Information Fusion 000 (2015) 1–15

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Social big data: Recent achievements and new challenges

Gema Bello-Orgaz [a], Jason J. Jung [b,*], David Camacho [a]

[a] *Computer Science Department, Universidad Autónoma de Madrid, Spain*
[b] *Department of Computer Engineering, Chung-Ang University, Seoul, Republic of Korea*

ARTICLE INFO

Article history:
Received 23 July 2015
Revised 7 August 2015
Accepted 17 August 2015
Available online xxx

Keywords:
Big data
Data mining
Social media
Social networks
Social-based frameworks and applications

ABSTRACT

Big data has become an important issue for a large number of research areas such as data mining, machine learning, computational intelligence, information fusion, the semantic Web, and social networks. The rise of different big data frameworks such as Apache Hadoop and, more recently, Spark, for massive data processing based on the MapReduce paradigm has allowed for the efficient utilisation of data mining methods and machine learning algorithms in different domains. A number of libraries such as Mahout and SparkMLib have been designed to develop new efficient applications based on machine learning algorithms. The combination of big data technologies and traditional machine learning algorithms has generated new and interesting challenges in other areas as social media and social networks. These new challenges are focused mainly on problems such as data processing, data storage, data representation, and how data can be used for pattern mining, analysing user behaviours, and visualizing and tracking data, among others. In this paper, we present a revision of the new methodologies that is designed to allow for efficient data mining and information fusion from social media and of the new applications and frameworks that are currently appearing under the "umbrella" of the social networks, social media and big data paradigms.

1. Introduction

Data volume and the multitude of sources have experienced exponential growth, creating new technical and application challenges; data generation has been estimated at 2.5 Exabytes (1 Exabyte = 1.000.000 Terabytes) of data per day [1]. These data come from everywhere: sensors used to gather climate, traffic and flight information, posts to social media sites (Twitter and Facebook are popular examples), digital pictures and videos (YouTube users upload 72 hours of new video content per minute [2]), transaction records, and cell phone GPS signals, to name a few. The classic methods, algorithms, frameworks, and tools for data management have become both inadequate for processing this amount of data and unable to offer effective solutions for managing the data growth. The problem of managing and extracting useful knowledge from these data sources is currently one of the most popular topics in computing research [3,4].

In this context, **big data** is a popular phenomenon that aims to provide an alternative to traditional solutions based on databases and data analysis. Big data is not just about storage or access to data; its solutions aim to analyse data in order to make sense of them and exploit their value. Big data refers to datasets that are terabytes to petabytes (and even exabytes) in size, and the massive sizes of these datasets extend beyond the ability of average database software tools to capture, store, manage, and analyse them effectively.

The concept of big data has been defined through the **3V model**, which was defined in 2001 by Laney [5] as: *"high-volume, high-velocity and high-variety information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision making"*. More recently, in 2012, Gartner [6] updated the definition as follows: *"Big data is high volume, high velocity, and/or high variety information assets that require new forms of processing to enable enhanced decision making, insight discovery and process optimization"*. Both definitions refer to the three basic features of big data: **V**olume, **V**ariety, and **V**elocity. Other organisations, and big data practitioners (e.g., researchers, engineers, and so on), have extended this 3V model to a 4V model by including a new "V": **V**alue [7]. This model can be even extended to 5Vs if the concepts of **V**eracity is incorporated into the big data definition.

Summarising, this set of $*$V-models provides a straightforward and widely accepted definition related to what is (and what is not) a big-data-based problem, application, software, or framework. These concepts can be briefly described as follows [5,7]:

- ***V**olume*: refers to large amounts of any kind of data from any different sources, including mobile digital data creation devices and digital devices. The benefit from gathering, processing, and analysing these large amounts of data generates a number

[*] Corresponding author. Tel.: +821020235863.
E-mail addresses: gema.bello@uam.es (G. Bello-Orgaz), j2jung@gmail.com, j2jung@yahoo.com, j3ung@cau.ac.kr (J.J. Jung), david.camacho@uam.es (D. Camacho).

http://dx.doi.org/10.1016/j.inffus.2015.08.005
1566-2535/© 2015 Elsevier B.V. All rights reserved.

of challenges in obtaining valuable knowledge for people and companies (see Value feature).

- ***Velocity***: refers to the speed of data transfers. The data's contents are constantly changing through the absorption of complementary data collections, the introduction of previous data or legacy collections, and the different forms of streamed data from multiple sources. From this point of view, new algorithms and methods are needed to adequately process and analyse the online and streaming data.
- ***Variety***: refers to different types of data collected via sensors, smartphones or social networks, such as videos, images, text, audio, data logs, and so on. Moreover, these data can be structured (such as data from relational databases) or unstructured in format.
- ***Value***: refers to the process of extracting valuable information from large sets of social data, and it is usually referred to as *big data analytics*. Value is the most important characteristic of any big-data-based application, because it allows to generate useful business information.
- ***Veracity***: refers to the correctness and accuracy of information. Behind any information management practice lie the core doctrines of data quality, data governance, and metadata management, along with considerations of privacy and legal concerns.

Some examples of potential big data sources are the Open Science Data Cloud [8], the European Union Open Data Portal, open data from the U.S. government, healthcare data, public datasets on Amazon Web Services, etc. **Social media** [9] has become one of the most representative and relevant data sources for big data. Social media data are generated from a wide number of Internet applications and Web sites, with some of the most popular being Facebook, Twitter, LinkedIn, YouTube, Instagram, Google, Tumblr, Flickr, and WordPress. The fast growth of these Web sites allow users to be connected and has created a new generation of people (maybe a new kind of society [10]) who are enthusiastic about interacting, sharing, and collaborating using these sites [11]. This information has spread to many different areas such as everyday life [12] (e-commerce, e-business, e-tourism, hobbies, friendship, ...), education [13], health [14], and daily work.

In this paper, we assume that *social big data* comes from joining the efforts of the two previous domains: *social media* and *big data*. Therefore, **social big data** will be based on the analysis of vast amounts of data that could come from multiple distributed sources but with a **strong focus on social media**. Hence, social big data analysis [15,16] is inherently interdisciplinary and spans areas such as data mining, machine learning, statistics, graph mining, information retrieval, linguistics, natural language processing, the semantic Web, ontologies, and big data computing, among others. Their applications can be extended to a wide number of domains such as health and political trending and forecasting, hobbies, e-business, cybercrime, counterterrorism, time-evolving opinion mining, social network analysis, and humanmachine interactions. The concept of *social big data* can be defined as follows:

"Those processes and methods that are designed to provide sensitive and relevant knowledge to any user or company from social media data sources when data sources can be characterised by their different formats and contents, their very large size, and the online or streamed generation of information."

The gathering, fusion, processing and analysing of the big social media data from unstructured (or semi-structured) sources to extract value knowledge is an extremely difficult task which has not been completely solved. The classic methods, algorithms, frameworks and tools for data management have became inadequate for processing the vast amount of data. This issue has generated a large number of open problems and challenges on social big data domain related to different aspects as knowledge representation, data management, data processing, data analysis, and data visualisation [17]. Some of these challenges include accessing to very large quantities of unstructured data (management issues), determination of how much data is enough for having a large quantity of high quality data (quality versus quantity), processing of data stream dynamically changing, or ensuring the enough privacy (ownership and security), among others. However, given the very large heterogeneous dataset from social media, one of the major challenges is to identify the valuable data and how analyse them to discover useful knowledge improving decision making of individual users and companies [18].

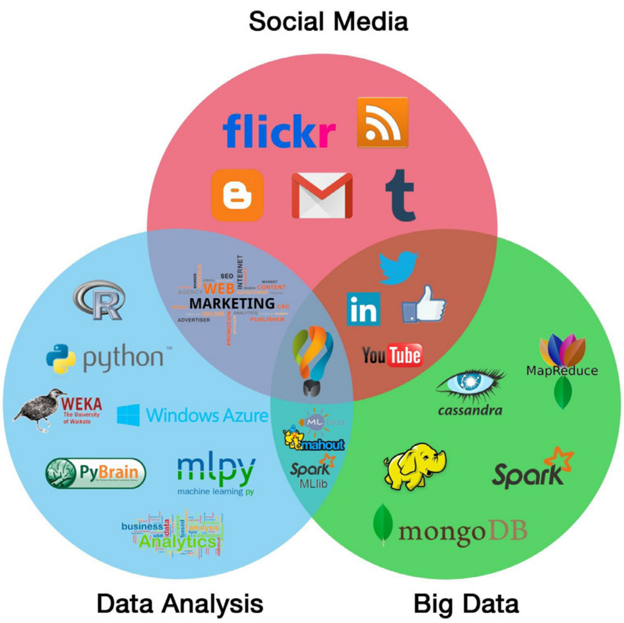

Fig. 1. The conceptual map of Social BigData.

In order to analyse the social media data properly, the traditional analytic techniques and methods (**data analysis**) require adapting and integrating them to the new big data paradigms emerged for massive data processing. Different big data frameworks such as Apache Hadoop [19] and Spark [20] have been arising to allow the efficient application of data mining methods and machine learning algorithms in different domains. Based on these big data frameworks, several libraries such as Mahout [21] and SparkMLib [22] have been designed to develop new efficient versions of classical algorithms. This paper is focused on review those new methodologies, frameworks, and algorithms that are currently appearing under the big data paradigm, and their applications to a wide number of domains such as e-commerce, marketing, security, and healthcare.

Finally, summarising the concepts mentioned previously, Fig. 1 shows the conceptual representation of the three basic social big data areas: **social media** as a natural source for data analysis; **big data** as a parallel and massive processing paradigm; and **data analysis** as a set of algorithms and methods used to extract and analyse knowledge. The intersections between these clusters reflect the concept of mixing those areas. For example, the intersection between big data and data analysis shows some machine learning frameworks that have been designed on top of big data technologies (Mahout [21], MLBase [23,24], or SparkMLib [22]). The intersection between data analysis and social media represents the concept of current Web-based applications that intensively use social media information, such as applications related to marketing and e-health that are described in Section 4. The intersection between big data and social media is reflected in some social media applications such as LinkedIn, Facebook, and Youtube that are currently using big data technologies (MongoDB, Cassandra, Hadoop, and so on) to develop their Web systems.

Finally, the centre of this figure only represents the main goal of any social big data application: knowledge extraction and exploitation.

The rest of the paper is structured as follows; Section 2 provides an introduction to the basics on the methodologies, frameworks, and software used to work with big data. Section 3 provides a description of the current state of the art in the data mining and data analytic techniques that are used in social big data. Section 4 describes a number of applications related to marketing, crime analysis, epidemic intelligence, and user experiences. Finally, Section 5 describes some of the current problems and challenges in social big data; this section also provides some conclusions about the recent achievements and future trends in this interesting research area.

2. Methodologies for social big data

Currently, the exponential growth of social media has created serious problems for traditional data analysis algorithms and techniques (such as data mining, statistics, machine learning, and so on) due to their high computational complexity for large datasets. This type of methods does not properly scale as the data size increases. For this reason, the methodologies and frameworks behind the big data concept are becoming very popular in a wide number of research and industrial areas.

This section provides a short introduction to the methodology based on the MapReduce paradigm and a description of the most popular framework that implements this methodology, Apache Hadoop. Afterwards Apache Spark is described as emerging big data framework that improves the current performance of the Hadoop framework. Finally, some implementations and tools for big data domain related to distributed data file systems, data analytics, and machine learning techniques are presented.

2.1. MapReduce and the big data processing problem

MapReduce [25,26] is presented as one of the most efficient big data solutions. This programming paradigm and its related algorithms [27], were developed to provide significant improvements in large-scale data-intensive applications in clusters [28]. The programming model implemented by MapReduce is based on the definition of two basic elements: **mappers** and **reducers**. The idea behind this programming model is to design map functions (or mappers) that are used to generate a set of intermediate key/value pairs, after which the reduce functions will merge (reduce can be used as a shuffling or combining function) all of the intermediate values that are associated with the same intermediate key. The key aspect of the MapReduce algorithm is that if every map and reduce is independent of all other ongoing maps and reduces, then the operations can be run in parallel on different keys and lists of data.

