Mining of Massive Datasets

Anand Rajaraman
Jeffrey David Ullman

CAMBRIDGE

The popularity of the Web and Internet commerce provides many extremely large datasets from which information can be gleaned by data mining. This book focuses on practical algorithms that have been used to solve key problems in data mining and can be used on even the largest datasets.

It begins with a discussion of the map-reduce framework, an important tool for parallelizing algorithms automatically. The tricks of locality-sensitive hashing are explained. This body of knowledge, which deserves to be more widely known, is essential when seeking similar objects in a very large collection without having to compare each pair of objects. Stream processing algorithms for mining data that arrives too fast for exhaustive processing are also explained. The PageRank idea and related tricks for organizing the Web are covered next. Other chapters cover the problems of finding frequent itemsets and clustering, each from the point of view that the data is too large to fit in main memory. The final chapters cover two applications: recommendation systems and Web advertising, each vital in e-commerce.

Written by two authorities in database and web technologies, this book will be essential for students and practitioners alike.

Cover illustration: © S. Ullman 2011.

Anand Rajaraman is a Senior Vice President at Walmart Global eCommerce. He heads up the newly created @WalmartLabs, focused on the intersection of social, mobile, and retail. Anand joined Walmart in 2011 when Walmart acquired Kosmix, the startup he co-founded in 2005. He also teaches a class on web-scale data mining at the Computer Science Department at Stanford University

Anand is an active member of the Silicon Valley ecosystem, serving as investor, advisor and Board member to several startups. He is a Founding Partner of Cambrian Ventures, an early-stage venture capital firm. Anand's investments include Aster Data (acquired by Teradata), Kaltix and Transformic (both acquired by Google), Neoteris (acquired by Juniper Networks), Facebook, and Efficient Frontier.

Prior to founding Cambrian in 2000, Anand was Director of Technology at Amazon.com, where he helped launch the transformation of Amazon from retailer into a retail platform. He also is an inventor of the concept underlying Amazon Mechanical Turk. Anand came to Amazon.com in 1998 through the acquistion of Junglee, an Internet shopping pioneer that he co-founded in 1996, and where he served as Chief Technology Officer.

Anand obtained his Bachelor's degree in Computer Science and Engineering from the Indian Institute of Technology, Madras, where he won the President of India Gold Medal for graduating at the top of his class, and his MS and Ph.D. in Computer Science from Stanford University. Anand has been featured in articles in *Business Week*, *The New York Times*, *San Francisco Chronicle*, and other leading national publications. He has been a vocal proponent for using data to improve all aspects of business, and his blog Datawocky is devoted to this topic.

Jeffrey David Ullman is the Stanford W. Ascherman Professor of Computer Science (Emeritus) at Stanford University. He is also the CEO of Gradiance.

Ullman's research interests include database theory, data integration, data mining, and education using the information infrastructure. He is one of the founders of the field of database theory, and was the doctoral advisor of an entire generation of students who later became leading database theorists in their own right. He was also the Ph.D. advisor of Sergey Brin, one of the co-founders of Google, and served on Google's technical advisory board.

In 1995 he was inducted as a Fellow of the Association for Computing Machinery and in 2000 he was awarded the Knuth Prize. Ullman is also the co-recipient (with John Hopcroft) of the 2010 IEEE John von Neumann Medal, for "laying the foundations for the fields of automata and language theory and many seminal contributions to theoretical computer science."

CAMBRIDGE
UNIVERSITY PRESS
www.cambridge.org

ISBN 978-1-107-01535-7

Rajaraman and Ullman

Mining of Massive Datasets

CAMBRIDGE

Mining of Massive Datasets

ANAND RAJARAMAN
@WalmartLabs

JEFFREY DAVID ULLMAN
Stanford University


CAMBRIDGE
UNIVERSITY PRESS

CAMBRIDGE UNIVERSITY PRESS
Cambridge, New York, Melbourne, Madrid, Cape Town,
Singapore, São Paulo, Delhi, Tokyo, Mexico City

Cambridge University Press
The Edinburgh Building, Cambridge CB2 8RU, UK

Published in the United States of America by Cambridge University Press, New York

www.cambridge.org
Information on this title: www.cambridge.org/9781107015357

First published 2012

Printed in the United Kingdom at the University Press, Cambridge

A catalogue record for this publication is available from the British Library

ISBN 978-1-107-01535-7 Hardback

Mining of Massive Datasets

The popularity of the Web and Internet commerce provides many extremely large datasets from which information can be gleaned by data mining. This book focuses on practical algorithms that have been used to solve key problems in data mining and can be used on even the largest datasets.

