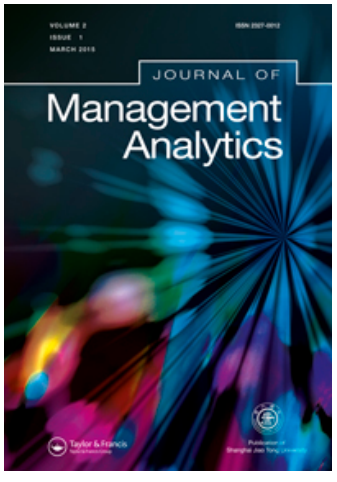

Journal of Management Analytics
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tjma20

Big data analytics and business analytics

Lian Duan[a] & Ye Xiong[a]
[a] College of Computing Sciences, New Jersey Institute of Technology, Newark, NJ 07102, USA
Published online: 19 Mar 2015.

Journal of Management Analytics, 2015
Vol. 2, No. 1, 1–21, http://dx.doi.org/10.1080/23270012.2015.1020891

Taylor & Francis
Taylor & Francis Group

Big data analytics and business analytics

Lian Duan* and Ye Xiong

College of Computing Sciences, New Jersey Institute of Technology, Newark, NJ 07102, USA

(*Received 18 July 2014; revised 9 December 2014; accepted 16 February 2015*)

Over the past few decades, with the development of automatic identification, data capture and storage technologies, people generate data much faster and collect data much bigger than ever before in business, science, engineering, education and other areas. Big data has emerged as an important area of study for both practitioners and researchers. It has huge impacts on data-related problems. In this paper, we identify the key issues related to big data analytics and then investigate its applications specifically related to business problems.

Keywords: big data analytics; business analytics; management analytics; business intelligence; marketing analytics

1. Introduction

Any research progress in business, science, engineering, education, sociology and other areas is either driven or supported by data. Although data alone are cheap and ubiquitous, what makes data a valuable asset is the useful information hidden inside them. Since there are many different types of useful hidden information which require different analytical techniques to find, these analytical techniques become an indispensable complement to data. Data analytics is the all-encompassing term for any analysis on any type of data. As such, data analytics can be widely applied to almost any area; it has abundant applications in business, and "business analytics" is considered the general term for any data analytics in business problems.

Due to the technical limitations before computers were invented, researchers had a limited capacity to collect, store and process data. Given any problem they were working on, they were limited to collecting and analyze the data that are considered to be directly related to the problem, with a small sample size. With recent technology revolutions, people generate data much faster and collect data much bigger than ever before. Big data is usually characterized by three Vs: volume, velocity and variety (Russom, 2011; Zhou, Chawla, Jin and Williams, 2014). Volume refers to the large amount of data. Data analytics benefits from the high volume of data as statistical reliability is better when the population size increases. In addition, a predictive method with hundreds of factors can predict better than the one with only a few input factors. Velocity refers to the rate at which data are generated. Online sales, mobile computing, smartphones and social networks have significantly increased the information flow. Velocity raises the significant challenges on real-time

*Corresponding author. Email: lian.duan@njit.edu

applications such as recommender systems, advertising bidding and anomaly detection. It requires related models to run in the background and make optimal or near-optimal decisions in real time. Variety refers to the mix of different data sources in different formats. Big data can be a transaction record in the traditional relational database, a text message from social networks or an image from a camera. When aggregating all the information to search for hidden useful patterns, it requires a middle layer to extract summarized information from heterogeneous data that can be directly used by different models.

Generally speaking, big data benefits our data analytics in two ways. First, statistical reliability is better when the population size increases from the high volume of data. Second, models can be potentially improved if they properly include more related factors because nothing can be isolated in nature, and objects can be related to each other in a very unintuitive way. Therefore, big data provides us the opportunity to tackle classical questions in a more panoramic way. However, big data brings not only more relevant signals, but also irrelevant noises. In order to gain big benefits from big data, first we need to overcome big challenges from big data on data capture, storage, transfer, sharing, search, analysis and visualization technologies. Otherwise, we will either lose data with signals or lose signals in the noise. In the following sections, the general challenges and progress in big data analytics will be discussed, and then the related business applications and their specific challenges in business analytics will be introduced.

2. Big data analytics

Big data is an all-encompassing term for any technique to handle the challenges of large data sets. These challenges include capture, storage, transfer, sharing, search, analysis and visualization. Among those challenges, capture, storage, transfer and sharing are related to system infrastructure, while search, analysis and visualization are related to data analytic methods. In addition, different types of data will raise new challenges to both system infrastructure and data analytic methods. In the following, system infrastructure, data analytic methods and challenges of different data type will be discussed, respectively.

2.1 System infrastructure

System infrastructure deals with data management, which includes any technique to capture, store, transfer and process data.

2.1.1 Data capture

Before computers were invented, all the data were manually collected and documented in papers. The amount of manually generated data is very limited. As people started to process data in computers, manually collecting data became the bottleneck, and more and more techniques for automatic data collection were developed. These techniques include sensor networks (Akyildiz, Su, Sankarasubramaniam and Cayirci, 2002), software logs (Moreta & Telea, 2007), social media (Kaplan & Haenlein, 2010), cameras (Baak, Müller, Bharaj, Seidel and Theobalt, 2013), radio-frequency identification (RFID) readers (Roberts, 2006), and global positioning systems (GPS; Kaplan & Hegarty, 2005). In addition, as more and more

people realize the power of big data, they are more motivated to invent new devices to collect the data, which are considered useful. For example, Google glasses (Wagner, 2013) are used to collect almost all-encompassing human activity data including what they see, what they say and where they go. Smartphones (Lee, Jin, & Park, 2011) are another type of devices with a bundle of cutting-edge technologies, such as a GPS receiver, 3G/4G radio interface, Wi-Fi, various sensors and a powerful computing system. In addition, three-dimensional (3D) cameras (Jansen, Temmermans, & Deklerck, 2007) are utilized to recognize facial expressions to infer cognitive statures, eye-tracking devices (Duchowski, 2002) are utilized to recognize the part of content that people pay attention to and various wearable sensor devices are used to monitor different real-time human statuses.

2.1.2 Storage

Data may be stored in different platforms and formats. Data warehouses (DW) consolidate information gathered from different sources and save summarized information for multidimensional queries. After data warehouses are constructed, data analysts can perform online analytical processing (OLAP; Berson & Smith, 1997), create business reports and visualize data. Comparing to Structured Query Language (SQL) that is used to create, store and update traditional relational databases, the NoSQL on nonrelational, distributed and horizontally scalable databases has emerged and has been used in real-time web applications (Cattell, 2011) due to the change of data capture and process in the big data. Whereas relational databases consist of related tables, records and fields, NoSQL databases contain nonstructural data in the form of key values, graphs or documents. Unlike traditional relational databases, NoSQL databases are categorized based on schema and storage strategies such as document stores, key-value stores, BigTable implementations and graph databases (Cattell, 2011; McCreary & Kelly, 2013).

When data is counted in terabytes or petabytes, traditional data computing models can no longer meet these new requirements. NoSQL solutions can handle large quantities of data and distribute it to a large number of data servers. There are two features of NoSQL solutions: (1) elasticity, the ability to add and remove new data nodes as desired without interrupting service; and (2) fault tolerance, the ability to make data safe and as available as possible, even when hardware fails (Carme & Jiménez, 2011). There are many NoSQL solutions available, such as Cassandra (key values), HBase, Hypertable, MongoDB (document stores), Voldemort, Membase and CouchDB, etc. (Carme & Jiménez, 2011; Silva, Beroud, Costa and Oliveira, 2014). These solutions can be used at different scales. For instance, Cassandra, HBase and Hypertable are used by major players with clusters of several hundred data nodes. MongoDB, Rlak, Voldemort and Membase are appropriate where a reasonable amount of data needs to be managed, typically less than a terabyte (Carme & Jiménez, 2011).

With the increasing focus on big data analytics, there is a trend to combine the traditional data warehouse with big data systems in a logical data warehousing architecture, which requires careful data classification to ensure that data can be analyzed quickly and productively. A consequence of the workload-centric approach is the shift from the single-platform enterprise data warehouse (EDW) to a physically distributed data warehouse environment (DWE; Russom, 2013), which consists of multiple platform types, ranging from the traditional warehouse to new platforms

like DW appliances, columnar database management systems (DBMSs), NoSQL DBMSs, MapReduce tools and Hadoop Distributed File System (HDFS; Russom, 2013; Sagiroglu & Sinanc, 2013).