Although three functions, Map(), Combining()/Shuffling(), and Reduce(), are the basic processes in any MapReduce approach, usually they are decomposed as follows:

1. Prepare the **input**: The MapReduce system designates map processors (or *worker nodes*), assigns the input key value **K1** that each processor would work on, and provides each processor with all of the input data associated with that key value.
2. The **Map()** step: Each worker node applies the Map() function to the local data and writes the output to a temporary storage space. The Map() code is run exactly once for each **K1** key value, generating output that is organised by key values **K2**. A master node arranges it so that for redundant copies of input data only one is processed.
3. The **Shuffle()** step: The map output is sent to the reduce processors, which assign the **K2** key value that each processor should work on, and provide that processor with all of the map-generated data associated with that key value, such that all data belonging to one key are located on the same worker node.
4. The **Reduce()** step: Worker nodes process each group of output data (per key) in parallel, executing the user-provided Reduce() code; each function is run exactly once for each **K2** key value produced by the map step.
5. Produce the **final output**: The MapReduce system collects all of the reduce outputs and sorts them by **K2** to produce the final outcome.

Fig. 2 shows the classical "word count problem" using the MapReduce paradigm. As Fig. 2 shown, initially a process will split the data into a subset of chunks that will later be processed by the *mappers*. Once the key/values are generated by mappers, a shuffling process is used to mix (combine) these key values (combining the same keys in the same worker node). Finally, the *reduce* functions are used to count the words that generate a common output as a result of the algorithm. As a result of the execution or wrappers/reducers, the output will generate a sorted list of word counts from the original text input.

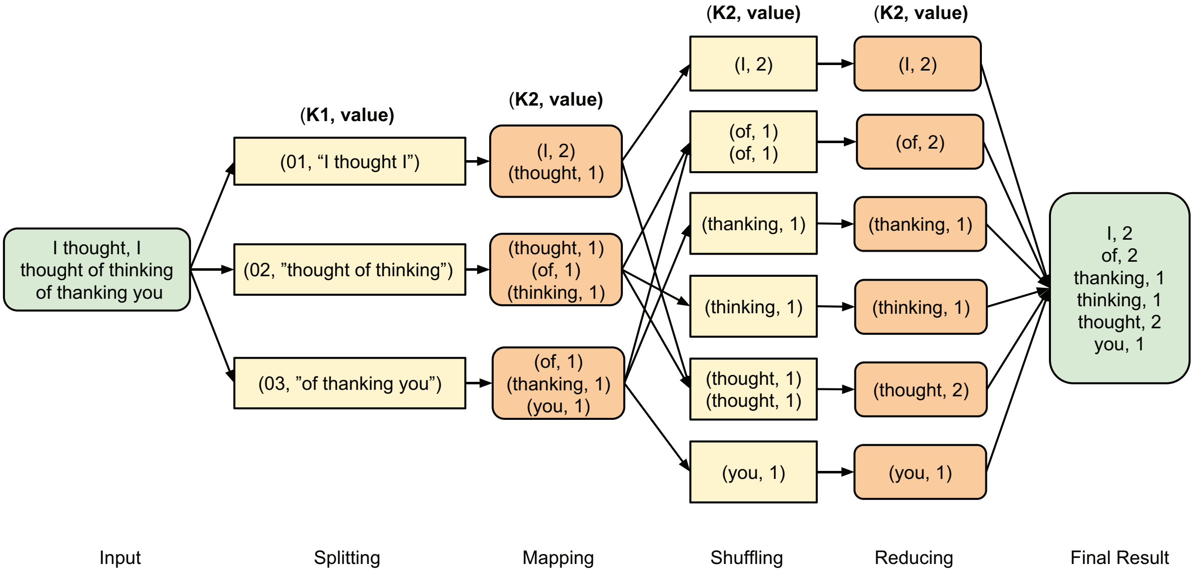

Fig. 2. The MapReduce processes for counting words in a text.

Please cite this article as: G. Bello-Orgaz et al., Social big data: Recent achievements and new challenges, Information Fusion (2015), http://dx.doi.org/10.1016/j.inffus.2015.08.005

Finally, and before the application of this paradigm, it is essential to understand if the algorithms can be translated to mappers and reducers or if the problem can be analysed using traditional strategies. MapReduce provides an excellent technique to work with large sets of data when the algorithm can work on small pieces of that dataset in parallel, but if the algorithm cannot be mapped into this methodology, it may be "*trying to use a sledgehammer to crack a nut*".

2.2. Apache Hadoop

Any MapReduce system (or framework) is based on a MapReduce engine that allows for implementing the algorithms and distributing the parallel processes. Apache Hadoop [19] is an open-source software framework written in Java for the distributed storage and distributed processing of very large datasets using the MapReduce paradigm. All of the modules in Hadoop have been designed taking into account the assumption that hardware failures (of individual machines or of racks of machines) are commonplace and thus should be automatically managed in the software by the framework. The core of Apache Hadoop comprises a storage area, the Hadoop Distributed File System (HDFS), and a processing area (MapReduce).

The HDFS (see Section 2.4.1) spreads multiple copies of the data across different machines. This not only offers reliability without the need for RAID-controlled disks but also allows for multiple locations to run the mapping. If a machine with one copy of the data is busy or offline, another machine can be used. A job scheduler (in Hadoop, the *JobTracker*) keeps track of which MapReduce jobs are executing; schedules individual maps; reduces intermediate merging operations to specific machines; monitors the successes and failures of these individual tasks; and works to complete the entire batch job. The HDFS and the job scheduler can be accessed by the processes and programs that need to read and write data and to submit and monitor the MapReduce jobs. However, Hadoop presents a number of limitations:

1. For maximum parallelism, you need the maps and reduces to be stateless, to not depend on any data generated in the same MapReduce job. You *cannot control the order* in which the maps run or the reductions.
2. Hadoop is very inefficient (in both CPU time and power consumed) if you are *repeating similar searches* repeatedly. A database with an index will always be faster than running a MapReduce job over un-indexed data. However, if that index needs to be regenerated whenever data are added, and data are being added continually, MapReduce jobs may have an edge.
3. In the Hadoop implementation, *reduce operations do not take place until all of the maps have been completed* (or have failed and been skipped). As a result, you do not receive any data back until the entire mapping has finished.
4. There is a general assumption that the output of the reduce is smaller than the input to the map. That is, you are taking a large data source and generating smaller final values.

2.3. Apache Spark

Apache Spark [20] is an open-source cluster computing framework that was originally developed in the AMPLab at University of California, Berkeley. Spark had over 570 contributors in June 2015, making it a very high-activity project in the Apache Software Foundation and one of the most active big data open source projects. It provides high-level APIs in Java, Scala, Python, and R and an optimised engine that supports general execution graphs. It also supports a rich set of high-level tools including Spark SQL for SQL and structured data processing, Spark MLlib for machine learning, GraphX for graph processing, and Spark Streaming.

The Spark framework allows for reusing a working set of data across multiple parallel operations. This includes many iterative machine learning algorithms as well as interactive data analysis tools. Therefore, this framework supports these applications while retaining the scalability and fault tolerance of MapReduce. To achieve these goals, Spark introduces an abstraction called resilient distributed datasets (RDDs). An RDD is a read-only collection of objects partitioned across a set of machines that can be rebuilt if a partition is lost. In contrast to Hadoops two-stage disk-based MapReduce paradigm (mappers/reducers), Sparks in-memory primitives provide performance up to 100 times faster for certain applications by allowing user programs to load data into a clusters memory and to query it repeatedly. One of the multiple interesting features of Spark is that this framework is particularly well suited to machine learning algorithms [29].

From a distributed computing perspective, Spark requires a cluster manager and a distributed storage system. For cluster management, Spark supports stand-alone (native Spark cluster), Hadoop YARN, and Apache Mesos. For distributed storage, Spark can interface with a wide variety, including the Hadoop Distributed File System, Apache Cassandra, OpenStack Swift, and Amazon S3. Spark also supports a pseudo-distributed local mode that is usually used only for development or testing purposes, when distributed storage is not required and the local file system can be used instead; in this scenario, Spark is running on a single machine with one executor per CPU core.

2.4. Other MapReduce implementations and software

A list related to big data implementations and MapReduce-based applications was generated by Mostosi [30]. Although the author finds that *"It is [the list] still incomplete and always will be"*, his "BigData Ecosystem Table" [31] contains more than 600 references related to different big data technologies, frameworks, and applications and, to the best of this authors knowledge, is one of the best (and more exhaustive) lists related to available big data technologies. This list comprises 33 different topics related to big data, and a selection of those technologies and applications were chosen. Those topics are related to: *distributed programming, distributed files systems*, a *document data model*, a key-value data model, a *graph data model, machine learning, applications, business intelligence*, and *data analysis*. This selection attempts to reflect some of the recent popular frameworks and software implementations that are commonly used to develop efficient MapReduce-based systems and applications.

2.4.1. Distributed programming & distributed filesystems

- **Apache Pig**. Pig provides an engine for executing data flows in parallel on Hadoop. It includes a language, Pig Latin, for expressing these data flows. Pig Latin includes operators for many of the traditional data operations (join, sort, filter, etc.), as well as the ability for users to develop their own functions for reading, processing, and writing data.
- **Apache Storm**. Storm is a complex event processor and distributed computation framework written basically in the Clojure programming language [32]. It is a distributed real-time computation system for rapidly processing large streams of data. Storm is an architecture based on a master-workers paradigm, so that a Storm cluster mainly consists of master and worker nodes, with coordination done by Zookeeper [33].
- **Stratosphere** [34]. Stratosphere is a general-purpose cluster computing framework. It is compatible with the Hadoop ecosystem:, accessing data stored in the HDFS and running with Hadoops new cluster manager YARN. The common input formats of Hadoop are supported as well. Stratosphere does not use Hadoops MapReduce implementation; it is a completely new system that brings its own runtime. The new runtime allows for defining more advanced operations that include more transformations than only map and

reduce. Additionally, Stratosphere allows for expressing analysis jobs using advanced data flow graphs, which are able to resemble common data analysis task more naturally.

- **Apache HDFS**. The most extended and popular distributed file system for MapReduce frameworks and applications is the Hadoop Distributed File System. The HDFS offers a way to store large files across multiple machines. Hadoop and HDFS were derived from the Google File System (GFS) [35].

2.4.2. Document data model & graph data model

- **Apache Cassandra**. Cassandra is a recent open source fork of a **stand-alone distributed non-SQL DBMS** system that was initially coded by Facebook, derived from what was known of the original Google BigTable [36] and Google File System designs [35]. Cassandra uses a system inspired by Amazons Dynamo for storing data, and MapReduce can retrieve data from Cassandra. Cassandra can run without the HDFS or on top of it (the DataStax fork of Cassandra).
- **Apache Giraph**. Giraph is an iterative **graph processing system** built for high scalability. It is currently used at Facebook to analyse the social graph formed by users and their connections. Giraph was originated as the open-source counterpart to Pregel [37], the graph processing framework developed at Google (see Section 3.1 for a further description).
- **MongoDB**. MongoDB is an open-source **document-oriented database system** and is part of the NoSQL family of database systems [38]. It provides high performance, high availability, and automatic scaling. Instead of storing data in tables as is done in a classical relational database, MongoDB stores structured data as JSON-like documents, which are data structures composed of fields and value pairs. Its index system supports faster queries and can include keys from embedded documents and arrays. Moreover, this database allows users to distribute data across a cluster of machines.