It begins with a discussion of the map-reduce framework, an important tool for parallelizing algorithms automatically. The tricks of locality-sensitive hashing are explained. This body of knowledge, which deserves to be more widely known, is essential when seeking similar objects in a very large collection without having to compare each pair of objects. Stream processing algorithms for mining data that arrives too fast for exhaustive processing are also explained. The PageRank idea and related tricks for organizing the Web are covered next. Other chapters cover the problems of finding frequent itemsets and clustering, each from the point of view that the data is too large to fit in main memory. The final chapters cover two applications: recommendation systems and Web advertising, each vital in e-commerce.

Written by two authorities in database and web technologies, this book will be essential for students and practitioners alike.

Preface

This book evolved from material developed over several years by Anand Rajaraman and Jeff Ullman for a one-quarter course at Stanford. The course CS345A, titled "Web Mining," was designed as an advanced graduate course, although it has become accessible and interesting to advanced undergraduates.

What the Book Is About

At the highest level of description, this book is about data mining. However, it focuses on data mining of very large amounts of data, that is, data so large it does not fit in main memory. Because of the emphasis on size, many of our examples are about the Web or data derived from the Web. Further, the book takes an algorithmic point of view: data mining is about applying algorithms to data, rather than using data to "train" a machine-learning engine of some sort. The principal topics covered are:

(1) Distributed file systems and map-reduce as a tool for creating parallel algorithms that succeed on very large amounts of data.

(2) Similarity search, including the key techniques of minhashing and locality-sensitive hashing.

(3) Data-stream processing and specialized algorithms for dealing with data that arrives so fast it must be processed immediately or lost.

(4) The technology of search engines, including Google's PageRank, link-spam detection, and the hubs-and-authorities approach.

(5) Frequent-itemset mining, including association rules, market-baskets, the A-Priori Algorithm and its improvements.

(6) Algorithms for clustering very large, high-dimensional datasets.

(7) Two key problems for Web applications: managing advertising and recommendation systems.

Prerequisites

CS345A, although its number indicates an advanced graduate course, has been found accessible by advanced undergraduates and beginning masters students. In the future, it is likely that the course will be given a mezzanine-level number. The prerequisites for CS345A are:

(1) The first course in database systems, covering application programming in SQL and other database-related languages such as XQuery.
(2) A sophomore-level course in data structures, algorithms, and discrete math.
(3) A sophomore-level course in software systems, software engineering, and programming languages.

Exercises

The book contains extensive exercises, with some for almost every section. We indicate harder exercises or parts of exercises with an exclamation point. The hardest exercises have a double exclamation point.

Support on the Web

You can find materials from past offerings of CS345A at:

`http://infolab.stanford.edu/~ullman/mining/mining.html`

There, you will find slides, homework assignments, project requirements, and in some cases, exams.

Acknowledgements

Cover art is by Scott Ullman. We would like to thank Foto Afrati and Arun Marathe for critical readings of the draft of this manuscript. Errors were also reported by Leland Chen, Shrey Gupta, Xie Ke, Haewoon Kwak, Brad Penoff, Philips Kokoh Prasetyo, Mark Storus, Tim Triche Jr., and Roshan Sumbaly. The remaining errors are ours, of course.

A. R.
J. D. U.
Palo Alto, CA
June, 2011

Contents

1 Data Mining

In this intoductory chapter we begin with the essence of data mining and a discussion of how data mining is treated by the various disciplines that contribute to this field. We cover "Bonferroni's Principle," which is really a warning about overusing the ability to mine data. This chapter is also the place where we summarize a few useful ideas that are not data mining but are useful in understanding some important data-mining concepts. These include the TF.IDF measure of word importance, behavior of hash functions and indexes, and identities involving e, the base of natural logarithms. Finally, we give an outline of the topics covered in the balance of the book.

1.1 What is Data Mining?

The most commonly accepted definition of "data mining" is the discovery of "models" for data. A "model," however, can be one of several things. We mention below the most important directions in modeling.

1.1.1 Statistical Modeling

Statisticians were the first to use the term "data mining." Originally, "data mining" or "data dredging" was a derogatory term referring to attempts to extract information that was not supported by the data. Section 1.2 illustrates the sort of errors one can make by trying to extract what really isn't in the data. Today, "data mining" has taken on a positive meaning. Now, statisticians view data mining as the construction of a *statistical model*, that is, an underlying distribution from which the visible data is drawn.

