Preface

Big is not absolute; it is relative. A person can be big relative to other people. A person is tiny compared to a mountain and gigantic compared to an ant. The same is true of data. Data is big relative to data that is easy to handle using current approaches. A few rows of data in a spreadsheet can be big if existing approaches rely on human visual inspection. Likewise, all data ever typed by humans can be small for a system designed to handle all data flowing over all communication networks.

Big Data describes a new era in the digital age in which the volume, velocity, and variety of data created across a wide range of fields (e.g., internet search, healthcare, finance, social media, defense, ...) are increasing at a rate well beyond our ability to analyze the data. Tools such as spreadsheets, databases, matrices, and graphs have been developed to address these challenges. The common theme amongst these tools is the need to store and operate on data as whole sets instead of as individual data elements. This book describes the common mathematical foundations of these data sets (associative arrays) that apply across many applications and technologies. Associative arrays unify and simplify data, leading to rapid solutions to volume, velocity, and variety problems. Understanding the mathematical underpinnings of data will allow the reader to see past the differences that lie on the surface of these tools and to leverage their mathematical similarities to solve the hardest data challenges. Specifically, understanding associative arrays (1) reduces the effort required to pass data between steps in a data processing system, (2) allows steps to be interchanged with full confidence that the results will be unchanged, and (3) makes it possible to recognize when steps can be simplified or eliminated.

A modern professional career spans decades. It is normal to work in many fields, with an ever-changing set of tools applied to a variety of data. The goal of this book is to provide you, the reader, with the concepts and techniques that will allow you to adapt to new data whenever it grows outside of your routine. The ideas discussed are applicable across the full spectrum of data sizes. Although specific tools are referred to in this book (see d4m.mit.edu), the mathematical concepts are independent of the tools and can be implemented with a variety of technologies. This book covers several of the primary viewpoints on data (spreadsheets, databases, matrices, and graphs) that encompass data and applications spanning a large part of human activity. Spreadsheets are used by more than 100 million people every day. Databases are used in nearly every digital transaction on Earth. Matrices and graphs are employed nearly every time this data is analyzed.

The purpose of collecting data is not simply to fill archives, but to generate insight that leads to new solutions of practical problems. Nothing handles big like mathematics. Mathematics is at ease with both the infinite and the infinitesimal. For this reason, a mathematical approach to data lies at the very heart of the scientific method

$$\text{theory} + \text{experiment} = \text{discovery}$$

Mathematics is theory made manifest. Likewise, data is the principal product of experi-

ment. A mathematical approach to data is the quickest path to bringing theory and experiment together. The primary tools for this merger (i.e., the "+" in the above formula) are computers that transform mathematics into operations on data in the form of computer bits.

This book will discuss mathematics, data, and computations that have been proven on real-world applications: science, engineering, bioinformatics, healthcare, banking, finance, computer networks, text analysis, social media, electrical networks, transportation, and building controls. The most interesting data sets that provide the most enthralling examples are extremely valuable and extremely private. Companies are interested in this data so they can sell you the products you want. Using this data, companies, stores, banks, hospitals, utilities, and schools are able to provide goods and services that are tailored specifically to you and allow you to live a demonstrably longer and happier life. Fortunately, such data is not readily available to be distributed by anyone who wishes to write a book on the topic. Thus, while it is possible to talk about the results of analyzing such data in general terms, it will not be possible to use the data that is most compelling to you and to the global economy. In addition, such examples would be quickly outdated as the field is rapidly moving. The examples in the book will be principally drawn from art and music. These topics are both compelling, readily shared, and have a long history of being interesting. Finally, it is worth mentioning that big data is big. It is not possible to use realistically sized examples given the limitations of the number of characters on a page. Fortunately, this is where mathematics comes to the rescue. In mathematics one can say that

$$\mathbf{c}(i) = \mathbf{a}(i) + \mathbf{b}(i)$$

for all $i = 1, ..., N$ and *know* this to be true. The ability to exactly predict the large-scale emergent behavior of a system from its small-scale properties is one of the most powerful properties of mathematics. Thus, while the examples in this book are tiny compared to real applications, by learning the key mathematical concepts, the reader can be confident that they apply to data at all scales. That a few mathematical concepts can span a diverse set of applications over many sizes is perhaps the most fundamental idea in this book.