2.4.3. Machine learning

- **Apache Mahout** [21]. The Mahout(TM) Machine Learning (ML) library is an Apache(TM) project whose main goal is to build scalable libraries that contain the implementation of a number of the conventional ML algorithms (dimensionality reduction, classification, clustering, and topic models, among others). In addition, this library includes implementations for a set of recommender systems (user-based and item-based strategies). The first versions of Mahout implemented the algorithms built on the Hadoop framework, but recent versions include many new implementations built on the Mahout-Samsara environment, which runs on Spark and H2O. The new Spark-item similarity implementations enable the next generation of co-occurrence recommenders that can use entire user click streams and contexts in making recommendations.
- **Spark MLlib** [22]. MLlib is Sparks scalable machine learning library, which consists of common learning algorithms and utilities, including classification, regression, clustering, collaborative filtering, and dimensionality reduction, as well as underlying optimization primitives. It supports writing applications in Java, Scala, or Python and can run on any Hadoop2/YARN cluster with no preinstallation. The first version of MLlib was developed at UC Berkeley by 11 contributors, and it provided a limited set of standard machine learning methods. However, MLlib is currently experiencing dramatic growth, and it has over 140 contributors from over 50 organisations.
- **MLBase** [23]. The MLbase platform consists of three layers: *ML Optimizer*, *MLlib*, and *MLI*. *ML Optimizer* (currently under development) aims to automate the task of ML pipeline construction. The optimizer solves a search problem over the feature extractors and ML algorithms included in MLI and MLlib. *MLI* [24] is an experimental API for feature extraction and algorithm development that introduces high-level ML programming abstractions. A prototype of MLI has been implemented against Spark and serves as a test bed for MLlib. Finally, *MLlib* is Apache Sparks distributed ML library. MLlib was initially developed as part of the MLbase project, and the library is currently supported by the Spark community.

2.4.4. Applications & business intelligence & data analysis

- **Apache Nutch**. Nutch is a highly extensible and scalable open source web crawler software project, specifically, a search engine based on Lucene (a Web crawler is an Internet bot that systematically browses the World Wide Web, usually for Web indexing). It can process various document types (plain text, XML, OpenDocument, Word, Excel, Powerpoint, PDF, RTF, MP3) that are all parsed by the Tika plugin. Currently, the project has two versions. Nutch 1.x is relying on Apache Hadoop data structures, which are excellent for batch processing. Nutch 2.x differs in the data storage, which is performed using Apache Gora to manage persistent object mappings. This allows for incorporating a flexible model/stack to store everything (fetch time, status, content, parsed text, outlinks, inlinks, etc.) into a number of NoSQL storage solutions.
- **Apache Zeppelin**. Zeppelin is a Web-based notebook that enables interactive data analytics; it is an open source data analysis environment that runs on top of Apache Spark. Current languages included in the Zeppelin interpreter are Scala, Python, SparkSQL, Hive, Markdown, and Shell. Zeppelin can dynamically create some input forms in your notebook and provides some basic charts to show the results, and the notebook URL can be shared among collaborators.
- **Pentaho**. Pentaho is an open source data integration (Kettle) tool that delivers powerful extraction, transformation, and loading capabilities using a groundbreaking, metadata-driven approach. It also provides analytics, reporting, visualisation, and a predictive analytics framework that is directly designed to work with Hadoop nodes. It provides data integration and analytic platforms based on Hadoop in which datasets can be streamed, blended, and then automatically published into one of the popular analytic databases.
- **SparkR**. There is an important number of R-based applications for MapReduce and other big data applications. R [39] is a popular and extremely powerful programming language for statistics and data analysis. SparkR provides an R frontend for Spark. It allows users to interactively run jobs from the R shell on a cluster, automatically serializes the necessary variables to execute a function on the cluster, and also allows for easy use of existing R packages.

3. Social data analytic methods and algorithms

Social big data analytic can be seen as the set of algorithms and methods used to extract relevant knowledge from social media data sources that could provide heterogeneous contents, with very large size, and constantly changing (stream or online data). This is inherently interdisciplinary and spans areas such as data mining, machine learning, statistics, graph mining, information retrieval, and natural language among others. This section provides a description of the basic methods and algorithms related to network analytics, community detection, text analysis, information diffusion, and information fusion, which are the areas currently used to analyse and process information from social-based sources.

3.1. Network analytics

Today, society lives in a connected world in which communication networks are intertwined with daily life. For example, social networks are one of the most important sources of social big data;

Please cite this article as: G. Bello-Orgaz et al., Social big data: Recent achievements and new challenges, Information Fusion (2015), http://dx.doi.org/10.1016/j.inffus.2015.08.005

specifically, Twitter generates over 400 million tweets every day [40]. In social networks, individuals interact with one another and provide information on their preferences and relationships, and these networks have become important tools for collective intelligence extraction. These connected networks can be represented using graphs, and network analytic methods [41] can be applied to them for extracting useful knowledge.

Graphs are structures formed by a set of vertices (also called nodes) and a set of edges, which are connections between pairs of vertices. The information extracted from a social network can be easily represented as a graph in which the vertices or nodes represent the users and the edges represent the relationships among them (e.g., a re-tweet of a message or a favourite mark in Twitter). A number of network metrics can be used to perform social analysis of these networks. Usually, the importance, or influence, in a social network is analysed through *centrality measures*. These measures have high computational complexity in large-scale networks. To solve this problem, focusing on a large-scale graph analysis, a second generation of frameworks based on the MapReduce paradigm has appeared, including **Hama, Giraph** (based on Pregel), and **GraphLab** among others [42].

Pregel [37] is a graph-parallel system based on the Bulk Synchronous Parallel model (BSP) [43]. A BSP abstract computer can be interpreted as a set of processors that can follow different threads of computation in which each processor is equipped with fast local memory and interconnected by a communication network. According to this, the platform based on this model comprises three major components:

- Components capable of processing and/or local memory transactions (i.e., processors).
- A network that routes messages between pairs of these components.
- A hardware facility that allows for the synchronisation of all or a subset of components.

Taking into account this model, a BSP algorithm is a sequence of global supersteps that consists of three components:

1. *Concurrent computation*: Every participating processor may perform local asynchronous computations.
2. *Communication*: The processes exchange data from one processor to another, facilitating remote data storage capabilities.
3. *Barrier synchronisation*: When a process reaches this point (the barrier), it waits until all other processes have reached the same barrier.

Hama [44] and Giraph are two distributed graph processing frameworks on Hadoop that implement Pregel. The main difference between the two frameworks is the matrix computation using the MapReduce paradigm. Apache Giraph is an iterative graph processing system in which the input is a graph composed of vertices and directed edges. Computation proceeds as a sequence of iterations (supersteps). Initially, every vertex is active, and for each superstep, every active vertex invokes the “Compute Method” that will implement the graph algorithm that will be executed. This means that the algorithms implemented using Giraph are vertex oriented. Apache Hama does not only allow users to work with Pregel-like graph applications. This computing engine can also be used to perform compute-intensive general scientific applications and machine learning algorithms. Moreover, it currently supports YARN, which is the resource management technology that lets multiple computing frameworks run on the same Hadoop cluster using the same underlying storage. Therefore, the same data could be analysed using MapReduce or Spark.

In contrast, GraphLab is based on a different concept. Whereas Pregel is a one-vertex-centric model, this framework uses vertex-to-node mapping in which each vertex can access the state of adjacent vertices. In Pregel, the interval between two supersteps is defined by the run time of the vertex with the largest neighbourhood. The GraphLab approach improves this splitting of vertices with large neighbourhoods across different machines and synchronises them.

Finally, Elser and Montresor [42] present a study of these data frameworks and their application to graph algorithms. The k-core decomposition algorithm is adapted to each framework. The goal of this algorithm is to compute the centrality of each node in a given graph. The results obtained confirm the improvement achieved in terms of execution time for these frameworks based on Hadoop. However, from a programming paradigm point of view, the authors recommend Pregel-inspired frameworks (a vertex-centric framework), which is the better fit for graph-related problems.

3.2. Community detection algorithms

The community detection problem in complex networks has been the subject of many studies in the field of data mining and social network analysis. The goal of the community detection problem is similar to the idea of graph partitioning in graph theory [45,46]. A cluster in a graph can be easily mapped into a community. Despite the ambiguity of the community definition, numerous techniques have been used for detecting communities. Random walks, spectral clustering, modularity maximization, and statistical mechanics have all been applied to detecting communities [46]. These algorithms are typically based on the topology information from the graph or network. Related to graph connectivity, each cluster should be connected; that is, there should be multiple paths that connect each pair of vertices within the cluster. It is generally accepted that a subset of vertices forms a good cluster if the induced sub-graph is dense and there are few connections from the included vertices to the rest of the graph [47]. Considering both connectivity and density, a possible definition of a graph cluster could be a connected component or a maximal clique [48]. This is a sub-graph into which no vertex can be added without losing the clique property.

One of the most well-known algorithms for community detection was proposed by Girvan and Newman [49]. This method uses a new similarity measure called “edge betweenness” based on the number of the shortest paths between all vertex pairs. The proposed algorithm is based on identifying the edges that lie between communities and their successive removal, achieving the isolation of the communities. The main disadvantage of this algorithm is its high computational complexity with very large networks.

Modularity is the most used and best known quality measure for graph clustering techniques, but its computation is an NP-complete problem. However, there are currently a number of algorithms based on good approximations of modularity that are able to detect communities in a reasonable time. The first greedy technique to maximize modularity was a method proposed by Newman [50]. This was an agglomerative hierarchical clustering algorithm in which groups of vertices were successively joined to form larger communities such that modularity increased after the merging. The update of the matrix in the Newman algorithm involved a large number of useless operations owing to the sparseness of the adjacency matrix. However, the algorithm was improved by Clauset et al. [51], who used the matrix of modularity variations to arrange for the algorithm to perform more efficiently.

Despite the improvements to and modifications of the accuracy of these greedy algorithms, they have poor performance when they are compared against other techniques. For this reason, Newman reformulated the modularity measure in terms of eigenvectors by replacing the Laplacian matrix with the modularity matrix [52], called the spectral optimization of modularity. This improvement must also be applied in order to improve the results of other optimization techniques [53,54].

Random walks can also be useful for finding communities. If a graph has a strong community structure, a random walker spends a long time inside a community because of the high density of internal edges and the consequent number of paths that could be followed. Zhou and Lipowsky [55], based on the fact that walkers move preferentially towards vertices that share a large number of neighbours, defined a proximity index that indicates how close a pair of vertices is to all other vertices. Communities are detected with a procedure called NetWalk, which is an agglomerative hierarchical clustering method by which the similarity between vertices is expressed by their proximity.

A number of these techniques are focused on finding disjointed communities. The network is partitioned into dense regions in which nodes have more connections to each other than to the rest of the network, but it is interesting that in some domains, a vertex could belong to several clusters. For instance, it is well-known that people in a social network for natural memberships in multiple communities. Therefore, the overlap is a significant feature of many real-world networks. To solve this problem, fuzzy clustering algorithms applied to graphs [56] and overlapping approaches [57] have been proposed.

Xie et al. [58] reviewed the state of the art in overlapping community detection algorithms. This work noticed that for low overlapping density networks, SLPA, OSLOM, Game, and COPRA offer better performance. For networks with high overlapping density and high overlapping diversity, both SLPA and Game provide relatively stable performance. However, test results also suggested that the detection in such networks is still not yet fully resolved . A common feature that is observed by various algorithms in real-world networks is the relatively small fraction of overlapping nodes (typically less than 30%), each of which belongs to only 2 or 3 communities.

3.3. Text analytics

A significant portion of the unstructured content collected from social media is text. Text mining techniques can be applied for automatic organization, navigation, retrieval, and summary of huge volumes of text documents [59–61]. This concept covers a number of topics and algorithms for text analysis including natural language processing (NLP), information retrieval, data mining, and machine learning [62].

Information extraction techniques attempt to extract entities and their relationships from texts, allowing for the inference of new meaningful knowledge. These kinds of techniques are the starting point for a number of text mining algorithms. A usual model for representing the content of documents or text is the vector space model. In this model, each document is represented by a vector of frequencies of remaining terms within the document [60]. The term frequency (TF) is a function that relates the number of occurrences of the particular word in the document divided by the number of words in the entire document. Another function that is currently used is the inverse document frequency (IDF); typically, documents are represented as TF-IDF feature vectors. Using this data representation, a document represents a data point in n-dimensional space where n is the size of the corpus vocabulary.

Text data tend to be sparse and high dimensional. A text document corpus can be represented as a large sparse TF-IDF matrix, and applying dimensionality reduction methods to represent the data in compressed format [63] can be very useful. Latent semantic indexing [64] is an automatic indexing method that projects both documents and terms into a low-dimensional space that attempts to represent the semantic concepts in the document. This method is based on the singular value decomposition of the term-document matrix, which constructs a low-ranking approximation of the original matrix while preserving the similarity between the documents. Another family of dimension reduction techniques is based on probabilistic topic models such as latent Dirichlet allocation (LDA) [65]. This technique provides the mechanism for identifying patterns of term co-occurrence and using those patterns to identify coherent topics. Standard LDA implementations of the algorithm read the documents of the training corpus numerous times and in a serial way. However, new, efficient, parallel implementations of this algorithm have appeared [66] in attempts to improve its efficiency.

