Chapter 5 Big Data Analysis

In this section, we present methods, architectures, and tools for big data analysis.

The analysis of big data mainly involves analytical methods for traditional data and big data, analytical architecture for big data, and software used for mining and analysis of big data.

- Traditional Data Analysis
- Big Data Analytic Methods
- Architecture for Big Data Analysis
- Tools for Big Data Mining and Analysis

Big Data Analysis: Traditional Data Analysis

5.1 Traditional Data Analysis

Several representative traditional data analysis methods are examined in the following, many of which are from statistics and computer science.

- Cluster Analysis
- Factor Analysis
- Correlation Analysis and Regression Analysis
- A/B Testing: also called bucket testing. It is a technology for determining plans to improve target variables by comparing the tested group. Big data will require a large number of tests to be executed and analyzed, to ensure sufficient scale of the groups for detecting the significant differences between the control group and the treatment group.
- Statistical Analysis
- Data Mining: Classification, Estimation, Prediction, Affinity grouping or association rules, Clustering, and Description and Visualization

Big Data Analysis-Big Data Analytic Methods

5.2 Big Data Analytic Methods

In the dawn of the big data era, people are concerned with how to rapidly extract key information from massive data so as to bring value to enterprises and individuals. At present, the main processing methods of big data are shown as follows.

- Bloom Filter
- Hashing
- Index
- Triel
- Parallel Computing

Big Data Analysis-Big Data Analytic Methods

Bloom Filter

Bloom Filter is actually a bit array and a series of Hash functions. The principle of Bloom Filter is to store Hash values of data other than data itself by utilizing a bit array, which is in essence a bitmap index that uses Hash functions to conduct lossy compression storage of data. It has such advantages as high space efficiency and high query speed. Bloom Filter applies to big data applications that allow a certain misrecognition rate.

Hashing

It is a method that essentially transforms data into shorter fixed-length numerical values or index values. Hashing has such advantages as rapid reading, writing, and high query speed, but a sound Hash function is hard to be found.

Big Data Analysis-Big Data Analytic Methods

Hashing is a process of converting input data (or a 'message') into a fixed-size string of characters, which is typically a sequence of numbers and letters. The output, often called a hash value or hash code, is generated using a hash function. The primary goal of hashing is to produce a unique hash value for each unique input.

Key characteristics of a good hash function include:

- Deterministic: The same input will always produce the same hash value.
- Efficient: The hash function should be computationally efficient to calculate.
- Avalanche Effect: A small change in the input should produce a significantly different hash value.
- Fixed Output Length: The hash function should produce a fixed-size output, regardless of the size of the input.

Big Data Analysis-Big Data Analytic Methods

Hashing is widely used in various computer science applications, including:

- Data Integrity: Hash functions are used to verify the integrity of data by generating a hash value (checksum) of the original data. If the data changes, the hash value will also change.
- Password Storage: Hash functions are employed to securely store passwords. Instead of storing actual passwords, systems store their hash values. During login attempts, the system hashes the entered password and compares it to the stored hash.
- Hash Tables: Hashing is fundamental to the implementation of hash tables, a data structure used for efficient data retrieval. It allows for quick lookups, insertions

Big Data Analysis-Big Data Analytic Methods

Index

Index is always an effective method to reduce the expense of disc reading and writing, and improve insertion, deletion, modification, and query speeds in both traditional relational databases that manage structured data, and technologies that manage semi-structured and unstructured data.

Triel

Triel is also called trie tree, a variant of Hash Tree. It is mainly applied to rapid retrieval and word frequency statistics. The main idea of Triel is to utilize common prefixes of character strings to reduce comparison on character strings to the greatest extent, so as to improve query efficiency.

Parallel Computing

Compared to traditional serial computing, parallel computing refers to utilizing several computing resources to complete a computation task. Its basic idea is to decompose a problem and assign them to several independent processes to be independently completed, so as to achieve coprocessing. Presently, some classic parallel computing models include MPI (Message Passing Interface), MapReduce, and Dryad.

5.3 Architecture for Big Data Analysis

Due to the wide range of sources and variety, different structures, and the broad application fields of big data, different analytical architectures shall be considered for big data with different application requirements.