2.1.3 Networking

Businesses often have multiple computing sites. Networks are used to connect these geographically dispersed sites and provide for the exchange of data and other resources. Networking provides many benefits of both centralized and distributed environments: more localized processing and control with shared data, processing power and equipment. The Internet connection might not be suitable to transfer big data among geographically distributed sites. One concern with big data is that the network should be fast enough to access this data in an efficient way, applying massive parallelism for both computation and storage (Merelli, Pérez-Sánchez, Gesing and D'Agostino, 2014). A fast network can not only allow data to be transferred quickly, but also improve performance of the shuffle phase of MapReduce application.

There are many different types of networks that serve as the telecommunications infrastructure for the Internet and the intranets and extranets of internetworked enterprises, such as wide-area networks (WANs) and local-area networks (LANs; Stamper & Case, 1994). Most WANs and LANs are interconnected using client/server, network computing, peer-to-peer and Internet networking technologies (Stamper & Case, 1994). Originally, LANs were installed to connect mainframe and minicomputers, and most LANs were implemented for two reasons: high-speed data transfer and resource sharing in a local area (Stamper & Case, 1994). WANs were mainly used to overcome distance, to overcome the computational limitations of a single computer, and to provide departmental computing. WAN optimization improves the efficiency of data transfer over a WAN.

While the computing hardware is integral to the system infrastructure for data storage and processing, the networking hardware is needed to connect the different systems to transfer data. One direction of network technology is to optimize the ability to effectively distribute processing resources such as hardware, software and data, as well as the use, management and control of these resources. There are also many technologies to be applied to limit bandwidth utilization. For instance, deduplication can reduce data transferred, and compression can shrink the size of the data (He, Li, & Zhang, 2010).

2.1.4 Distributed system parallel computing

Systems can be distributed in various ways. A distributed file system allows access to files from multiple hosts sharing via a computer network, which makes it possible for multiple users on multiple machines to share files and storage resources. When these distributed systems are used, processing is distributed. Distributed systems are becoming viable processing systems. Data can also be distributed over two or many nodes, such as on file servers, SQL servers, workstations and so on (Stamper & Case, 1994). Distributed data and distributed transactions may have a significant impact on the use of network resources (Stamper & Case, 1994). The key to a distributed system is making resource distribution transparent to the users.

Big data requires a different processing approach by using massive parallelism on readily available hardware. Parallel computing (Quinn, 1994) is widely adopted for

processing large-scale data in industry. Because of capacity needs, mainframe computers with multiple processors are needed to store and manage data warehouses, and support parallel processing (sometimes called multiprocessing) of different data at the same time and runing at high speeds. The implementation of parallel processing enables different methods to improve overall performance of extract, transform and load (ETL) processes when dealing with large volumes of data (Stonebraker, et al., 2010). Parallel processing, usually in the form of massively parallel processing (MPP; Metropolis, 1986), can enable high performance with data-intense operations (Russom, 2013).

With the ability to store data on large clusters of services, there is a need for tools that can process these data. Prior to the development of MapReduce (Schneider, 2012), the input data was usually large and the computations had to be distributed across hundreds or thousands of machines in order to finish certain tasks in a reasonable amount of time. Technologies like MPP database systems and MapReduce provide new methods of analyzing data that are complementary to the capabilities provided by traditional relational databases (Madden, 2012). The distributed parallel architecture distributes data across multiple processing units, and parallel processing units provide data much faster by improving processing speeds.

MapReduce (Condie, Conway, Alvaro, Hellerstein, Elmeleegy and Sears, 2010; Schneider, 2012) is a distributed processing framework that can give multithreaded parallelism to hand-coded routines written in various programming languages (Java, Pig, Japl or R). The MapReduce framework can parallelize computations across multiple cores of the same machine, and execute the routines using parallel processing to access the massive file and data repositories managed by an HDFS cluster. As a common scalable distributed computing model, MapReduce (Condie et al., 2010; Dean & Ghemawat, 2008; Fang, Pan, & Cui, 2012) realizes large-scale data processing for cloud computing and reduces data transfer as much as possible. The processing of complex parallel computing running on a large-scale cluster is abstracted to two functions: Map and Reduce (Sagiroglu & Sinanc, 2013). In the Map step, queries are split and distributed across parallel nodes and processed in parallel. In the Reduce step, the results of the Map processes are gathered and delivered (Shunnar & Raver, 2014). With MapReduce, complex big data problems can be broken down into small units of work and then processed in parallel (Schneider, 2012).

2.2 Data analytic methods

System infrastructure deals with storing, extracting, transforming and loading data. Once the information is made available, data analytic methods are used to search useful information. There are three types of data analytic methods: descriptive analytics, predictive analytics and prescriptive analytics. Descriptive analytic methods are used to understand what has happened in the data regarding its key indicators. Descriptive analytics is used to understand the reasons behind past success or failure. It is the first stage of data analytics and still the majority of the current business analytics applications. The next stage of data analytics is predictive analytics, which can be used to forecast future events based on past patterns. The final stage is prescriptive analytics, which uses optimization and other mathematical

models to identify the best actions and decisions. The performance of prescriptive analytics heavily relies on those of descriptive analytics and predictive analytics, as they determine the values of important parameters in prescriptive analytics.

2.2.1 Descriptive analytics

Descriptive analytics is used to provide a summary of descriptive statistics for a given sample, rather than an estimation of the ground true population value. It is considered a straightforward presentation of facts. The common descriptive statistics include mean, mode, median, standard deviation, range, stem, histogram and others. The results are usually displayed via graphics, charts and lines. For example, the company has its products, customers and sales information captured and stored in structured relationship databases. Then, periodically, this information can be extracted, transformed and stored into data warehouses. Then managers can investigate the correlation or the trend of all the possible attributes. For example, they can investigate popular products against gender and age. This type of data analysis helps to decide how to arrange shelves, recommend products and offer discounts. These simple types of descriptive statistics are related to business intelligence, and OLAP.

Besides the above simple descriptive statistics, we have more complicated methods to describe hidden patterns in big data. These complicated methods include associations, clustering and generative/graphical models.

2.2.1.1 Associations. The first research on searching associations in big data is frequent itemset mining. The Apriori algorithm (Ye & Chiang, 2006) was proposed to search sets of objects appearing together frequently in a data set. It makes use of an important downward-closed property to prune the exponential superset search space. If itemset is a subset of itemset, for any record that contains, it must also contains. However, the converse may or may not be true. Therefore, the number of records containing must be greater than or equal to that of. If the number of records containing one set is lower than the threshold, there is no need to check all its supersets which will turn out to be lower than the threshold at the end. Besides the Apriori algorithm, a lot of different frequent itemset search algorithms, such as frequent pattern (FP)-growth (Han, Pei, & Yin, 2000) and Equivalence CLAss Transformation (ECLAT) (Zaki, 2000), were proposed.

Although these algorithms can speed up the search procedure, a more important issue is its effectiveness. Using co-occurrence for associations has following drawbacks (Brin, Motwani, & Silverstein, 1997). For example, if both A and B happen in 90% of the records, A and B will co-occur in 81% of the records when they are independent from each other. As high co-occurrence does not necessarily mean associations, the concept of correlated itemsets has attracted more attention recently, and there have been several challenges on this topic. All related correlated itemset search algorithms can be generalized as a comparison between the actual co-occurrence and the expected co-occurrence under the assumption of independence. Although correlation has been studied in statistics for centuries, big data raises issues of both effectiveness and efficiency.

