' models can be combined to build a strong predictive model. Boosting is described in Bishop [1], Murphy [17], and Hastie et al. [6]. Schapiere and Freund [21] is a comprehensive text on the topic by the inventors of the method.

Genetic algorithms have been used in machine learning to train models, especially in cases where the objective function is multi-modal. Goldberg [3] discusses the theory and application of genetic algorithms with a focus on integer encoding of variables. Haupt and Haupt [7] discuss genetic algorithms where variables are encoded with real values.

Wavelets have been used to analyze data in both time and frequency domains. The text by Strang and Nguyen [23] covers wavelets from the viewpoint of communication theory.

Minimum description length (Grunwald [5]) attempts to find maximum compression of data and a basis for model selection.

2.3 Other machine learning resources

IEEE [8] conducts extensive publishing and conference activity related to machine learning. IEEE journals related to machine learning include the following:

- *Transactions on Pattern Analysis and Machine Intelligence*
- *Transactions on Intelligent Transportation Systems*
- *Transactions on Fuzzy Systems*
- *Transactions on Big Data*
- *Computational Intelligence Magazine*
- *Transactions on Neural Networks and Learning Systems*

The *Journal of Machine Learning Research* [10] is an online journal in which articles appear as soon as they are approved for publication. Articles in the journal are frequently cited in papers on machine learning.

Microsoft Research [14] publishes their machine learning research for free online.

Several publishers have specialty areas in artificial intelligence and machine learning. These include MIT Press [15], Morgan Kaufmann [16], Cambridge University Press [2], and O'Reilly [18].

2.4 References – machine learning

1. **Bishop, Christopher M. *Pattern recognition and machine learning*.** New York: Springer, 2006. Although this book (referred to in the literature as PRML) can be regarded as an introductory machine learning text similar to Hastie et al., it is primarily directed at pattern recognition with a strong Bayesian viewpoint. Both supervised and unsupervised methods are covered.

2. **Cambridge University Press.** https://www.cambridge.org/

3. **Goldberg, David E. Genetic Algorithms in Search, Optimization, and Machine Learning**. Reading, MA: Addison-Wesley, 1989. A good explanation of genetic algorithms and why they work. The focus is on problems with integer-valued variables, which may somewhat limit its use.

4. **Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. *Deep Learning***. Cambridge, MA: MIT Press, 2016. Deep learning is a recent advance in machine learning whereby a computer can learn build up complex concepts out of simpler ones. This book provides a comprehensive introduction to the field. There is extensive background material in mathematics (linear algebra, analysis), statistics, and information theory, so the book is largely self-contained. A companion web site provides additional exercises, links to resources, and lecture slides.

5. **Grunwald, Peter D. *The Minimum Description Length Principle***. Cambridge, MA: MIT Press, 2007. In one sense, the goal of any modeling exercise should be to represent the real world as economically as possible. Minimum description Length (MDL) is an emerging method for inductive inference and machine learning that seeks to provide maximum compression of data and a basis for model selection. A useful feature of this book: various summaries and boxes that highlight the most important concepts. Although the book is self-contained, readers would benefit most by having some previous familiarity with information theory and coding such as that provided in Cover and Thomas [32] or MacKay [13].

6. **Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*.** 2nd ed. New York: Springer, 2009. An excellent introduction to machine learning methods. The book begins with a discussion of regression and classification methods from a machine learning viewpoint. Next comes a discussion of supervised learning methods including kernel methods, boosting, and neural networks; the discussion also includes an assessment of different learning methods on characteristics such as ability to deal with missing data, scalability, and predictive power. These chapters are followed by an extensive discussion of unsupervised methods including spectral methods, PCA, and clustering. The second addition has several new chapters on random forests, ensemble learning, undirected graphical models, and high-dimensional problems (p >> N).

7. **Haupt, Randy L. and Sue Ellen Haupt. *Practical Genetic Algorithms***. Hoboken, N.J.: Wiley, 2004. A practical guide to implementing genetic algorithms. Good coverage of optimization problems with real (as opposed to integer) variable values.

8. **IEEE** https://www.ieee.org/ IEEE is the world's largest technical professional organization. IEEE includes a number of societies, some of which deal with machine learning; these include the Computer Society, Computational Intelligence Society, and the Intelligent Transportation Systems Society. The IEEE Xplore Digital Library, available as a separate subscription, provides access to thousands of technical articles on machine learning.

9. **James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani**. ***An Introduction to Statistical Learning – With Applications in R*.** New York: Springer, 2013. A more elementary text than Hastie et al. [6], this book was originally written as a teaching text for a machine learning course at Stanford. Topics covered include regression, classification, resampling methods, regularization, general additive models, and unsupervised learning. Each chapter contains lab exercises with web references for supporting software and datasets.

10. ***Journal of Machine Learning Research***. http://www.jmlr.org/ Formerly a "paper journal", JLMR started publishing papers online in 2005. Papers are published online as soon as they are accepted by reviewers. It is interesting to note that JLMR has an ISI 2004 impact factor of 3.420, the highest among all machine learning journals.