Unsupervised machine learning methods can be applied to any text data without the need for a previous manual process. Specifically, clustering techniques are widely studied in this domain to find hidden information or patterns in text datasets. These techniques can automatically organise a document corpus into clusters or similar groups based on a blind search in an unlabelled data collection, grouping the data with similar properties into clusters without human supervision. Generally, document clustering methods can be mainly categorized into two types [67]: *partitioning* algorithms that divide a document corpus into a given number of disjoint clusters that are optimal in terms of some predefined criteria functions [68] and *hierarchical* algorithms that group the data points into a hierarchical tree structure or a dendrogram [69]. Both types of clustering algorithms have strengths and weaknesses depending on the structure and characteristics of the dataset used. In Zhao and Karypis [70], a comparative assessment of different clustering algorithms (partitioning and hierarchical) was performed using different similarity measures on high-dimensional text data. The study showed that partitioning algorithms perform better and can also be used to produce hierarchies of higher quality than those returned by the hierarchical ones.

In contrast, the classification problem is one of the main topics in the supervised machine learning literature. Nearly all of the well-known techniques for classification, such as decision trees, association rules, Bayes methods, nearest neighbour classifiers, SVM classifiers, and neural networks, have been extended for automated text categorisation [71]. *Sentiment classification* has been studied extensively in the area of opinion mining research, and this problem can be formulated as a classification problem with three classes, positive, negative and neutral. Therefore, most of the existing techniques designed for this purpose are based on classifiers [72].

However, the emergence of social networks has created massive and continuous streams of text data. Therefore, new challenges have been arising in adapting the classic machine learning methods, because of the need to process these data in the context of a one-pass constraint [73]. This means that it is necessary to perform data mining tasks online and only one time as the data come in. For example, the online spherical k-means algorithm [74] is a segment-wise approach that was proposed for streaming text clustering. This technique splits the incoming text stream into small segments that can be processed effectively in memory. Then, a set of k-means iterations is applied to each segment in order to cluster them. Moreover, in order to consider less important old documents during the clustering process, a decay factor is included.

3.4. Information diffusion models and methods

One of the most important roles of social media is to spread information to social links. With the large amount of data and the complex structures of social networks, it has been even more difficult to understand how (and why) information is spread by social reactions (e.g., *retweeting* in Twitter and *like* in Facebook). It can be applied to various applications, e.g., viral marketing, popular topic detection, and virus prevention [75].

As a result, many studies have been proposed for modelling the information diffusion patterns on social networks. The characteristics of the diffusion models are (*i*) the topological structure of the network (a sub-graph composed of a set of users to whom the information has been spread) and (*ii*) temporal dynamics (the evolution of the number of users whom the information has reached over time) [76].

According to the analytics, these diffusion models can be categorized into explanatory and predictive approaches [77].

- *Explanatory models*: The aim of these models is to discover the hidden spreading cascades once the activation sequences are collected. These models can build a path that can help users to easily understand how the information has been diffused. The NETINT method [78] has applied sub-modular, function-based iterative optimisation to discover the spreading cascade (path) that maximises the likelihood of the collected dataset. In particular, for working with missing data, a k-tree model [79] has been proposed to estimate the complete activation sequences.
- *Predictive models*: These are based on learning processes with the observed diffusion patterns. Depending on the previous diffusion patterns, there are two main categories of predictive models: (*i*) structure-based models (graph-based approaches) and (*ii*) content-analysis-based models (non-graph-based approaches).

Moreover, there are more existing approaches to understanding information diffusion patterns. The projected greedy approach for non-sub-modular problems [80] was recently proposed to populate the useful seeds in social networks. This approach can identify the partial optimisation for understanding the information diffusion. Additionally, an evolutionary dynamics model was presented in [81, 82] that attempted to understand the temporal dynamics of information diffusion over time.

One of the relevant topics for analysing information diffusion patterns and models is the concept of time and how it can be represented and managed. One of the popular approaches is based on time series. Any time series can be defined as a chronological collection of observations or events. The main characteristics of this type of data are large size, high dimensionality, and continuous change. In the context of data mining, the main problem is how to represent the data. An effective mechanism for compressing the vast amount of time series data is needed in the context of information diffusion. Based on this representation, different data mining techniques can be applied such as pattern discovery, classification, rule discovery, and summarisation [83]. In Lin et al. [84], a new symbolic representation of time series is proposed that allows for a dimensionality/numerosity reduction. This representation is tested using different classic data mining tasks such as clustering, classification, query by content, and anomaly detection.

Based on the mathematical models mentioned above, we need to compare a number of various applications that can support users in many different domains. One of the most promising applications is detecting meaningful social events and popular topics in society. Such meaningful events and topics can be discovered by well-known text processing schemes (e.g., *TF-IDF*) and simple statistical approaches (e.g., LDA, Gibbs sampling, and the TSTE method [85]). In particular, not only the time domain but also the frequency domain have been exploited to identify the most frequent events [86].

3.5. Information fusion for social big data

The social big data from various sources needs to be fused for providing users with better services. These fusion can be done in different ways and affect to different technologies, methods and even research areas. Two of these possible areas are Ontologies and Social Networks, next how previous areas could benefit from information fusion in social big data are briefly described:

- **Ontology-based fusion**. Semantic heterogeneity is an important issue on information fusion. Social networks have inherently different semantics from other types of network. Such semantic heterogeneity includes not only linguistic differences (e.g., between ‘reference’ and ‘bibliography’) but also mismatching between conceptual structures. To deal with these problems, in [87] ontologies are exploited from multiple social networks, and more importantly, semantic correspondences obtained by ontology matching methods.

  More practically, semantic meshup applications have been illustrated. To remedy the data integration issues of the traditional web mashups, the semantic technologies uses the linked open data (LOD) based on RDF data model, as the unified data model for combining, aggregating, and transforming data from heterogeneous data resources to build linked data mashups [88].
- **Social network integration**. Next issue is how to integrate the distributed social networks. As many kinds of social networking services have been developed, users are joining multiple services for social interactions with other users and collecting a large amount of information (e.g., statuses on Facebook and tweets on Twitter). An interesting framework has been proposed for a social identity matching (SIM) method across these multiple SNS [89]. It means that the proposed approach can protect user privacy, because only the public information (e.g., username and the social relationships of the users) is employed to find the best matches between social identities. Particularly, cloud-based platform has been applied to build software infrastructure where the social network information can be shared and exchanged [90].

4. Social-based applications

The social big data analysis can be applied to social media data sources for discovering relevant knowledge that can be used to improve the decision making of individual users and companies [18]. In this context, business intelligence can be defined as those techniques, systems, methodologies, and applications that analyse critical business data to help an enterprise better understand its business and market and to support business decisions [91]. This field includes methodologies that can be applied to different areas such as e-commerce, marketing, security, and healthcare [18]; more recent methodologies have been applied to treat social big data. This section provides short descriptions of some applications of these methodologies in domains that intensively use social big data sources for business intelligence.

4.1. Marketing

Marketing researchers believe that big social media analytics and cloud computing offer a unique opportunity for businesses to obtain opinions from a vast number of customers, improving traditional strategies. A significant market transformation has been accomplished by leading e-commerce enterprises such Amazon and eBay through their innovative and highly scalable e-commerce platforms and recommender systems.

Social network analysis extracts user intelligence and can provide firms with the opportunity for generating more targeted advertising and marketing campaigns. Maurer and Wiegmann [92] show an analysis of advertising effectiveness on social networks. In particular, they carried out a case study using Facebook to determine users perceptions regarding Facebook ads. The authors found that most of the participants perceived the ads on Facebook as annoying or not helpful for their purchase decisions. However, Trattner and Kappe [93] show how ads placed on users social streams that have been generated by the Facebook tools and applications can increase the number of visitors and the profit and ROI of a Web-based platform. In addition, the authors present an analysis of real-time measures to detect the most valuable users on Facebook.

A study of microblogging (Twitter) utilization as an eWOM (electronic word-of-mouth) advertising mechanism is carried out in Jansen et al. [94]. This work analyses the range, frequency, timing, and

Table 1
Basic features related to social big data applications in marketing area.

Authors	Ref. num.	Summary	Methods
Trattner and Kappe	[93]	Targeted advertising on Facebook	Real-time measures to detect the most valuable users
Jansen et al.	[94]	Twitter as eWOM advertising mechanism	Sentiment analysis
Asur et al.	[95]	Using Twitter to forecast box-office revenues for movies	Topics detection, sentiment analysis
Ma et al.	[96]	Viral marketing in social networks	Social network analysis, information diffusion models

content of tweets in various corporate accounts. The results obtained show that 19% of microblogs mention a brand. Of the branding microblogs, nearly 20% contained some expression of brand sentiments. Therefore, the authors conclude that microblogging reports what customers really feel about the brand and its competitors in real time, and it is a potential advantage to explore it as part of companies overall marketing strategies. Customers brand perceptions and purchasing decisions are increasingly influenced by social media services, and these offer new opportunities to build brand relationships with potential customers. Another approach that uses Twitter data is presented in Asur et al. [95] to forecast box-office revenues for movies. The authors show how a simple model built from the rate at which tweets are created about particular topics can outperform market-based predictors. Moreover, the sentiment extraction from Twitter is used to improve the forecasting power of social media.

Because of the exponential growth use of social networks, researchers are actively attempting to model the dynamics of viral marketing based on the information diffusion process. Ma et al. [96] proposed modelling social network marketing using heat diffusion processes. Heat diffusion is a physical phenomenon related to heat, which always flows from a position with higher temperature to a position with lower temperature. The authors present three diffusion models along with three algorithms for selecting the best individuals to receive marketing samples. These models can diffuse both positive and negative comments on products or brands in order to simulate the real opinions within social networks. Moreover, the authors complexity analysis shows that the model is also scalable to large social networks. Table 1 shows a brief summary of the previously described applications, including the basic functionalities for each and their basic methods.

4.2. Crime analysis

Criminals tend to have repetitive pattern behaviours, and these behaviours are dependent upon situational factors. That is, crime will be concentrated in environments with features that facilitate criminal activities [97]. The purpose of crime data analysis is to identify these crime patterns, allowing for detecting and discovering crimes and their relationships with criminals. The knowledge extracted from applying data mining techniques can be very useful in supporting law enforcement agencies.

Communication between citizens and government agencies is mostly through telephones, face-to-face meetings, email, and other digital forms. Most of these communications are saved or transformed into written text and then archived in a digital format, which has led to opportunities for automatic text analysis using NLP techniques to improve the effectiveness of law enforcement [98]. A decision support system that combines the use of NLP techniques, similarity measures, and classification approaches is proposed by Ku and Leroy [99] to automate and facilitate crime analysis. Filtering reports and identifying those that are related to the same or similar crimes can provide useful information to analyse crime trends, which allows for apprehending suspects and improving crime prevention.

Traditional crime data analysis techniques are typically designed to handle one particular type of dataset and often overlook geospatial distribution. Geographic knowledge discovery can be used to discover patterns of criminal behaviour that may help in detecting where, when, and why particular crimes are likely to occur. Based on this concept, Phillips and lee [100] present a crime data analysis technique that allows for discovering co-distribution patterns between large, aggregated and heterogeneous datasets. In this approach, aggregated datasets are modelled as graphs that store the geospatial distribution of crime within given regions, and then these graphs are used to discover datasets that show similar geospatial distribution characteristics. The experimental results obtained in this work show that it is possible to discover geospatial co-distribution relationships among crime incidents and socio-economic, socio-demographic and spatial features.

