The Book Review Column [1]
by **Nicholas Tran** (ntran@scu.edu)
Department of Mathematics & Computer Science, Santa Clara University

1 Notable new releases

Graph Theory in America: The First Hundred Years (Princeton University Press, 2023) by Robin Wilson, John J. Watkins, and David J. Parks chronicles the first hundred years of graph theory in North America, starting with the arrival of James Joseph Sylvester at Johns Hopkins University in 1876 and ending with the proof of the four color theorem by Kenneth Appel and Wolfgang Haken in 1976.

Computability and Complexity: Foundations and Tools for Pursuing Scientific Applications (Springer, 2024) by decorated author Rod Downey is a new undergraduate textbook that covers advanced topics such as parameterized complexity, generic case complexity, and smoothed analysis.

How to Think about Algorithms, 2nd ed. (Cambridge University Press, 2024) by Jeff Edmonds aims to teach students to think like a designer of algorithms. This new edition includes a short chapter on machine learning algorithms and more than 150 new exercises. A review of the first edition of this book appeared in this column in 2009.

Differential Privacy (The MIT Press, 2025) by Simson L. Garfinkel, a senior data scientist at the US Department of Homeland Security, explains the algorithmic technology used by Google, Apple, and the US Census Bureau to protect the privacy of individuals in large datasets.

2 This column

If you teach Automata Theory, you will want to check out Bill Gasarch's reviews of *The Logical Approach to Automatic Sequences: Exploring Combinatorics on Words with Walnut* by Jeffrey Shallit and *200 Problems on Languages, Automata, & Computation* by Filip Murlak, Damian Niwiński, and Wojciech Rytter. These thought-provoking books may very well change the way you teach your course.

Serious students of data science should read Chinmay Hegde's review of *Mathematical Analysis of Machine Learning Algorithms* by Tong Zhang. This rigorous and comprehensive book is an important resource for those who want to understand the mathematical underpinnings of machine learning algorithms.

3 How to contribute

Don't wear white after Labor Day! Read a book and write a review for SIGACT News instead. Either choose from the books listed below, or propose your own. In either case, the publisher will send you a free copy of the book. Guidelines and a LaTeX template can be found at `https://algoplexity.com/~ntran`.

BOOKS THAT NEED REVIEWERS FOR THE SIGACT NEWS COLUMN

Algorithms, Complexity, & Computability

1. Chen, H. (2023). *Computability and Complexity.* The MIT Press.
2. Ferragina. P. (2023). *Pearls of Algorithm Engineering.* Cambridge University Press.
3. Downey, R. (2024). *Computability and Complexity: Foundations and Tools for Pursuing Scientific Applications.* Springer.
4. Edmonds, J. (2024). *How to Think about Algorithms, 2nd ed.* Cambridge University Press.

Miscellaneous Computer Science & Mathematics

1. Grechuk, B. (2019) *Theorems of the 21st Century.* Springer.
2. Nahin, P. (2021). *When Least Is Best: How Mathematicians Discovered Many Clever Ways to Make Things as Small (or as Large) as Possible.* Princeton University Press.
3. Chayka, K. (2024). *Filterworld: How Algorithms Flattened Culture.* Doubleday.
4. Valiant, L. (2024). *The Importance of Being Educable: A New Theory of Human Uniqueness.* Princeton University Press.

Discrete Mathematics and Computing

1. Ross, S., & Peköz, E. (2023). *A Second Course in Probability.* Cambridge University Press.

Cryptography and Security

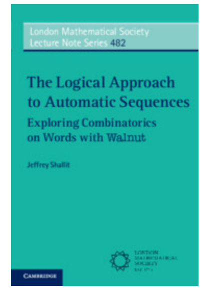
1. Garfinkel, S. (2025). *Differential Privacy.* The MIT Press.

Combinatorics and Graph Theory

1. Landman, B., Luca, F., Nathanson, M., Nešetřil, J., & Robertson, A. (Eds.). (2022). *Number Theory and Combinatorics: A Collection in Honor of the Mathematics of Ronald Graham.* De Gruyter.

2. Wilson, R., Watkins, J., & Parks, D. (2023). *Graph Theory in America: The First Hundred Years*. Princeton University Press.

