Contents

Preface

Is there a role for mathematics in the era of data science, and if so, what is it? In the natural sciences, the *unreasonable effectiveness of mathematics* [45] came from its ability to create new concepts that correspond to hidden structures in nature, and to capture essential features of complex phenomena with simple, almost skeletal models with amazing predictive skills. As modern science has come to rely more and more on data, mathematicians have been building bridges between the data and the underlying reality via mathematical models that were then validated on the data. As data began to have a life of their own, without the need of being an expression of an underlying object in a framework of a model, they became themselves the object of interest. Within this new data paradigm, mathematics play a big role when it comes to summarizing data sets, detecting latent structures, and extracting specific information from the data. As new data-driven applications show the *unreasonable effectiveness of data* [14], the contribution of mathematicians to the data science world continues to grow, and a more clearly defined profile of the mathematics of data science has begun to emerge.

This book is on the mathematics of data science, and thus the mathematical perspective will shape the presentation of the material, without forgetting the data science driver behind it. The coupling between mathematics and data science is highlighted in every chapter, with the exception of the first one, where we provide a brief review of linear algebra. Some concepts in data science keep reappearing in different contexts, like recurring themes around which different algorithms are designed. The three dominating data science themes that will be studied from a mathematical perspective in this book are data reduction and visualization, clustering, and classification.

Data science can be approached from different, complementary directions. Many popular and successful techniques in data analysis are rooted in statistics, a discipline with the tradition of considering data as the object of intrinsic interest. There are a number of data science books that look at the field from a statistical point of view; see, e.g., [16]. In this book we take an alternative approach, based for the most part, but not solely, on linear algebra, without pursuing the corresponding, more common, statistical interpretation. This choice has been motivated by the desire for a consistent, concise, and coherent narrative, and to highlight different uses of mathematics, especially linear algebra, in data science. In line with this choice of perspective, whenever possible we will organize the data in the form of vectors and assemble them as the columns of the data matrix, favoring a treatment in linear algebraic terms. We will leverage the close connection with linear algebra to interpret and visualize the data geometrically. While linear algebra has a prominent role in this book, some of the algorithms that we will consider do not have a natural linear algebra formulation, relying instead on some optimization techniques that will be introduced as needed.

The material in this book can be used for a one-semester course for advanced undergraduates or beginning graduate students. Readers with a solid linear algebra background may skip Chapter 1, which provides a brief linear algebra refresher, presenting a collection of definitions and results needed later. Readers who are less familiar with linear algebra might find this material

helpful, and, moreover, it will establish the notation for the subsequent chapters. Ideally, this book will enable the reader to implement from scratch all the algorithms, thus having the ability to customize as desired. To facilitate the implementation part, we have interspersed the presentation with segments of the MATLAB code that was used to test the algorithms and produce the graphics. The choice of using MATLAB as a reference language arises from its close affinity with the notions of linear algebra, acknowledging that in modern data science, languages like R, Python, and Julia have a prominent role.

Chapter 2 is dedicated to the principal component analysis (PCA) of a data set. The singular value decomposition being the backbone of PCA means that there is a strong mathematical and linear algebraic foundation behind the PCA algorithm. In this chapter we highlight two different, important uses of this method. Often PCA is used to detect the effective dimensionality of data sets that may live in subspaces of high-dimensional spaces, and the PCA algorithm provides a natural way to find a lower-dimensional representation of the data, or to find the best low-rank approximation of the data. While providing a reduction in the number of parameters needed to describe each data point, PCA also finds feature vectors that summarize the aggregate of the data and are ranked in terms of decreasing importance. This may be a convenient way to get a sense of the data set as a whole. Another important use of PCA is to visualize high-dimensional data by projecting them onto the directions of maximum spread. This is used throughout the remainder of the book as the default visualization tool. In many applications, preprocessing of the data is carried out by PCA, motivating our decision to make this the first data science algorithm to be considered.

In Chapter 3, we discuss two basic clustering algorithms: the classic k-means algorithm and a modification known as the k-medoids algorithm. These algorithms, whose mathematical underpinning is not nearly as strong as for the PCA, let the data organize themselves into clusters, or classes, without providing any external guidelines about the grouping. For this reason, the k-means and k-medoids algorithms are examples of nonsupervised learning methods. Since the data may or may not naturally separate into clusters, the mathematical theory backing these algorithms is to some extent guided by heuristic considerations. Since searching for natural clusters is often one of the first tasks when dealing with a new data set and these are some of the most popular methods for doing it, it is important to understand their merits and limitations.

Chapter 4 continues on the theme of data clustering by presenting an algorithm for finding the directions along which clusters in the data are most clearly seen. In this case we assume that the data have been partitioned into k classes, and that an annotation recording the class of each data point is available. The algorithm, known as linear discriminant analysis (LDA), finds those directions in space where the projections of each cluster in the data is as compact as possible and the different clusters are maximally separated. Once these directions have been determined, they can be used to visualize high-dimensional data in lower-dimensional spaces, with the maximization of the separation of the clusters as the main criterion. The LDA algorithm is formulated as an eigenvalue problem, and the separating directions are further analyzed for suitably reducing the dimensionality of the data.