This book is divided into three parts: (I) Data Processing, (II) Data Foundations, and (III) Data Transformations. The book will unfold so that a variety of readers can find it useful. Wherever possible, the relevant mathematical concepts are introduced in the context of big data to make them easily accessible. In fact, this book is a practical introduction to many of the more useful concepts found in matrix mathematics, graph theory, and abstract algebra.

Part I Data Processing introduces the concept of the associative array in practical terms that are accessible to a wide audience. Part I includes examples showing how associative arrays encompass spreadsheets, databases, matrices, and graphs. Next, the associative array manipulation system D4M (Dynamic Distributed Dimensional Data Model) is described along with of some of its successful results. Finally, several chapters describe applications of associative array to graph analysis and machine learning systems. The goal of Part I is to make it apparent that associative arrays are a powerful tool for creating interfaces to data processing systems. Associative array-based interfaces work because of their strong mathematical foundations that enable the structure of associative arrays to predict with mathematical precision (as opposed to relying on trial and error).

Part II Data Foundations provides a mathematically rigorous definition of associative arrays and describes the properties of associative arrays that emerge from this definition. Part II begins with definitions of associative arrays in terms of sets. The structural properties of associative arrays are then enumerated and compared with the properties of matri-

ces and graphs. Part II concludes with a survey of the diverse behavior of associative arrays under a variety of transformations (such as contraction and rotation) that are the building blocks of more complex algorithms. The ability to predict the impact of an associative array transformation is critical to the effectiveness of associative arrays.

Part III Data Transformations shows how concepts of linearity can be extended to encompass associative arrays. Linearity provides powerful tools for analyzing the behavior of associative array transformations. Part III starts with defining maps and bases for associative arrays. Eigenvalues and eigenvectors are then introduced and discussed. Part III ends with a discussion of the extension of associative arrays to higher dimensions.

In recognition of the severe time constraints of professional readers, each chapter is mostly self-contained. Forward and backward references to other chapters are limited, and key terms are redefined as needed. The reader is encouraged to consult the table of contents and the index to find more detailed information on concepts that might be covered in less detail in a particular chapter. Each chapter begins with a short summary of its content. Specific examples are given to illustrate concepts throughout each chapter. References and appendices are also contained in each chapter. This arrangement allows the professional reader to pick up and refer to any particular chapter as needed.

While most algorithms are presented mathematically, when working code examples are required, these are expressed in D4M. The D4M software package is an open-source toolbox that runs in the MATLAB, GNU Octave, and Julia programming languages. D4M is the first practical implementation of associative array mathematics and has been used in diverse applications. D4M has a complete set of documentation, example programs, tutorial slides, and many hours of instructional videos that are all available online (see d4m.mit.edu). The D4M examples in the book are written in MATLAB, and some familiarity with MATLAB is helpful (see [Higham & Higham 2005, Moler 2004, Kepner 2009] for an introduction). Notationally, associative arrays and their corresponding operations that are specifically referring to the D4M use of associative arrays will be written using sans serif font, such as

$$\mathsf{C} = \mathsf{A} + \mathsf{B}$$

Likewise, associative arrays and their corresponding operations that are specifically referring to the mathematical use of associative arrays will be written using serif font, such as

$$\mathbf{C} = \mathbf{A} \oplus \mathbf{B}$$

A complete summary of the notation in the book is given in the Appendix.

This book is suitable as either the primary or supplemental book for a class on big data, algorithms, data structures, data analysis, linear algebra, or abstract algebra. The material is useful for engineers, scientists, mathematicians, computer scientists, and software engineers.

References

[Higham & Higham 2005] Desmond J. Higham and Nicholas J. Higham, *MATLAB Guide*, Second Edition, SIAM, Philadelphia, 2005 (Cited on p. xi)

[Kepner 2009] Jeremy Kepner, *Parallel MATLAB for Multicore and Multinode Computers*, SIAM, Philadelphia, 2009 (Cited on pp. xi, 46)

[Moler 2004] Cleve Moler, *Numerical Computing with MATLAB*, SIAM, Philadelphia, 2004 (Cited on p. xi)