Review of [1]



The Logical Approach to Automatic Sequences:
Exploring Combinatorics on Words with `Walnut`
by **Jeffrey Shallit**

Cambridge University Press, 2022
Paperback, $90.00; eBook, $90.00; 374 pages

Review by **William Gasarch** (gasarch@umd.edu)

1 Introduction

This book is concerned with the following question:

Given a sequence s, does it have property p?

That is a rather broad question. What kind of sequences are allowed? How are they to be given? What kind of property is allowed? How are they given?

Here is an example of a question considered in the book.

Sequence Let

$X_1 = 1$

$X_2 = 0$

For all $i \geq 3$, $X_i = X_{i-1}X_{i-2}$ (concatenation).

The *Infinite Fibonacci Sequence* is $\lim_{i\to\infty} X_i$.

Question Is the following true: For every n there is a palindrome of length n that is a subword of the Infinite Fibonacci Sequence.

You might approach this by generating by hand (say) the first 20 bits. Or perhaps write a program to generate the first (say) 100 bits. And then see how it looks. You might write a program that generates the first (say) 1000 bits and then tries to find palindromic subwords of length $1, 2, 3, \ldots$ and see how far you get. These would all be aiming at a human-readable proof where the computer *helps you* to find the answer, and perhaps the proof.

Or you might just ask some program for the answer directly.

Really? Can we do that? Is there an algorithm to do that? Is it fast enough to be worth coding up? Did someone code it up? Is there a user's manual for it? Is there a book that ties all of this together so it tells you what such a package can do, why it works, and how to use it? The answers are yes, yes, yes, yes, and yes.

Véronique Bruyère et al. [2] describe an algorithm for this problem. The algorithm is fast enough so that it could be coded up. Hamoon Mousavi [4] has coded it up in a package called **`Walnut`** , which is freely available. The reference given is to a user's manual. The book under review ties it all together.

More precisely, `Walnut` is a computer package that will, given a sequence (we will say what kind of sequence and how it is given later) and a question about that sequence (we will say what kind of question and how it is given later) outputs the answer to that question.

This book does the following:

1. Gives many sequences, questions about sequences, and the answers.
2. Describes the theory behind `Walnut` including the proofs of decidability when needed.
3. Instructs the reader on how to use `Walnut`. The reader can then use `Walnut` and investigate questions about sequences.

2 What Kind of Sequences Can You Ask About?

We first define a variant of DFAs that also has an output.

Definition A *Deterministic Finite Automata with Output (DFAO)* is a tuple $(Q, \Sigma, \Gamma, s, \delta, \tau)$ such that

1. Q is a finite set of states.
2. Σ and Γ are finite alphabets.
3. $s \in Q$.
4. $\delta : Q \times \Sigma \to Q$.
5. $\tau : Q \to \Gamma$
6. If $x \in \Sigma^*$ then we can run it through the DFAO in the usual way and get to a state q. Note that we do not have a notion of accept state. Instead we think of x as being mapped to $\tau(q)$. Hence a DFAO computes a function from Σ^* to Γ.
7. Let $k \in \mathsf{N}$. If $\Sigma = \{0, \ldots, k-1\}$ then we interpret the input as a base-k number and hence the DFAO can be identified with a function from N to Γ. (Alternatively the DFAO can be identified with an infinite sequence of elements of Γ.) We call such a DFAO a *k-DFAO*.

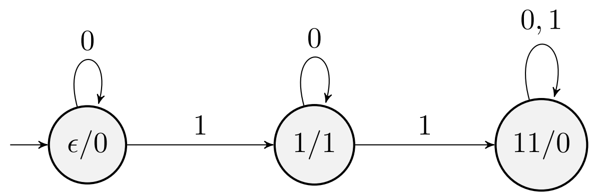
We now define the type of sequence that `Walnut` can be asked about.

Definition Let Γ be a finite alphabet. A *k-automatic sequence* is an element $x \in \Gamma^\omega$ such that there is a k-DFAO that, on input n (a base-k number), ends in a state labeled $x(n)$.

Example Let x be the characteristic sequence of the powers of 2. So the sequence begins

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	1	1	0	1	0	0	0	1	0	0	0	0	0	0	0	1

The following is a 2-DFAO for this sequence. Each state has the name of the state (one of $\{\epsilon, 1, 11\}$) and then the label (one of $\{0, 1\}$). On input a number n in base 2, the DFAO ends in a state labeled 1 iff the nth bit of the sequence is 1 iff n is a power of 2.