Another analytical technique that is now in high use by law enforcement agencies to visually identify where crime tends to be highest is the *hotspot mapping*. This technique is used to predict where crime may happen, using data from the past to inform future actions. Each crime event is represented as a point, allowing for the geographic distribution analysis of these points. A number of mapping techniques can be used to identify crime hotspots, such as: point mapping, thematic mapping of geographic areas, spatial ellipses, grid thematic mapping, and kernel density estimation (KDE), among others. Chainey et al. [101] conducted a comparative assessment of these techniques, and the results obtained showed that KDE was the technique that consistently outperformed the others. Moreover, the authors offered a benchmark to compare with the results of other techniques and other crime types, including comparisons between advanced spatial analysis techniques and prediction mapping methods. Another novel approach using spatio-temporally tagged tweets for crime prediction is presented by Gerber [102]. This work shows the use of Twitter, applying a linguistic analysis and statistical topic modelling to automatically identify discussion topics across a city in the United States. The experimental results showed that adding Twitter data improved crime prediction performance versus a standard approach based on KDE.

Finally, the use of data mining in fraud detection is very popular, and there are numerous studies on this area. ATM phone scams are one well-known type of fraud. Kirkos et al. [103] analysed the effectiveness of data mining classification techniques (decision trees, neural networks and Bayesian belief networks) for identifying fraudulent financial statements, and the experimental results concluded that Bayesian belief networks provided higher accuracy for fraud classification. Another approach to detecting fraud in real-time credit card transactions was presented by Quah and Sriganesh [104]. The system these authors proposed uses a self-organization map to filter and analyse customer behaviour to detect fraud. The main idea is to detect the patterns of the legal cardholder and of the fraudulent transactions through neural network learning and then to develop rules for these two different behaviours. One typical fraud in this area is the ATM phone scams that attempts to transfer a victims money into fraudulent accounts. In order to identify the signs of fraudulent accounts and the patterns of fraudulent transactions, Li et al. [105] applied Bayesian classification and association rules. Detection rules are developed based on the identified signs and applied to the design of a fraudulent account detection system. Table 2 shows a brief summary of all of the applications that were previously mentioned, providing a description of the basic functionalities of each and their main methods.

Table 2
Basic features related to social big data applications in crime analysis area.

Authors	Ref. num.	Summary	Methods
Phillips and Lee	[100]	Decision support system (DSS) to analyse crime trends allowing to catch suspects	NLP, Similarity measures, classification
Ku and Leroy	[99]	Technique to discover geospatial co-distribution relations among crime incidents	Network analysis
Chainey et al.	[101]	Comparative assessment of mapping techniques to predict where crimes may happen	Spatial analysis, mapping methods
Gerber	[102]	Identify discussion topics across a city in the United States to predict crimes	Linguistic analysis, statistical topic modelling
Kirkos et al.	[103]	Identification of fraudulent financial statements	Classification (decision trees, neural networks and Bayesian belief networks)
Quah and Sriganesh	[104]	Detect fraud detection in real-time credit card transactions	Neural network learning, association rules
Li et al.	[105]	Identify the signs of fraudulent accounts and the patterns of fraudulent transactions	Bayesian classification, association rules

4.3. Epidemic intelligence

Epidemic intelligence can be defined as the early identification, assessment, and verification of potential public health risks [106] and the timely dissemination of the appropriate alerts. This discipline includes surveillance techniques for the automated and continuous analysis of unstructured free text or media information available on the Web from social networks, blogs, digital news media, and official sources.

Text mining techniques have been applied to biomedical text corpora for named entity recognition, text classification, terminology extraction, and relationship extraction [107]. These methods are human language processing algorithms that aim to convert unstructured textual data from large-scale collections to a specific format, filtering them according to need. They can be used to detect words related to diseases or their symptoms in published texts [108]. However, this can be difficult because the same word can refer to different things depending upon context. Furthermore, a specific disease can have multiple associated names and symptoms, which increases the complexity of the problem. Ontologies can help to automate human understanding of key concepts and the relationships between them, and they allow for achieving a certain level of filtering accuracy. In the health domain, it is necessary to identify and link term classes such as diseases, symptoms, and species in order to detect the potential focus of disease outbreaks. Currently, there are a number of available biomedical ontologies that contain all of the necessary terms. For example, the BioCaster ontology [109] is based on the OWL Semantic Web language, and it was designed to support automated reasoning across terms in 12 languages.

The increasing popularity and use of microblogging services such as Twitter are recently a new valuable data source for Web-based surveillance because of their message volume and frequency. Twitter users may post about an illness, and their relationships in the network can give us information about whom they could be in contact with. Furthermore, user posts retrieved from the public Twitter API can come with GPS-based location tags, which can be used to locate the potential centre of disease outbreaks. A number of works have already appeared that show the potential of Twitter messages to track and predict outbreaks. A document classifier to identify relevant messages was presented in Culotta [110]. In this work, Twitter messages related to the flu were gathered, and then a number of classification systems based on different regression models to correlate these messages with CDC statistics were compared; the study found that the best model had a correlation of 0.78 (simple model regression). Aramaki [111] presented a comparative study of various machine-learning methods to classify tweets related to influenza into two categories: positive and negative. Their experimental results showed that the SVM model that used polynomial kernels achieved the highest accuracy (FMeasure of 0.756) and the lowest training time.

Well-known *regression models* were evaluated on their ability to assess disease outbreaks from tweets in Bodnar and Salathé [112]. Regression methods such as linear, multivariable, and SVM were applied to the raw count of tweets that contained at least one of the keywords related to a specific disease, in this case ”flu”. The models also validated that even using irrelevant tweets and randomly generated datasets, regression methods were able to assess disease levels comparatively well.

A new *unsupervised machine learning approach* to detect public health events was proposed in Fisichella et al. [113] that can complement existing systems because it allows for identifying public health events even if no matching keywords or linguistic patterns can be found. This new approach defined a generative model for predictive event detection from documents by modelling the features based on trajectory distributions.

However, in recent years, a number of *surveillance systems* have appeared that apply these social mining techniques and that have been widely used by public health organizations such as the World Health Organization (WHO) and the European Centre for Disease Prevention and Control [114]. Tracking and monitoring mechanisms for early detection are critical in reducing the impact of epidemics through rapid responses.

One of the earliest surveillance systems is the **Global Public Health Intelligence Network (GPHIN)** [115] developed by the Public Health Agency of Canada in collaboration with the WHO. It is a secure, Web-based, multilingual warning tool that continuously monitors and analyses global media data sources to identify information about disease outbreaks and other events related to public healthcare. The information is filtered for relevance by an automated process and is then analysed by Public Health Agency of Canada GPHIN officials. From 2002 to 2003, this surveillance system was able to detect the outbreak of SARS (severe acute respiratory syndrome).

From the BioCaster ontology in 2006 arose the **BioCaster system** [116] for monitoring online media data. The system continuously analyses documents reported from over 1700 RSS feeds, Google News, WHO, ProMED-mail, and the European Media Monitor, among other data sources. The extracted text is classified based on its topical relevance and plotted onto a Google map using geo-information. The system has four main stages: topic classification, named entity recognition, disease/location detection, and event recognition. In the first stage, the texts are classified as relevant or non-relevant using a naive Bayes classifier. Then, for the relevant document corpora, entities of interest from 18 concept types based on the ontology related to diseases, viruses, bacteria, locations, and symptoms are searched.

HealthMap project [117] is a global disease alert map that uses data from different sources such as Google News, expert-curated discussions such as ProMED-mail, and official organization reports such as those from the WHO or Euro Surveillance, an automated real-time system that monitors, organises, integrates, filters, visualises, and disseminates online information about emerging diseases.

Another system that collects news from the Web related to human and animal health and that plots the data on Google Maps is **EpiSpider** [118]. This tool automatically extracts information on infectious disease outbreaks from multiple sources including ProMed-mail and medical Web sites, and it is used as a surveillance system by

Table 3
Basic features related to social big data applications in health care area.

Authors	Ref. num.	Summary	Methods
Culotta	[110]	Track and predict outbreak detection using Twitter	Classification (regression models)
Aramaki et al.	[111]	Classify tweets related to influenza	Classification
Bodnar and Salathé	[112]	Assess disease outbreaks from tweets	Regression methods
Fisichella et al.	[113]	Detect public health events	Modelling trajectory distributions
GPHIN	[115]	Identify information about disease outbreaks and other events related to public healthcare	Classification documents for relevance
BioCaster	[116]	Monitoring online media data related to diseases, viruses, bacteria, locations and symptoms	Topic classification, named entity recognition, event recognition
HealthMap	[117]	Global disease alert map	Mapping techniques
EpiSpider	[118]	Human and animal disease alert map	Topic and location detection

Table 4
Basic features related to social big data applications in user experiences-based visualisation.

Authors	Ref. num.	Summary	Methods
GGobi	[123]	Visualisation program for exploring high-dimensional data	Supervised Classification, Unsupervised Classification, Inference
MIMO	[124]	Visualisation Framework for Real Time Decision Making in a Multi-Input Multi-Output System	Bayesian causal network, Decision Making Tools
Insense	[126]	Collecting user experiences into a continually growing and adapting multimedia diary.	Classification of patterns in sensor readings from a camera, microphone, and accelerometers
Many Eyes	[127]	Creating visualisations in collaborative environment from upload data sets	Visualisation layout algorithms
TweetPulse	[128]	Building social pulse by aggregating identical user experiences	Visualising temporal dynamics of the thematic events

public healthcare organizations, a number of universities, and health research organizations. Additionally, this system automatically converts the topic and location information from the reports into RSS feeds.

Finally, Lyon et al. [119] conducted a comparative assessment of these three systems (BioCaster, EpiSpider, and HealthMap) related to their ability to gather and analyse information that is relevant to public health. EpiSpider obtained more relevant documents in this study. However, depending on the language of each system, the ability to acquire relevant information from different countries differed significantly. For instance, Biocaster gives special priority to languages from the Asia-Pacific region, and EpiSpider only considers documents written in English. Table 3 shows a summary of the previous applications and their related functionalities and methods.

4.4. User experiences-based visualisation

Big data from social media needs to be visualised for better user experiences and services. For example, the large volume of numerical data (usually in tabular form) can be transformed into different formats. Consequently, user understandability can be increased. The capability of supporting timely decisions based on visualising such big data is essential to various domains, e.g., business success, clinical treatments, cyber and national security, and disaster management [120]. Thus, user-experience-based visualisation has been regarded as important for supporting decision makers in making better decisions. More particularly, visualisation is also regarded as a crucial data analytic tool for social media [121]. It is important for understanding users needs in social networking services.

There have been many visualisation approaches to collecting (and improving) user experiences. One of the most well-known is interactive data analytics. Based on a set of features of the given big data, users can interact with the visualisation-based analytics system. Such systems are R-based software packages [122] and **GGobi** [123]. Moreover, some systems have been developed using statistical inferences. A Bayesian inference scheme-based multi-input/multi-output (**MIMO**) system [124] has been developed for better visualisation.

We can also consider life-logging services that record all user experiences [125], which is also known as quantify-self. Various sensors can capture continuous physiological data (e.g., mood, arousal, and blood oxygen levels) together with user activities. In this context, life caching has been presented as a collaborative social action of storing and sharing users life events in an open environment. More practically, this collaborative user experience has been applied to gaming to encourage users. Systems such as **Insense** [126] are based on wearable devices and can collect users experiences into a continually growing and adapting multimedia diary. The inSense system uses the patterns in sensor readings from a camera, a microphone, and accelerometers to classify the users activities and automatically collect multimedia clips when the user is in an interesting situation.

Moreover, visualisation systems such as **Many Eyes** [127] have been designed to upload datasets and create visualisations in collaborative environments, allowing users to upload data, create visualisation of that data, and leave comments on both the visualisation and the data, providing a medium to foment discussion among users. Many Eyes is designed for ordinary people and does not require any extensive training or prior knowledge to take full advantage of its functionalities.

Other visual analytics tools have shown some graphical visualisations for supporting efficient analytics of the given big data. Particularly, **TweetPulse** [128] has built social pulses by aggregating identical user experiences in social networks (e.g., Twitter), and visualised temporal dynamics of the thematic events. Finally, Table 4 provides a summary of those applications related to the methods used for visualisation based on user experiences.

5. Conclusions and open problems

With the large number and rapid growth of social media systems and applications, social big data has become an important topic in a broad array of research areas. The aim of this study has been to provide a holistic view and insights for potentially helping to find the most relevant solutions that are currently available for managing knowledge in social media.