11. **Kelleher, John D., Brian Mac Namee, and Aiofe D'Arcy. *Fundamentals of Machine Learning for Predictive Data Analytics*.** Cambridge, MA: MIT Press, 2015. This book provides an introduction to predictive data analytics. It was written for a course taught by the authors, and contains sufficient material for several types of courses on machine learning. Several types of approaches to learning are covered: information-based, similarity-based, probability-based, and error based. A particularly nice feature of the book is its in-depth discussion of two case studies of applications of predictive analytics. No coverage of unsupervised methods.

12. **Koller, Daphne and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*.** Cambridge, MA: MIT Press, 2009. This book provides comprehensive coverage of probabilistic graphical models. The book is divided into four main sections: representation, inference, learning, and actions and decisions. Although the book is self-contained, it requires a fair degree of mathematical capability to reap the full benefits; a good working knowledge of Bayes' theorem, conditional probabilities, and conditional independence is essential to grasping the core concepts presented in the book. Koller offers a course based on this book via the Coursera web site.

13. **MacKay, David J.C. *Information Theory, Inference, and Learning Algorithms*.** Cambridge, UK: Cambridge University Press, 2003. (Online version available for reading only at http://www.inference.phy.cam.ac.uk/itprnn/book.pdf) An interesting combination of information theory, coding theory, probability, and neural networks. This book would be a useful follow-on to Cover and Thomas [32] as well as a useful precursor to books on minimum description length, such as the ones by Grunwald [32].

14. **Microsoft Research**. https://www.microsoft.com/en-us/research/ Microsoft Research offers a number of free publications and downloads on machine learning.

15. **MIT Press.** https://mitpress.mit.edu/ MIT Press has published a number of books on artificial intelligence and machine learning.

16. **Morgan Kaufmann.** https://www.elsevier.com/books-and-journals/morgan-kaufmann Morgan Kaufmann is another publisher that

17. **Murphy, Kevin P. *Machine Learning – A Probabilistic Perspective*.** Cambridge, MA: MIT Press, 2012. A comprehensive introduction to machine learning that covers both supervised and unsupervised methods. Topics of particular interest that are not usually covered in other machine learning books include graphical models and deep learning. An accompanying web site contains data and Matlab code.

18. **O'Reilly.** https://www.oreilly.com/ O'Reilly publishes a number of books on machine learning, programming, and computer science. They have a subscription service, Safari, that provides online access to thousands of books, many of which deal with machine

learning. O'Reilly also sponsors a number of professional conferences on artificial intelligence and machine learning.

19. **Pearl, Judea.** ***Causality: Models, Reasoning and Inference*****.** 2nd ed. Cambridge, UK: Cambridge University Press, 2009. A thorough discussion of probabilistic approaches to causal modeling. Highly recommended for anybody engaged in statistical research and machine learning. This book is a useful precursor to Pearl's book on probabilistic learning [13].

20. **Pearl, Judea.** ***Probabilistic Reasoning in Intelligent Systems*****.** San Francisco: Morgan Kaufmann, 1988. One of the first, and still one of the best, books on Bayesian networks. Pearl goes into great detail explaining the differences between logic-based and probability-based approaches to machine learning.

21. **Schapiere, Robert E. and Yoav Freund.** ***Boosting: Foundations and Algorithms*****.** Cambridge, MA: MIT Press, 2014. Boosting is a method by which weighted predictions from a weak classifier can be combined to yield a high-performing classifier. This book, by the original developers of the method, goes into detail in describing a number of different approaches to boosting.

22. **Schölkopf, Bernhard and Alexander J. Smola.** ***Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*****.** Cambridge, MA: MIT Press, 2001. Thorough treatment of kernel methods and support vector machines. Both classification and regression SVMs are discussed. The problem sets at the end of each chapter add significantly to the value of this book.

23. **Strang, Gilbert and Truong Nguyen. Wavelets and Filter Banks. 2nd ed.** Wellesley, MA: Wellesley-Cambridge Press, 1996. Wavelets from an electrical engineering viewpoint.

24. **Suthaharan, Shan.** ***Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning*****.** New York: Springer, 2016. As the title says, this book focuses on algorithms for data classification. Methods include support vector machines, decision tree learning and random forests. There is also an extensive discussion of dimensionality reduction. What distinguishes this book from others in this category is the depth of coverage of data systems for handling big data; there is an entire chapter devoted to Hadoop.

25. **Sutton, Richard S. and Andrew G Barto.** ***Reinforcement Learning: An Introduction***. 2nd ed. Cambridge, MA: MIT Press, 2018. Reinforcement learning falls in-between supervised and unsupervised learning. Unlike unsupervised learning, the data are labeled; unlike supervised learning, the training results are only labeled "success" or "fail". Reinforcement learning has been used in diverse areas such as game playing (chess, backgammon) and training self-flying drones. This book remains one of the best introductions to the subject. The second edition is a revision and update of a 20-year-old classic that reflects the considerable advances in this field.