- Real-Time vs. Offline Analysis Big data analysis can be classified into real-time analysis and offline analysis. Real-time analysis includes (a) parallel processing clusters using traditional relational databases, and (b) memory-based computing platforms. For example, Greenplum from EMC and HANA from SAP are all real-time analysis architectures. Offline analysis generally conducts analysis by importing big data of logs into a special platform through data acquisition tools. Under the big data setting, many Internet enterprises utilize the offline analysis architecture based on Hadoop in order to reduce the cost of data format conversion and improve the efficiency of data acquisition. Examples include Facebook's open source tool Scribe, LinkedIn's open source tool Kafka, Taobao's open source tool Timetunnel, and Chukwa of Hadoop, etc.

Big Data Analysis-Architecture for Big Data Analysis

- Analysis at Different Levels Big data analysis can also be classified into memory level analysis, Business Intelligence (BI) level analysis, and massive level analysis, which are examined in the following.
  Memory-Level: Memory-level analysis is for the case when the total data volume is within the maximum level of the memory of a clusters. Memory-level analysis is extremely suitable for real-time analysis. MongoDB is a representative memory-level analytical architecture.
  BI: BI analysis is for the case when the data scale surpasses the memory level but may be imported into the BI analysis environment. Currently, mainstream BI products are provided with data analysis plans supporting the level over TB.
  Massive: Massive analysis for the case when the data scale has completely surpassed the capacities of BI products and traditional relational databases. At present, most massive analysis utilize HDFS of Hadoop to store data and use MapReduce for data analysis. Most massive analysis belongs to the offline analysis category.

Big Data Analysis-Architecture for Big Data Analysis

- Analysis with Different Complexity The time and space complexity of data analysis algorithms differ greatly from each other according to different kinds of data and application demands. For example, for applications that are amenable to parallel processing, a distributed algorithm may be designed and a parallel processing model may be used for data analysis.

5.4 Tools for Big Data Mining and Analysis

Many tools for big data mining and analysis are available, including professional and amateur software, expensive commercial software, and free open source software.

- R
- Excel
- Rapid-I Rapidminer
- KNIME
- Weka/Pentaho

Analytics Models for Data Science

Analytics Models for Data Science

The ultimate goal of data science is to turn raw data into data products. Data analytics is the science of examining the raw data with the purpose of making correct decisions by drawing meaningful conclusions. The key differences of traditional analytics versus Big Data Analytics are shown in Table.

Analytics Models for Data Science

Concept	Traditional analytics	Big Data Analytics
Focus on	• Descriptive analytics • Diagnosis analytics	• Predictive analytics • Data science • Innovate with machine learning
Datasets	• Limited datasets • Less types of data • Cleansed data • Structured data	• Large-scale/unlimited datasets • More types of data • Raw data • Semi-structured/unstructured data
Data models	• Simple data models	• Complex data models
Data architecture	• Centralized database architecture in which complex and large problems are solved in a single system	• Distributive database architecture in which complex and large problems are solved by dividing into many chunks
Data schema	• Fixed/static schema for data storage	• Dynamic schema for data storage

AMDS-1.2 Data Model

Data Model

Datamodel is a process of arriving at the diagram for deep understanding, organizing and storing the data for service, access and use. The process of representing the data in a pictorial format helps the business and the technology experts to understand the data and get to know how to use the data. This section deals with data science and four computing models of data analytics

- Data Products
- Data Munging
- Descriptive Analytics
- Predictive Analytics
- Data Science
- Network Science

AMDS-1.2.1 Data Products

Data Products

Three types of data products are as follows:

- Data used to predict,
- Data used to recommend,
- Data used to benchmark

Data Munging

Data munging, sometimes referred to as data wrangling, is the process of converting and mapping data from one format (raw data) to another format (desired format) with the purpose of making it more suitable, easier and valuable for analytics. Once data retrial is done from any source, for example, from the web, it needs to be stored in an easy-to-use format. Suppose a data source provides reviews in terms of rating in stars (1–5 stars); this can be mapped with the response variable of the form $X \epsilon \{1, 2, 3, 4, 5\}$. Another data source provides reviews using thumb rating system, thumbs-up and thumbs-down; this can be inferred with a response variable of the form $X \epsilon \{positive, negative\}$. In order to make a combined decision, first data source response (five-point star rating) representation has to be converted to the second form (two-point logical rating), by considering one and two stars as negative and three, four and five stars as positive. This process often requires more time allocation to be delivered with good quality.