As such, we have investigated the state-of-the-art technologies and applications for processing the big data from social media. These technologies and applications were discussed in the following aspects: (*i*) What are the main methodologies and technologies that are available for gathering, storing, processing, and analysing big data from social media? (*ii*) How does one analyse social big data to discover meaningful patterns? and (*iii*) How can these patterns be exploited as smart, useful user services through the currently deployed examples in social-based applications?

More practically, this survey paper shows and describes a number of existing systems (e.g., frameworks, libraries, software applications) that have been developed and that are currently being used in various domains and applications based on social media. The paper has avoided describing or analysing those straightforward applications such as Facebook and Twitter that currently intensively use big data technologies, instead focusing on other applications (such as those related to marketing, crime analysis, or epidemic intelligence) that could be of interest to potential readers.

Although it is extremely difficult to predict which of the different issues studied in this work will be the next "*trending topic*" in social big data research, from among all of the problems and topics that are currently under study in different areas, we selected some "open topics" related to privacy issues, streaming and online algorithms, and data fusion visualisation, providing some insights and possible future trends.

5.1. Privacy issues

In the era of online big data and social media, protecting the privacy of the users on social media has been regarded as an important issue. Ironically, as the analytics introduced in this paper become more advanced, the risk of privacy leakage is growing.

As such, many privacy-preserving studies have been proposed to address privacy-related issues. We can note that there are two main well-known approaches. The first one is to exploit "k-anonymity", which is a property possessed by certain anonymised data [129]. Given the private data and a set of specific fields, the system (or service) has to make the data practically useful without identifying the individuals who are the subjects of the data. The second approach is "differential privacy", which can provide an efficient way to maximise the accuracy of queries from statistical databases while minimising the chances of identifying its records [130].

However, there are still open issues related to privacy. Social identification is the important issue when social data are merged from available sources, and secure data communication and graph matching are potential research areas [89]. The second issue is evaluation. It is not easy to evaluate and test privacy-preserving services with real data. Therefore, it would be particularly interesting in the future to consider how to build useful benchmark datasets for evaluation.

Moreover, we have to consider this data privacy issues in many other research areas. In the context of law (also, international law) enforcement, data privacy must be prevented from any illegal usages, whereas governments tend to trump the user privacy for the purpose of national securities [131].

Also, developing educational program for technicians (also, students) is important [132]. It is still open issue on how (and what) to design the curriculum for the data privacy.

5.2. Streaming and online algorithms

One of the current main challenges in data mining related to big data problems is to find adequate approaches to analysing massive amounts of **online data** (or **data streams**). Because classification methods require previous labelling, these methods also require great effort for real-time analysis. However, because unsupervised techniques do not need this previous process, clustering has become a promising field for real-time analysis, especially when these data come from social media sources. When data streams are analysed, it is important to consider the analysis goal in order to determine the best type of algorithm to be used. We were able to divide data stream analysis into two main categories:

- *Offline analysis*: We consider a portion of data (usually large data) and apply an offline clustering algorithm to analyse the data.
- *Online analysis*: The data are analysed in real time. These kinds of algorithms are constantly receiving new data and are not usually able to keep past information.

A new generation of online [133,134] and streaming [135,136] algorithms is currently being developed in order to manage social big data challenges, and these algorithms require high scalability in both memory consumption [137] and time computation. Some new developments related to traditional clustering algorithms, such as the K-mean [138], EM [139], which has been modified to work with the MapReduce paradigm, and more sophisticated approaches based on graph computing (such as spectral clustering), are currently being developed [140–142] into more efficient versions from the state-of-the-art algorithms [143,144].

5.3. Methods for data fusion & data visualisation

Finally, data fusion and data visualisation are two clear challenges in social big data. Although both areas have been intensively studied with regard to large, distributed, heterogeneous, and streaming data fusion [145] and data visualisation and analysis [146], the current, rapid evolution of social media sources jointly with big data technologies creates some particularly interesting challenges related to:

- Obtaining more reliable methods for fusing the multiple features of multimedia objects for social media applications [147].
- Studying the dynamics of individual and group behaviour, characterising patterns of information diffusion, and identifying influential individuals in social networks and other social media-based applications [148].
- Identifying events [149] in social media documents via clustering and using similarity metric learning approaches to produce high-quality clustering results [150].
- The open problems and challenges related to visual analytics [146], especially related to the capacity to collect and store new data, are rapidly increasing in number, including the ability to analyse these data volumes[151], to record data about the movement of people and objects at a large scale [152], and to analyse spatio-temporal data and solve spatio-temporal problems in social media [153], among others.

Acknowledgements

This work has been supported by several research grants: Comunidad Autónoma de Madrid under CIBERDINE S2013/ICE-3095 project; Spanish Ministry of Science and Education under grant TIN2014-56494-C4-4-P; Savier Open Innovation Project (Airbus Defence & Space, FUAM-076915), and by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP): (NRF-2013K2A1A2055213, NRF-2014R1A 2A2A05007154).

References

[1] IBM, Big Data and Analytics, 2015. URL http://www-01.ibm.com/software/data/bigdata/what-is-big-data.html
[2] Infographic, The Data Explosion in 2014 Minute by Minute, 2015. URL http://aci.info/2014/07/12/the-data-explosion-in-2014-minute-by-minute-infographic
[3] X. Wu, X. Zhu, G.-Q. Wu, W. Ding, Data mining with big data, IEEE Trans. Knowl. Data Eng. 26 (1) (2014) 97–107.
[4] A. Cuzzocrea, I.-Y. Song, K.C. Davis, Analytics over large-scale multidimensional data: the big data revolution!, in: Proceedings of the ACM 14th International Workshop on Data Warehousing and OLAP, ACM, 2011, pp. 101–104.
[5] D. Laney, 3D Data Management: Controlling Data Volume, Velocity, and Variety, Technical Report, 2001.URL http://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf (accessed August 2015).
[6] M.A. Beyer, D. Laney, The Importance of 'Big Data': A Definition, Gartner, Stamford, CT (2012).
[7] I.A.T. Hashema, I. Yaqooba, N.B. Anuara, S. Mokhtara, A. Gania, S.U. Khanb, The rise of big data on cloud computing: review and open research issues, Inf. Syst. 47 (2015) 98–115.

[8] R.L. Grossman, Y. Gu, J. Mambretti, M. Sabala, A. Szalay, K. White, An overview of the open science data cloud, in: Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing, HPDC '10, ACM, New York, NY, USA, 2010, pp. 377–384, doi:10.1145/1851476.1851533.

[9] N. Khan, I. Yaqoob, I.A.T. Hashem, Z. Inayat, W.K.M. Ali, M. Alam, M. Shiraz, A. Gani, Big data: survey, technologies, opportunities, and challenges, The Sci. World J. 2014 (2014) 1–18.

[10] N. Couldry, Media, Society, World: Social Theory and Digital Media Practice, Polity, 2012.

[11] T. Correa, A.W. Hinsley, H.G. De Zuniga, Who interacts on the web?: the intersection of users' personality and social media use, Comput. Hum. Behav. 26 (2) (2010) 247–253.

[12] A.M. Kaplan, M. Haenlein, Users of the world, unite! the challenges and opportunities of social media, Bus. Horizons 53 (1) (2010) 59–68.

[13] P.A. Tess, The role of social media in higher education classes (real and virtual)–a literature review, Comput. Hum. Behav. 29 (5) (2013) A60–A68.

[14] M. Salathé, D.Q. Vu, S. Khandelwal, D.R. Hunter, The dynamics of health behavior sentiments on a large online social network, EPJ Data Sci. 2 (1) (2013) 1–12.

[15] E. Cambria, D. Rajagopal, D. Olsher, D. Das, Big social data analysis, Big Data Comput. 13 (2013) 401–414.

[16] L. Manovich, Trending: the promises and the challenges of big social data, Debates Digit. Hum. (2011) 460–475.

[17] S. Kaisler, F. Armour, J.A. Espinosa, W. Money, Big data: Issues and challenges moving forward, in: Proceedings of 46th Hawaii International Conference on System Sciences (HICSS), IEEE, 2013, pp. 995–1004.

[18] H. Chen, R.H. Chiang, V.C. Storey, Business intelligence and analytics: from big data to big impact, MIS Q. 36 (4) (2012) 1165–1188.

[19] T. White, Hadoop: The Definitive Guide, O'Reilly Media, 2009.

[20] M. Zaharia, M. Chowdhury, M.J. Franklin, S. Shenker, I. Stoica, Spark: Cluster computing with working sets, in: Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing,HotCloud'10, USENIX Association, Berkeley, CA, USA, 2010, p. 10. http://dl.acm.org/citation.cfm?id=1863103.1863113.

[21] S. Owen, R. Anil, T. Dunning, E. Friedman, Mahout in Action, 1, Manning Publications, 2011. URL http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/1935182684

[22] X. Meng, J. Bradley, B. Yavuz, E. Sparks, S. Venkataraman, D. Liu, J. Freeman, D. Tsai, M. Amde, S. Owen, et al., MLlib: machine learning in apache spark, 2015, pp. 1–7, arXiv:1505.06807.

[23] T. Kraska, A. Talwalkar, J.C. Duchi, R. Griffith, M.J. Franklin, M.I. Jordan, Mlbase: a distributed machine-learning system, in: Proceedings of Sixth Biennial Conference on Innovative Data Systems Research, Asilomar CIDR, CA, USA, January 6-9, 2013, 2013.

[24] E.R. Sparks, A. Talwalkar, V. Smith, J. Kottalam, X. Pan, J.E. Gonzalez, M.J. Franklin, M.I. Jordan, T. Kraska, MLI: an API for distributed machine learning, in: Proceedings of IEEE 13th International Conference on Data Mining, Dallas, TX, USA, December 7-10, 2013, 2013, pp. 1187–1192, doi:10.1109/ICDM.2013.158.

[25] J. Dean, S. Ghemawat, Mapreduce: simplified data processing on large clusters, in: Proceedings of the 6th Conference on Symposium on Operating Systems Design and Implementation, OSDI'04, USENIX Association, 2004.

[26] J. Dean, S. Ghemawat, Mapreduce: simplified data processing on large clusters, Commun. ACM 51 (1) (2008) 107–113, doi:10.1145/1327452.1327492.

[27] K. Shim, Mapreduce algorithms for big data analysis, Proc. VLDB Endow. 5 (12) (2012) 2016–2017.

[28] M. Zaharia, A. Konwinski, A.D. Joseph, R. Katz, I. Stoica, Improving mapreduce performance in heterogeneous environments, in: Proceedings of the 8th USENIX Conference on Operating Systems Design and Implementation, OSDI'08, USENIX Association, Berkeley, CA, USA, 2008, pp. 29–42. http://dl.acm.org/citation.cfm?id=1855741.1855744.

[29] R.S. Xin, J. Rosen, M. Zaharia, M.J. Franklin, S. Shenker, I. Stoica, Shark: Sql and rich analytics at scale, in: Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data, SIGMOD '13, ACM, New York, NY, USA, 2013, pp. 13–24, doi:10.1145/2463676.2465288.

[30] A. Mostosi, Useful stuff, 2015. http://blog.andreamostosi.name/big-data/

[31] A. Mostosi, The big-data ecosystem table, 2015. URL http://bigdata.andreamostosi.name/

[32] C. Emerick, B. Carper, C. Grand, Clojure Programming, O'Really, 2011.

[33] M. Burrows, The chubby lock service for loosely-coupled distributed systems, in: Proceedings of the 7th Symposium on Operating Systems Design and Implementation, OSDI '06, USENIX Association, Berkeley, CA, USA, 2006, pp. 335–350. http://dl.acm.org/citation.cfm?id=1298455.1298487.

[34] A. Alexandrov, R. Bergmann, S. Ewen, J.-C. Freytag, F. Hueske, A. Heise, O. Kao, M. Leich, U. Leser, V. Markl, F. Naumann, M. Peters, A. Rheinländer, M.J. Sax, S. Schelter, M. Höger, K. Tzoumas, D. Warneke, The stratosphere platform for big data analytics, VLDB J. 23 (6) (2014) 939–964, doi:10.1007/s00778-014-0357-y.

[35] S. Ghemawat, H. Gobioff, S.-T. Leung, The google file system, in: Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles, SOSP '03, ACM, New York, NY, USA, 2003, pp. 29–43, doi:10.1145/945445.945450.