3 Tools of the trade – math, stat, programming, and all that

Machine learning draws on a number of disciplines, including several branches of mathematics, statistics, information theory, and, of course, computation. These areas are covered little if at all in typical undergraduate curricula in civil engineering and planning, although some schools are beginning to recognize the importance of data analytics in these fields.[1]

This section provides references on the background needed to understand machine learning, as well as implementation tools for machine learning. Topics include:

- Mathematical and statistical background
- Numerical analysis
- Computers and programming
- Available machine learning environments

3.1 Mathematics and statistics

The necessary mathematical background for machine learning falls into two main areas:

- Linear algebra
- Analysis

Some of the works cited above contain introductory chapters that introduce the necessary concepts. Garrity [33] (Chapters 1 and 2, and Section 6.4) can be considered a 'one-stop shop' for a quick introduction to the necessary mathematical background. The book by Mahajan [48] and its associated course on MIT OpenCourseWare provide good exercises in how to think about mathematical approaches and approximating answers.

When thinking about mathematics:

- **Focus on the concepts rather than computation.** Computers are there to do the computational work for you. What are most important for you to understand are the 'what' and the 'why' of mathematical ideas.

- **Think geometrically.** Drawing or imagining pictures, especially when dealing with linear algebra, can help you understand what is going on.

The necessary statistical background includes:

- Basic probability
- Bayesian statistics
- Information theory

[1] For example, MIT now offers an undergraduate civil engineering major with a minor in computer science. https://cee.mit.edu/majorcee_minorcs/

3.1.1 Linear algebra

The following concepts in linear algebra are important to understand for machine learning:

- Vector spaces
- Vectors and matrices, operations on vectors and matrices
- Dimension, rank, and nullspace
- Eigenvalues, singular value decomposition
- Fundamental theorem of linear algebra.

The books by Bishop [1] and Goodfellow et al. [4] have particularly good coverage of linear algebra concepts required for machine learning. Strang [23] and its associated course on the MIT OpenCourseWare site provide a thorough grounding in linear algebra.

3.1.2 Analysis

Analysis can be thought of as providing the underpinnings of calculus and further advanced mathematics. Important concepts from analysis include:

- Set theory
- Sequences and limits
- Continuity
- Manifolds

Most books on machine learning take it for granted that the reader has at least a passing familiarity with these concepts. The appendix in Schapiere et al. [21] provides adequate coverage of most of these concepts. Rudin [61] is a standard undergraduate text in analysis that provides everything and more that you will need on the subject for machine learning. Mattuck [49] is much less intense than Rudin's book, but it provides a more than adequate background for machine learning. Alcock's book [26] is intended to be a companion to an analysis course that emphasizes the 'why' more than the 'what' of analysis, but it covers enough of the basic ideas to serve as a stand-alone reference for purposes of machine learning.

3.1.3 Probability

Probability theory is at the heart of many modern machine learning methods. Key concepts to know include the following:

- Random variables: discrete, continuous; expectation and variance
- Chain rule
- Important discrete and continuous distributions, including binomial, Dirichlet, exponential, Gaussian
- Functions of random variables
- Independence
- Mixture distributions
- Conditional distributions and conditional independence

Bertsekas and Tsitsklis [28] is intended as a text for engineering students encountering probability for the first time. The texts by Bishop [1] and Murphy [17] introduce key concepts of probability and important probability distributions.

3.1.4 Statistics

Statistics in machine learning is almost entirely Bayesian. In fact, some authors, like Murphy [17], go out of their way to point out why traditional (frequentist) statistics is inappropriate for machine learning. And, as pointed out in Section 0 above, some areas of machine learning such as Bayesian networks are grounded in Bayesian statistics. For much of the 20th century Bayesian statistics was ignored or vigorously opposed by frequentist statisticians because of its supposed 'subjectivity'; McGrayne's book [52] is worth reading for a discussion of the issues between frequentists and Bayesians.

Bayes' theorem is the key concept in Bayesian statistics: i.e.,

prior knowledge + new data = better knowledge

The biggest barrier to implementing Bayesian methods was that except in special cases, there was no analytical solution to most problems. The advent of low-cost computing combined with the development of efficient numerical tools for Bayesian analysis like BUGS (now JAGS [41]) have made it possible to carry out Bayesian analysis for a wide range of problems.

Gelman et al. [34] is one of the most comprehensive works available on Bayesian analysis, and is recommended for a thorough grounding in the subject. Kruschke [46] is a good reference for those wanting to get a quick start in implementing Bayesian analysis. McElreath [51] is a text on Bayesian statistics with examples using the Stan language.

3.1.5 Information theory

Information theory deals with the quantification of information content of data. Machine learning makes use of a number of concepts from information theory, such as entropy, mutual information, and Kullback-Leibler divergence.

Cover and Thomas [32] is a standard text on information theory that provides more than the minimum needed for machine learning; Chapter 2 contains a thorough introduction to the basic concepts, and should be more than sufficient for machine learning. MacKay's text [13] discusses information theory from the perspective of machine learning. Bishop [1] and Murphy [17] each provide a brief background in information theory.