Recall that `Walnut` takes as input a sequence and question about that sequence. The way to input a sequence is by inputting the DFAO for it.

3 What Kinds of Questions Can You Ask?

We define an extension of Presburger Arithmetic.
Definition We define formulas inductively. Note that there are two kinds of variables.

1. A *term* is either (1) a constant $c \in \mathsf{N}$, (2) a variable a (the domain of a is N), (3) if s, t are terms then $s + t$ is a term. (4) $x[t]$ where t is a term (x represents a sequence of natural numbers, so, for example, this could be $x[3]$ or $x[i]$ or $x[i+j]$). Terms 1, 2, 3 are in Presburger arithmetic. Term 4 is the extension.

2. If s, t are terms then $s = t$ and $s < t$ are atomic formulas.

3. If $\phi_1(x_1, \ldots, x_n)$ and $\psi(y_1, \ldots, y_m)$ are formulas then the following are formulas:

   (a) $\neg\phi_1(x_1, \ldots, x_n)$.

   (b) $\phi_1(x_1, \ldots, x_n) \vee \psi(y_1, \ldots, y_m)$. (We can write $\wedge$ by using De Morgan's law.)

   (c) For $1 \le i \le n$, $(\exists x_i)[\phi_1(x_1, \ldots, x_n)]$. (We can write $\forall$ by using $\forall = \neg\exists\neg$.)

4. A *Sentence* is a formula with no free variables.

5. A *Question* is a formula with only one variable of sequence type, and that variable is free, together with what sequence you want. For example if you want to know if the Thue-Moore Sequence is cube-free you would have the question with x as the sequence, and another place where you would specify that x is the Thue-Moore Sequence.

Note We have omitted other extensions where additional functions are added.
Examples of Questions

1. $(\forall i)(\exists j)[(j > i) \wedge x[i] = 0]$.

   This question asks if the sequence x is 0 infinitely often.

2. $(\exists n)(\exists i)[n \ge 1 \wedge [(\forall j)[j < n \rightarrow x[i+j] = x[i+j+n]]]$.

   This question asks if the sequence x has a square, which is a subword of the form yy.

The following Theorem is the key to `Walnut`.
Theorem The following problem is decidable: Given a sequence via the DFAO (the alphabet is base-k digits for some k) and a question, determine if the answer to the question is yes.

4 Pros and Cons of This Approach to Mathematics

Hilbert (in today's language) wanted an algorithm that would, given a problem in math, output the answer. Gödel showed this could not be done. However, there were some subsets of math that were decidable (e.g., Presburger arithmetic). Are these systems useful for finding out answers to math problems? Before reading this book I would have answered No and made the following points:

1. You cannot state anything of interest in these systems. I give one exception; however, my second point will negate it. Rabin showed S2S is decidable [5] (see Gurevich and Harrington [3] or the book by Börger, Erich Grädel, and Yuri Gurevich [1] for an easier proof). Rabin points out that the theorem of Wolfe stating "Every Gale-Stewart game with a set in F_σ is determinate" is expressible in S2S. This theorem came out around 10 years before Rabin's paper (there are a few other math problems of interest that can be stated in S2S). If Wolfe's result was still an open problem then, could the decidability of S2S have solved it?

2. Many theories that are decidable have terrible run times and are quite complicated to code up. I doubt that S2S has ever been coded up.

3. One counter-thought: The theory WS2S has been coded up and has been used to verify hardware and low-level code. See here: `https://www.brics.dk/mona/`. While this is of course interesting, it's not proving *mathematical* theorems of interest.

`Walnut` is a strong counter-example to the notion that decidable theories cannot be used to prove real theorems. As such it provides *two* uses in a course in automata theory: (1) obviously the students can use it to ask questions about sequences, and (2) the teacher can point to it as a decidable theory that can actually be used.

Having said this, I now list PROS and CONS to using `Walnut`.

PRO: We can get answers to questions of interest.

CON: Do these proofs give us insights into what is really going on? (I do not have the intuition for why, for every n, the Infinite Fibonacci Sequence has a palindromic subword of length n).

COUNTER: Once you know a statement is true, then you may be able to find a human-readable proof.