Data Munging

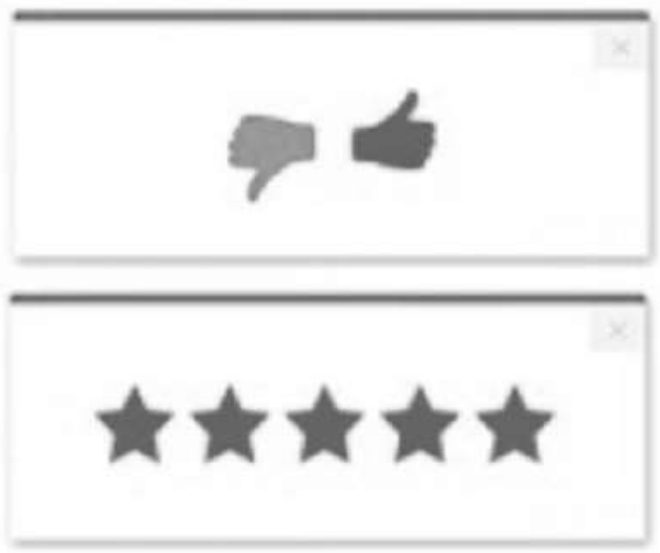


Figure: Thumb and star rating system

AMDS-1.2.3 Descriptive Analytics

Descriptive Analytics

Descriptive analytics is the process of summarizing historical data and identifying patterns and trends.

The descriptive analysis of data provides the following:

- Information about the certainty/uncertainty of the data,
- Indications of unexpected patterns,
- Estimates and summaries and organize them in graphs, charts and tables and
- Considerable observations for doing formal analysis.

Once the data is grouped, different statistical measures are used for analyzing data and drawing conclusions. The data was analyzed descriptively in terms of

1. Measures of probability,
2. Measures of central tendency,
3. Measures of variability,
4. Measures of divergence from normality,
5. Graphical representation.

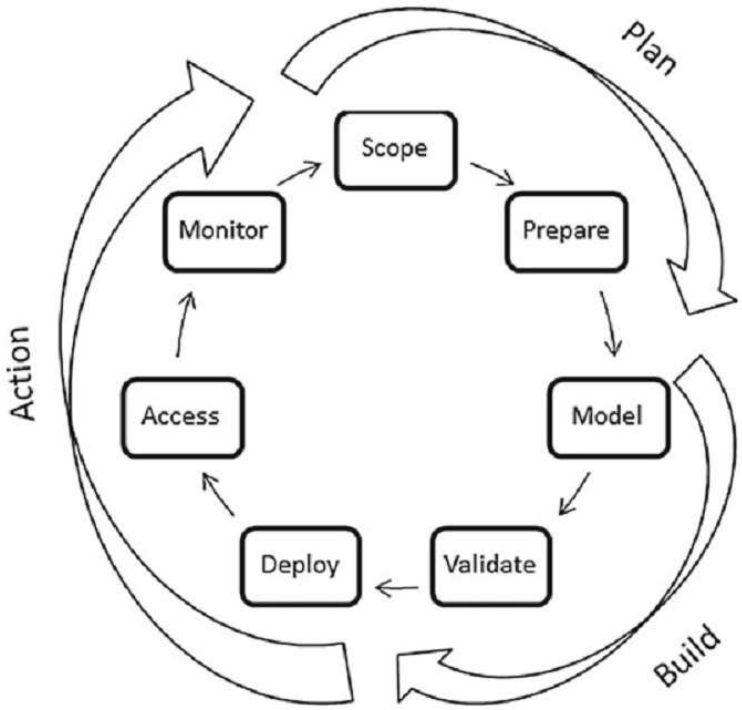
AMDS-1.2.4 Predictive Analytics

Predictive Analytics

Predictive analytics is defined to have data modeling for making confident predictions about the future or any unknown events using business forecasting and simulation which depends upon the observed past occurrences. The predictive modeling process for a wide range of businesses is depicted well in the following figure.

Sentiment analysis is the most common model of predictive analytics. This model takes plain text as input and provides sentiment score as output which in turn determines whether the sentiment is neutral, positive or negative. The best example of predictive analytics is to compute the credit score that helps financial institutions like banking sectors to decide the probability of a customer paying credit bills on time.

Predictive Analytics



Predictive Analytics

Other models for performing predictive analytics are -

- Time series analysis,
- Econometric analysis,
- Decision trees,
- Naive Bayes classifier,
- Ensembles,
- Boosting,
- Support vector machines,
- Linear and logistic regression,
- Artificial neural network,
- Natural language processing,
- Machine learning.

Data Science

Data science is a field associated with data capturing, cleansing, preparation, align- ment and analysis to extract information from the data to solve any kind of problem. It is a combination of statistics, mathematics and programming.