EXAMPLE 1.1 Suppose our data is a set of numbers. This data is much simpler than data that would be data-mined, but it will serve as an example. A statistician might decide that the data comes from a Gaussian distribution and use a formula to compute the most likely parameters of this Gaussian. The mean and standard deviation of this Gaussian distribution completely characterize the distribution and would become the model of the data.

1.1.2 Machine Learning

There are some who regard data mining as synonymous with machine learning. There is no question that some data mining appropriately uses algorithms from machine learning. Machine-learning practitioners use the data as a training set, to train an algorithm of one of the many types used by machine-learning practitioners, such as Bayes nets, support-vector machines, decision trees, hidden Markov models, and many others.

There are situations where using data in this way makes sense. The typical case where machine learning is a good approach is when we have little idea of what we are looking for in the data. For example, it is rather unclear what it is about movies that makes certain movie-goers like or dislike it. Thus, in answering the "Netflix challenge" to devise an algorithm that predicts the ratings of movies by users, based on a sample of their responses, machine-learning algorithms have proved quite successful. We shall discuss a simple form of this type of algorithm in Section 9.4.

On the other hand, machine learning has not proved successful in situations where we can describe the goals of the mining more directly. An interesting case in point is the attempt by WhizBang! Labs[1] to use machine learning to locate people's resumes on the Web. It was not able to do better than algorithms designed by hand to look for some of the obvious words and phrases that appear in the typical resume. Since everyone who has looked at or written a resume has a pretty good idea of what resumes contain, there was no mystery about what makes a Web page a resume. Thus, there was no advantage to machine-learning over the direct design of an algorithm to discover resumes.

1.1.3 Computational Approaches to Modeling

More recently, computer scientists have looked at data mining as an algorithmic problem. In this case, the model of the data is simply the answer to a complex query about it. For instance, given the set of numbers of Example 1.1, we might compute their average and standard deviation. Note that these values might not be the parameters of the Gaussian that best fits the data, although they will almost certainly be very close if the size of the data is large.

There are many different approaches to modeling data. We have already mentioned the possibility of constructing a statistical process whereby the data could have been generated. Most other approaches to modeling can be described as either

(1) Summarizing the data succinctly and approximately, or
(2) Extracting the most prominent features of the data and ignoring the rest.

We shall explore these two approaches in the following sections.

[1] This startup attempted to use machine learning to mine large-scale data, and hired many of the top machine-learning people to do so. Unfortunately, it was not able to survive.

1.1.4 Summarization

One of the most interesting forms of summarization is the PageRank idea, which made Google successful and which we shall cover in Chapter 5. In this form of Web mining, the entire complex structure of the Web is summarized by a single number for each page. This number, the "PageRank" of the page, is (oversimplifying somewhat) the probability that a random walker on the graph would be at that page at any given time. The remarkable property this ranking has is that it reflects very well the "importance" of the page – the degree to which typical searchers would like that page returned as an answer to their search query.

Another important form of summary – clustering – will be covered in Chapter 7. Here, data is viewed as points in a multidimensional space. Points that are "close" in this space are assigned to the same cluster. The clusters themselves are summarized, perhaps by giving the centroid of the cluster and the average distance from the centroid of points in the cluster. These cluster summaries become the summary of the entire data set.

EXAMPLE 1.2 A famous instance of clustering to solve a problem took place long ago in London, and it was done entirely without computers.[2] The physician John Snow, dealing with a Cholera outbreak plotted the cases on a map of the city. A small illustration suggesting the process is shown in Fig. 1.1.

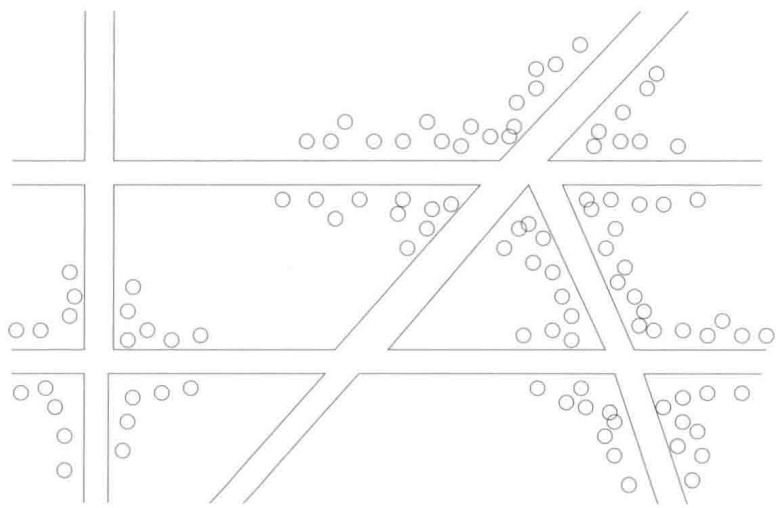

Figure 1.1 Plotting cholera cases on a map of London

The cases clustered around some of the intersections of roads. These intersections were the locations of wells that had become contaminated; people who lived nearest these wells got sick, while people who lived nearer to wells that had not been contaminated did not get sick. Without the ability to cluster the data, the cause of Cholera would not have been discovered.