PRO: There are times when you really just want the answer. I am thinking of proving that competing actors for a resource all get served (a version of the dining philosophers problem). I wonder if `Walnut` could be used on such problems.

PRO: Speculation: Is `Walnut` doing the dreary part of the proof freeing up us humans to do other things?

CON: Suppose we had a program that could tell us if P=NP, and we asked it and it said P $\neq$ NP. Well... we already knew that. Are we enlightened?

5 Opinion

You should buy this book for the following reasons.

1. For your own enlightenment.

2. For examples of concepts you can teach in a course in automata theory. That is, even if you do not plan to use or teach `Walnut`, there is much of interest in the book.

References

[1] E. Börger, E. Grädel, and Y. Gurevich. *The Classical Decision Problem.* Perspectives in Mathematical Logic. Springer, 1997.

[2] V. Bruyère, G. Hansel, C. Michaux, and R. Villemaire. Logic and p-recognizable sets of integers. *Bulletin of the Belgian Math. Soc.– Simon Stevin*, 1(2):191–238, 1994. Corrigendum 1:577, 1994.

[3] Y. Gurevich and L. Harrington. Trees, automata, and games. In *Proceedings of the Fourteenth Annual ACM Symposium on the Theory of Computing,* San Francisco CA, pages 60–65, 1982.

[4] H. Mousavi. Automatic theorem proving in Walnut, 2016. `https://arxiv.org/abs/1603.06017`.

[5] M. Rabin. Decidabilty of second-order theories of automata on infinite trees. *Transactions of the American Math Society*, 141:1–35, 1969. `www.jstor.org/stable/1995086`.

Review of [1]

200 Problems on Languages, Automata, & Computation
by **Filip Murlak, Damian Niwiński, Wojciech Rytter**

Cambridge University Press, 2023
Hardback, $90.00; Paperback, $39.99; eBook, $39.99; 254 pages

Review by **William Gasarch** (gasarch@umd.edu)

1 Introduction

The title of the book describes its content so well that I have little to add except this: Since the phrase *Languages, Automata, & Computation* may mean different things to different people, I list the chapters:

1. Part I: Problems
   (a) Words, Numbers, Graphs
   (b) Regular Languages
   (c) Context-Free Languages
   (d) Theory of Computation (This chapter includes computational complexity.)
2. Part II: Solutions
   (a) Words, Numbers, Graphs
   (b) Regular Languages
   (c) Context-Free Languages
   (d) Theory of Computation
3. Further Reading
4. Index

2 Summary

Each problem in this book has one of four markings (if you count NO marking as a marking).

1. Easy- a five-pointed star with white interior.

2. Intermediate-unmarked,

3. Hard-$\star$,

4. Very Hard-$\star\star$.

Solutions are included, which is either good or bad depending on how you intend to use it. Since some of the problems are hard, I like that there are solutions.

Are these good problems? Absolutely yes. There were many that I have not seen or thought of before, and note that I have taught automata theory $\aleph_0$ times and TAed it four times. There is a wide range of problems. The chapter headings help with finding problems in various topics. The $\star$-system helps with finding problems of various difficulties.

The rest of this review will be a few problems from each chapter and an opinion. When I give a problem, I paraphrase it since there are legal issues with quoting a book verbatim.

Elsewhere in this book review column is a review of *The Logical Approach to Automatic Sequences: Exploring Combinatorics on Words with Walnut* by Jeffrey Shallit. That book describes a program called *Walnut* which can be used to solve many problems having to do with sequences. Some of the problems in the book *200 Problems* can probably be solved automatically with *Walnut* (including some mentioned in this review). This is *not* a criticism; however, the user of *200 Problems* should be aware of this issue. There is a large discussion to be had about what-tool-do-we-allow-our-students-to-use, which we will not engage in here.

2.1 Words, Numbers, Graphs

I expected this chapter to have all easy and intermediate problems. It did not! It has two easy, two intermediate, two hard, and one very hard problems.

$\star\star$ **Definition** The *Thue-Morse sequence* is defined as follows: The first symbol is 0. Assume you have the first n symbols t_n. The next n symbols are t'_n, defined as t_n with the 0's changed to 1's and the 1's changed to 0's. (The problem gives two definitions and asks you to prove they are equivalent.)

