4

Advanced statistical analysis of large-scale Web-based data

Wienke Strathern, Raji Ghawi, Jürgen Pfeffer
Technical University of Munich

People leave millions of digital traces in the big data ecosystem. This ecosystem is a huge network with millions of daily personal transactions. And each of these transactions leaves traces that may be compiled into comprehensive information about individual and group behaviour (Lazer *et al* 2009, 2020). The capacity to collect huge amounts of data transforms the way people and organisations work and behave; hence, the market starts to react faster and increasingly anticipates traditional or other data sources. Data-driven computational economics capture changes in market, attitude and consumer behaviour over time and in real time. The quantitative techniques of machine learning have been applied to demonstrate a shift from a discretionary to a quantitative investment style (Kolanovic and Krishnamachari 2017). An increasing share of human interaction, communication and culture is recorded as digital text. Text is used as an input to economic research. Statistical methods and deep learning methods are applied to digital texts, as such data provides a rich repository of information about economic and social activity (Gentzkow *et al* 2019; Gentzkow and Shapiro 2010). More interesting for behavioural economics are the large-scale studies of social behaviour (Ruths and Pfeffer 2014). Research on big data analytics for economy and finance, especially quantitative finance, has been widely conducted (Ginsberg *et al* 2009; Engelberg and Parsons 2011; Goel *et al* 2010; Bańbura *et al* 2013; Cook *et al* 2011). A variety of studies have focused on social media data as a data source for finance and for decision makers. Bollen *et al* (2011) used Twitter data to predict

changes in stock market prices. Vermeer *et al* (2019) used machine learning and social media data from Facebook to better understand electronic word-of-mouth and its implications for brands. Ciulla *et al* (2012) used Twitter data to predict social events during elections to anticipate the voting outcome. Twitter data is used in financial market prediction (Mao *et al* 2011) and commonly used as a news source in mainstream media (Moon and Hadley 2014).

We refer to Laney (2001) and define big data by the following features.

- **Volume:** the size of collected and stored files, tables, numbers, etc.
- **Velocity:** the speed of transmitted data in real time or near real time.
- **Variety:** the number of different formats, ie, structured (structured query language (SQL) tables, comma-separated values (CSV) files), semi-structured (JavaScript object notation (JSON) or hypertext markup language (HTML)) or unstructured (social media post, video message).

According to Kolanovic and Krishnamachari (2017), we can differentiate big data sources as follows.

- **Data generated by individuals,** such as social media posts, product reviews and Internet search trends. Mostly recorded through textual mediums, such data is often unstructured and distributed across multiple platforms. We can further classify this data into data from social media, specialised sites such as business-reviewing websites (eg, e-commerce groups), Web searches and personalised data, data from personal inboxes, etc.
- **Data generated by business processes,** such as company exhaust data, commercial transactions, credit card data and order book data. This data refers to data produced or collected by corporations and public entities. An important subcategory is transaction records such as credit card data. Corporate data can be a byproduct or "exhaust" of corporate record-keeping such as banking records, supermarket scanner data and supply chain data. Data generated in this way is often highly structured (compared with individual data) and can act as a leading indicator for business metrics, which tend to be reported at a

significantly lower frequency. Business-processed data can, for example, arise from public agencies.

- **Data generated by sensors,** such as satellite images, foot and car traffic and ship location, is data collected mechanically through sensors embedded in various devices. The data generated is either structured or unstructured and is often much larger in size than either individual or process-generated data streams. An example would be satellite imaging used to monitor economic activities (construction, shipping, commodity production, etc). Geolocation data can be used to track foot traffic in retail stores (smartphone data, if allowed) or ships in ports. Other examples of sensors include cameras fixed at a location of interest and weather and pollution sensors. The practice of embedding microprocessors and networking technology into all personal and commercial electronic devices – the concept of the Internet of Things (IoT) – is the next step for sensor-generated data.

There have been three important trends that enabled big data analytics (Kolanovic and Krishnamachari 2017). The availability of different data sources and a possible application of quantitative strategies can be a huge informational advantage in complex systems. An exponential increase in the data available and an increase in computing power and data storage capacity at reduced cost (cloud computing) increases access to data. There have been increasingly fast developments in the advancement of machine learning methods to analyse complex data sets. One of the biggest advantages is the ability to collect large quantities of data and analyse it in real time. Simultaneously, there has been significant growth in the methodological advancements in pattern recognition and function approximation.