3.2 Algorithms and numerical analysis

3.2.1 Algorithms

Algorithms and data structures are at the heart of all machine learning work. This is where 'the rubber meets the road'. Corman et al. [] is one of the most comprehensive, yet accessible, works on algorithms; examples are written in pseudocode so that they can be eas-

ily implemented in the language of the reader's choice. Cormen et al. also discuss **randomized algorithms**, which often provide faster, 'almost as good' solutions that may not be achievable by traditional algorithms; Motwani and Raghavan [55] is devoted entirely to randomized algorithms. Sedgwick [63] is another useful reference on algorithms.

The series of texts by Knuth [45] has long been regarded as a classic on algorithms. The discussion on random number generation in Volume 2 is a must-read for anybody who uses random numbers in their work.

3.2.2 Numerical analysis

Almost any machine learning application will require some use of numerical analysis. As discussed below, there are software packages in machine learning that will do the numerical analysis (e.g., optimization, testing) for you. But you still need to know what is going on inside the computer so that you can fix things when they go wrong, as they inevitably will at times.

Press et al. [57] is an extensive reference on numerical analysis algorithms that shows implementations in computer code (C++ in the latest edition; earlier editions were written in C and Fortran). The book is especially valuable for discussions on traps one can fall into and how to avoid them.

The books by Nocedal and Wright [56] and Luenberger and Ye [47] cover both unconstrained and constrained **optimization** methods.

All numerical implementations will use **floating point numbers**, which can present some unpleasant surprises for the programmer who is unaware of the issues involved; for example, one should never, never use equality comparisons for floating point numbers. The paper by Goldberg [38] should be read and understood by anybody who deals with floating point numbers.

Random number generation is used in many machine learning applications. But a number of random number generators in use are faulty. Volume 2 of Knuth [45] has an extensive discussion on random number generation, including the pitfalls of some of the most popular random number generators. The GNU site on random number generation algorithms [37] is also worth accessing for a more modern discussion of the issues.

3.3 Programming, languages, and software

3.3.1 Programming

Machine learning requires programming. Nobody else can do it for you. You will have to learn how to program: this means not just hacking together some code, but writing software that will work under all conditions and can be easily maintained. Some advice based on hard experience gained over the years:

- Learn at least two programming languages. There is no one best language for all applications. Some are better at data handling; others are better for computationally-intensive applications.

- Take a software engineering approach to any code you write. It may take slightly longer to develop the original code, but the additional time will more than repay itself in reducing the effort needed for testing and debugging. And when you come back to code that you wrote six months or a year ago, you will still be able to understand what you wrote.

- Learn object-oriented programming. Object-oriented programming is the modern approach to software engineering. Among other things it provides mechanisms for data abstraction, data hiding, and developing reusable code.

- Tie in to the user communities to get answers to your problems and see examples of how other programmers work. Stack Overvflow [64] is a particularly good source of information for software and programming issues. Code Project [30] provides a number of articles and applications, many of which deal with machine learning, written in a variety of programming languages.

- Avoid language wars! A lot of ink – and electrons – have been wasted on debating which is the 'best' language. (Stack Overflow will now delete any conversations with the themes like 'my language is better than yours because …'.) Choose a language or languages that work best for you and learn them. (Of course, if you are looking for jobs in machine learning you will need to pay attention to what is being sought in the job market.) You will find that once you learn one language well it is not too difficult to pick up others.

Two very good books on programming are those by McConnell [50] and Hunt and Thomas [40]. McConnell's book is good to have at one's side when programming.

3.3.2 Programming languages

There are a number of programming languages that are being used in machine learning. R and Python are two of the top languages in data engineering. Here is a list of some of them, in no particular order:

- **R** – R is an open source version of the S language for data analysis that was developed at Bell Labs. It has excellent capabilities for statistics, data analysis, and graphics. R is especially popular in the academic environment; most statistics texts published these days are based on R. The **ggplot2** package provides one of the best data visualization capabilities in any language. There are over 10,000 add-on packages available for R, many of which deal with machine learning. The R community provides excellent support for the language, and new versions are coming out every quarter. R is available through the R web site [59].

- **Python** – Python is an object-oriented language that is rapidly becoming popular in the commercial world because of machine learning oriented features like SciPy, NumPy, and Pandas. Python can be downloaded from the Python web site [58] or as part of the Anaconda package [27], which also automatically includes SciPy, NumPy, and Pandas.

- **Java** – Java is billed as a 'write once, run anywhere' object-oriented language that is one of the most widely used languages in enterprise applications. Java source code is compiled to an intermediate language that runs on the Java Virtual Machine. The language is fairly similar to C++, although it has its own idioms. Java is maintained by Oracle, but resides on its own web site [42].

- **Scala** – Scala was designed to address some of the criticisms of Java. It has become popular for machine learning applications. It combines both functional and object-oriented programming. It is designed to work with both Java and Javascript. Scala is another language that has been popular with data scientists and machine learning specialists. Scala can be found on its web site [62].