On the effectiveness side, the actual co-occurring probability and the expected co-occurring probability are estimated from data, which is affected by random noise

effects. For example, when we flip a fair coin 10 times, we have a chance to get heads 10 times in a row. When we get the sample of 10 heads in a row, the probability of getting heads through the traditional maximum likelihood estimation based on the observed data will be 100%, which is different from the fact of a fair coin. Similarly, when we estimate the correlation degree from big data, some seemingly correlated patterns might not be actually correlated. In order to handle the randomness effect, Dunning introduced a more statistically reliable method, likelihood ratio (Dunning, 1993), which outperforms other correlation methods in text mining. Bate and colleagues (Bate et al., 1998) proposed a Bayesian confidence propagation neural network (BCPNN) to generate a continuity correction version of Lift (Brin, Motwani and Silverstein, 1997). Compared with Lift, the variance of BCPNN is suppressed by continuity correction and generates more reliable results. In addition to randomness effects, different correlation search methods provide totally different results. In order to handle this chaos, researchers have tried to categorize methods. If methods are properly categorized, users only need to check the performance of the typical method in each category, instead of all the possible methods. Two very influential papers (Geng & Hamilton, 2006; Tan, Kumar, & Srivastava, 2004) tried to categorize methods according to their different property satisfaction. However, two methods can retrieve very different patterns even if they satisfy the same set of properties. Instead, one recent study (Tew, Giraud-Carrier, Tanner and Burton, 2014) tried to categorize methods directly according to the final result similarity. No matter how methods are categorized, two fundamental questions are still not answered: first, which method can achieve better results? Second, if there is a result difference between two methods, what is the difference? In order to answer these two fundamental questions, Duan and Street (2009) conducted analysis on correlation properties and upper bounds of different methods to understand the difference.

On the efficiency side, searching correlated itemsets raises two additional challenges compared to search frequent itemsets. First, correlated itemsets do not have a downward-closed property to prune the exponential superset search space. For this issue, Duan and Street (2009) proposed a fully correlated itemset framework, in which any two subsets are correlated. This framework can not only decouple correlation methods from the need for efficient search, but can also rule out the itemsets with irrelevant items. Second, the pruning procedure must start with pairs instead of single items. As the number of items and records in a data set increases, checking all of the possible pairs is still computationally expensive. The most significant progress in speeding up correlated pair search was made by researchers (Xiong, Shekhar, Tan and Kumar, 2006; Xiong, Zhou, Brodie and Ma, 2008). They made use of the upper bound of the Pearson correlation coefficient. The computation of the upper bound is much cheaper than the computation of the exact correlation. However, their work is only related to the Pearson correlation coefficient. Duan, Street and Liu (2013) extended this work to any correlation method, and proposed a token-ring algorithm to speed up the search.

2.2.1.2 Clustering. Clustering is the process of assigning records into groups. No matter how a clustering algorithm is designed, its goal can be summarized into two objectives: (1) objects in the same group are similar to each other, and (2) objects in different groups are different from each other. As different methods strike a different balance between two objectives, a lot of clustering algorithms are proposed, and they

can be categorized into five categories (Fahad et al., 2014): partitioning-based, hierarchical-based, density-based, grid-based and model-based.

The partitioning-based clustering uses an iterative relocation procedure to move objects from one group to another. The most well known partitioning-based clustering algorithm is K-means (MacQueen, 1967). It randomly selects objects as centers and assigns each object to its closest center. When all the objects are assigned to their closest center, it recalculates the center of each group and reassigns objects to the new closest centers. It stops when the centers will not change any more. Other clustering algorithms in the partitioning-based category include K-modes (Huang & Ng, 1999), PAM (Product of Arthur Meyerhoff) (Huang, 1998), CLARA (clustering large applications) (Ng & Han, 1994), CLARANS (Clustering Large Applications based on a RANdomized Search) (Ng & Han, 2002), and FCM (Fuzzy c-means clustering) (Bezdek, Ehrlich, & Full, 1984). Partitioning-based clustering holds the assumption that clusters are spherical-shaped, and has trouble detecting arbitrary shape clusters.

In partitioning-based clustering, each object must belong to exactly one group. However, objects can belong to multiple hierarchical structures. For example, one student can belong to the Department of Management Sciences, and he/she can also belong to the School of Business. A hierarchical-based clustering algorithm organizes objects in a way that can be represented in a tree structure. The hierarchical tree can be constructed either in an agglomerative (bottom-up) or divisive (top-down) fashion. An agglomerative clustering starts with one object for each cluster and recursively merges the two most similar clusters. A divisive clustering starts with all the objects in one cluster and recursively splits clusters. Besides hierarchical structures, hierarchical clustering is capable of finding clusters of arbitrary shapes. There are two major drawbacks for hierarchical-based clustering. First, generally speaking, the complexity of hierarchical clustering is not less than as it requires getting the result at different granularities. Second, the mistakes made in the current agglomerative or divisive step will propagate to all the following steps. BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) (Zhang, Ramakrishnan, & Livny, 1996), CURE (Clustering Using Re-presentatives) (Guha, Rastogi, & Shim, 1998), ROCK (Robust Clustering using Links) (Guha, Rastogi, & Shim, 2000) and Chameleon (clustering using inter connectivity) (Karypis, Han & Kumar, 1999) are the well-known algorithms in this category.

Although hierarchical-based clustering can find arbitrary shape clusters, it only considers cluster similarity and ignores cluster interconnectivity. Therefore, outliers are still assigned to the closest cluster. To discover outliers and clusters with arbitrary shapes, density-based clustering methods, which regard clusters as dense regions separated by sparse regions, have been developed since the 1990s. DBSCAN (Density-Based Spatial Clustering of Applications with Noise) (Ester, Kriegel, Sander and Xu, 1996) is the first density-based clustering algorithm which grows clusters according to a density-based connectivity analysis. DBSCAN uses a global density parameter to find clusters. However, different clusters might have very different densities, which require different local density parameters in the data space. To solve the issues of clusters with different densities, LDBSCAN (Local-Density Based Spatial Clustering of Applications with Noise) (Duan, Xu, Guo, Lee and Yan, 2007) combines the concepts of DBSCAN and LOF (Local Outlier Factor) (Breunig, Kriegel, Ng and Sander, 2000) to discover clusters and outliers in different local

density regions. Other popular density-based clustering methods include OPTICS (Ordering Points to Identify the Clustering Structure) (Ankerst, Breunig, Kriegel and Sander, 1999), and DENCLUE (Density-based Clustering) (Hinneburg & Gabriel, 2007).

All of the above clustering methods operate on objects in the data set, and the computational cost is mainly determined by the number of objects in the data set. In order to handle big data efficiently, grid-based clustering methods were proposed. For grid-based clustering methods, the data space is divided into grids and each grid is represented by the statistical values of the objects in it. Then, grid-based clustering methods perform the clustering on the grid instead of on each object, and the computational cost of a grid-based clustering method depends on the size of the grid, which is usually much less than the size of the objects. The typical grid-based clustering methods include Wave-Cluster (Sheikholeslami, Chatterjee, & Zhang 1998), STING (Statistical Information Grid) (Wang, Yang, & Muntz, 1997), CLIQUE (Clustering in Quest) (Zaki, Parthasarathy, Ogihara, and Li, 1997), and OptiGrid (Hinneburg & Keim, 1999).

For partitioning-based, hierarchical-based, density-based and grid-based clustering, clusters are discovered through some straightforward math functions. However, straightforward math functions do not handle statistical randomness very well, and make many type-1 errors. Therefore, model-based clustering methods utilize statistical approaches to generate robust clustering results. Model-based clustering methods are generally more accurate but more computationally expensive. The popular model-based clustering methods include EM (Expectation and Maximization) (Celeux & Govaert, 1992), COBWEB (an unsupervised conceptual clustering algorithm using the cobweb theory in economics) (Arifovic, 1994), CLASSIT (a conceptual clustering algorithm) (Gennari, Langley, & Fisher, 1989), and SOMs (Self-Organizing Maps) (Crane & Hewitson, 2003).