1. Show that the Thue-Morse sequence is *cube-free*. That is, there is no subword of the form www where $w \in \{0,1\}^+$. (The Thue-Morse sequence is actually *strongly cube-free*: there is no subword of the form $bwbwb$ where $b \in \{0,1\}$ and $w \in \{0,1\}^+$.)

   They give a hint, but I will not. So perhaps my version is $\star\star\star$.

2. Construct a sequence over a 4-letter alphabet that is square-free. That is, there is no subword of the form ww where $w \in \{0,1\}^+$.

3. Construct a sequence over a 3-letter alphabet that is square-free.

4. Is there a sequence over a 2-letter alphabet that is square-free?

2.2 Regular Languages

1. Let $\Sigma = \{0,\ldots,9\}$ and view the input as a number in base 10. Show that

$$\{w \colon w \equiv 0 \pmod 7\}$$

is regular.

2. $\star$ An *infix* of a word $w = \sigma_1 \cdots \sigma_n$ is any string of the form $\sigma_i \cdots \sigma_j$ where $i \leq j$. A language L is *closed under infix* if every infix of every word in L is also in L.

   Give an example of an infinite language that is closed under infix but does not contain an infinite regular language as a subset.

3. $\star\star$ Let $\Sigma = \{a, b\}$. If w is a word over Σ and $\sigma \in \{a, b\}$ then let $\#_\sigma(w)$ be the number of σ's in w.

   Give an algorithm for the following (it need not be efficient):

   (a) Given a DFA for L, determine whether for all $w \in L$, $\#_a(w) = \#_b(w)$.

   (b) Given a DFA for L, determine whether there exists an infinite number of $w \in L$ such that $\#_a(w) = \#_b(w)$.

2.3 Context-Free Languages

1. Give an algorithm to determine whether $L(G)$ is infinite for a given CFG G.

2. $\star$ Show that for every context-free grammar G there is a constant C such that every non-empty word w generated by G has a derivation of length at most $C|w|$.

3. $\star$ Show that the set of palindromes cannot be recognized by a deterministic pushdown automaton.

2.4 Theory of Computation

1. $\star$ Let $X \subseteq \mathsf{N}$. Show that X is decidable iff either X is finite or X is the image of a computable strictly-increasing function. (I think this question should be 0 stars.)

2. $\star$ Is the following problem decidable: given $u, v \in \Sigma^*$ and a number k, is there a string w of length at least k such that $\#_u(w) = \#_v(w)$?

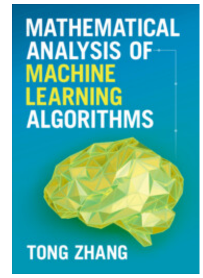
3. Show that the following problem is NP-complete: given a regular expression α over Σ ($|\Sigma|$ might be large), is there a word $w \in L(\alpha)$ such that every letter in Σ appears in it. (I think this question should be $\star$.)

3 Opinion

I began reading this book skeptical that I would find problems in it that I had not already seen. I was wrong. There are many great problems in this book, some for your students as homework, some to be the basis of lectures you will give your students (I will definitely teach the last problem on Theory of Computation that I stated the next time I teach Automata Theory), and some for your own edification.

The only criticisms are that there is not enough problems on NP-completeness or complexity theory in general.

Review of [1]



Mathematical Analysis of
Machine Learning Algorithms
by **Tong Zhang**

Cambridge University Press, 2023
Hardback, 479 pages, $54.99

Review by **Chinmay Hegde**
New York University

1 Overview

Progress in machine learning as an academic discipline has been relentless. Over the last two-plus decades, the field has evolved from being the purview of a small set of experts in academia and top industrial labs to impacting (arguably) all of computer science, data science, applied mathematics, and beyond. The center-of-gravity of scholarly research has been shifting from mathematical ideas, concepts, and theories towards concrete applications, systems, and real-world use cases.

Within this context, the new textbook *Mathematical Analysis of Machine Learning Algorithms* by Professor Tong Zhang is a *tour de force*. The book provides an introduction to a variety of mathematical tools that underpin modern machine learning techniques. It serves as a reminder that the foundations of machine learning have been (and continue to be) derived from mathematical analysis, and that we as a community should preserve and enrich these elements. In many ways, this book echoes earlier classical texts such as [HTFF09].