- **Julia** – Julia is a high-level general-purpose programming language that was developed to carry out high-performance numerical analysis. It is a fairly new language – version 1.0 was released in August 2018 – but has attracted a number of high-profile users whose need for computational speed is paramount. Julia can be downloaded from its web site [43].

- **Octave** – Octave is an open-source platform based on Matlab, and has most of the functionality of Matlab. Like Matlab, it is particularly useful for matrix computations. Some machine learning developers use Octave to prototype their algorithms, which they later implement in another language (e.g., Python, R) for more efficient running. Both Linux and Windows versions of Octave are available [36].

The following languages are not designed for machine learning, but can provide useful capabilities working in conjunction with languages like those listed above.

- **C++** -- C++ is an object-oriented language based on the C language. C++ compilers are available for Windows and Linux platforms. Microsoft Visual Studio [54] includes C++. The GNU Compiler Collection [35] includes C++ for both Linux and Windows platforms. C++ can produce fast executing code, but it has some significant weaknesses that have limited its popularity for machine learning. In particular: 1) memory management is nonexistent, requiring a lot of effort on the part of the programmer to avoid memory leaks, and 2) generics in C++ are not true generics, leading to significant 'code bloat'.

- **C#** -- C# was developed as Microsoft's flagship language for developing Windows applications, although versions are now available for Linux. It is an object-oriented language that also contains a number of functional programming features. Like Java, C# source code is compiled to intermediate code that runs on a virtual machine (.NET). Although not particularly designed for machine learning, C# can be quite useful for processing data in preparation for machine learning. Among the nice features of C# are: 1) implementation of true generics, which means that generics can be part of a dynamic link library, 2) a large set of generic container classes (lists, stacks, dictionaries, hash tables, etc.), and 3) LINQ, a data query facility that can be used instead of SQL for many applications and that can be used to query a number of different types of data sources (databases, spreadsheets, text files, in-memory arrays, etc.). C# is available as part of Microsoft Visual Studio [54].

- **C** – C is a systems programming language that was developed at Bell Labs in the late 1960s and early 1970s. C is a procedural language, unlike object-oriented languages like C++ and Java. It is one of the most widely used languages in the world, and is used on a number of different platforms. The GNU Compiler Collection [35] and Microsoft Visual Studio [54] both contain C compilers. C can be used to write efficient numeric and data processing routines that can be called from languages like Python or R. Because it provides access to the internal workings of a computer, it is useful to think of C as essentially a high-level assembly language.

3.3.3 Machine learning environments

Recently several machine learning environments have become available for general use:

- **TensorFlow** [66] was developed by Google. It includes capabilities for deep learning and reinforcement learning. APIs are available in a number of languages including Python (most well developed) and C++.

- **Microsoft Cognitive Toolkit** [53] contains tools for a number of different machine learning applications including neural networks and time series.

- **Keras** [44] is a Python-based tool to support convolutional neural networks and deep learning. It can now work in conjunction with TensorFlow.

3.4 References – tools of the trade

26. **Alcock, Lara.** ***How to Think About Analysis***. Oxford, UK: Oxford University Press, 2014. This book was written to help out undergraduates taking their first course in analysis. It is intended to be a precursor to, or a companion for, an analysis course. That said, the book provides good coverage of elementary concepts such as limits and continuity. It may be a good alternative if one would rather not slog through more detailed treatments such as Rudin [61].

27. **Anaconda**. https://www.anaconda.com/ Anaconda is a downloadable platform for R and Python that includes additional tools such as Jupyter Notebook and other utilities.

28. **Bertsekas, Dmitri P. and John N. Tsitsklis.** ***Introduction to Probability***. 2nd ed. Belmont, MA: Athena Scientific, 2008. An introduction to probability that requires some familiarity with single-variable and multivariable calculus. This is a companion text to an introductory course on probability at MIT, available on edX [68] or the MIT OpenCourseWare site at https://ocw.mit.edu/resources/res-6-012-introduction-to-probability-spring-2018/index.htm?utm_source=OCWDept&utm_medium=CarouselSm&utm_campaign=FeaturedCourse .

29. **Buuren, Stef van.** ***Flexible Imputation of Missing Data***. Boca Raton, FL: CRC Press, 2012. Missing data are the norm, rather than the exception in the real world. Van Buuren provides an up-to-date guide on the latest data imputation methods. Contains numerous examples in R.

30. **CodeProject**. https://www.codeproject.com/ Contains practical examples if coding in a number of different programming languages. There is a section on artificial intelligence.

31. **Cormen, Thomas H. et al. Introduction to Algorithms. 3rd ed.** Cambridge, MA: MIT Press, 2009. This book has evolved into the quintessential introductory text on algorithms. Provides an encyclopedic coverage of most algorithms used in computer science. Extensive coverage of graphical and multithreaded algorithms.