[36] F. Chang, J. Dean, S. Ghemawat, W.C. Hsieh, D.A. Wallach, M. Burrows, T. Chandra, A. Fikes, R.E. Gruber, Bigtable: a distributed storage system for structured data, in: Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation, OSDI '06, USENIX Association, Berkeley, CA, USA, 2006, p. 15. http://dl.acm.org/citation.cfm?id=1267308.1267323.

[37] G. Malewicz, M.H. Austern, A.J. Bik, J.C. Dehnert, I. Horn, N. Leiser, G. Czajkowski, Pregel: a system for large-scale graph processing, in: Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, SIGMOD '10, ACM, New York, NY, USA, 2010, pp. 135–146, doi:10.1145/1807167.1807184.

[38] K. Chodorow, MongoDB: The Definitive Guide, O'Reilly Media, Inc., 2013.

[39] M.J. Crawley, The R Book, 1st, Wiley Publishing, 2007.

[40] S. Bennett, Twitter now seeing 400 million tweets per day, increased mobile ad revenue, says ceo, 2012. URL http://www.adweek.com/socialtimes/twitter-400-million-tweets

[41] L. Ott, M. Longnecker, R.L. Ott, An Introduction to Statistical Methods and Data Analysis, 511, Duxbury Pacific Grove, CA, 2001.

[42] B. Elser, A. Montresor, An evaluation study of bigdata frameworks for graph processing, in: Proceedings of IEEE International Conference on Big Data, IEEE, 2013, pp. 60–67.

[43] L.G. Valiant, A bridging model for parallel computation, Commun. ACM 33 (8) (1990) 103–111, doi:10.1145/79173.79181.

[44] S. Seo, E.J. Yoon, J. Kim, S. Jin, J.-S. Kim, S. Maeng, Hama: an efficient matrix computation with the mapreduce framework, in: Proceedings of the Second International Conference on Cloud Computing Technology and Science (CloudCom), IEEE, 2010, pp. 721–726.

[45] A. Clauset, Finding local community structure in networks, Phys. Rev. E 72 (2005) 026132, doi:10.1103/PhysRevE.72.026132.

[46] F. Santo, Community detection in graphs, Phys. Rep. 486 (3-5) (2010) 75–174, doi:10.1016/j.physrep.2009.11.002.

[47] R. Kannan, S. Vempala, A. Veta, On clusterings-good, bad and spectral, in: Proceedings of the 41st Annual Symposium on Foundations of Computer Science, FOCS '00, IEEE Computer Society, Washington, DC, USA, 2000, pp. 367–377.

[48] I.M. Bomze, M. Budinich, P.M. Pardalos, M. Pelillo, The maximum clique problem, in: Handbook of Combinatorial Optimization, Kluwer Academic Publishers, 1999, pp. 1–74.

[49] M. Girvan, M.E.J. Newman, Community structure in social and biological networks, Proc. Natl. Acad. Sci. 99 (12) (2002) 7821–7826.

[50] M.E.J. Newman, Fast algorithm for detecting community structure in networks, Phys. Rev. E 69 (6) (2004) 066133+, doi:10.1103/physreve.69.066133.

[51] A. Clauset, M.E. Newman, C. Moore, Finding community structure in very large networks, Phys. Rev. E 70 (6) (2004) 066111.

[52] M.E. Newman, Modularity and community structure in networks, Proc. Natl. Acad. Sci. 103 (23) (2006) 8577–8582.

[53] T. Richardson, P.J. Mucha, M.A. Porter, Spectral tri partitioning of networks, Phys. Rev. E 80 (3) (2009) 036111.

[54] G. Wang, Y. Shen, M. Ouyang, A vector partitioning approach to detecting community structure in complex networks, Comput. Math. Appl. 55 (12) (2008) 2746–2752.

[55] H. Zhou, R. Lipowsky, Network brownian motion: a new method to measure vertex-vertex proximity and to identify communities and subcommunities, in: Computational Science-ICCS 2004, Springer, 2004, pp. 1062–1069.

[56] Y. Dong, Y. Zhuang, K. Chen, X. Tai, A hierarchical clustering algorithm based on fuzzy graph connectedness, Fuzzy Sets Syst. 157 (13) (2006) 1760–1774.

[57] G. Bello-Orgaz, H.D. Menéndez, D. Camacho, Adaptive k-means algorithm for overlapped graph clustering, Int. J. Neural Syst. 22 (05) (2012) 1250018.

[58] J. Xie, S. Kelley, B.K. Szymanski, Overlapping community detection in networks: the state-of-the-art and comparative study, ACM Comput. Surv. (CSUR) 45 (4) (2013) 43.

[59] O. Zamir, O. Etzioni, Web document clustering: a feasibility demonstration, in: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '98, ACM, New York, NY, USA, 1998, pp. 46–54, doi:10.1145/290941.290956.

[60] W.B. Frakes, R.A. Baeza-Yates (Eds.), Information Retrieval: Data Structures & Algorithms, Prentice-Hall, 1992.

[61] C.D. Manning, P. Raghavan, H. Schütze, Introduction to Information Retrieval, Cambridge University Press, New York, NY, USA, 2008.

[62] X. Hu, H. Liu, Text analytics in social media, in: Mining Text Data, Springer, 2012, pp. 385–414.

[63] S. Wold, K. Esbensen, P. Geladi, Principal component analysis, Chemom. Intell. Lab. Syst. 2 (1) (1987) 37–52.

[64] S.C. Deerwester, S.T. Dumais, T.K. Landauer, G.W. Furnas, R.A. Harshman, Indexing by latent semantic analysis, JAsIs 41 (6) (1990) 391–407.

[65] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent Dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.

[66] L. Yao, D. Mimno, A. McCallum, Efficient methods for topic model inference on streaming document collections, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2009, pp. 937–946.

[67] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, ACM Comput. Surv. 31 (3) (1999) 264–323, doi:10.1145/331499.331504.

[68] B. Larsen, C. Aone, Fast and effective text mining using linear-time document clustering, in: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and data mining, KDD '99, ACM, New York, NY, USA, 1999, pp. 16–22, doi:10.1145/312129.312186.

[69] Y. Zhao, G. Karypis, Evaluation of hierarchical clustering algorithms for document datasets, in: Proceedings of the Eleventh International Conference on Information and Knowledge Management, CIKM '02, ACM, New York, NY, USA, 2002, pp. 515–524, doi:10.1145/584792.584877.

[70] Y. Zhao, G. Karypis, Empirical and theoretical comparisons of selected criterion functions for document clustering, Mach. Learn. 55 (3) (2004) 311–331, doi:10.1023/B:MACH.0000027785.44527.d6.

[71] F. Sebastiani, Machine learning in automated text categorization, ACM Comput. Surv. (CSUR) 34 (1) (2002) 1–47.

[72] B. Pang, L. Lee, S. Vaithyanathan, Thumbs up?: sentiment classification using machine learning techniques, in: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing, 10, Association for Computational Linguistics, 2002, pp. 79–86.

[73] C.C. Aggarwal, Data Streams: Models and Algorithms, 31, Springer Science & Business Media, 2007.

[74] S. Zhong, Efficient online spherical k-means clustering, in: Proceedings of the 2005 IEEE International Joint Conference on Neural Networks, IJCNN'05, 5, IEEE, 2005, pp. 3180–3185.

[75] W. Chen, C. Wang, Y. Wang, Scalable influence maximization for prevalent viral marketing in large-scale social networks, in: B. Rao, B. Krishnapuram, A. Tomkins, Q. Yang (Eds.), Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, , July 25-28, 2010, ACM, Washington, DC, USA, 2010, pp. 1029–1038, doi:10.1145/1835804.1835934.

[76] D.T. Nguyen, J.J. Jung, Real-time event detection on social data stream, Mobile Netw. Appl. 20 (4) (2015) 475–486, doi:10.1007/s11036-014-0557-0.

[77] A. Guille, H. Hacid, C. Favre, D.A. Zighed, Information diffusion in online social networks: a survey, SIGMOND Rec. 42 (2) (2013) 17–28.

[78] M. Gomez-Rodriguez, J. Leskovec, A. Krause, Inferring networks of diffusion and influence, ACM Trans. Knowl. Discov. Data 5 (4) (2012) 21, doi:10.1145/2086737.2086741.

[79] E. Sadikov, M. Medina, J. Leskovec, H. Garcia-Molina, Correcting for missing data in information cascades, in: I. King, W. Nejdl, H. Li (Eds.), Proceedings of the 4th International Conference on Web Search and Web Data Mining (WSDM 2011), Hong Kong, China, February 9-12, 2011, ACM, 2011, pp. 55–64, doi:10.1145/1935826.1935844.

[80] E. Anshelevich, A. Hate, M. Magdon-Ismail, Seeding influential nodes in non-submodular models of information diffusion, Auton. Agents Multi-Agent Syst. 29 (1) (2015) 131–159.

[81] C. Jiang, Y. Chen, K.R. Liu, Graphical evolutionary game for information diffusion over social networks, IEEE J. Sel. Top. Signal Process. 8 (4) (2014b) 524–536.

[82] C. Jiang, Y. Chen, K.R. Liu, Evolutionary dynamics of information diffusion over social networks, IEEE Trans. Signal Process. 62 (17) (2014a) 4573–4586.

[83] T.-c. Fu, A review on time series data mining, Eng. Appl. Artif. Intell. 24 (1) (2011) 164–181.

[84] J. Lin, E. Keogh, S. Lonardi, B. Chiu, A symbolic representation of time series, with implications for streaming algorithms, in: Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, ACM, 2003, pp. 2–11.

[85] M. Cataldi, L.D. Caro, C. Schifanella, Emerging topic detection on twitter based on temporal and social terms evaluation, in: Proceedings of the 10th International Workshop on Multimedia Data Mining, ACM, New York, NY, USA, 2010, pp. 1–10, doi:10.1145/1814245.1814249.

[86] D.T. Nguyen, J.J. Jung, Privacy-preserving discovery of topic-based events from social sensor signals: an experimental study on twitter, Sci. World J. 2014 (2014) 1–5.

[87] J.J. Jung, Integrating social networks for context fusion in mobile service platforms, J. Univers. Comput. Sci. 16 (15) (2010) 2099–2110.

[88] H.H. Hoang, T.N.-P. Cung, D.K. Truong, D. Hwang, J.J. Jung, Semantic information integration with linked data mashups approaches, Int. J. Distrib. Sens. Networks 2014 (2014) 1–12. Article ID 813875

[89] N.H. Long, J.J. Jung, Privacy-aware framework for matching online social identities in multiple social networking services, Cybern. Syst. 46 (1-2) (2015) 69–83.

[90] S. Caton, C. Haas, K. Chard, K. Bubendorfer, O.F. Rana, A social compute cloud: allocating and sharing infrastructure resources via social networks, IEEE Trans. Serv. Comput. 7 (3) (2014) 359–372.

[91] T.H. Davenport, J.G. Harris, Competing on Analytics: The New Science of Winning, Harvard Business Press, 2007.

[92] C. Maurer, R. Wiegmann, Effectiveness of Advertising on Social Network Sites: A Case Study on Facebook, Springer, 2011.

[93] C. Trattner, F. Kappe, Social stream marketing on Facebook: a case study, Int. J. Soc. Humanist. Comput. 2 (1-2) (2013) 86–103.

[94] B.J. Jansen, M. Zhang, K. Sobel, A. Chowdury, Twitter power: tweets as electronic word of mouth, J. Am. Soc. Inf. Sci. Tech. 60 (11) (2009) 2169–2188.

[95] S. Asur, B. Huberman, et al., Predicting the future with social media, in: Proceedings of International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM, 1, IEEE, 2010, pp. 492–499.

[96] H. Ma, H. Yang, M.R. Lyu, I. King, Mining social networks using heat diffusion processes for marketing candidates selection, in: Proceedings of the 17th ACM Conference on Information and Knowledge Management, ACM, 2008, pp. 233–242.

[97] R. Wortley, L. Mazerolle, Environmental Criminology and Crime Analysis, Willan, 2013.