2.2.1.3 Generative Models. In statistics, a generative model is a model for randomly generating data according to its rules and parameters. Any observed real-life data are generated by its underlying mechanisms. When the predefined generative model is closer to the underlying mechanisms of generating the observed real-life data, the calculated statistical likelihood of getting the observed real-life data will be higher. Therefore, through maximizing the likelihood, related algorithms can discover the hidden structure and the fitting parameters in complex conditional distributions to describe unstructured information succinctly. Two general branches of generative models are Bayesian networks (Jensen, 1996) and Markov networks (Roller, 2004). Bayesian networks are a directed acyclic graph model represented by a factorization of joint conditional probability of all related variables. One popular method in Bayesian networks is called the Latent Dirichlet allocation (Blei, Ng, & Jordan, 2003) for understanding the hidden group information of text documents. A Markov network (Roller, 2004) is an undirected graph with many repeated states. In simpler Markov networks, the state is directly observable, and the state transition probabilities are the only parameters. In a hidden Markov network (Rabiner, 1989), the state is not directly observable, but the final data, dependent on the hidden state, are observable. A hidden Markov model is considered a generalization of a mixture model, where latent variables control the mixture component to be selected for each data and are related through a Markov process rather than independent

from each other. Such models are especially useful in temporal pattern recognition such as speech, handwriting, gestures and bioinformatics.

2.2.2 Predictive analytics

Predictive analytics uses statistical models to predict future behaviors based on the assumption that what has happened in the past will continue to happen in the future. The common predictive methods include regression, decision tree, Bayesian statistics, neural network, Support Vector Machine (SVM) and nearest neighbor.

2.2.2.1 Regression. Regression models have a long history in statistics and are widely used for many applications. The linear regression model (Kutner, Nachtsheim, & Neter, 2004) is the basic model. Parameters in a linear regression model are adjusted to minimize residuals. Ordinary least squares estimation results in the best linear unbiased estimates of parameters if Gauss–Markov assumptions are satisfied. When the output variable is discrete instead of continuous, some discrete choice models, including logistic regression (Hosmer & Lemeshow, 2004), multinomial logit (Hausman & McFadden, 1984) and probit regression (Rosett & Nelson, 1975), are proposed to solve these type of discrete output variables. When applying regression models to time-series data, we need to take the internal structure such as autocorrelation and seasonal variation into consideration. The two commonly used regression models are autoregressive models (Akaike, 1969) and moving average models (Said & Dickey, 1984). Regression models are very popular because they can predict any type of values and handle any type of data. In addition, parameters related to each variable can be used to explain the impact of each variable on the final output. If they are properly used and assumptions are satisfied, regression models can generate good results. However, it requires a good understanding of related applications to use regression models.

2.2.2.2 Decision tree. Decision tree algorithms recursively partition records into smaller subsets to each branch according to related attributes. The most important step for decision trees is how to select an attribute for splitting. Three popular methods are information gain (Quinlan, 1986), gain ratio (Quinlan, 1993) and gini index (Breiman, Friedman, Olshen and Stone, 1984). Decision trees are more naturally designed to handle nominal attributes and predict nominal values, but they can be applied to handle numeric attributes and predict numeric values. In order to handle numeric values, one must add an additional layer to discretize numeric values into different ranges (Kerber, 1992). In order to predict numeric values, decision trees need to integrate regression model to achieve this function (Quinlan, 1992).

2.2.2.3 Bayesian Statistics. Bayesian statistics use the Bayes theorem for prediction. The classical method in this category is naïve Bayesian (Wang, Garrity, Tiedje and Cole, 2007), which assumes attributes are independent from each other. Since the assumption of independence is usually violated in practical applications, Bayesian network is proposed to solve the problem. It illustrates the dependent relationships among attributes through a directed acyclic network. The most challenging part for Bayesian networks is how to construct the dependent relationships among attributes. This can be specified by humans or learned from data (Pearl, 2000). Generally

speaking, the learned topology from data is not very good. If the number of related attributes is not too big, the best way of constructing it is by experts who have a good understanding of the data. If the network can be correctly constructed, its performance can surpass many state-of-the-art prediction models.

2.2.2.4 Neural network. A neural network is composed of connected artificial neurons connected by weighted edges. The most popular one is called back-propagation (Bryson and Ho, 1975), which iteratively adjusts the edge weight according to the gap between the predicted value and the actual value. Neural networks are more naturally designed for numeric value prediction, but they can use the same logistic transformation function in regression models to predict binary value. If the neural network only contains an input and an output layer, the related neural network is exactly a linear regression model. If we add one hidden layer, the related neural network is a polynomial function. With traditional techniques, performance will not increase much if the model goes beyond two hidden layers. However, with the development of the current deep learning techniques (Arel, Rose, & Karnowski, 2010), different groups of data will be mapped to different layers to process, and the performance of a deep learning neural network is the current best classifier for big data. However, compared to regression, decision tree and Bayesian statistics, it is hard to explain how results are produced. If users only care about the prediction performance, neural network is a good choice.

2.2.2.5 Support vector machine. The original support vector machine (Vapnik, 1963) uses a linear separation hyperplane to separate two classes. However, some data are not linearly separable on their original space. With some non-linear transformation to map the original data to a higher dimension, the transformed data might be linearly separable. Therefore, people proposed different non-linear kernel functions to map the original data into a higher linearly separable dimension space. Three commonly used kernel functions include polynomial function (Klee & Minty, 1970), Gaussian radial basis function (Broomhead & Lowe, 1988) and sigmoid function (Yin, Goudriaan, Lantinga, Vos and Spiertz, 2003). However, non-linear transformation does not necessarily generate better results, and it depends on the characteristics of the data set. As SVM uses a linear separation hyperplane to separate two classes, it is less prone to overfitting compared to other methods. SVMs were originally designed for binary value prediction. However, with the proper transformation with its formula, an SVM can also be used to predict numeric values (Suykens & Vandewalle, 1999).

2.2.2.6 Nearest neighbor. The nearest neighbor classifier (Vapnik & Vapnik, 1998) predicts values using the target of similar historical records. Two important factors that impact the final performance include distance functions and feature selection. Euclidean distance is used when the absolute value is important, while cosine distance is used when the vector angle is more important. The nearest neighbor classifier doesn't need to build a model in the training phase, but is time consuming in the prediction phase.

2.2.3 Prescriptive analytics

Prescriptive analytics uses mathematical programming, heuristic search and simulation modeling to identify the optimal actions. For example, companies need to determine the optimal price for the highest profit. With regard to the property of objective functions and constraints, mathematical programming has many different types of problems. Linear programming (Dantzig, 1998; Hadley & Hadley, 1962) studies the case where the objective function and constraints are linear functions. Integer programming (Wolsey, 1998) studies linear programming questions where some variables are constrained to be integer values. Quadratic programming (Yuryevich & Wong, 1999) studies the quadratic objective function, but constraints are still linear. Nonlinear programming (Bertsekas, 2004) studies the general case where either the objective function or the constraints contain nonlinear parts. Stochastic programming (Kali & Wallace, 1994) studies the case where some parameters depend on random statistical variables. Combinatorial programming (Murty, 1976) is related to the problems where related variables are all discrete. Dynamic programming (Bellman, 1956) studies the questions that can be split into smaller subproblems. To solve the above different types of mathematical programming questions, different methods are proposed. Simplex algorithm (Klee & Minty, 1970) is the classical algorithm for linear programming. Some iterative methods are proposed to evaluate Hessians, gradients or the objective values to guide the current solution approaching the optimal solution. Those classical methods in this category include Quasi-Newton methods (Shanno, 1970), interior point methods (Potra & Wright, 2000), gradient descent (Burges et al., 2005) and ellipsoid methods (Shcrader, 1981). However, for some problems that are too complicated to get the optimal solution, some heuristic methods can provide approximate solutions. The classical heuristic methods include memetic algorithm (Moscato, Cotta, & Mendes, 2004), genetic algorithms (Goldberg, 2006), hill climbing (Tsamardinos, Brown, & Aliferis, 2006), particle swarm optimization (Kennedy, 2010), ant colony optimization (Dorigo & Birattari, 2010), simulated annealing (Van Laarhoven & Aarts, 1987) and tabu search (Glover & Laguna, 1999). For some questions where it is hard to generate their objective functions, as their final outputs are determined by the random interactions of many factors, simulations will be utilized to get the proximate solutions.

2.3 Challenges of unstructured data

As we mentioned earlier, big data contain more unstructured data than structured data. Those unstructured data include text data, graph data and time-series data. They raise challenges not only regarding data storage techniques but also regarding data analytics techniques. There are two types of efforts to handle the challenges. The first type is to transfer unstructured data into a structured format. Then all the classical methods can be used on the transferred data. The second type is to develop the new method to handle unstructured data.

