Foundations of Data Science[*]

Avrim Blum, John Hopcroft, and Ravindran Kannan

Thursday 4th January, 2018

[*] Copyright 2015. All rights reserved

Contents

1 Introduction 9

2 High-Dimensional Space 12

2.1 Introduction . . . 12
2.2 The Law of Large Numbers . . . 12
2.3 The Geometry of High Dimensions . . . 15
2.4 Properties of the Unit Ball . . . 17
2.4.1 Volume of the Unit Ball . . . 17
2.4.2 Volume Near the Equator . . . 19
2.5 Generating Points Uniformly at Random from a Ball . . . 22
2.6 Gaussians in High Dimension . . . 23
2.7 Random Projection and Johnson-Lindenstrauss Lemma . . . 25
2.8 Separating Gaussians . . . 27
2.9 Fitting a Spherical Gaussian to Data . . . 29
2.10 Bibliographic Notes . . . 31
2.11 Exercises . . . 32

3 Best-Fit Subspaces and Singular Value Decomposition (SVD) 40

3.1 Introduction . . . 40
3.2 Preliminaries . . . 41
3.3 Singular Vectors . . . 42
3.4 Singular Value Decomposition (SVD) . . . 45
3.5 Best Rank-k Approximations . . . 47
3.6 Left Singular Vectors . . . 48
3.7 Power Method for Singular Value Decomposition . . . 51
3.7.1 A Faster Method . . . 51
3.8 Singular Vectors and Eigenvectors . . . 54
3.9 Applications of Singular Value Decomposition . . . 54
3.9.1 Centering Data . . . 54
3.9.2 Principal Component Analysis . . . 56
3.9.3 Clustering a Mixture of Spherical Gaussians . . . 56
3.9.4 Ranking Documents and Web Pages . . . 62
3.9.5 An Application of SVD to a Discrete Optimization Problem . . . 63
3.10 Bibliographic Notes . . . 65
3.11 Exercises . . . 67

4 Random Walks and Markov Chains 76

4.1 Stationary Distribution . . . 80
4.2 Markov Chain Monte Carlo . . . 81
4.2.1 Metropolis-Hasting Algorithm . . . 83
4.2.2 Gibbs Sampling . . . 84
4.3 Areas and Volumes . . . 86

4.4 Convergence of Random Walks on Undirected Graphs 88
4.4.1 Using Normalized Conductance to Prove Convergence 94
4.5 Electrical Networks and Random Walks 97
4.6 Random Walks on Undirected Graphs with Unit Edge Weights 102
4.7 Random Walks in Euclidean Space 109
4.8 The Web as a Markov Chain 112
4.9 Bibliographic Notes 116
4.10 Exercises 118

5 Machine Learning 129
5.1 Introduction 129
5.2 The Perceptron algorithm 130
5.3 Kernel Functions 132
5.4 Generalizing to New Data 134
5.5 Overfitting and Uniform Convergence 135
5.6 Illustrative Examples and Occam's Razor 138
5.6.1 Learning Disjunctions 138
5.6.2 Occam's Razor 139
5.6.3 Application: Learning Decision Trees 140
5.7 Regularization: Penalizing Complexity 141
5.8 Online Learning 141
5.8.1 An Example: Learning Disjunctions 142
5.8.2 The Halving Algorithm 143
5.8.3 The Perceptron Algorithm 143
5.8.4 Extensions: Inseparable Data and Hinge Loss 145
5.9 Online to Batch Conversion 146
5.10 Support-Vector Machines 147
5.11 VC-Dimension 148
5.11.1 Definitions and Key Theorems 149
5.11.2 Examples: VC-Dimension and Growth Function 151
5.11.3 Proof of Main Theorems 153
5.11.4 VC-Dimension of Combinations of Concepts 156
5.11.5 Other Measures of Complexity 156
5.12 Strong and Weak Learning - Boosting 157
5.13 Stochastic Gradient Descent 160
5.14 Combining (Sleeping) Expert Advice 162
5.15 Deep Learning 164
5.15.1 Generative Adversarial Networks (GANs) 170
5.16 Further Current Directions 171
5.16.1 Semi-Supervised Learning 171
5.16.2 Active Learning 174
5.16.3 Multi-Task Learning 174
5.17 Bibliographic Notes 175