The theme of understanding the structure and internal organization of data that cannot be visualized directly continues in Chapter 5. The idea behind the self-organizing maps (SOM) algorithm is to approximate high-dimensional data with a set of prototype vectors connected by a neighborhood structure that is described in terms of a low- (one- or two-)dimensional lattice. The topological relations among the prototypes provided by the lattice are preserved as they experience the data one point at a time and distribute themselves to capture their organization. In this chapter we present the details of the SOM algorithm and show a few different ways to interpret the prototypes and their way of capturing the organization of the data.

Chapter 6 addresses the problem of summarizing and representing data sets where all the data components, or attributes, are nonnegative numbers, in a way that respects the structure

of the data. The nonnegative matrix factorization (NMF) addresses this problem, providing an approximation of each data vector as the linear combination with nonnegative coefficients of feature vectors with nonnegative entries. In NMF the data are described in terms of prototypes that also are nonnegative, thus readily interpretable in the terms originally used to describe the data. While a rank k NMF factorization of the data cannot be a better approximation than a rank k PCA approximation, in the latter the orthogonality of the feature vector prevents them normally from being nonnegative. Therefore, when preserving the nonnegativity in the data representation is important, NMF is superior to PCA. In both cases, the feature vectors can provide a quick summary of the aggregated data, and in NMF they are amenable to the same interpretation as the original data.

In Chapter 7 we begin addressing the classification problem by building elementary classifiers inspired by some of the algorithms introduced in the previous chapters. This will lead to the k-nearest neighbor algorithm, PCA and LDA classifiers, and the learning vector quantifier (LVQ). These methods are referred to as supervised learning algorithms because they use training sets with the data partitioned into classes with known annotations to instruct the algorithm. Methods to test and evaluate the performance of classification algorithm are also introduced.

Data science has been very helpful in the study of large corpora, that is, data sets comprising text documents. Chapter 8 presents data science techniques specifically developed for text files. Preprocessing text documents is an important step: After pruning from the documents stop words that add nothing to the context, and reducing words to the stems from which they originate, the corpus is represented numerically as a term-document matrix with nonnegative entries. At this point, given a query, retrieval algorithms are used to subdivide the data set into the two classes of relevant and nonrelevant documents. Data science methods for sets of images are presented in Chapter 9, where we focus especially on how to handle texture images. The preprocessing of images to represent them in a unified framework is very important. For black-and-white images, we coarsen the representation by limiting the range of the gray scale, then compute the gray level co-occurrence matrix (GLCM), which allows us to represent each image as a $k \times k$ matrix of nonnegative entries regardless of the original size or shape of the image. At this point, methods for numerical data sets can be applied.

In Chapter 10 we look at the popular tree-based classifiers and their extensions: random tree and random forest algorithms. Tree classifiers are very popular because of the relative simplicity of the algorithm, which ultimately can be reduced to a string of yes/no answers. The advantage of them is also their high interpretability. This chapter, which is less mathematically based than the previous ones, has been included for completeness and to acknowledge the popularity of the method.

Chapter 11 discusses a widely used binary classification algorithm, the support vector machine (SVM). The algorithm relies on quadratic optimization tools with constraints. We have included a short section on primal-dual methods and convex optimization, restricted to the special case needed to implement the algorithm. Of particular interest is the kernel extension of the algorithm which adds a significant amount of flexibility beyond the original SVM algorithm looking for linear separators.

Chapter 12 is dedicated to the page ranking, in particular the algorithm known as PageRank. This algorithm, where linear algebra gets the lion's share of the merit, opens the way towards network analysis and random processes in networks. We decided to present the page ranking algorithm last because, unlike the methods of the earlier chapters, it does not provide generic and useful tools to analyze any given data set, yet it is a method that every data-oriented applied mathematician—and every web searcher—should know. After all, this is the driver behind the success of some of the most successful web search engines!

The selection of topics excludes some popular algorithms in data science such as the various versions of neural networks. A solid mathematical theory of neural networks is still emerging,

and delving into the details, including understanding the role of the architecture and underlying optimization methods, requires another type of approach.

At the end of each chapter, we have some notes and comments for readers who are interested in reading more about the material. Acknowledging the fact that data science applications keep growing and expanding, practice exercises have not been included as part of the book, but are posted and periodically updated at https://bookstore.siam.org/di01/bonus, including resources for implementation in Python.

The final form of this book greatly profited from the feedback from the students who took our Mathematics of Data Science course at Case Western Reserve University in the past decade, at the University of Naples "Federico II" in the spring of 2018 and 2019, and at Milan Polytechnic University in 2019. Many thanks to them, and to the three anonymous reviewers whose many suggestions helped us improve the presentation of the material.

Index