With regard to the efforts in the first type, the most popular transformation in text data is the assumption of a bag of words (Wallach, 2006). It ignores the word sequence and emphasis on the word used. Then each document can be transformed into a structured record with each word as an attribute. The simplest way to generate the value for each attribute (word) is to count how many times the related word is

used in the document. Another popular method is called the tf-idf (term frequency-inverse document frequency) method (Ramos, 2003). In addition to how many times the related word is used in the current document, tf-idf also considers how often the related word is used in other documents. If a word is more frequently used in the current document and more rarely used in other documents, this word is more unique and important to the current document. In addition, LDA (Linear Discriminant Analysis) (Yang & Yang, 2003) is another technique to transform text into its related topic. Besides text data, there are other works to transfer graph data and time-series data. For example, Laplacian transformation (Widder, 1945) is used to map graph data into key eigenvector dimensions.

The second type is to develop new methods for unstructured data. For example, Okapi (Robertson, Walker, Jones, Hancock-Beaulieu and Gatford, 1995) is proposed to find related documents. PageRanking (Langville & Meyer, 2011) is proposed to find important nodes in graphs, and modularity-based methods (Cui, Zhang, & Rao, 2014) are proposed to find clusters in graphs. As this type of research is ongoing, there are many different models proposed each year.

3. Business analytics

As big data provide huge potentials, many companies are still looking for better ways to gain value from their data to be competitive in the market. MIT Sloan Management Review partnered with the IBM Institute for Business Value to conduct a survey of nearly 3000 executives, managers and analysts working across more than 30 industries and 100 countries (LaValle, Lesser, Shockley, Hopkins and Kruschwitz, 2013). One key finding is that top-performing organizations use analytics instead of intuition five times more than do lower performers. Six out of 10 respondents considered using data effectively to achieve competitive differentiation as the biggest challenge. In order to gain better analytics-driven insights, business analytics must be closely linked to business strategy and embedded into organizational processes so that action can be taken at the right time. It requires every effort on the way from manufacturing and new product design to credit approvals and customer relationship managements. As follows, we will briefly describe several business analytics applications in different areas.

According to the American Marketing Association (AMS)'s 2014 Marketing Analytics Survey (Udell, 2014), marketing analytics can be defined as the tools, technology and processes that enable marketers to measure and assess their marketing efforts' effectiveness, to analyze customer data, to measure and assess website traffic and engagement, and to mine insights from various big data sources. It can be helpful in better-targeted social-influencer marketing, crossing-selling, direct marketing, market and customer-base segmentation, market basket analysis, product mix analysis, discovery of associate products of affinity, recognition of sales and marketing opportunities, churn identification and analysis, anticipation of customer behavior (preferences, purchasing habits and customer affinity groups), prediction of customer lifetime value, understanding of market sentiment trending, trends and seasonality forecast, measurement of market signals and so on (Chase, 2014; Cokins, 2014; Giudici, 2005; Russom, 2011; Seng & Chen, 2010; Udell, 2014).

In terms of marketing analytics, there are a broad range of applications, which include: (1) market intelligence on consumers and communities. Chau and Xu (2012)

proposed a framework for gathering business intelligence by automatically collecting and analyzing blog content and bloggers' interaction networks. (2) Market intelligence in predicting customers' profiles. Park, Huh, Oh and Han (2012) developed a social network-based inference model for validating customer profile data. (3) Market intelligence on environmental scanning. Lau, Liao, Wong and Chiu (2012) designed a due diligence scorecard model that leverages collective web intelligence to enhance company mergers and acquisitions (M&A) decision making. (4) Market intelligence in anticipating customer behavior. For instance, telecommunications companies can use the model of the historical data of interaction with a particular company (Fayyad & Uthurusamy, 2002) to predict churn – the likelihood that a customer will drop service and subscribe with other competitors (Cokins, 2014). In addition, Amazon can recommend items that customers would likely be interested in purchasing based on customer shopping patterns, ratings and data of other customers with similar preferences (Cokins, 2014). Macy's conducts marketing campaigns and performs promotion analysis to understand the impact of past promotions and to predict changes in future promotions, for optimal results (Cokins, 2014).

With growing trends in globalization, supply chain management has become more important. Big data analytics can be used to identify the best suppliers, evaluate supplier performance based on cost and quality, increase supply chain visibility and integration, manage transportation, improve production yield, enable optimized inventory planning, input sourcing and scheduling of manufacturing processes (Banerjee, Bandyopadhyay, & Acharya, 2014; Downing, 2010; Seng & Chen, 2010; Trkman, McCormack, De Oliveira and Ladeira, 2010; Waller & Fawcett, 2013b), etc. For instance, Walmart can determine which items are popular in different geographic regions by monitoring social media and studying point-of-sale data for forecasting and inventory management (Cokins, 2014; Waller & Fawcett, 2013a). Chae and Olson (2013) proposed a framework of business analytics for supply chain analytics (SCA) as information technology (IT)-enabled, analytical dynamic capabilities, including data management capability (DMC), analytical supply chain process capability (APC) and supply chain performance management capability (SPC). First, DMC is the IT-enabled capability of data acquisition and transformation. The technologies, including enterprise resource planning (ERP), inter-organizational systems (IOS), RFID, and etc., can be used for supply chain data collection and storage. OLAP analysis and data warehouses can be used to transform transactional data into analytical data for supply chain planning and performance management. Also, APC is the analytical capability for four processes of a supply chain (planning, sourcing, making and delivering), which include simulation (e.g., data envelopment analysis) and data management tapping transactional data to automate the Statistical CAMELS Off-site Rating (SCOR) monitoring. Moreover, SPC as a dynamic capability is enabled by analytical techniques and tools, such as statistics and simulation, visualization tools to alert of potential problems through supply chain metrics or key performance indicators (KPIs) (Chae and Olson 2013).

As for finance, big data analytics can be mainly applied in financial fraud detection (FFD), which is categorized as bank fraud, insurance fraud, securities and commodities fraud and other related financial fraud (Bay, Kumaraswamy, Anderle, Kumar and Steier, 2006; Ngai, Hu, Wong, Chen and Sun, 2011). Other applications

include bad debt collection, credit rating, financial statement fraud prediction, automated decisions of real-time business processes like loan approval, systemic risk analysis and management in banking systems and so on (Kirkos, Spathis, & Manolopoulos, 2007; Ngai et al., 2011; Ravisankar, Ravi, Raghava Rao and Bose, 2011; Russom, 2011; Seng & Chen, 2010). For instance, Abbasi, Albrecht, Vance and Hansen (2012) used a design science approach to develop MetaFraud, a meta-learning framework for enhanced financial fraud detection. Kirkos, Spathis and Manolopoulos (2007) conducted a study to investigate the usefulness of decision trees, neural networks and Bayesian belief networks in the identification of fraudulent financial statements. Hu, Zhao, Hua and Wong (2012) developed a network approach to risk management (NARM) for modeling and analyzing systemic risk in banking systems.

In the accounting function, big data analytics can be helpful in providing sales and marketing with more reliable and accurate customer profitability information (Cokins, 2014). Management accountants become more proactive with increased adoption of analytics and performance improvement methods (Cokins, 2014). For instance, there is a shift away from reporting profitability by product and service line toward providing a more encompassing view of channel and customer profitability reporting. Accountants can include churn in calculating customer lifetime value (CLV) to prioritize which customers are most attractive to retain and grow (Cokins, 2014). Also, strategy maps are developed to report and monitor both financial and nonfinancial KPIs. In addition, traditional cost-center budgeting and cost variance control is changed toward driver-based rolling financial forecasts with the integration of predictive analytics into business processes (Cokins, 2014).

4. Conclusion

In order to achieve competitive differentiation and survive in the business world, companies must face the strategic and operational challenges in the era of big data. It requires them to (1) invest in data infrastructures to capture and store valuable data, (2) link business analytics to each business strategy and organizational process, and (3) keep up with the evolving techniques in big data and create an effective educational module for employees.

References

Abbasi, A., Albrecht, C., Vance, A. & Hansen, J. (2012). Metafraud: A meta-learning framework for detecting financial fraud. *MIS Quarterly*, *36*(4), 1293–1327.