5.18 Exercises 176

6 Algorithms for Massive Data Problems: Streaming, Sketching, and Sampling 181

6.1 Introduction 181

6.2 Frequency Moments of Data Streams 182

6.2.1 Number of Distinct Elements in a Data Stream 183

6.2.2 Number of Occurrences of a Given Element. 186

6.2.3 Frequent Elements 187

6.2.4 The Second Moment 189

6.3 Matrix Algorithms using Sampling 192

6.3.1 Matrix Multiplication using Sampling 193

6.3.2 Implementing Length Squared Sampling in Two Passes 197

6.3.3 Sketch of a Large Matrix 197

6.4 Sketches of Documents 201

6.5 Bibliographic Notes 203

6.6 Exercises 204

7 Clustering 208

7.1 Introduction 208

7.1.1 Preliminaries 208

7.1.2 Two General Assumptions on the Form of Clusters 209

7.1.3 Spectral Clustering 211

7.2 k-Means Clustering 211

7.2.1 A Maximum-Likelihood Motivation 211

7.2.2 Structural Properties of the k-Means Objective 212

7.2.3 Lloyd's Algorithm 213

7.2.4 Ward's Algorithm 215

7.2.5 k-Means Clustering on the Line 215

7.3 k-Center Clustering 215

7.4 Finding Low-Error Clusterings 216

7.5 Spectral Clustering 216

7.5.1 Why Project? 216

7.5.2 The Algorithm 218

7.5.3 Means Separated by $\Omega(1)$ Standard Deviations 219

7.5.4 Laplacians 221

7.5.5 Local spectral clustering 221

7.6 Approximation Stability 224

7.6.1 The Conceptual Idea 224

7.6.2 Making this Formal 224

7.6.3 Algorithm and Analysis 225

7.7 High-Density Clusters 227

7.7.1 Single Linkage 227

7.7.2 Robust Linkage . . . 228
7.8 Kernel Methods . . . 228
7.9 Recursive Clustering based on Sparse Cuts . . . 229
7.10 Dense Submatrices and Communities . . . 230
7.11 Community Finding and Graph Partitioning . . . 233
7.12 Spectral clustering applied to social networks . . . 236
7.13 Bibliographic Notes . . . 239
7.14 Exercises . . . 240

8 Random Graphs 245
8.1 The $G(n,p)$ Model . . . 245
8.1.1 Degree Distribution . . . 246
8.1.2 Existence of Triangles in $G(n,d/n)$ . . . 250
8.2 Phase Transitions . . . 252
8.3 Giant Component . . . 261
8.3.1 Existence of a giant component . . . 261
8.3.2 No other large components . . . 263
8.3.3 The case of $p < 1/n$ . . . 264
8.4 Cycles and Full Connectivity . . . 265
8.4.1 Emergence of Cycles . . . 265
8.4.2 Full Connectivity . . . 266
8.4.3 Threshold for $O(\ln n)$ Diameter . . . 268
8.5 Phase Transitions for Increasing Properties . . . 270
8.6 Branching Processes . . . 272
8.7 CNF-SAT . . . 277
8.7.1 SAT-solvers in practice . . . 278
8.7.2 Phase Transitions for CNF-SAT . . . 279
8.8 Nonuniform Models of Random Graphs . . . 284
8.8.1 Giant Component in Graphs with Given Degree Distribution . . . 285
8.9 Growth Models . . . 286
8.9.1 Growth Model Without Preferential Attachment . . . 287
8.9.2 Growth Model With Preferential Attachment . . . 293
8.10 Small World Graphs . . . 294
8.11 Bibliographic Notes . . . 299
8.12 Exercises . . . 301

9 Topic Models, Nonnegative Matrix Factorization, Hidden Markov Models, and Graphical Models 310
9.1 Topic Models . . . 310
9.2 An Idealized Model . . . 313
9.3 Nonnegative Matrix Factorization - NMF . . . 315
9.4 NMF with Anchor Terms . . . 317
9.5 Hard and Soft Clustering . . . 318

9.6 The Latent Dirichlet Allocation Model for Topic Modeling 320
9.7 The Dominant Admixture Model 322
9.8 Formal Assumptions 324
9.9 Finding the Term-Topic Matrix 327
9.10 Hidden Markov Models 332
9.11 Graphical Models and Belief Propagation 337
9.12 Bayesian or Belief Networks 338
9.13 Markov Random Fields 339
9.14 Factor Graphs 340
9.15 Tree Algorithms 341
9.16 Message Passing in General Graphs 342
9.17 Graphs with a Single Cycle 344
9.18 Belief Update in Networks with a Single Loop 346
9.19 Maximum Weight Matching 347
9.20 Warning Propagation 351
9.21 Correlation Between Variables 351
9.22 Bibliographic Notes 355
9.23 Exercises 357