This advanced textbook, geared towards graduate students and doctoral candidates, encapsulates several broad families of concepts in machine learning theory. Zhang is a renowned machine learning researcher, and I have been reading his papers since my early graduate student years. The breadth and depth of coverage mirror Zhang's own research contributions, reflecting an illustrious career spent at the forefront of machine learning theory. This also adds an extra layer of authority to the text, as readers may benefit from insights gleaned from several decades of research.

The book's scope is ambitious. It surveys a number of topics, each representing a distinct neighborhood in the landscape of machine learning theory. Starting with Probably Approximately Correct (PAC) learning and the statistical elements of generalization theory, the textbook moves onto kernel methods and nonlinear data analysis. Generalized linear models, which form the backbone of many predictive algorithms used in practice, are examined next. The treatment of neural networks is brief but offers a peek into the dizzying array of recent theoretical developments that have attempted to probe deeper into the building blocks of modern artificial intelligence. The last quarter of the book ventures into the fields of online learning and reinforcement learning, both areas being at the forefront of several current research agendas.

What struck me about this book is its unwavering focus on mathematical rigor. Zhang makes no concessions to the reader and does not fall prey to handwavy arguments and simplification at the expense of mathematical precision. This approach is undoubtedly demanding, but I believe this will help readers gain a deeper understanding of the subject matter.

Early in the book, the author makes it clear that he will avoid lengthy descriptions of standard machine learning algorithms and applications. The reader is assumed to be familiar with the basics and does not need to be convinced why they should be interested in the techniques that are being discussed. This is a bold choice, but one that I like: I appreciated the author clearly flagging the book's intended role, which is that it is a complement to (rather than a replacement for) more application-oriented textbooks.

For the same reason, the book's approach may present challenges for some readers. The dense mathematical content and theoretical focus requires a high level of mathematical maturity, perseverance, and discourse. This book is undoubtedly *not* meant for casual perusal, but rather a work that will reward thinking and reflection.

Despite this challenging style (or perhaps because of it!), *Mathematical Analysis of Machine Learning Algorithms* will greatly benefit those seeking to understand the theoretical foundations of machine learning at the highest level. It fills a distinct niche in the existing literature. I think the book will stand as a testament to the depth and maturity of machine learning as a field. It challenges readers to elevate their understanding of learning algorithms beyond mere applications to grasp the fundamental principles that may drive further innovation. For those willing to embark on this intellectual journey, the rewards will be substantial.

2 Book Contents

The book is organized into 18 chapters. I will take the liberty to loosely cluster these chapters into seven sections. Here is a brief overview of each section:

1. **Introduction and Probability Theory** (Chapters 1-2): The first couple of chapters lay the groundwork for the rest of the book. The author introduces the fundamental concepts of machine learning and reviews essential probability theory. These chapters ensure that readers will have the necessary mathematical foundation to absorb the more advanced topics that follow later in the book.

2. **Generalization Theory** (Chapters 3-6) The next several chapters focus on the theoretical question of why and how machine learning methods perform well when presented with new data. To address this, the book introduces the concepts of risk minimization, uniform convergence, hypothesis classes and their VC-dimension, covering numbers, and Rademacher complexity. These concepts are essential for any researcher interested in understanding the statistical limits of model generalization from training data to unseen examples.

3. **Stability and Model Selection** (Chapters 7-8) These chapters provide an alternative lens to view generalization, focusing algorithmic stability and model selection techniques. While classical generalization theory focuses on establishing risk bounds based on the complexity of the hypothesis class, algorithmic stability obtains risk bounds based on the learning algorithm. Model selection is essential for choosing the best model among various alternatives.

4. **Advanced Learning Techniques** (Chapters 9-11) The book then moves onto specific families of machine learning algorithms. The next few chapters cover theoretical elements of several powerful machine learning techniques: kernel methods, additive models, random feature models, and neural networks. It provides a theoretical treatment of these models, explaining their mathematical foundations and properties. Here is also where the author (briefly) touches on capacity questions of machine learning models, the Universal Approximation Theorem, and the intriguing phenomenon of double descent.