Akaike, H. (1969). Fitting autoregressive models for prediction. *Annals of the Institute of Statistical Mathematics*, *21*(1), 243–247.

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y. & Cayirci, E. (2002). A survey on sensor networks. *Communications Magazine, IEEE*, *40*(8), 102–114.

Ankerst, M., Breunig, M. M., Kriegel, H.-P. & Sander, J. (1999). OPTICS: Ordering points to identify the clustering structure. In *ACM Sigmod Record (Special Interest Group on Management of Data)*, 49–60: Association for Computing Machinery (ACM).

Arel, I., Rose, D. C., & Karnowski, T. P. (2010). Deep machine learning – A new frontier in artificial intelligence research. *Computational Intelligence Magazine, IEEE*, *5*(4), 13–18.

Arifovic, J. (1994). Genetic algorithm learning and the cobweb model. *Journal of Economic Dynamics and Control*, *18*(1), 3–28.

Baak, A., Muller, M., Bharaj, G., Seidel, H. P. & Theobalt, C. (2011). A data-driven approach for real-time full body pose reconstruction from a depth camera. In *Proceedings of the IEEE International Conference on Computer Vision*, 1092–99. Barcelona.

Banerjee, A., Bandyopadhyay, T., & Acharya, P. (2014). Data analytics: Hyped up aspirations or true potential? *Vikalpa: The Journal for Decision Makers*, *38*(4), 1–11.

Bate, A., Lindquist, M., Edwards, I., Olsson, S., Orre, R., Lansner, A. & De Freitas, R. M. (1998). A Bayesian neural network method for adverse drug reaction signal generation. *European Journal of Clinical Pharmacology*, *54*(4), 315–321.

Bay, S., Kumaraswamy, K., Anderle, M. G., Kumar, R. & Steier, D. M. (2006). Large scale detection of irregularities in accounting data. In *ICDM '06: Proceedings of the Sixth International Conference on Data Mining*, 75–86. Washington, DC, USA: IEEE Computer Society.

Bellman, R. (1956). Dynamic programming and Lagrange multipliers. *Proceedings of the National Academy of Sciences of the United States of America*, *42*(10), 767.

Bertsekas, D. P. (1999). *Nonlinear programming*. Belmont, MA: Athena Scientific.

Berson, A., & Smith, S. J. (1997). *Data warehousing, data mining, and OLAP*. *McGraw-Hill series on data warehousing and data management*, New York: McGraw-Hill.

Bezdek, J. C., Ehrlich, R., & Full, W. (1984). FCM: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, *10*(2), 191–203.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *The Journal of Machine Learning Research*, *3*, 993–1022.

Breiman, L., Friedman, H., Olshen, R. A. & Stone, C. J. (1984). *Classification and regression trees*. The Wadsworth Statistics/Probability Series. Belmont, CA: Wadsworth International Group.

Breuniq, M. M., Kriegel, H. P., Sander, J. & Ng, R. T. (2000). LOF: Identifying density-based local outliers. In *ACM Sigmod Record (Special Interest Group on Management of Data)*, 93–104: Association for Computing Machinery (ACM).

Brin, S., Motwani, R., & Silverstein, C. (1997). Beyond market baskets: Generalizing association rules to correlations. In *ACM Sigmod Record (Special Interest Group on Management of Data)*, 265–76: Association for Computing Machinery (ACM).

Broomhead, D. S., & Lowe, D. (1988). *Radial basis functions, multi-variable functional interpolation and adaptive networks*. London: DTIC Document.

Bryson, A. E. (1975). *Applied optimal control: Optimization, estimation and control*. Washington DC: Hemisphere Publishing Corporation.

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N. & Hullender, G. (2005). Learning to rank using gradient descent. In *ICML 2005 - Proceedings of the 22nd International Conference on Machine Learning*, 89–96: Association for Computing Machinery (ACM).

Carme, J., & Jiménez, F. J. R. (2011). *Open source solutions for big data management*. Bezons, France: AtoS.

Cattell, R. (2011). *Scalable SQL and NoSQL data stores*. ACM Sigmod Record (Special Interest Group on Management of Data), *39*(4), 12–27.

Celeux, G., & Govaert, G. (1992). A classification EM algorithm for clustering and two stochastic versions. *Computational Statistics & Data Analysis*, *14*(3), 315–332.

Chae, B., & Olson, D. L. (2013). Business analytics for supply chain: A dynamic-capabilities framework. *International Journal of Information Technology & Decision Making*, *12*(1), 9–26.

Chase, J. C. W. (2014). Innovations in business forecasting: Predictive analytics. *Journal of Business Forecasting*, *33*(2), 26–32.

Chau, M., & Xu, J. (2012). Business intelligence in blogs: Understanding consumer interactions and communities. *MIS Quarterly*, *36*(4), 1189–1216.

Cokins, G. (2014). Mining the past to see the future. *Strategic Finance*, *96*(11), 23–30.

Condie, T., Conway, N., Alvaro, P., Hellerstein, J. M., Elmeleegy, K. & Sears, R. (2010). MapReduce online. In *Proceedings of the 7th USENIX conference on Networked systems design and implementation*, 21–21. San Jose, California: USENIX Association.

Crane, R. G., & Hewitson, B. C. (2003). Clustering and upscaling of station precipitation records to regional patterns using self-organizing maps (SOMs). *Climate Research*, *25*(2), 95–107.
Cui, Y., Zhang, B., & Rao, G. (2014). A new iterative modularity-based method for graph clustering on scalable networks. *Applied Mechanics & Materials*, 644–650, 2562.
Dantzig, G. B. (1998). *Linear programming and extensions*. Princeton, NJ: Princeton University Press.
Dean, J., & Ghemawat, S. (2008). MapReduce: Simplified data processing on large clusters. *Communications of the Association for Computing Machinery (ACM)*, *51*(1), 107–113.
Dorigo, M., & Birattari, M. (2010). Ant colony optimization. In *Encyclopedia of machine learning* (pp. 36–39). New York, NY, USA: Springer.
Downing, C. E. (2010). Is web-based supply chain integration right for your company? *Communications of the Association for Computing Machinery (ACM)*, *53*(5), 134–137.
Duan, L., Xu, L., Guo, F., Lee, J. & Yan, B. (2007). A local-density based spatial clustering algorithm with noise. *Information Systems*, *32*(7), 978–986.
Duan, L., Street, W. N., Liu, Y., Xu, S. & Wu, B. (2014). Selecting the right correlation measure for binary data. *ACM Transactions on Knowledge Discovery from Data*, *9*(2), 13:1.
Duan, L., & Street, W. N. (2009). *Finding maximal fully-correlated itemsets in large databases*. In *Proceedings - IEEE International Conference on Data Mining*, ICDM, 770–75.
Duan, L., Street, W. N., & Liu, Y. (2013). Speeding up correlation search for binary data. *Pattern Recognition Letters*, *34*(13), 1499–1507.
Duchowski, A. T. (2002). A breadth-first survey of eye-tracking applications. *Behavior Research Methods, Instruments, & Computers*, *34*(4), 455–470.
Dunning, T. (1993). Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, *19*(1), 61–74.
Ester, M., Kriegel, H.-P., Sander, J. & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, ed. E. Simoudis, J. Han, & U. M. Fayyad, 226–31. Portland: AAAI Press.
Fahad, A., Alshatri, N., Tari, Z., Alamri, A., Zomaya, A. Y., Khalil, I., … Bouras, A. (2014). *A survey of clustering algorithms for big data: Taxonomy & empirical analysis*. *Emerging Topics in Computing, IEEE Transactions on 2*, no 3: 267–79.
Fang, W., Pan, W., & Cui, Z. (2012). View of MapReduce: Programming model, methods, and its applications. *IETE Technical Review*, *29*(5), 380–387.
Fayyad, U., & Uthurusamy, R. (2002). Evolving data into mining solutions for insights. *Communications of the Association for Computing Machinery (ACM)*, *45*(8), 28–31.
Geng, L., & Hamilton, H. J. (2006). Interestingness measures for data mining: A survey. *ACM Computing Surveys (CSUR)*, *38*(3), 9.
Gennari, J. H., Langley, P., & Fisher, D. (1989). Models of incremental concept formation. *Artificial Intelligence*, *40*(1), 11–61.
Giudici, P. (2005). *Applied data mining: statistical methods for business and industry*. England: John Wiley & Sons.
Glover, F., & Laguna, M. (1999). *Tabu search*. Boston: Kluwer Academic Publishers.
Goldberg, D. E. (2006). *Genetic algorithms in search, optimization and machine learning*. Singapore: Pearson Education India.
Guha, S., Rastogi, R., & Shim, K. (1998). CURE: An efficient clustering algorithm for large databases. In *ACM Sigmod Record (Special Interest Group on Management of Data)*, 73–84: Association for Computing Machinery (ACM).
Guha, S., Rastogi, R., & Shim, K. (2000). ROCK: A robust clustering algorithm for categorical attributes. *Information Systems*, *25*(5), 345–366.
Hadley, G., & Hadley, G. (1962). *Linear programming*. vol. 4. Reading, MA: Addison–Wesley.
Han, J., Pei, J., & Yin, Y. (2000). Mining frequent patterns without candidate generation. In *ACM Sigmod Record (Special Interest Group on Management of Data)*, 1–12: Association for Computing Machinery (ACM).
Hausman, J., & McFadden, D. (1984). Specification tests for the multinomial logit model. *Econometrica* 52, no 5: 1219–1240. *Journal of the Econometric Society*.

