1 Introduction

Nowadays, we are creating a huge amount of data every day from all kinds of devices, in different formats, from independent or connected applications. This flood of big data has outpaced our capability to process, analyze, store, and understand these datasets. This rapid expansion is accelerated by the dramatic increase in acceptance of social networking applications, which allow users to create content freely and increase the already huge size of the Web.

Furthermore, with mobile phones becoming the sensory gateway to get real-time data on people from different aspects, the vast amount of data that mobile carriers can potentially process to improve our daily life has significantly outpaced our past call data record-based processing, which was designed only for billing purposes. We can foresee that Internet of Things applications will raise the scale of data to an unprecedented level. People and devices (from home coffee machines to cars, to buses, railway stations, and airports) are all loosely connected. Trillions of such connected components will generate a huge data ocean, and valuable information must be discovered from the data to help improve our quality of life and make our world a better place. For example, after we get up every morning, in order to optimize our commute time to work and complete the optimization before we arrive at the office, the system needs to process information from traffic, weather, construction, police activities, and our calendar schedules, and perform deep optimization under tight time constraints.

To deal with this staggeringly large quantity of data, we need fast and efficient methods that operate in real time using a reasonable amount of resources.

1.1 Big Data

It is not really useful to define *big data* in terms of a specific dataset size, for example, on the order of petabytes. A more useful definition is that the dataset is too large to be managed without using nonstandard algorithms or technologies, particularly if it is to be used for extracting knowledge.

While twenty years ago people were struggling with gigabytes, at the time of writing this book the corresponding memory unit in table 1.1 is between the terabyte and the petabyte. There is no question that in a further twenty years, we will be a few lines down from this point.

Big data was characterized by Laney in [154] by the three Vs of big data management:

Table 1.1
Memory units in multiples of bytes.

Memory unit	Decimal Size	Binary size
kilobyte (kB, KB)	10^3	2^{10}
megabyte (MB)	10^6	2^{20}
gigabyte (GB)	10^9	2^{30}
terabyte (TB)	10^{12}	2^{40}
petabyte (PB)	10^{15}	2^{50}
exabyte (EB)	10^{18}	2^{60}
zettabyte (ZB)	10^{21}	2^{70}
yottabyte (YB)	10^{24}	2^{80}

- Volume: There is more data than ever before, and its size continues to increase, but not the percentage of data that our tools can process.
- Variety: There are many different types of data, such as text, sensor data, audio, video, graph, and more, from which we would like to extract information.
- Velocity: Data is arriving continuously in streams, and we are interested in obtaining useful information from it in real time.

Other Vs have been added since then:

- Variability: The structure of the data, or the way users want to interpret the data, changes over time.
- Value: Data is valuable only to the extent that it leads to better decisions, and eventually to a competitive advantage.
- Validity and Veracity: Some of the data may not be completely reliable, and it is important to manage this uncertainty.

Gartner [200] summarizes this in his definition of big data in 2012 as "high volume, velocity and variety information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision making."

Applications of big data should allow people to have better services and better customer experiences, and also be healthier:

- Business: Customer personalization and churn detection (customers moving from one company to a rival one).
- Technology: Reducing processing time from hours to seconds.
- Health: Mining people's medical records and genomics data, to monitor and improve their health.
- Smart cities: Cities focused on sustainable economic development and high quality of life, with wise management of natural resources.

As an example of the usefulness of big data mining, we refer to the work by Global Pulse [236], which uses big data to improve life in developing countries. Global Pulse is a United Nations initiative, functioning as an innovative lab, whose strategy is to mine big data for:

1. Researching innovative methods and techniques for analyzing real-time digital data to detect early emerging vulnerabilities.
2. Assembling a free and open-source technology toolkit for analyzing real-time data and sharing hypotheses.
3. Establishing an integrated, global network of Pulse Labs, to pilot the approach at the country level.

The big data mining revolution is not restricted to the industrialized world, as mobile devices are spreading in developing countries as well. It is estimated that there are over five billion mobile phones, and that 80% are located in developing countries.

1.1.1 Tools: Open-Source Revolution

The big data phenomenon is intrinsically related to the open-source software revolution. Large companies such as Yahoo!, Twitter, LinkedIn, Google, and Facebook both benefitted from and contributed to open-source projects. Some examples are:

- Apache Hadoop [16], a platform for data-intensive distributed applications, based on the MapReduce programming model and the Hadoop Distributed File system (HDFS). Hadoop allows us to write applications that quickly process large amounts of data in parallel on clusters of computing nodes.
- Projects related to Apache Hadoop [260]: Apache Pig, Apache Hive, Apache HBase, Apache ZooKeeper, Apache Cassandra, Cascading, Scribe, and

Apache Mahout [17], which is a scalable machine learning and data mining open-source software based mainly on Hadoop.