One way of assessing how an organization currently interacts with data and determines where they fit to achieve increasingly accurate hindsight, insight and foresight and its different ecosystems in four fundamental ways (i) descriptive analytics, (ii) diagnostic analytics, (iii) predictive analytics and (iv) prescriptive analytics by Gartner's analytics maturity model.

Data Science

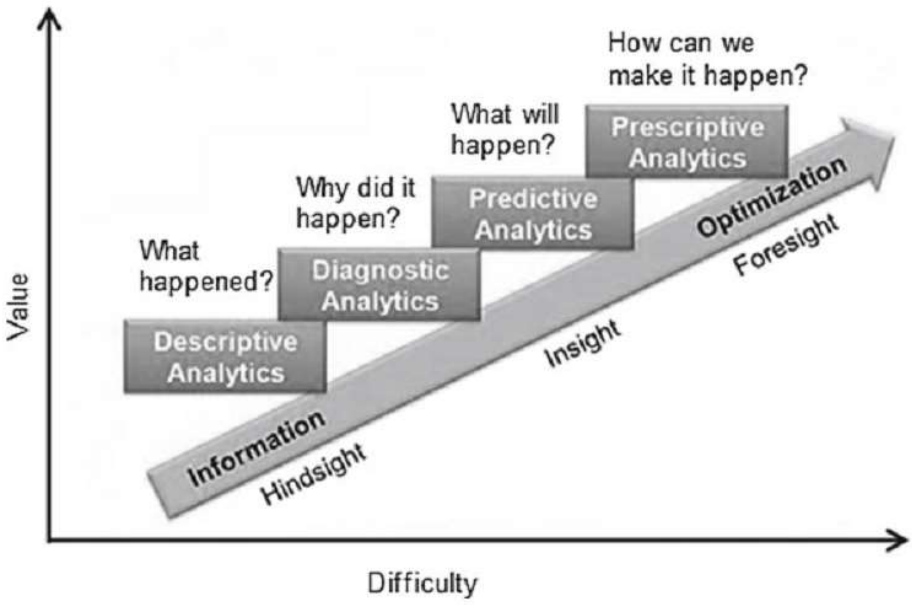


Figure: Gartner's analytics maturity model

Data Science

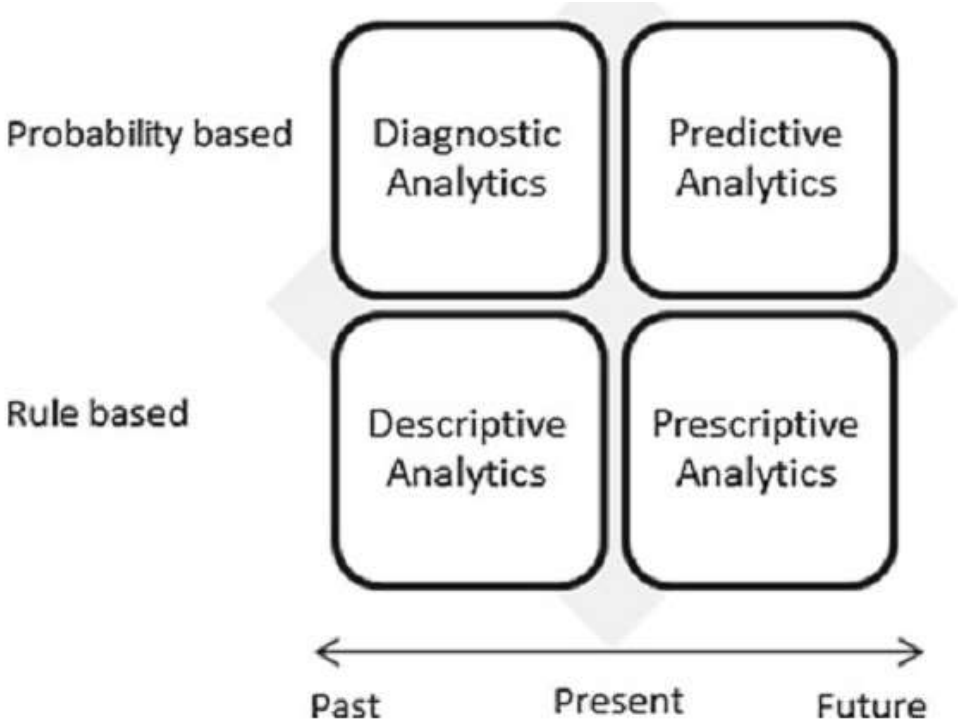


Figure: Four types of analytics

Data Science

Key differences of four types of analytics

Descriptive analytics	Diagnostic analytics	Predictive analytics	Prescriptive analytics
Backward focus	Backward focus	Forward focus	Forward focus
Rule-based	Probability-based	Probability-based	Rule-based
Live data that is comprehensive, accurate and good visualization	Find out the root cause of the problem and remove all confusing information	Historical pattern to predict specific outcomes using algorithms	Applying advanced analytical techniques to make specific recommendations