10 Other Topics 360

10.1 Ranking and Social Choice 360
10.1.1 Randomization 362
10.1.2 Examples 363
10.2 Compressed Sensing and Sparse Vectors 364
10.2.1 Unique Reconstruction of a Sparse Vector 365
10.2.2 Efficiently Finding the Unique Sparse Solution 366
10.3 Applications 368
10.3.1 Biological 368
10.3.2 Low Rank Matrices 369
10.4 An Uncertainty Principle 370
10.4.1 Sparse Vector in Some Coordinate Basis 370
10.4.2 A Representation Cannot be Sparse in Both Time and Frequency Domains 371
10.5 Gradient 373
10.6 Linear Programming 375
10.6.1 The Ellipsoid Algorithm 375
10.7 Integer Optimization 377
10.8 Semi-Definite Programming 378
10.9 Bibliographic Notes 380
10.10Exercises 381

11 Wavelets 385
11.1 Dilation 385
11.2 The Haar Wavelet 386
11.3 Wavelet Systems 390
11.4 Solving the Dilation Equation 390
11.5 Conditions on the Dilation Equation 392
11.6 Derivation of the Wavelets from the Scaling Function 394
11.7 Sufficient Conditions for the Wavelets to be Orthogonal 398
11.8 Expressing a Function in Terms of Wavelets 401
11.9 Designing a Wavelet System 402
11.10Applications 402
11.11 Bibliographic Notes 402
11.12 Exercises 403

12 Appendix 406
12.1 Definitions and Notation 406
12.2 Asymptotic Notation 406
12.3 Useful Relations 408
12.4 Useful Inequalities 413
12.5 Probability 420
12.5.1 Sample Space, Events, and Independence 420
12.5.2 Linearity of Expectation 421
12.5.3 Union Bound 422
12.5.4 Indicator Variables 422
12.5.5 Variance 422
12.5.6 Variance of the Sum of Independent Random Variables 423
12.5.7 Median 423
12.5.8 The Central Limit Theorem 423
12.5.9 Probability Distributions 424
12.5.10Bayes Rule and Estimators 428
12.6 Bounds on Tail Probability 430
12.6.1 Chernoff Bounds 430
12.6.2 More General Tail Bounds 433
12.7 Applications of the Tail Bound 436
12.8 Eigenvalues and Eigenvectors 437
12.8.1 Symmetric Matrices 439
12.8.2 Relationship between SVD and Eigen Decomposition 441
12.8.3 Extremal Properties of Eigenvalues 441
12.8.4 Eigenvalues of the Sum of Two Symmetric Matrices 443
12.8.5 Norms 445
12.8.6 Important Norms and Their Properties 446
12.8.7 Additional Linear Algebra 448
12.8.8 Distance between subspaces 450

12.8.9 Positive semidefinite matrix 451
12.9 Generating Functions 451
12.9.1 Generating Functions for Sequences Defined by Recurrence Relationships 452
12.9.2 The Exponential Generating Function and the Moment Generating Function 454
12.10Miscellaneous 456
12.10.1Lagrange multipliers 456
12.10.2Finite Fields 457
12.10.3Application of Mean Value Theorem 457
12.10.4Sperner's Lemma 459
12.10.5Prüfer 459
12.11Exercises 460

Index **466**

1 Introduction

Computer science as an academic discipline began in the 1960's. Emphasis was on programming languages, compilers, operating systems, and the mathematical theory that supported these areas. Courses in theoretical computer science covered finite automata, regular expressions, context-free languages, and computability. In the 1970's, the study of algorithms was added as an important component of theory. The emphasis was on making computers useful. Today, a fundamental change is taking place and the focus is more on a wealth of applications. There are many reasons for this change. The merging of computing and communications has played an important role. The enhanced ability to observe, collect, and store data in the natural sciences, in commerce, and in other fields calls for a change in our understanding of data and how to handle it in the modern setting. The emergence of the web and social networks as central aspects of daily life presents both opportunities and challenges for theory.

While traditional areas of computer science remain highly important, increasingly researchers of the future will be involved with using computers to understand and extract usable information from massive data arising in applications, not just how to make computers useful on specific well-defined problems. With this in mind we have written this book to cover the theory we expect to be useful in the next 40 years, just as an understanding of automata theory, algorithms, and related topics gave students an advantage in the last 40 years. One of the major changes is an increase in emphasis on probability, statistics, and numerical methods.

Early drafts of the book have been used for both undergraduate and graduate courses. Background material needed for an undergraduate course has been put in the appendix. For this reason, the appendix has homework problems.