- Apache Spark [253], a data processing engine for large-scale data, running on the Hadoop infrastructure. Spark powers a stack of libraries including SQL and DataFrames, MLlib for machine learning, GraphX, and Spark Streaming. These libraries can be combined easily in the same application.
- Apache Flink [62], a streaming dataflow engine that provides data distribution, communication, and fault tolerance for distributed computations over data streams. Flink includes several APIs for creating applications that use the Flink engine. If Apache Spark is a batch data processing engine that can emulate streaming data processing with Spark Streaming using micro-batches of data, Apache Flink is a streaming data processing engine that can perform batch data processing.
- Apache Storm [168], software for streaming data-intensive distributed applications, similar to Apache S4 and Apache Samza.
- TensorFlow [1], an open-source package for machine learning and deep neural networks.

1.1.2 Challenges in Big Data

There are many challenges for the future in big data management and analytics, arising from the very nature of data: large, diverse, and evolving [128]. Some of the challenges that researchers and practitioners will have to deal with in the years to come are:

- Analytics architecture. It is not clear yet how an optimal architecture of an analytics system should be built to deal with historical data and with real-time data at the same time. A first proposal was the Lambda architecture of Nathan Marz [169]. The Lambda architecture solves the problem of computing arbitrary functions on arbitrary data in real time by decomposing the problem into three layers: the batch layer, the serving layer, and the speed layer. It combines in the same system Hadoop for the batch layer and Storm for the speed layer. A more recent proposal is the Kappa architecture, proposed by Kreps from LinkedIn [152]. It simplifies the Lambda architecture, removing the batch processing system.
- Evaluation. It is important to achieve significant statistical results, and not be fooled by randomness. If the "multiple hypothesis problem" is not properly

cared for, it is easy to go wrong with huge datasets and thousands of questions to answer at once, as Efron explains [95]. Also, it will be important to avoid the trap of focusing only on technical measures such as error or speed instead of on eventual real-world impact, as discussed by Wagstaff [242]: arguing against those who believe that big data is all hype is only possible by regularly publishing applications that meet reasonable criteria for a challenge-problem in the sense explained in that paper.

- Distributed mining. Many data mining techniques are not trivial to parallelize. To have distributed versions of some methods, substantial research is needed with both practical experiments and theoretical analysis.
- Time-evolving data. Data may be evolving over time, so it is important that the big data mining techniques are able to adapt to, and in some cases explicitly detect, change. Many data stream mining techniques in this book are motivated by exactly this requirement [110].
- Compression. When dealing with big data, the quantity of space needed to store it is very relevant. There are two main approaches: compression, where we lose no information; and sampling, where we choose data that we deem representative. Using compression, we will use more time and less space, so we can consider it as a transformation from time to space. Using sampling, we are losing information, but the gains in space may be in orders of magnitude. For example Feldman et al. [99] use coresets to reduce the complexity of big data problems; a coreset is a small subset of the data that provably approximates the original data for a given problem.
- Visualization. A main issue in big data analysis is how to visualize the results. Presenting information from large amounts of data in a way that is understandable to humans is quite a challenge. It requires new techniques and frameworks to tell stories, such as those covered in the beautiful book *The Human Face of Big Data* [228].
- Hidden big data. Large quantities of useful data are in fact useless because they are untagged, file-based, and unstructured. The 2012 IDC study on big data [117] explained that, in 2012, 23% (643 exabytes) of the digital universe would be useful for big data if tagged and analyzed. However, at that time only 3% of the potentially useful data was tagged, and even less was analyzed. The figures have probably gotten worse in recent years. The Open Data and Semantic Web movements have emerged, in part, to make us aware and improve on this situation.

1.2 Real-Time Analytics

One particular case of the big data scenario is real-time analytics. It is important for organizations not only to obtain answers to queries immediately, but to do so according to the data that has just arrived.

1.2.1 Data Streams

Data streams are an algorithmic abstraction to support real-time analytics. They are sequences of items, possibly infinite, each item having a timestamp, and so a temporal order. Data items arrive one by one, and we would like to build and maintain models, such as patterns or predictors, of these items in real time. There are two main algorithmic challenges when dealing with streaming data: the stream is large and fast, and we need to extract information in real time from it. That means that usually we need to accept approximate solutions in order to use less time and memory. Also, the data may be evolving, so our models have to adapt when there are changes in the data.