Diagnostic Analytics

It is a form of analytics which examines data to answer the question 'why did it happen?' It is kind of root cause analysis that focuses on the processes and causes, key factors and unseen patterns. Steps for diagnostic analytics:

- Identify the worthy problem for investigation.
- Perform the Analysis: This step finds a statistically valid relationship between two datasets, where the upturn or downturn occurs. Diagnostic analytics can be done by the following techniques: Multiregression, Self-organizing maps, Cluster and factor analysis, Bayesian clustering, k-nearest neighbors, Principal component analysis, Graph and affinity analysis.
- Filter the Diagnoses: Analyst must identify the single or at most two influential factors from the set of possible causes.

Prescriptive Analytics

This model determines what actions to take in order to change undesirable trends. Prescriptive analytics is defined as deriving optimal planning decisions given the predicted future and addressing questions such as 'what shall we do?' and 'why shall we do it? Prescriptive analytics is based on -

- Optimization that helps achieving the best outcomes,
- Stochastic optimization that helps understanding how to identify data uncertainties to make better decisions and accomplish the best outcome.

Prescriptive analytics is a combination of data, business rules and mathematical models. It uses optimization and simulation models such as sensitivity and scenario analysis, linear and nonlinear programming and Monte Carlo simulation.

Network Science

In networked systems, standardized graph-theoretic methods are used for analyzing data. These methods have the assumption that we precisely know the microscopic details such as how the nodes are interconnected with each other. Mapping network topologies can be costly, time consuming, inaccurate, and the resources they demand are often unaffordable, because the datasets comprised billions of nodes and hundreds of billions of links in large decentralized systems like Internet. This problem can be addressed by combining methods from statistical physics and random graph theory.

The following steps are used for analyzing large-scale networked systems -

- Step 1:To compute the aggregate statistics of interest. It includes the number of nodes, the density of links, the distribution of node degrees, diameter of a network, shortest path length between pair of nodes, correlations between neighboring nodes, clustering coefficient, connectedness, node centrality and node influence. In distributed systems, this can be computed and esti- mated efficiently by sensible sampling techniques.

- Step 2: To determine the statistical entropy ensembles. This can be achieved by probability spaces by assigning probabilities to all possible network realizations that are consistent with the given aggregate statistics using analytical tools like computational statistics methods of metropolis sampling.
- Step 3: Finally, derive the anticipated properties of a system based on aggregate statistics of its network topology.

AMDS-1.3 Computing Models

Big data grows super-fast in four dimensions (4Vs: volume, variety, velocity, and veracity) and needs advanced data structures, new models for extracting the details, and novel algorithmic approaches for computation. This section focuses on data structure for big data, feature engineering, and computational algorithms.

- Data Structures for Big Data
- Feature Engineering for Structured Data
  - Feature Construction
  - Feature Extraction
  - Feature Selection (Algorithms for feature selection..)
  - Feature Learning
  - Ensemble Learning (Bagging, Boosting, Stacking)

AMDS-1.3 Computing Models

AMDS-1.3 Computing Models...

- Computational Algorithm: K-means clustering, Association rule mining, Linear regression, Logistic regression, C4.5, Support vector machine (SVM), Apriori, Expectation–maximization (EM), AdaBoost, and Naïve Bayesian.
- Programming Models
- Parallel Programming: Map and Reduce
- Functional Programming
- Distributed Programming

AMDS-1.3.1 Data Structures for Big Data

AMDS-1.3.1 Data Structures for Big Data

In big data, special data structures are required to handle huge datasets. Hash tables, train/atrain, and tree-based structures like B trees and K-D trees are best suited for handling big data.

- Hash table
- Tree-based Data Structure
- Train and Atrian

Hash table: Hash tables use the hash function to compute the index and map keys to values. This section deals with four commonly used probabilistic data structures:

- Membership query—Bloom filter
- HyperLogLog
- Count–min sketch
- MinHash

AMDS-1.3.1 Data Structures for Big Data..