Machine learning methods are often extensions of well-known statistical methods; supervised learning methods attempt to establish a relationship between two data sets and use one data set to predict the other. The underlying concepts of machine learning methods are often as simple as regression models, improved to contain changing market regimes, data outliers and correlated variables. Unsupervised machine learning methods try to understand the underlying structure of data and identify the main patterns. Supervised machine learning methods try to find a rule that can be used to predict a variable (Kolanovic and Krishnamachari 2017). However, skills, infrastructure, market intuition and experiences in complex economic

and financial systems are required in order to handle and evaluate big data and gain insights about the economic drivers behind the data.

In this chapter we showcase the machine learning methods for analysing large-scale data and debate the strengths and weaknesses of these methods. First, we discuss in detail the machine learning methods used to work with big data. Then we discuss an application of these methods with representative data to illustrate advanced statistical analysis.

MACHINE LEARNING METHODS

New methods are needed to tackle the complexity and volume of new data sets. For instance, the automated analysis of unstructured data such as images and social media is not possible with standard analytical tools (eg, spreadsheets). Machine learning methods can be used to analyse big data, as well as to more efficiently analyse traditional data sets. Artificial intelligence (AI) is a broader scheme enabling machines to tackle complex problems in complex systems. In many cases, when a computing problem needs to be solved we often write a program that manually specifies a series of programming steps which need to be run to solve that particular problem. We can instruct a computer to perform certain operations based on a fixed set of rules. For instance, in finance, we can instruct a computer to sell an asset if the asset price drops by a certain amount (stop loss). This works well for a vast number of computing problems. However, not all problems lend themselves to being solved effectively by writing a handcrafted program or a set of rules. Image classification, speech recognition (converting human speech to text) and authorship identification (inferring the author of a document) are examples of tasks that cannot be accurately carried out by writing down a set of rules in a programming language.

Given how complex and delicate those problems are, writing by hand a set of program rules that could solve them would be a tremendous task. Even then, such a hand-crafted system would still likely be inflexible and not very robust at recognising different types of objects (images, speech or text). Moreover, if the system is required to be customised such that it could recognise new objects or other features that had not been encoded in the existing rules, we would have

to write a whole new set of rules, which would be a prohibitively difficult task.

Giving a machine a large number of complex rules for automating tasks is referred to as "symbolic AI". With this "symbolic AI", the machine will freeze the first time it encounters a situation that does not exactly match a set of pre-programmed rules. Machine learning, on the other hand, gives us technology that allows us to automatically learn these complex rules efficiently from labelled examples, called training data, in a way that is much more accurate and flexible than attempting to program all the rules by hand. The goal of machine learning is to enable computers to learn from their experience in certain tasks, and to improve their performance automatically as they gain more experience. This experience can take the form of data in a lot of different formats or situations, such as the labelled examples that are used to train the system's initial structure.

In machine learning, the computer is given an input (set of variables and data sets) and an output that is a consequence of the input variables. The machine then finds or "learns" a rule that links the input and output. The success of this learning task can be tested with respect to its ability to gain useful knowledge of the relationship between the variables and predict outcomes in as yet unseen situations. That is, since it is unlikely any future examples would match what was in the training set exactly, the primary goal of effective machine learning algorithms is to be able to generalise: to correctly predict or recognise new objects that were not seen during training.

Machine learning is a part of the broader fields of computer science and statistics. Statistical methods give machine learning ways to infer conclusions from data (learn from data) and also to estimate how reliable those conclusions are. Computer science methods, on the other hand, give machine learning algorithms the computing power (including effective large-scale computational architectures and algorithms for capturing, manipulating, indexing, combining and performing predictions with data) to solve problems.

Machine learning tasks can be categorised into two main types. The first type is known as "supervised learning", where the goal is to predict some output variable (a predefined label) associated with each input item. The second type deals with data that has no predefined labels, hence the name "unsupervised learning". Here the goal is to find structure in the data by finding some commonality in

its features (Kolanovic and Krishnamachari 2017; Domingos 2012). When we apply machine learning, using either a supervised or an unsupervised approach, a typical workflow consists of three basic components: representation, evaluation and optimisation.

- **Representation.** The first step in solving a problem with machine learning is to figure out how to represent the learning problem in terms of something the computer can handle. This serves two purposes.

    1. The representation of the data (eg, what features to use): we need to convert each input object, which we often call a sample, into a set of features that describe the object.
    2. The choice of the learning algorithm to apply: we need to pick a learning model, typically the type of classifier that you want the system to learn.

- **Evaluation.** The second step is to decide on an evaluation method that provides some type of quality or accuracy score for the predictions or the output of the machine learning algorithm. An evaluation function (or scoring function) is needed to assess and compare the effectiveness of different algorithms (models), and hence to distinguish good ones from bad ones. For example, a good classifier will have a high accuracy, making a high percentage of predictions matching the correct "true" label.