5. **Lower Bounds** (Chapter 12): Much of the book focuses on upper bounds on how well machine learning algorithms may perform; this chapter deviates from this narrative and asks the question: "What's the best possible performance achievable?", introducing information-theoretic tools such as Fano's Inequality. Understanding these lower bounds is crucial for recognizing the inherent limitations of learning and the hardness of certain problems.

6. **Sequential Learning and Online Methods** (Chapters 13-15) These chapters make the transition standard (offline) learning to sequential (online) learning settings. There is a brief interlude again where the author revisits the basics of sequential random variables, online learning algorithms, and posterior averaging techniques. All these are important for analyzing streaming data and adapting models in real-time.

7. **Decision Making Under Uncertainty** (Chapters 16-18) The final section of the book explores topics at the intersection of machine learning and decision theory. It covers multi-armed bandits, contextual bandits, and reinforcement learning (RL). This serves as a very accessible theoretical treatment of bandit and RL algorithms, which are vital for making decisions in uncertain environments while learning from feedback.

3 Critiques

This book is a remarkable achievement. However, as with any book with an expansive scope, there are certain aspects that are uniquely excellent and other aspects that could potentially be improved in future editions.

Let me start by highlighting the book's best aspects. I enjoyed the book's comprehensive coverage of the topic; it offers both a broad and deep exploration of machine learning theory, covering foundational concepts and advanced research ideas alike. I particularly enjoyed the last three chapters on online algorithms and reinforcement learning; apart from a small handful of great books such as [AJKS19], this area has received relatively lesser attention from the machine learning theory community and may benefit from more comprehensive treatises.

I already mentioned above that Zhang's commitment to mathematical precision and theoretical depth is commendable and will be immensely rewarding to the dedicated reader. The book also benefits from Zhang's extensive research contributions: it is filled with insights that bridge the gap between theoretical concepts and cutting-edge research.

While the book's strengths are numerous, I believe there are two main aspects where future editions could potentially be enhanced:

1. **Optimization Aspects**: This might be my own personal bias seeping through, but I felt that the book could benefit from a more comprehensive treatment of *optimization aspects* in

the context of machine learning. This is itself a vast area of theoretical study, but somewhat under-emphasized (in my view) within the contents of this book.

Research over the last ten years (highlighted in [NTS15], but likely dating back much earlier) has pointed to intricate relationship between optimization and generalization in machine learning, particularly through the lens of implicit bias. Expanding on this relationship could provide readers with the key understanding of how optimization strategies impact learning outcomes and generalization performance.

Addressing this subject as thoroughly as the other topics of this book will be no small feat. But I believe that this set of topics could further enrich the book's already comprehensive theoretical treatment and align it with current research trends.

2. **Chapter Organization and Narrative Flow**: The current sequence of chapters, while logical, may sometimes feel disconnected to readers. It took me a little bit of time to appreciate why Chapters 7-8 (Stability and Model Selection) and Chapter 12 were placed in their respective locations in the book.

   To improve this aspect, the author could consider introducing a set of recurring, real-world inspired problems or examples that thread through the different sections. These "running examples" could serve multiple purposes:

   - They would provide practical context for the theoretical concepts, helping readers understand the real-world implications of the mathematical principles discussed.
   - They could act as a unifying element, tying together different chapters and sections.
   - Such examples could make the material more engaging and accessible, especially for readers who are application-oriented.

4 Conclusion

The book stands as a monumental achievement, and Zhang deserves high praise for this work. It spans both advanced research and academic instruction, while also pushing the boundaries of our understanding of the mathematical foundations underlying modern machine learning techniques. Zhang's expertise shines through every page. I hope that it will solidify the foundational thinking of the next wave of machine learning researchers who will undoubtedly shape the future of the field.

No doubt there will be room for minor enhancements in future editions. But the current work is already an indispensable resource for graduate students, researchers, and anyone seeking a rigorous understanding of machine learning. This book is a testament to both the author's extensive knowledge and to the maturation of machine learning as a scientific and mathematical discipline. I will recommend it very highly to my colleagues and students.

References

[AJKS19] Alekh Agarwal, Nan Jiang, Sham M Kakade, and Wen Sun. Reinforcement learning: Theory and algorithms. *CS Dept., UW Seattle, Seattle, WA, USA, Tech. Rep*, 32:96, 2019.

[HTFF09] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, volume 2. Springer, 2009.

[NTS15] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*, 2015.