Membership query—Bloom filter: A Bloom filter proposed by Burton Howard Bloom in 1970 is a space-efficient probabilistic data structure that allows one to reduce the number of exact checks, which is used to test whether an element is 'a member' or 'not a member' of a set. Here, the query returns the probability with the result either 'may be in set' or 'definitely not in set.'

Bit vector is the base data structure for a Bloom filter. Each empty Bloom filter is a bit array of 'm' bits and is initially unset.

0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	2	3	4	5	6	7	8	9	10	11	12

When an element is added to the filter, it is hashed by 'k' functions, $h_1, h_2 ... h_k$ mod by 'm', resulting in 'k' indices into the bit array, and the respective index is set to '1'.

AMDS-1.3 Computing Models

What is the bit (binary digit) in computer?

A bit (binary digit) is the smallest unit of data that a computer can process and store. A bit is always in one of two physical states, similar to an on/off light switch. The state is represented by a single binary value, usually a 0 or 1. However, the state might also be represented by yes/no, on/off or true/false.

In other words- "A bit is a binary digit, the smallest increment of data on a computer. A bit can hold only one of two values: 0 or 1, corresponding to the electrical values of off or on, respectively."

Because bits are so small, you rarely work with information one bit at a time. Bits are usually assembled into a group of eight to form a byte. A byte contains enough information to store a single ASCII character, like "h"."

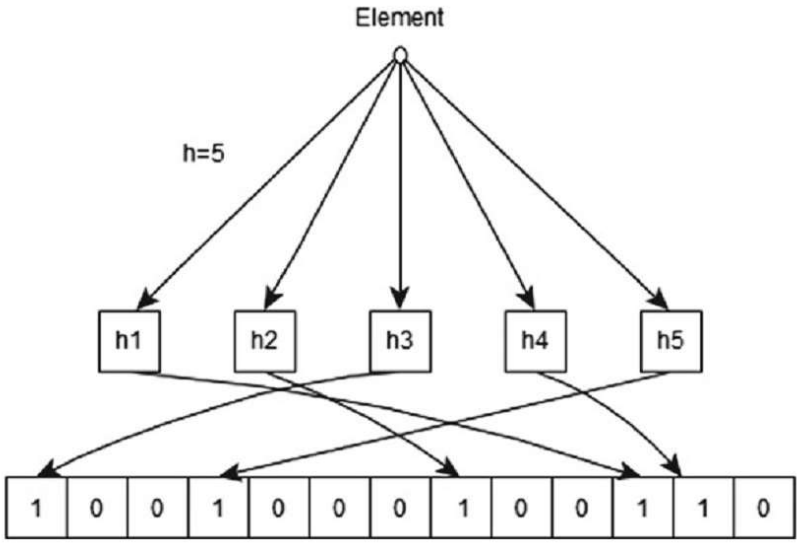


Figure: Bloom filter hashing.

To query the membership of an element, we hash the element again with the same hashing functions and check if each corresponding bit is set. If any one of them is zero, then conclude the element is not present.

Suppose you are generating an online account for a shopping Web site and you are asked to enter a username during sign-up; as you entered, you will get an immediate response, 'Username already exists'. Bloom filter data structure can perform this task very quickly by searching from the millions of registered users.

AMDS-1.3 Computing Models

Consider you want to add a username 'Smilie' into the dataset and five hash functions, h_1, h_2, h_3, h_4, h_5 are applied on the string.

$h_1("Smilie")\%13 = 10$
$h_2("Smilie")\%13 = 4$
$h_3("Smilie")\%13 = 0$
$h_4("Smilie")\%13 = 11$
$h_5("Smilie")\%13 = 6$

Set the bits to 1 for the indices 10, 4, 0, 11, and 6.