He, Q., Li, Z., & Zhang, X. (2010). Data deduplication techniques. In *2010 International Conference on Future Information Technology and Management Engineering, FITME 2010*, 430–33. Changzhou IEEE.

Hinneburg, A., & Gabriel, H.-H. (2007). Denclue 2.0: Fast clustering based on kernel density estimation. In *Proceedings of the Seventh International Symposium on Intelligent Data Analysis*, ed. Berthold, MR, Shawe-Taylor, J., Lavrac, N., 70–80. Ljubljana, Slovenia: Springer.

Hinneburg, A., & Keim, D. A. (1999). *Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering*. In *Proceedings of the 25th International Conference on Very Large Data Bases*, ed. Tkinson, MP, Orlowska, M. E., Valduriez, P., Zdonik, S. B., Brodie, M. L., 506–17. Morgan Kaufmann, Edinburgh, UK.

Hosmer Jr, D. W., & Lemeshow, S. (2004). *Applied logistic regression*. New York: John Wiley & Sons.

Hu, D., Zhao, J. L., Hua, Z., & Wong, M. C. (2012). Network-based modeling and analysis of systemic risk in banking systems. *MIS Quarterly*, *36*(4), 1269–1291.

Huang, Z. (1998). Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, *2*(3), 283–304.

Huang, Z., & Ng, M. K. (1999). A fuzzy k-modes algorithm for clustering categorical data. *IEEE Transactions on Fuzzy Systems*, *7*(4), 446–452.

Jansen, B., Temmermans, F., & Deklerck, R. (2007). 3D human pose recognition for home monitoring of elderly. In *29th Annual International Conference of the IEEE Engineering in Medicine and Biology*, 4049–51: IEEE.

Jensen, F. V. (1996). *An introduction to Bayesian networks*. Vol. 210. London, UK: UCL Press.

Kali, P., & Wallace, S. W. (1994). *Stochastic programming*. New York: John Wiley & Sons.

Kaplan, A. M., & Haenlein, M. (2010). Users of the world, unite! The challenges and opportunities of social media. *Business Horizons*, *53*(1), 59–68.

Kaplan, E., & Hegarty, C. (2005). *Understanding GPS: Principles and applications*. Norwood, MA: Artech House.

Karypis, G., Han, E.-H., & Kumar, V. (1999). Chameleon: Hierarchical clustering using dynamic modeling. *Computer*, *32*(8), 68–75.

Kennedy, J. (2010). Particle swarm optimization. In *Encyclopedia of machine learning* (pp. 760–766). US: Springer.

Kerber, R. (1992). Chimerge: Discretization of numeric attributes. In *Proceedings of the tenth national conference on artificial intelligence*, 123–28. San Jose, CA: AAAI Press.

Kirkos, E., Spathis, C., & Manolopoulos, Y. (2007). Data mining techniques for the detection of fraudulent financial statements. *Expert Systems with Applications*, *32*(4), 995–1003.

Klee, V., & Minty, G. J. (1970). *How good is the simplex algorithm*. ed. Shisha, O, 159–79. New York: Inequalities Ill, Academic Press.

Kutner, M. H., Nachtsheim, C., & Neter, J. (2004). *Applied linear regression models*. Boston; New York: McGraw–Hill/Irwin.

Langville, A. N., & Meyer, C. D. (2011). *Google's PageRank and beyond: The science of search engine rankings*. Princeton, NJ: Princeton University Press.

Lau, R. Y., Liao, S. S., Wong, K.-F., & Chiu, D. K. (2012). Web 2.0 environmental scanning and adaptive decision support for business mergers and acquisitions. *MIS Quarterly*, *36*(4), 1239–1268.

Lavalle, S., Lesser, E., Shockley, R., Hopkins, M. S., & Kruschwitz, N. (2013). Big data, analytics and the path from insights to value. *MIT Sloan Management Review*, 52, no 2: 21–32.

Lee, J., Jin, A., & Park, Y. (2011). Next-generation smartphones technology trend analysis. In *KIPS Conference* vol. 18(1).

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, ed. Neyman, LMLCJ, 281–97. Oakland, CA, USA: University of California Press.

Madden, S. (2012). From databases to big data. *IEEE Internet Computing*, *16*(3), 4–6.

McCreary, D., & Kelly, A. (2103). *Making sense of NoSQL*. Greenwich, CT: Manning Publications.
Merelli, I., Pérez-Sánchez, H., Gesing, S., & D'agostino, D. (2014). Managing, analysing, and integrating big data in medical bioinformatics: Open problems and future perspectives. *BioMed Research International* 2014: 1–13.
Metropolis, N. (1986). Massively parallel processing. *Journal of Scientific Computing*, *1*(2), 115–116.
Moreta, S., & Telea, A. (2007). Multiscale visualization of dynamic software logs. In *Proceedings of the 9th Joint Eurographics - IEEE VGTC Symposium on Visualization (EuroVis 2007)*. ed. K. Museth, KM, T. Möller, T. Möller, A. Ynnerman & A. Ynnerman, 11–18. Norrköping, Sweden: Aire-la-Ville: Eurographics Association.
Moscato, P., Cotta, C., & Mendes, A. (2004). Memetic algorithms. In *New optimization techniques in engineering*, ed. Onwubolu, G. C., Babu, B. V., 53–85: Springer-Verlag, Berlin Heidelberg.
Murty, K. G. (1976). *Linear and combinatorial programming*. Vol. 7. New York: Wiley.
Ng, R. T., & Han, J. (1994). Efficient and effective clustering methods for spatial data mining. In *Proceedings of the 20th International Conference on Very Large Data Bases*, 144–55: Morgan Kaufmann Publishers Inc.
Ng, R. T., & Han, J. (2002). Clarans: A method for clustering objects for spatial data mining. *IEEE Transactions on Knowledge and Data Engineering*, *14*(5), 1003–1016.
Ngai, E. W. T., Hu, Y., Wong, Y. H., Chen, Y., & Sun, X. (2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decision Support Systems*, *50*(3), 559–569.
Park, S.-H., Huh, S.-Y., Oh, W., & Han, S. P. (2012). A social network-based inference model for validating customer profile data. *MIS Quarterly*, *36*(4), 1217–1237.
Pearl, J. (2000). *Causality: models, reasoning and inference*. vol. 29. Cambridge, U.K.: Cambridge University Press.
Potra, F. A., & Wright, S. J. (2000). Interior-point methods. *Journal of Computational and Applied Mathematics*, *124*(1), 281–302.
Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, *1*(1), 81–106.
Quinlan, J. R. (1992). Learning with continuous classes. In *5th Australian joint conference on artificial intelligence (AI 1992)*, 343–348: Singapore.
Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. San Mateo: Morgan Kaufmann Publishers.
Quinn, M. J. (1994). *Parallel computing: Theory and practice*. vol. 8. New York: McGraw–Hill.
Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, *77*(2), 257–286.
Ramos, J. (2003). Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning (ICML)*. Piscataway, NJ.
Ravisankar, P., Ravi, V., Raghava Rao, G., & Bose, I. (2011). Detection of financial statement fraud and feature selection using data mining techniques. *Decision Support Systems*, *50*(2), 491–500.
Roberts, C. M. (2006). Radio frequency identification (RFID). *Computers & Security*, *25*(1), 18–26.
Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M. M., & Gatford, M. (1995). *Okapi at TREC-3*. In *Proceedings of the Third Text REtrieval Conference(TREC-3), NIST Special Publication 500-225*, 109–26. Washington D.C.
Taskar, B., Guestrin, C., & Koller, D. (2004). Max-margin Markov networks. In *Advances in neural information processing systems (NIPS 16)*, ed. S Thrun, LSaBS, 25–32: MIT Press, Cambridge, MA.
Rosett, R. N., & Nelson, F. D. (1975). Estimation of the two-limit probit regression model. *Econometrica: Journal of the Econometric Society*, 43, no 1: 141–146.
Russom, P. (2011). *Big data analytics*. Renton, WA: The Data Warehousing Institute (TDWI) Best Practices Report.
Russom, P. (2013). *Managing big data*. Renton, WA: The Data Warehousing Institute (TDWI) Best Practices Report.