Modern data in diverse fields such as information processing, search, and machine learning is often advantageously represented as vectors with a large number of components. The vector representation is not just a book-keeping device to store many fields of a record. Indeed, the two salient aspects of vectors: geometric (length, dot products, orthogonality etc.) and linear algebraic (independence, rank, singular values etc.) turn out to be relevant and useful. Chapters 2 and 3 lay the foundations of geometry and linear algebra respectively. More specifically, our intuition from two or three dimensional space can be surprisingly off the mark when it comes to high dimensions. Chapter 2 works out the fundamentals needed to understand the differences. The emphasis of the chapter, as well as the book in general, is to get across the intellectual ideas and the mathematical foundations rather than focus on particular applications, some of which are briefly described. Chapter 3 focuses on singular value decomposition (SVD) a central tool to deal with matrix data. We give a from-first-principles description of the mathematics and algorithms for SVD. Applications of singular value decomposition include principal component analysis, a widely used technique which we touch upon, as well as modern applications to statistical mixtures of probability densities, discrete optimization, etc., which are described in more detail.

Exploring large structures like the web or the space of configurations of a large system with deterministic methods can be prohibitively expensive. Random walks (also called Markov Chains) turn out often to be more efficient as well as illuminative. The stationary distributions of such walks are important for applications ranging from web search to the simulation of physical systems. The underlying mathematical theory of such random walks, as well as connections to electrical networks, forms the core of Chapter 4 on Markov chains.

One of the surprises of computer science over the last two decades is that some domain-independent methods have been immensely successful in tackling problems from diverse areas. Machine learning is a striking example. Chapter 5 describes the foundations of machine learning, both algorithms for optimizing over given training examples, as well as the theory for understanding when such optimization can be expected to lead to good performance on new, unseen data. This includes important measures such as the Vapnik-Chervonenkis dimension, important algorithms such as the Perceptron Algorithm, stochastic gradient descent, boosting, and deep learning, and important notions such as regularization and overfitting.

The field of algorithms has traditionally assumed that the input data to a problem is presented in random access memory, which the algorithm can repeatedly access. This is not feasible for problems involving enormous amounts of data. The streaming model and other models have been formulated to reflect this. In this setting, sampling plays a crucial role and, indeed, we have to sample on the fly. In Chapter 6 we study how to draw good samples efficiently and how to estimate statistical and linear algebra quantities, with such samples.

While Chapter 5 focuses on supervised learning, where one learns from labeled training data, the problem of unsupervised learning, or learning from unlabeled data, is equally important. A central topic in unsupervised learning is clustering, discussed in Chapter 7. Clustering refers to the problem of partitioning data into groups of similar objects. After describing some of the basic methods for clustering, such as the k-means algorithm, Chapter 7 focuses on modern developments in understanding these, as well as newer algorithms and general frameworks for analyzing different kinds of clustering problems.

Central to our understanding of large structures, like the web and social networks, is building models to capture essential properties of these structures. The simplest model is that of a random graph formulated by Erdös and Renyi, which we study in detail in Chapter 8, proving that certain global phenomena, like a giant connected component, arise in such structures with only local choices. We also describe other models of random graphs.

Chapter 9 focuses on linear-algebraic problems of making sense from data, in particular topic modeling and non-negative matrix factorization. In addition to discussing well-known models, we also describe some current research on models and algorithms with provable guarantees on learning error and time. This is followed by graphical models and belief propagation.

Chapter 10 discusses ranking and social choice as well as problems of sparse representations such as compressed sensing. Additionally, Chapter 10 includes a brief discussion of linear programming and semidefinite programming. Wavelets, which are an important method for representing signals across a wide range of applications, are discussed in Chapter 11 along with some of their fundamental mathematical properties. The appendix includes a range of background material.

A word about notation in the book. To help the student, we have adopted certain notations, and with a few exceptions, adhered to them. We use lower case letters for scalar variables and functions, bold face lower case for vectors, and upper case letters for matrices. Lower case near the beginning of the alphabet tend to be constants, in the middle of the alphabet, such as i, j, and k, are indices in summations, n and m for integer sizes, and x, y and z for variables. If A is a matrix its elements are a_{ij} and its rows are $\mathbf{a_i}$. If $\mathbf{a_i}$ is a vector its coordinates are a_{ij}. Where the literature traditionally uses a symbol for a quantity, we also used that symbol, even if it meant abandoning our convention. If we have a set of points in some vector space, and work with a subspace, we use n for the number of points, d for the dimension of the space, and k for the dimension of the subspace.

The term "almost surely" means with probability tending to one. We use $\ln n$ for the natural logarithm and $\log n$ for the base two logarithm. If we want base ten, we will use $\log_{10}$. To simplify notation and to make it easier to read we use $E^2(1-x)$ for $\big(E(1-x)\big)^2$ and $E(1-x)^2$ for $E\left((1-x)^2\right)$. When we say "randomly select" some number of points from a given probability distribution, independence is always assumed unless otherwise stated.