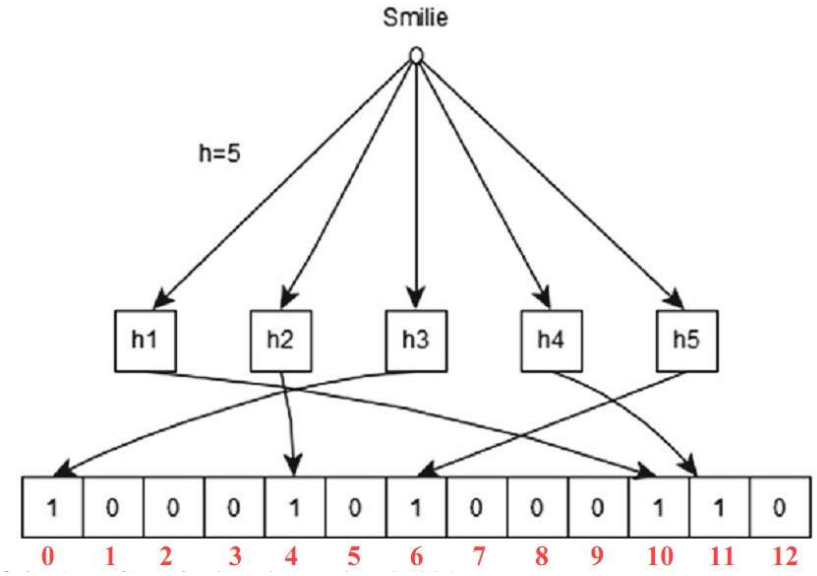


Figure: Bloom filter after inserting a string 'Smilie'

AMDS-1.3 Computing Models

Similarly, enter the next username 'Laughie' by applying the same hash functions.

$h_1("Laughie")\%13 = 3$
$h_2("Laughie")\%13 = 5$
$h_3("Laughie")\%13 = 8$
$h_4("Laughie")\%13 = 10$
$h_5("Laughie")\%13 = 12$

Set the bits to 1 for the indices 3, 5, 8, 10, and 12.

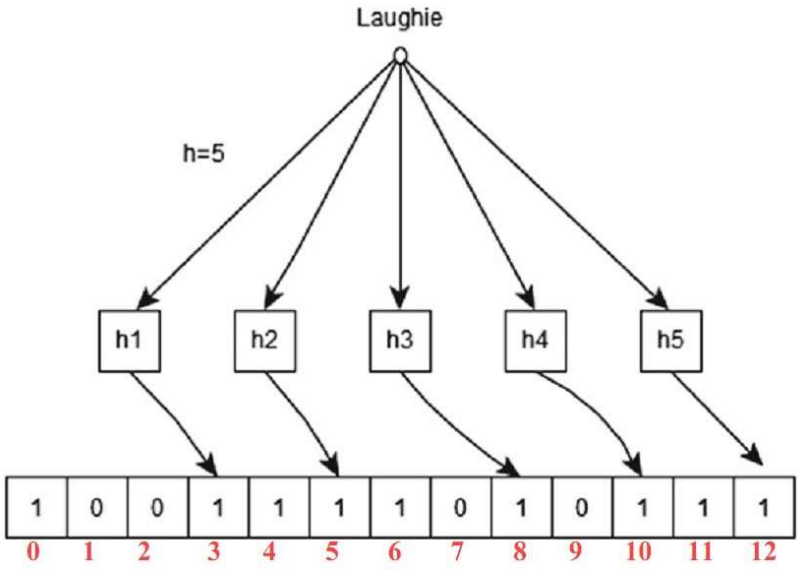


Figure: Bloom filter after inserting a string 'Laughie'.

Now check the availability of the username 'Smilie' is presented in filter or not. For performing this task, apply hashing using h_1, h_2, h_3, h_4, h_5 functions on the string and check if all these indices are set to 1. If all the corresponding bits are set, then the string is 'probably present.' If any one of them indicates 0, then the string is 'definitely not present.'

Let us consider another new username 'Bean'. Suppose we want to check whether 'Bean' is available or not. The result after applying the hash functions h_1, h_2, h_3, h_4, h_5 is as follows:

$h_1("Bean")\%13 = 6$
$h_2("Bean")\%13 = 4$
$h_3("Bean")\%13 = 0$
$h_4("Bean")\%13 = 11$
$h_5("Bean")\%13 = 12$

If we check the bit array after applying hash function to the string 'Bean,' bits at these indices are set to 1 but the string 'Bean' was never added earlier to the Bloom filter. As the indices are already set by some other elements, Bloom filter incorrectly claims that 'Bean' is present and thus will generate a false-positive result.