Said, S. E., & Dickey, D. A. (1984). Testing for unit roots in autoregressive-moving average models of unknown order. *Biometrika*, *71*(3), 599–607.
Sagiroglu, S., & Sinanc, D. (2013). Big data: A review. In *Proceedings of the International Conference on Collaboration Technologies and Systems (CTS '13)*, 42–47. San Diego, Calif, USA: The Institute of Electrical and Electronics Engineers (IEEE).
Schneider, R. D. (2012). *Hadoop For Dummies®, Special Edition*. Canada: John Wiley & Sons.
Schrader, R. (1981). *Ellipsoid methods*, ed. Korte, B, 265–311. North-Holland, Amsterdam: Modern Applied Mathematics - Optimization and Operations Research.
Seng, J.-L., & Chen, T. C. (2010). An analytic approach to select data mining for business decision. *Expert Systems with Applications*, *37*(12), 8042–8057.
Shanno, D. F. (1970). Conditioning of quasi-Newton methods for function minimization. *Mathematics of Computation*, *24*(111), 647–656.
Sheikholeslami, G., Chatterjee, S., & Zhang, A. (1998). Wavecluster: A multi-resolution clustering approach for very large spatial databases. In *Proceedings of the 24th International Conference on Very Large Data Bases*, 428–39. New York City, New York: Morgan Kaufmann Publishers Inc.
Shunnar, J., & Raver, D. (2014). *Big data executive overview*. Ohio, USA: ITG Software Engineering.
Silva, L. A. B., Beroud, L., Costa, C., & Oliveira, J. L. (2014). Medical imaging archiving: A comparison between several NoSQL solutions. In *2014 IEEE-EMBS International Conference on Biomedical and Health Informatics, BHI 2014*, 65–68. Valencia: IEEE Computer Society.
Stamper, D. A., & Case, T. (1994). *Business data communications*. fourth edition ed. Redwood City, CA: Benjamin/Cummings Publishing Co.
Stonebraker, M., Abadi, D., Dewitt, D. J., Madden, S., Paulson, E., Pavlo, A., & Rasin, A. (2010). MapReduce and parallel DBMSs: Friends or foes? *Communications of the ACM*, *53* (1), 64–71.
Suykens, J. A., & Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural Processing Letters*, *9*(3), 293–300.
Tan, P.-N., Kumar, V., & Srivastava, J. (2004). Selecting the right objective measure for association analysis. *Information Systems*, *29*(4), 293–313.
Tew, C., Giraud-Carrier, C., Tanner, K., & Burton, S. (2014). Behavior-based clustering and analysis of interestingness measures for association rule mining. *Data Mining and Knowledge Discovery*, *28*(4), 1004–1045.
Trkman, P., Mccormack, K., De Oliveira, M. P. V., & Ladeira, M.B. (2010). The impact of business analytics on supply chain performance. *Decision Support Systems*, *49*(3), 318–327.
Tsamardinos, I., Brown, L. E., & Aliferis, C. F. (2006). The max–min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, *65*(1), 31–78.
Udell, M. (2014). From insight to action. *Marketing Insights*, *26*(6), 38–43.
Van Laarhoven, P. J., & Aarts, E. H. (1987). *Simulated annealing*. Netherlands: Springer.
Vapnik, V. (1963). Pattern recognition using generalized portrait method. *Automation and Remote Control*, *24*, 774–780.
Vapnik, V. N., & Vapnik, V. (1998). *Statistical learning theory*. Vol. 1. New York: Wiley.
Wagner, M. S. (2013). Google glass: A preemptive look at privacy concerns. *Journal on Telecommunications & High Technology Law*, *11*(2), 477–492.
Wallach, H. M. (2006). Topic modeling: Beyond bag-of-words. In *Proceedings of the 23rd international conference on machine learning*, 977–984: ACM (Association for Computing Machinery).
Waller, M. A., & Fawcett, S. E. (2013a). Click here for a data scientist: Big data, predictive analytics, and theory development in the era of a maker movement supply chain. *Journal of Business Logistics*, *34*(4), 249–252.
Waller, M. A., & Fawcett, S. E. (2013b). Data science, predictive analytics, and big data: A revolution that will transform supply chain design and management. *Journal of Business Logistics*, *34*(2), 77–84.

Wang, Q., Garrity, G. M., Tiedje, J. M., & Cole, J. R. (2007). Naïve Bayesian classifier for rapid assignment of rRNA sequences into the new bacterial taxonomy. *Applied and Environmental Microbiology*, *73*(16), 5261–5267.

Wang, W., Yang, J., & Muntz, R. (1997). STING: A statistical information grid approach to spatial data mining. In *Proceedings of the 23rd International Conference on Very Large Data Bases*, 186–95: Morgan Kaufmann Publishers Inc.

Widder, D. V. (1945). What is the Laplace transform? *American Mathematical Monthly*, 52, no 8: 419–425.

Wolsey, L. A. (1998). *Integer programming*. vol. 42. New York: Wiley.

Xiong, H., Shekhar, S., Tan, P.-N., & Kumar, V. (2006). TAPER: A two-step approach for all-strong-pairs correlation query in large databases. *IEEE Transactions on Knowledge and Data Engineering*, *18*(4), 493–508.

Xiong, H., Zhou, W., Brodie, M., & Ma, S. (2008). Top-k φ correlation computation. *INFORMS Journal on Computing*, *20*(4), 539–552.

Yang, J., & Yang, J.-Y. (2003). Why can LDA be performed in PCA transformed space? *Pattern Recognition*, *36*(2), 563–566.

Ye, Y., & Chiang, C.-C. (2006). A parallel apriori algorithm for frequent itemsets mining. In *Proceedings of the Fourth International Conference on Software Engineering Research, Management and Applications*, 87–94: The Institute of Electrical and Electronics Engineers (IEEE).

Yin, X., Goudriaan, J., Lantinga, E. A., Vos, J., & Spiertz, H. J. (2003). A flexible sigmoid function of determinate growth. *Annals of Botany*, *91*(3), 361–371.

Yuryevich, J., & Wong, K. P. (1999). Evolutionary programming based optimal power flow algorithm. *IEEE Transactions on Power Systems*, *14*(4), 1245–1250.

Zaki, M. J. (2000). Scalable algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering*, *12*(3), 372–390.

Zaki, M. J., Parthasarathy, S., Ogihara, M., & Li, W. (1997). New algorithms for fast discovery of association rules. In *3rd International Conference on Knowledge Discovery and Data Mining (KDD)*, 283–86.

Zhang, T., Ramakrishnan, R., & Livny, M. (1996). BIRCH: An efficient data clustering method for very large databases. In *ACM Sigmod Record (Special Interest Group on Management of Data)*, 103–14. Montreal, Canada: Association for Computing Machinery (ACM).

Zhou, Z.-H., Chawla, N. V., Jin, Y., & Williams, G. J. (2014). Big data opportunities and challenges: Discussions from data analytics perspectives. *IEEE Computational Intelligence Magazine*, *9*(4), 62.