- **Optimisation.** The third step is to search all possible models for the optimal model that gives the best evaluation outcome for that particular problem, ie, the highest-scoring model. This involves an iterative process, where we make an initial guess about what some good features are for solving the problem, and which classifier might be appropriate. We then train the system using training data, produce an evaluation and see how well the classifier works. Then, based on that evaluation, we refine the model and repeat the process.

Typically, data instances are represented as vectors. The components of vectors correspond to the features of the data instances. When a feature is binary (Boolean) or numeric, its values can be used

directly in the corresponding vector component. Non-numeric features need some sort of transformation to be used in vector components. Ordinal features comprise a finite set of discrete values with a ranked ordering between values, such as size (small, medium, large, etc). Ordinal features can be transformed using an integer encoding, for example, "small" = 1, "medium" = 2 and "large" = 3. Categorical features comprise a finite set of discrete values with no relationship between values, such as colour (red, green, blue, etc). Categorical features can be transformed using a technique called one-hot encoding, for example, "red" = $(1, 0, 0)$, "green" = $(0, 1, 0)$ and "blue" = $(0, 0, 1)$.

A common way to represent text in machine learning is the "vector space" model, where each document is represented as a vector whose elements correspond to words in the whole document collection (vocabulary). The values in the vector can be binary (1 for the presence of the word and 0 for the absence of the word). Alternatively, it is common to use the within-document term frequency (TF), which is the number of occurrences of the given term in the given document. Moreover, TF is typically combined with the inverse document frequency (IDF), which is a measure of how common or rare a word is across all documents. The TF–IDF scheme is the most popular scheme for text representation. Using this representation, the similarity between two text objects (sentences, paragraphs or documents) can be assessed using the dot product of the vectors representing them or, more commonly, using cosine similarity (Huang 2008).

When data instances are properly represented as vectors, they are ready to be used in machine learning algorithms. Many machine learning algorithms (in particular, clustering algorithms and the k-nearest neighbours (k-NN) classification algorithm) need some measure of distance (or similarity) between data instances.

Let two data instances be represented by two n-dimensional vectors, $\boldsymbol{A}$ and $\boldsymbol{B}$, with $a_1, a_2, \ldots, a_n$ the components of vector $\boldsymbol{A}$ that represent the values of the features (raw or transformed) of data instance A (such as the occurrence of words in text A), and $b_1, b_2, \ldots, b_n$ the components of vector $\boldsymbol{B}$. Distance measures that are commonly used in machine learning algorithms include the following.

- The Euclidean distance, which represents the shortest distance between two points

$$D(A,B) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2}$$

- The Manhattan distance, which is the sum of the absolute differences between points across all the dimensions

$$D(A,B) = \sum_{i=1}^{n}|a_i - b_i|$$

- The Minkowski distance, which is a generalised form of the Euclidean and Manhattan distances; a Minkowski distance of order p between two points is defined as

$$D(A,B) = \left(\sum_{i=1}^{n}|a_i - b_i|^p\right)^{1/p}$$

  If $p = 1$, the Minkowski distance reduces to the Manhattan distance. If $p = 2$, the Minkowski distance reduces to the Euclidean distance.

- Cosine similarity, which is a measure of similarity between two vectors defined to equal the cosine of the angle between them

$$\cos(A,B) = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2}\sqrt{\sum_{i=1}^{n} b_i^2}}$$

  The resulting similarity ranges from -1 to $+1$, where -1 means exactly the opposite, and $+1$ means exactly the same. The cosine distance is the complement of the cosine similarity, ie, $D_C(A,B) = 1 - S_C(A,B)$, where D_C is the cosine distance and S_C is the cosine similarity.

SUPERVISED LEARNING

The first type of machine learning methods is known as supervised learning. The goal is to predict some output variable that is associated with each input item. The output variable could be a category (with a finite number of possibilities), such as a spam or not-spam email, a fraudulent or not-fraudulent prediction for a credit card

transaction or the topic of a document (eg, sport, politics or the economy). In this case, we call this a classification problem within supervised learning, and the function that we learn is called the classifier. Conversely, if the output variable we want to predict is not a category, but a real-valued number such as the price of a house, then we call this a regression problem, and we are learning something called a regression function.

Regression

Regression is one of the most widely used machine learning tools. It allows us to make predictions from data by learning the relationship between features of the data and some observed, continuous-valued response. Regression is used in a very large number of applications, ranging from predicting stock prices to understanding gene regulatory networks. Regression aims to estimate the relationships between a dependent variable (outcome variable, or target) and a group of independent variables (predictors, or features). The most common type of regression is linear regression, where the relationship is typically in the form of a line (or a linear combination) that best approximates all the individual data points. On the other hand, in non-linear regression, observational data is modelled by a function that is a non-linear combination of the model parameters.