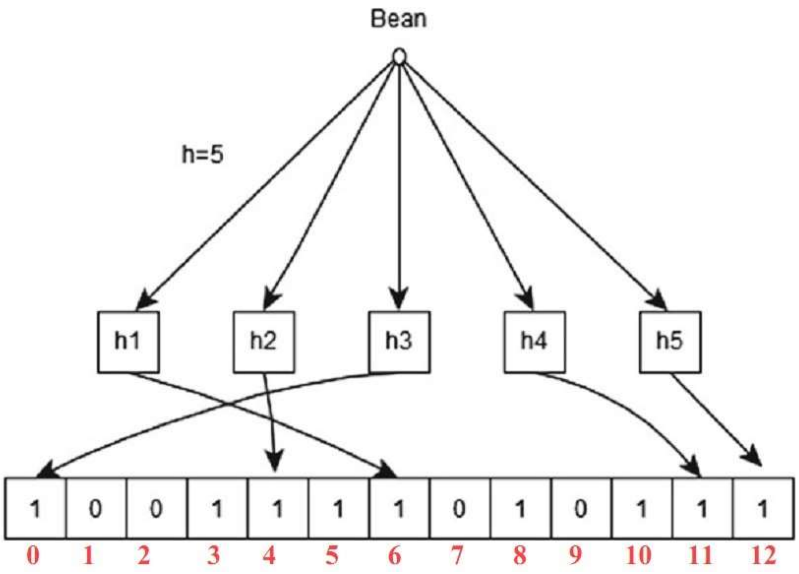


Figure: Bloom filter after inserting a string 'Bean'.

AMDS-1.3 Computing Models

The probability of false positivity 'P' can be calculated as:

$$P = \left(1 - \left[1 - \frac{1}{m}\right]^{kn}\right)^k$$

where, 'm' is the size of bit array,
'k' is the number of hash functions and
'n' is the number of expected elements to be inserted.

Size of bit array 'm' can be calculated as:

$$m = \frac{n \ln P}{(\ln 2)^2}$$

Optimum number of hash functions 'k' can be calculated as:

$$k = \frac{m}{n} \ln 2$$

where 'k' must be a positive integer.

AMDS-1.3 Computing Models

See details about the following topics in Books - "Big Data Analytics: Systems, Algorithms, Applications" by C. S. R. Prabhu, Aneesh Sreevallabh Chivukula, Aditya Mogadala · Rohit Ghosh, L. M. Jenila Livingston.

Cardinality—HyperLogLog
Frequency—Count–Min sketch
Similarity—MinHash
Tree-based Data Structure
K-D Trees
Train and Atrian

AMDS-1.3.2 Feature Engineering for Structured Data

See details about the following topics in Books - "Big Data Analytics: Systems, Algorithms, Applications" by C. S. R. Prabhu, Aneesh Sreevallabh Chivukula, Aditya Mogadala · Rohit Ghosh, L. M. Jenila Livingston.

AMDS-1.3.3 Computational Algorithm

Businesses are increasingly relying on the analysis of their massive amount of data to predict consumer response and recommend products to their customers. Classification, regression, and similarity matching are the fundamental principles on which many of the algorithms are used in applied data science to address big data issues.

- k-means clustering
- Association rule mining
- Linear regression
- Logistic regression
- C4.5, CART, etc.
- Support vector machine (SVM)
- Apriori
- Expectation–maximization (EM)
- AdaBoost
- Naive Bayesian

Predictive Modeling for Unstructured Data

PMUD-1 Predictive Modeling for Unstructured Data

PMUD-1 Predictive Modeling for Unstructured Data

- Background
- Applications of Predictive Modeling
  - Natural Language Processing
  - Computer Vision
  - Information Retrieval
  - Speech Recognition
- Feature Engineering
  - Feature Extraction and Weighing
  - Feature Selection
- Pattern Mining for Predictive Modeling
  - Probabilistic Graphical Models
  - Deep Learning
  - Convolutional Neural Networks (CNN)
  - Recurrent Neural Networks (RNNs)
  - Deep Boltzmann Machines (DBM)
  - Autoencoders

Machine Learning Algorithms for Big Data

MLABD-1 Machine Learning Algorithms for Big Data

- Background
- Generative Versus Discriminative Algorithms
- Supervised Learning for Big Data
  - Decision Trees
  - Logistic Regression
  - Regression and Forecasting
  - Supervised Neural Networks
  - Support Vector Machines
- Unsupervised Learning for Big Data
  - Spectral Clustering
  - Principal Component Analysis (PCA)
  - Latent Dirichlet Allocation (LDA)
  - Matrix Factorization
  - Manifold Learning

MLABD-1 Machine Learning Algorithms for Big Data.....

- Semi-supervised Learning for Big Data
  - Co-training
  - Label Propagation
  - Multiview Learning
- Reinforcement Learning Basics for Big Data
  - Markov Decision Process
  - Planning
  - Reinforcement Learning in Practice
- Online Learning for Big Data

ML

***** - /// - There are no End .. - /// - *****

****

****