1.2.2 Time and Memory

Accuracy, time, and memory are the three main resource dimensions of the stream mining process: we are interested in methods that obtain the maximum accuracy with minimum time and low total memory. It is possible, as we will show later, to reduce evaluation to a two-dimensional task, by combining memory and time in a single cost measure. Note also that, since data arrives at high speed, it cannot be buffered, so time to process one item is as relevant as the total time, which is the one usually considered in conventional data mining.

1.2.3 Applications

There are many scenarios of streaming data. Here we offer a few example areas:

- Sensor data and the Internet of Things: Every day, more sensors are used in industry to monitor processes, and to improve their quality. Cities are starting to implement huge networks of sensors to monitor the mobility of people and to check the health of bridges and roads, traffic in cities, people's vital constants, and so on.

- Telecommunication data: Telecommunication companies have large quantities of phone call data. Nowadays, mobile calls and mobile phone locations are huge sources of data to be processed, often in real-time.
- Social media: The users of social websites such as Facebook, Twitter, LinkedIn, and Instagram continuously produce data about their interactions and contributions. Topic and community discovery and sentiment analysis are but two of the real-time analysis problems that arise.
- Marketing and e-commerce: Sales businesses are collecting in real time large quantities of transactions that can be analyzed for value. Detecting fraud in electronic transactions is essential.
- Health care: Hospitals collect large amounts of time-sensitive data when caring for patients, for example, monitoring patient vital signs such as blood pressure, heart rate, and temperature. Telemedicine will also monitor patients when they are home, perhaps including data about their daily activity with separate sensors. Also, the system could have results of lab tests, pathology reports, X-rays, and digital imaging. Some of this data could be used in real time to provide warnings of changes in patient conditions.
- Epidemics and disasters: Data from streams originating in the Internet can be used to detect epidemics and natural disasters, and can be combined with official statistics from official centers for disease and disaster control and prevention [63].
- Computer security: Computer systems have to be protected from theft and damage to their hardware, software and information, as well as from disruption or misdirection of the services they provide, in particular, insider threat detection [11, 229] and intrusion detection [194, 195].
- Electricity demand prediction: Providers need to know some time in advance how much power their customers will be requesting. The figures change with time of day, time of year, geography, weather, state of the economy, customer habits, and many other factors, making it a complex prediction problem on massive, distributed data.

1.3 What This Book Is About

Among the many aspects of big data, this book focuses on mining and learning from data streams, and therefore on the techniques for performing data analytics on data that arrives in sequence at high speed. Of the Vs that define big data, the one we address most is therefore Velocity.

The techniques are illustrated in a hands-on way using MOA–Massive Online Analysis. MOA is the most popular open-source framework for data stream mining, with a very active growing community. It includes a collection of machine learning (ML) algorithms (classification, regression, clustering, pattern mining, outlier detection, change detection, and recommender systems) and tools for evaluation. Related to the WEKA project, MOA is also written in Java, while designed to scale to more demanding problems.

Part I is an introduction to the field of big data, an overview of the main techniques, and a first hands-on introduction to MOA usage. Part II presents in detail a good number of algorithms for stream mining, prioritizing those that have been implemented in MOA, and provides pointers to the relevant literature for the reader who is interested in more. Part III delves into MOA in more depth. It presents some additional hands-on exercises and some of the internal structure of MOA, oriented to readers who would like to add algorithms to MOA or modify existing ones, or use MOA through its API in their applications.

Finally, we would like to mention that there are other data stream mining techniques that are very useful and important, but that could not be covered in the book, such as matching problems, motif discovery, ranking/learning to rank, recommender systems, recurrent concept mining, geospatial data, and mining of large streaming graphs. Also, in real projects, aspects such as missing data, feature extraction, outlier detection, and forming training instances, among others, will be important; we do not cover them explicitly here as they tend to be highly problem-dependent.

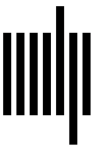
Machine Learning for Data Streams

with Practical Examples in MOA

By: Albert Bifet, Ricard Gavaldà, Geoffrey Holmes, Bernhard Pfahringer

Citation:

Machine Learning for Data Streams: with Practical Examples in MOA
By: Albert Bifet, Ricard Gavaldà, Geoffrey Holmes, Bernhard Pfahringer
DOI: 10.7551/mitpress/10654.001.0001
ISBN (electronic): 9780262346047
Publisher: The MIT Press
Published: 2023



The MIT Press

This book was set in Times Roman and Mathtime Pro 2 by the authors.

Printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Data is available

ISBN: 978-0-262-03779-2

10 9 8 7 6 5 4 3 2 1