Linear regression

A linear model expresses the target output value in terms of a sum of weighted input variables that predict the target value given an input data instance.

Let $(\boldsymbol{x}, y)$ be a data instance, where $\boldsymbol{x} = (x_0, x_1, \ldots, x_n)$ is a vector of features representing the input data instance and y is the target output value. The predicted output will be of the form $\hat{y} = \hat{w}_0 x_0 + \hat{w}_1 x_1 + \cdots + \hat{w}_n x_n + \hat{b}$, where $\hat{\boldsymbol{w}} = (\hat{w}_0, \hat{w}_1, \ldots, \hat{w}_n)$ is a vector of feature weights (model coefficients) and $\hat{b}$ is a constant bias term (intercept). The goal of the linear regression algorithm is to estimate the model parameters $\hat{\boldsymbol{w}}$ and $\hat{b}$.

A common method to estimate the model parameters is the ordinary least squares technique. The aim of this technique is to minimise the difference (the mean squared error) between the predicted value and the actual value of the target variable. Formally, the objective is to minimise the sum of $(y - \hat{y})^2$ over all the data instances in a data set.

There are several extensions to the ordinary least squares technique, such as the least absolute shrinkage and selection operator (Lasso) (Tibshirani 1996) and ridge regression (Hoerl and Kennard 2000), which aim to control the model complexity.

Non-linear regression

In non-linear regression, the relationship between the feature vector $\boldsymbol{x}$ of a data instance and the target output y takes the form of an arbitrary function, $y = f(\boldsymbol{x}, \boldsymbol{\beta})$. The function f is non-linear in the components of the vector of parameters $\boldsymbol{\beta}$. Examples of non-linear functions include exponential functions, logarithmic functions, trigonometric functions and power functions.

Other regression methods that are non-linear include polynomial regression and k-NN regression. Polynomial regression is a form of regression in which the relationship between the input $\boldsymbol{x}$ and the output y is modelled as an nth-degree polynomial in $\boldsymbol{x}$. k-nearest neighbours (k-NN) is a nonparametric method used for classification and regression. In k-NN regression, the output is the property value for the object. This value is the average of the values of the k nearest neighbours (Altman 1992).

Classification

The goal of the classification methods in the supervised learning group is to classify observations into distinct categories, ie, the target value is a discrete class value. Furthermore, classification can be binary or multi-class. In binary classification, the target value can be 0 (negative class) or 1 (positive class), eg, email classification as spam or not-spam. On the other hand, in multi-class classification, the target value is one of a set of discrete values, eg, labelling the topic of a document based on its text.

Logistic regression

Logistic regression is a classification algorithm that produces the output as a binary decision, eg, "spam" or "not-spam". It is the simplest adaptation of linear regression to a specific case when the output variable is binary (0 or 1). Logistic regression is derived via a simple change to ordinary linear regression. We first form a linear combination of the input variables (as in conventional regression) and then apply a function that maps this number to a value between 0 and 1. The mapping function is called the logistic function.

k-nearest neighbours

The *k*-NN algorithm is a non-parametric method, proposed by Thomas Cover, used for classification and regression (Cover and Hart 1967). In both cases, the input consists of the *k* closest training examples in the feature space. The output depends on whether *k*-NN is used for classification or regression. In *k*-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbours, with the object being assigned to the class that is most common among its *k* nearest neighbours (*k* is a positive integer, typically small).

Support-vector machines

Support-vector machines (SVMs) are one of the most popular classification algorithms. Their popularity stems from their ease of use and calibration. A support-vector machine constructs a hyperplane or set of hyperplanes in a high-dimensional space. The goal of an SVM is to separate the data hyperplane into non-overlapping parts. Intuitively, a good separation is achieved by the hyperplane that has the greatest distance to the nearest training-data point of any class (the so-called functional margin), since in general the larger the margin, the lower the generalisation error of the classifier (Hastie *et al* 2009).

Random forests

A random forest is a meta classifier that fits a number of decision-tree classifiers. A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences. A random forest classifier works by constructing a multitude of decision trees at training time on various subsamples of the data set. It uses averaging to improve the predictive accuracy and to control over-fitting. Thus, a random forest outputs the class that is the mode of the classes (the class that appears most often) of the individual trees (Ho 1995).