32. **Cover, Thomas M. and Joy A. Thomas.** ***Elements of Information Theory***. 2nd ed. New York: Wiley, 2006. Information theory is an important foundation for many areas of statistics and machine learning. This book by Cover and Thomas provides an excellent introduction to information theory and goes into considerable detail in some areas. Several topics in the book that are relevant to machine learning include information theory and statistics, Kolmogorov complexity, and data compression (relevant to minimum description length).

33. **Garrity, Thomas A.** ***All the Mathematics You Missed: But Need to Know for Graduate School.*** Cambridge, UK: Cambridge University Press, 2002. An excellent survey of various branches of mathematics, written mainly for graduate mathematics majors but also useful for graduate students in engineering and statistics. Covers the basic concepts in a number of areas useful for machine learning, including linear algebra, analysis, algorithms, and probability. The book provides an extensive guide – alas, now somewhat dated – to references for more in-depth follow-up reading on specific topics.

34. **Gelman, Andrew, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin.** ***Bayesian Data Analysis***. 3rd ed. Boca Raton, FL: CRC Press, 2014. A comprehensive text on Bayesian data analysis that covers all aspects of Bayesian data analysis, including modern approaches to Bayesian computation. The latest edition adds three new chapters on nonparametric models. The authors also provide access to STAN, a new Hamiltonian Monte Carlo package for carrying out numerical Bayesian analysis.

35. **GNU compiler collection (GCC).** https://www.gnu.org/software/gcc/ This collectin includes front ends for C, C++, Objective-C, Fortran, Ada, and Go, as well as libraries for these languages. Note that source code for some open source programs like R still have routines written in Fortran.

36. **GNU Octave.** https://www.gnu.org/software/octave/ GNU Octave is a freeware matrix processing environment that is largely compatible with Matlab. Includes both a GUI and command-line interfaces.

37. **GNU web page on random number generator algorithms.** https://www.gnu.org/software/gsl/manual/html_node/Random-number-generator-algorithms.html Random number generation is crucial to machine learning, statistical analysis, and simulation. Yet it is probably safe to say that many random number generators in use today are not very "random" at all. (For example: linear congruential random number generators are the most frequently used, yet are not capable of producing high-

quality streams of random numbers.) This page, part of the GNU Scientific Library, provides a link to two high-quality random number generators: the Mersenne Twister, and L'Ecuyer's WELL random number generator.

38. **Goldberg, David. What Every Computer Scientist Should Know About Floating-Point Arithmetic.** (Available on the Internet in several locations including the following: http://faculty.tarleton.edu/agapie/documents/cs_343_arch/papers/1991_Goldberg_FloatingPoint.pdf). Not knowing what goes inside the computer can, and often does, lead to problems for the user. This article goes into great detail on floating point representation, particularly the IEEE 754 standard, which is in common use today.

39. **Golub, Gene H. and Charles F. Van Loan. *Matrix Computations***. 4th ed. Baltimore: Johns Hop-kins University Press, 2013. A classic on implementing matrix computations. The latest edition covers parallel algorithms.

40. **Hunt, Andrew and David Thomas. The Pragmatic Programmer – *From Journeyman to Master***. Boston: Addison-Wesley, 2000. Exactly what it says. Covers many of the hands-on issues in code development that most books on the topic ignore.

41. **JAGS – Just Another Gibbs Sampler.** http://mcmc-jags.sourceforge.net/ JAGS is an open-source cross-platform implementation of the BUGS (Bayesian Analysis Using Gibbs Sampling) language. Versions are available for Windows, Linux, and Mac OS X. Both R and Python have packages that can interface with JAGS.

42. **Java programming language.** https://java.com/en/ Downloads for various versions of Java and programming environments.

43. **Julia Language**. http://julialang.org/ Julia is a high-level, high-performance dynamic programming language for technical computing that was created at MIT. The language is still in early development, but early testing shows that its speed of performance is superior to alternatives such as R and Matlab. It is designed to work with IJulia, an interactive graphical interface to Julia that was created in collaboration with the Jupyter project.

44. **Keras deep learning library.** https://developer.ibm.com/articles/cc-get-started-keras/ Keras is an open-source Python-based library of deep learning tools that can interface with TensorFlow [66].

45. **Knuth, Donald E. The Art of Computer Programming.** Reading, MA: Addison-Wesley. 4 vols. Volume 1: Fundamental Algorithms. Volume 2: Seminumerical Algorithms. Volume 3: Sorting and Searching. Volume 4A: Combinatorial Algorithms, Part 1. This comprehensive work has been regarded in the computer science world as the bible of algorithms. Volume 2 is particularly useful for its coverage of random number generation. One problem with this series is that the coding examples in volumes 1 – 3 are written in a 1960s-style assembly language created by the author, so they might be difficult to understand; Ruckert [60] rewrites these examples in a modern RISC-based assembly language, and is recommended as a supplement.