[2] See `http://en.wikipedia.org/wiki/1854_Broad_Street_cholera_outbreak`.

1.1.5 Feature Extraction

The typical feature-based model looks for the most extreme examples of a phenomenon and represents the data by these examples. If you are familiar with Bayes nets, a branch of machine learning and a topic we do not cover in this book, you know how a complex relationship between objects is represented by finding the strongest statistical dependencies among these objects and using only those in representing all statistical connections. Some of the important kinds of feature extraction from large-scale data that we shall study are:

(1) *Frequent Itemsets.* This model makes sense for data that consists of "baskets" of small sets of items, as in the market-basket problem that we shall discuss in Chapter 6. We look for small sets of items that appear together in many baskets, and these "frequent itemsets" are the characterization of the data that we seek. The orignal application of this sort of mining was true market baskets: the sets of items, such as hamburger and ketchup, that people tend to buy together when checking out at the cash register of a store or super market.

(2) *Similar Items.* Often, your data looks like a collection of sets, and the objective is to find pairs of sets that have a relatively large fraction of their elements in common. An example is treating customers at an on-line store like Amazon as the set of items they have bought. In order for Amazon to recommend something else they might like, Amazon can look for "similar" customers and recommend something many of these customers have bought. This process is called "collaborative filtering." If customers were single-minded, that is, they bought only one kind of thing, then clustering customers might work. However, since customers tend to have interests in many different things, it is more useful to find, for each customer, a small number of other customers who are similar in their tastes, and represent the data by these connections. We discuss similarity in Chapter 3.

1.2 Statistical Limits on Data Mining

A common sort of data-mining problem involves discovering unusual events hidden within massive amounts of data. This section is a discussion of the problem, including "Bonferroni's Principle," a warning against overzealous use of data mining.

1.2.1 Total Information Awareness

In 2002, the Bush administration put forward a plan to mine all the data it could find, including credit-card receipts, hotel records, travel data, and many other kinds of information in order to track terrorist activity. This idea naturally caused great concern among privacy advocates, and the project, called TIA, or

Total Information Awareness, was eventually killed by Congress, although it is unclear whether the project in fact exists under another name. It is not the purpose of this book to discuss the difficult issue of the privacy-security tradeoff. However, the prospect of TIA or a system like it does raise technical questions about its feasibility and the realism of its assumptions.

The concern raised by many is that if you look at so much data, and you try to find within it activities that look like terrorist behavior, are you not going to find many innocent activities – or even illicit activities that are not terrorism – that will result in visits from the police and maybe worse than just a visit? The answer is that it all depends on how narrowly you define the activities that you look for. Statisticians have seen this problem in many guises and have a theory, which we introduce in the next section.

1.2.2 Bonferroni's Principle

Suppose you have a certain amount of data, and you look for events of a certain type within that data. You can expect events of this type to occur, even if the data is completely random, and the number of occurrences of these events will grow as the size of the data grows. These occurrences are "bogus," in the sense that they have no cause other than that random data will always have some number of unusual features that look significant but aren't. A theorem of statistics, known as the *Bonferroni correction* gives a statistically sound way to avoid most of these bogus positive responses to a search through the data. Without going into the statistical details, we offer an informal version, *Bonferroni's principle*, that helps us avoid treating random occurrences as if they were real. Calculate the expected number of occurrences of the events you are looking for, on the assumption that data is random. If this number is significantly larger than the number of real instances you hope to find, then you must expect almost anything you find to be bogus, i.e., a statistical artifact rather than evidence of what you are looking for. This observation is the informal statement of Bonferroni's principle.

In a situation like searching for terrorists, where we expect that there are few terrorists operating at any one time, Bonferroni's principle says that we may only detect terrorists by looking for events that are so rare that they are unlikely to occur in random data. We shall give an extended example in the next section.

1.2.3 An Example of Bonferroni's Principle

Suppose there are believed to be some "evil-doers" out there, and we want to detect them. Suppose further that we have reason to believe that periodically, evil-doers gather at a hotel to plot their evil. Let us make the following assumptions about the size of the problem:

(1) There are one billion people who might be evil-doers.
(2) Everyone goes to a hotel one day in 100.