[98] O. Knutsson, E. Sneiders, A. Alfalahi, Opportunities for improving egovernment: using language technology in workflow management, in: Proceedings of the 6th International Conference on Theory and Practice of Electronic Governance, ICEGOV '12, ACM, New York, NY, USA, 2012, pp. 495–496, doi:10.1145/2463728.2463833.

[99] C.-H. Ku, G. Leroy, A decision support system: automated crime report analysis and classification for e-government, Gov. Inf. Q. 31 (4) (2014) 534–544.

[100] P. Phillips, I. Lee, Mining co-distribution patterns for large crime datasets, Expert Syst. Appl. 39 (14) (2012) 11556–11563.

[101] S. Chainey, L. Tompson, S. Uhlig, The utility of hotspot mapping for predicting spatial patterns of crime, Secur. J. 21 (1) (2008) 4–28.

[102] M.S. Gerber, Predicting crime using twitter and Kernel density estimation, Decis. Support Syst. 61 (2014) 115–125.

[103] E. Kirkos, C. Spathis, Y. Manolopoulos, Data mining techniques for the detection of fraudulent financial statements, Expert Syst. Appl. 32 (4) (2007) 995–1003.

[104] J.T. Quah, M. Sriganesh, Real-time credit card fraud detection using computational intelligence, Expert Syst. Appl. 35 (4) (2008) 1721–1732.

[105] S.-H. Li, D.C. Yen, W.-H. Lu, C. Wang, Identifying the signs of fraudulent accounts using data mining techniques, Comput. Hum. Behav. 28 (3) (2012) 1002–1013.

[106] C. Paquet, D. Coulombier, R. Kaiser, M. Ciotti, Epidemic intelligence: a new framework for strengthening disease surveillance in europe., Euro surveillance: bulletin europeen sur les maladies transmissibles European communicable disease bulletin 11 (12) (2005) 212–214.

[107] A.M. Cohen, W.R. Hersh, A survey of current work in biomedical text mining, Brief. Bioinform. 6 (1) (2005) 57–71.

[108] V. Lampos, N. Cristianini, Nowcasting events from the social web with statistical learning, ACM Trans. Intell. Syst. Technol. (TIST) 3 (4) (2012) 72.

[109] N. Collier, R.M. Goodwin, J. McCrae, S. Doan, A. Kawazoe, M. Conway, A. Kawtrakul, K. Takeuchi, D. Dien, An ontology-driven system for detecting global health events, in: Proceedings of the 23rd International Conference on Computational Linguistics, Association for Computational Linguistics, 2010, pp. 215–222.

[110] A. Culotta, Towards detecting influenza epidemics by analyzing twitter messages, in: Proceedings of the First Workshop on Social Media Analytics, ACM, 2010, pp. 115–122.

[111] E. Aramaki, S. Maskawa, M. Morita, Twitter catches the flu: detecting influenza epidemics using twitter, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2011, pp. 1568–1576.

[112] T. Bodnar, M. Salathé, Validating models for disease detection using twitter, in: Proceedings of the 22nd international conference on World Wide Web companion, International World Wide Web Conferences Steering Committee, 2013, pp. 699–702.

[113] M. Fisichella, A. Stewart, A. Cuzzocrea, K. Denecke, Detecting health events on the social web to enable epidemic intelligence, in: String Processing and Information Retrieval, Springer, 2011, pp. 87–103.

[114] D.M. Hartley, N.P. Nelson, R. Walters, R. Arthur, R. Yangarber, L. Madoff, J. Linge, A. Mawudeku, N. Collier, J.S. Brownstein, et al., The landscape of international event-based biosurveillance., Emerg. Health Threat. 3 (2010).

[115] E. Mykhalovskiy, L. Weir, The global public health intelligence network and early warning outbreak detection, Can. J. Public Health 97 (1) (2006) 42–44.

[116] N. Collier, S. Doan, A. Kawazoe, R.M. Goodwin, M. Conway, Y. Tateno, Q.-H. Ngo, D. Dien, A. Kawtrakul, K. Takeuchi, et al., Biocaster: detecting public health rumors with a web-based text mining system, Bioinformatics 24 (24) (2008) 2940–2941.

[117] J.S. Brownstein, C.C. Freifeld, B.Y. Reis, K.D. Mandl, Surveillance sans frontieres: internet-based emerging infectious disease intelligence and the healthmap project, PLoS Med. 5 (7) (2008) e151.

[118] M. Keller, M. Blench, H. Tolentino, C.C. Freifeld, K.D. Mandl, A. Mawudeku, G. Eysenbach, J.S. Brownstein, Use of unstructured event-based reports for global infectious disease surveillance, Emerg. Infect. Dis. 15 (5) (2009) 689.

[119] A. Lyon, M. Nunn, G. Grossel, M. Burgman, Comparison of web-based biosecurity intelligence systems: biocaster, epispider and healthmap, Transbound. Emerg. Dis. 59 (3) (2012) 223–232.

[120] D. Keim, H. Qu, K.-L. Ma, Big-data visualization, IEEE Comput. Gr. Appl. 33 (4) (2013) 20–21.

[121] X.P. Kotval, M.J. Burns, Visualization of entities within social media: toward understanding users' needs, Bell Labs Tech. J. 17 (4) (2013) 77–101.

[122] A. Miroshnikov, E.M. Conlon, Parallelmcmccombine: an r package for bayesian methods for big data and analytics, PLOS One 9 (9) (2014).

[123] D.F. Swayne, D.T. Lang, A. Buja, D. Cook, GGobi: evolving from XGobi into an extensible framework for interactive data visualization, Comput. Stat. Data Anal. 43 (4) (2003) 423–444, doi:10.1016/S0167-9473(02)00286-4.

[124] P. Ashok, D. Tesar, A visualization framework for real time decision making in a multi-input multi-output system, IEEE Syst. J. 2 (1) (2008) 129–145, doi:10.1109/JSYST.2008.916060.

[125] C. Gurrin, A.F. Smeaton, A.R. Doherty, Foundations and Trends in Information Retrieval, 8, Now Publishers, 2014, pp. pp.1–125.

[126] M. Blum, A. Pentland, G. Troster, Insense: interest-based life logging, Multimed. IEEE 13 (4) (2006) 40–48, doi:10.1109/MMUL.2006.87.

[127] F.B. Viegas, M. Wattenberg, F. van Ham, J. Kriss, M. McKeon, Manyeyes: a site for visualization at internet scale, IEEE Trans. Vis. Comput. Gr. 13 (6) (2007) 1121–1128, doi:10.1109/TVCG.2007.70577.

[128] D. Hwang, J.E. Jung, S. Park, H.T. Nguyen, Social data visualization system for understanding diffusion patterns on twitter: a case study on korean enterprises, Comput. Inform. 33 (3) (2014) 591–608.

[129] L. Sweeney, K-anonymity: a model for protecting privacy, Int. J. Uncertain. Fuzziness Knowledge-based Syst. 10 (5) (2002) 557–570.

[130] C. Dwork, Differential privacy: a survey of results, in: M. Agrawal, D. Du, Z. Duan, A. Li (Eds.), Proceedings of 5th International Conference on Theory and Applications of Models of Computation (TAMC 2008), Xi'an, China, April 25-29, Lecture Notes in Computer Science, 4978, Springer, 2008, pp. 1–19.

[131] C. Kuner, F.H. Cate, C. Millard, D.J.B. Svantesson, The challenge of 'big data' for data protection, Int. Data Priv. Law 2 (2) (2012) 47–49.

[132] S. Landau, Educating engineers: teaching privacy in a world of open doors, IEEE Secur. Priv. 12 (3) (2014) 66–70.

[133] A. Fiat, Online Algorithms: The State of the Art, in: A. Fiat, G.J. Woeginge (Eds.), Lecture Notes in Computer Science, 1442, 1998.

[134] K. Crammer, Y. Singer, Ultraconservative online algorithms for multiclass problems, J. Mach. Learn. Res. 3 (2003) 951–991.
[135] M. Charikar, L. O'Callaghan, R. Panigrahy, Better streaming algorithms for clustering problems, in: Proceedings of the Thirty-Fifth annual ACM Symposium on Theory of Computing, ACM, 2003, pp. 30–39.
[136] J. Cheng, Y. Ke, W. Ng, A survey on algorithms for mining frequent itemsets over data streams, Knowl. Inf. Syst. 16 (1) (2008) 1–27.
[137] H.D. Menéndez, D.F. Barrero, D. Camacho, A multi-objective genetic graph-based clustering algorithm with memory optimization, in: Proceedings of IEEE Congress on Evolutionary Computation (CEC), 2013, IEEE, 2013, pp. 3174–3181.
[138] W. Zhao, H. Ma, Q. He, Parallel k-means clustering based on mapreduce, in: Cloud Computing, Springer, 2009, pp. 674–679.
[139] C. Chu, S.K. Kim, Y.-A. Lin, Y. Yu, G. Bradski, A.Y. Ng, K. Olukotun, Map-reduce for machine learning on multicore, Adv. Neural Inf. Process. Syst. 19 (2007) 281.
[140] W.-Y. Chen, Y. Song, H. Bai, C.-J. Lin, E.Y. Chang, Parallel spectral clustering in distributed systems, IEEE Trans. Pattern Anal. Mach. Intell. 33 (3) (2011) 568–586.
[141] H.D. Menendez, D.F. Barrero, D. Camacho, A co-evolutionary multi-objective approach for a k-adaptive graph-based clustering algorithm, in: Proceedings of IEEE Congress on Evolutionary Computation (CEC), 2014, IEEE, 2014, pp. 2724–2731.
[142] H.D. Menendez, D. Camacho, Gany: a genetic spectral-based clustering algorithm for large data analysis, in: IEEE Congress on Evolutionary Computation (CEC), 2015, IEEE, 2015, pp. 640–647.
[143] A. Ng, M. Jordan, Y. Weiss, On Spectral Clustering: Analysis and an algorithm, in: T. Dietterich, S. Becker, Z. Ghahramani (Eds.), Advances in Neural Information Processing Systems, MIT Press, 2001, pp. 849–856. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.19.8100.
[144] F. Bach, M. Jordan, Learning spectral clustering, with application to speech separation, J. Mach. Learn. Res. 7 (2006) 1963–2001.URL http://jmlr.csail.mit.edu/papers/volume7/bach06b/bach06b.pdf
[145] R. Kumar, M. Wolenetz, B. Agarwalla, J. Shin, P. Hutto, A. Paul, U. Ramachandran, Dfuse: a framework for distributed data fusion, in: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, ACM, 2003, pp. 114–125.
[146] D. Keim, G. Andrienko, J.-D. Fekete, C. Görg, J. Kohlhammer, G. Melançon, Visual Analytics: Definition, Process, and Challenges, Springer, 2008.
[147] B. Cui, A.K. Tung, C. Zhang, Z. Zhao, Multiple feature fusion for social media applications, in: Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, ACM, 2010, pp. 435–446.
[148] E. Bakshy, I. Rosenn, C. Marlow, L. Adamic, The role of social networks in information diffusion, in: Proceedings of the 21st International Conference on World Wide Web, ACM, 2012, pp. 519–528.
[149] H. Becker, M. Naaman, L. Gravano, Event identification in social media., in: WebDB, 2009.
[150] H. Becker, M. Naaman, L. Gravano, Learning similarity metrics for event identification in social media, in: Proceedings of the Third ACM International Conference on Web Search and Data Mining, ACM, 2010, pp. 291–300.
[151] P.C. Wong, J. Thomas, Visual analytics, IEEE Comput. Gr. Appl. (5) (2004) 20–21.
[152] G. Andrienko, N. Andrienko, S. Wrobel, Visual analytics tools for analysis of movement data, ACM SIGKDD Explor. Newsl. 9 (2) (2007) 38–46.
[153] G. Andrienko, N. Andrienko, U. Demsar, D. Dransch, J. Dykes, S.I. Fabrikant, M. Jern, M.-J. Kraak, H. Schumann, C. Tominski, Space, time and visual analytics, Int. J. Geogr. Inf. Sci. 24 (10) (2010) 1577–1600.

Please cite this article as: G. Bello-Orgaz et al., Social big data: Recent achievements and new challenges, Information Fusion (2015), http://dx.doi.org/10.1016/j.inffus.2015.08.005