46. **Kruschke, John K.** ***Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan***. 2nd ed. Boston: Academic Press, 2015. As the title says, this is a tutorial on how to do Bayesian analysis, with an emphasis on coding Bayesian models in JAGS and Stan. Nevertheless, the author does not stint on the necessary background and rationale behind Bayesian analysis. The discussion on Bayesian vs frequentist analysis is particularly useful because it points out a number of areas where frequentists mistakenly assign a Bayesian interpretation to frequentist measures such as p-values and confidence intervals. The book is strongly oriented toward R, but it is still be useful for those doing Bayesian analysis in other environments such as Python.

47. **Luenberger, David G. and Yinyu Ye. Linear and Nonlinear Programming. 4th ed.** New York: Springer, 2008. This latest update of Luenberger's classic now contains an introduction to interior point methods. Luenberger provides extensive discussion on unconstrained optimization methods, carefully pointing out the advantages of each.

48. **Mahajan, Sanjoy.** ***Street-Fighting Mathematics: The Art of Educated Guessing and Opportunistic Problem Solving***. Cambridge, MA: MIT Press, 2010. A delightful book, written to encourage thinking about mathematical approaches rather than engaging in rigorous proofs. Ideal for engineers. An online version of this book is available on the MIT OpenCourseWare site for a course by the same name (MIT course18.098 / 6.099).

49. **Mattuck, Arthur P.** ***Introduction to Analysis.*** San Francisco: Pearson, 1998. Most analysis texts are written for mathematics majors. This text was written with non-mathematicians – e.g., physicists and engineers – in mind. The pace is slow and careful, but the book covers much of the material that can be found in more advanced texts like Rudin [61].

50. **McConnell, Steve. Code Complete. 2nd ed.** Redmond, WA: Microsoft Press, 2004. This is by far the best book on code construction, including variable naming, statement organization, program organization, design, and debugging. If you follow its precepts you will find that you will end up developing programs much quicker with much less debugging. And when you return to something you might have written several years before, you will be able to recognize what you wrote. No matter what language you write in, this book will help you. This is the one book that you should keep by your side when programming.

51. **McElreath, Richard.** ***Statistical Rethinking: A Bayesian Course with Examples in R and Stan.*** Boca Raton, FL: CRC Press, 2014. This book is oriented toward

52. **McGrayne, Sharon Bertsch.** ***The Theory That Would Not Die: How Bayes' Rule Cracked the Enigma Code, Hunted Down Russian Submarines, & Emerged Triumphant from Two Centuries of Controversy***. New Haven, CT: Yale University Press, 2011. This is a history book, not a text. What makes this book interesting is its narrative on how Bayesian statistics finally emerged from behind the shadows of frequentist statistics to take its place as the mainstream approach to statistics in the 21st century.

53. **Microsoft Cognitive Toolkit**. https://www.microsoft.com/en-us/cognitive-toolkit/ An open-source machine learning toolkit with interfaces in a number of languages including Python, C++, and .NET. Includes a variety of methods such as neural networks and time series analysis.

54. **Microsoft Visual Studio Community Edition**. https://visualstudio.microsoft.com/vs/community/ A fully functional free version that is available for use by academics, individual users, and small business only. Includes all the tools and languages in Microsoft Visual Studio Professional Edition, including C#, C++, and F#.

55. **Motwani, Rajeev and Prabhakar Raghavan. Randomized Algorithms.** Cambridge, UK: Cambridge University Press, 1995. For many applications, a randomized algorithm can be the simplest and fastest way to reach a near-optimal solution. This book provides a solid introduction to the topic, with example algorithms.

56. **Nocedal, Jorge and Stephen J. Wright. *Numerical Optimization***. 2nd ed. New York: Springer, c2006. Nonlinear optimization algorithms including non-derivative methods and interior-point methods.

57. **Press, William H., Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes: The Art of Scientific Computing.* 3rd ed. Cambridge, UK: Cambridge University Press, 2007.** (Earlier versions of the book can be found at http://numerical.recipes/.) This massive book is exactly what it says: a set of recipes for doing numerical computations. Reflecting the authors' backgrounds, the book is mainly toward an audience with a physics background. That said, the book covers a number of useful topics for developing machine learning programs including: matrix and vector computations, equation solving, optimization, classification algorithms and hidden Markov models (although this book should not be the primary reference for learning about these topics), Markov chain Monte Carlo, random number generation, and wavelets. The discussions in the book include coverage of how well the various algorithms work under different circumstances and how to avoid getting into computational trouble. The coding examples in the book are its biggest strength and its biggest weakness – they show how one might implement an algorithm, but they are poorly written.

58. **Python programming language.** https://www.python.org/ The main site for downloading Python. Contains links to references for learning about Python.

59. **R Project**. https://www.r-project.org/ This is the home page for the R project. If you aren't using R for machine learning, you should consider doing so. R has over 10,000 user-contributed add on packages, many of which deal with machine learning, e.g.: rattle, a user interface for developing machine learning models; e1071, a package of tools for machine learning; and kernlab, which contains functions for SVMs and RVMs. Support for R is available on the R list server (but be prepared for some snarky comments if you are asking a question that shows you didn't bother to use the help facility in R); other support can be found through several R groups on LinkedIn and in Stack Overflow.

60. **Ruckert, Martin.** ***The MMIX Supplement: Supplement to The Art Of Computer Pro-gramming, Volumes 1, 2, 3 by Donald E. Knuth***. Upper Saddle River, NJ: Addi-son-Wesley, 2015. Volumes 1-3 of Knuth [45] contain examples written in a 1960s-style assembly language, which is difficult to understand by modern standards. This book contains a rewrite of all those coding examples in MMIX, a RISC-based assembly lan-guage that is similar to modern assembly languages.

61. **Rudin, Walter.** ***Principles of Mathematical Analysis***. 3rd ed. New York: McGraw-Hill, 1976. This is a venerable reference that has stood the test of time and is a standard text in many upper-division courses on real analysis. The book covers standard topics such as limits, compactness, implicit function theorem, Lebesgue measure and integra-tion, and Stokes's theorem.

62. **Scala programming language.** https://www.scala-lang.org/ Scala is a programming language that runs on the Java Virtual Machine. It has a number of libraries specifi-cally for machine learning.

63. **Sedgewick, Robert.** ***Algorithms in Java***. Boston: Addison-Wesley, 2003. There are similar books by this author written for C and for C++. Covers basic algorithms, includ-ing graphs.

64. **Stack Overflow**. https://stackoverflow.com/ Stack Overflow is a general discussion web site that covers a wide range of topics on computing including programming languages, software development, and machine learning.

65. **Strang, Gilbert.** ***Introduction to Linear Algebra***. 5th ed. Wellesley, MA: Wellesley-Cambridge Press, 2016. An excellent introductory text on linear algebra that provides an excellent background in linear algebra for machine learning. Important concepts are carefully introduced and clearly explained. Includes examples of applications of linear algebra, including optimization and regression analysis. This is a companion text to Strang's course 18.06 on the MIT Open CourseWare site: https://ocw.mit.edu/courses/mathematics/18-06sc-linear-algebra-fall-2011/

66. **TensorFlow**. https://www.tensorflow.org/ TensorFlow is an open source machine learn-ing library created by Google. Numerical computations are represented by data flow graphs, where nodes represent mathematical operations and the graph edges represent the data arrays (tensors) communicated between nodes. Keras [] is a high-level interface to TensorFlow. The Python API is the most developed and tested of the available APIs, but other languages such as C, C++, and Java can be used as well to work with Tensor-Flow.

4 On-line learning

4.1 On-line learning sites

On-line learning is one of the best ways to learn about machine learning and the necessary background. Typical course content includes video lectures, short quizzes after each section,

and possibly intermediate or final exams. Courses offer official certificates of completion for a small fee; users who do not with to pay the fee can usually audit the courses for free.

Some courses follow a schedule and are only offered for a limited amount of time; although course content may be available in archived form later on. Other courses are self-paced and available on demand.

Coursera [67] and edX [68] in particular have a number of useful courses on machine learning, computer science, statistics, and mathematics. Udacity [71] is a commercial site that usually requires a fee for its courses.

MIT began its OpenCourseWare site [70] more than 15 years ago. Most courses on machine learning are taught in the Electrical Engineering and Computer Science Department, although machine learning courses are available from other department such as Management. Several courses in Mathematics and Economics departments may also be useful for the mathematical and statistical background they provide.

The Khan Academy [69] was created mainly to assist secondary school students. Some of its courses such as calculus and introductory computer science may be useful for those who require additional background review or study in mathematics or computer science.

4.2 References – online learning

67. **Coursera**. https://www.coursera.org/ A joint effort by a number of universities around the world (including Stanford, Princeton, and Brown), Coursera offers a number of online courses on line in machine learning, mathematics, statistics, and programming. Andrew Ng (Stanford) has a particularly good self-paced introduction to machine learning.

68. **edX**. https://www.edx.org/ Another on-line cooperative effort of universities around the world, including MIT, Cal, and Harvard; Microsoft is also part of the consortium. A somewhat broader offering of courses than Coursera.

69. **Khan Academy**. https://www.khanacademy.org/ Has a number of math and computer courses mainly at a secondary school level, including calculus and computer science. Nonetheless, some courses may be useful for those who wish to use machine learning but lack the necessary mathematical background.

70. **MIT OpenCourseWare**. https://ocw.mit.edu/index.htm MIT began to put its courses online for free beginning in 2002, becoming the first institution of its kind to do so. Now more than 1,200 courses are now available from all MIT departments. Course content varies: some offer only readings and problem sets, while others have video lectures available. (Some of these courses are offered on the edX site [68] as well, with much better coverage and course content.) Most machine learning courses are in the Electrical Engineering department, although other departments – including Economics, Management, and Mathematics – also have courses related to machine learning.

71. **Udacity**. https://www.udacity.com/ Udacity is a private venture, so many of its courses are fee-only. Udacity offers "nanodegrees" in a variety of areas, including machine

learning. One of their more interesting courses is on how to build a robot car, reflecting the interest of one of its founders, Sebastian Thrun.