Neural networks

Neural networks are complex models that try to mimic the way the human brain develops classification rules. A neural network consists of many different layers of neurons, with each layer receiving inputs from previous layers and passing outputs to further layers. A neural network is composed of artificial neurons (conceptually

derived from biological neurons). Each artificial neuron has inputs and produces a single output that can be sent to multiple other neurons. The inputs can be the feature values of a sample of data, or they can be the outputs of other neurons. The outputs of the final output neurons of the neural net accomplish the classification task.

Evaluation

Evaluation metrics for regression

As regression tasks seek to predict a continuous-valued response, the output is some numeric value. Evaluating the performance of a regression algorithm is hence based on assessing how the predicted values deviate from the actual values of the target variable. Various metrics are typically used to evaluate the results of the prediction.

- Mean squared error (MSE) is the average of the squared difference between the target value and the value predicted by the regression model

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- Root mean squared error (RMSE) is the square root of the averaged squared difference between the target value and the value predicted by the model

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

- Mean absolute error (MAE) is the average of the absolute difference between the target value and the value predicted by the model

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

- The R^2 or coefficient of determination is the proportion of the variance in the dependent variable that is predictable from the independent variables

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

where

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

Table 4.1 Confusion matrix.

	Actual	
Predicted	**Positive (1)**	**Negative (0)**
Positive (1)	True positives (TP)	False positives (FP)
Negative (0)	False negatives (FN)	True negatives (TN)

is the mean of the observed data.

In the best case, the predicted values would exactly match the observed values; in this case, the MSE, RMSE and MAE results are 0 and R^2 result is 1.

Evaluation metrics for classification

Classification tasks seek to predict a discrete class value for a target variable. Evaluating the performance of a classification algorithm is hence based on assessing the extent to which the predicted classes match the actual ones over all the instances in the data set.

Classification evaluation metrics are mainly based on a confusion matrix, which consists of two dimensions (actual and predicted) and sets of classes in both dimensions. In this matrix, the columns represent actual classifications, and the rows represent predicted ones.

Table 4.1 shows how the confusion matrix looks for a binary classification, where there are two classes, labelled positive and negative. Several terms are associated with the confusion matrix. True positives (TP) are the cases where the actual and predicted classes are both positive. True negatives (TN) are the cases where the actual and predicted classes are both negative. False positives (FP) are the cases where the actual class is negative while the predicted one is positive. Conversely, false negatives (FN) are the cases where the actual class is positive while the predicted one is negative. True positives and true negatives are the cases that are correctly classified, whereas false positives and false negatives are those that are predicted incorrectly by the model. Using these terms, the evaluation metrics for classification are defined as follows:

- Accuracy is the fraction of the total number of cases that are correctly classified

$$A = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}}$$

- Precision is the fraction of all predicted positive cases that are correctly classified as positive

$$P = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

- Recall is the fraction of all actual positive cases that are correctly classified as positive

$$R = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

- The F_1-score is the harmonic mean of recall and precision

$$F = \frac{2PR}{P + R}$$

For multi-class classification, the precision, recall and F_1-measure are calculated for each class. To combine the per-class scores into a single number, three methods are typically used.

- **Micro averaging:** first the values of TP, FN, TN and FP are summed over all instances, and then the performance measures are calculated using the accumulated values.
- **Macro averaging:** a simple arithmetic mean of per-class scores.
- **Weighted averaging:** similar to macro averaging, but the contribution of each class is weighted by the number of samples from that class.

UNSUPERVISED LEARNING

We have seen that supervised machine learning algorithms and techniques aim to develop models where the data has (previously known) labels, ie, the data has some target variables with specific values that are used to train the models (Bousquet *et al* 2004). However, when dealing with real-world problems, most of the time the data will not come with predefined labels. Therefore, there is a need to develop machine learning models that can classify data autonomously by finding commonality in the features. The main goal of unsupervised learning is to study the intrinsic structure of the data. The major applications of unsupervised learning include

- segmenting data sets by some shared attributes,
- detecting anomalies that do not fit into any group, and
- simplifying data sets by aggregating variables with similar attributes.

Clustering

The objective of clustering analysis is to find different groups within the data elements. To do this, clustering algorithms find a structure in the data so that elements of the same cluster (or group) are more similar to each other. Clustering algorithms have a wide range of applications, and are quite useful to solve real-world problems such as anomaly detection, recommending systems, document grouping or finding customers with common interests based on their purchases. Some of the most common clustering algorithms are the *K*-means, hierarchical clustering (agglomerative or divisive) and density-based spatial clustering of applications with noise (DBSCAN).

K-means

K-means clustering is a data mining technique used to group objects or data sets into clusters based on their similarities. The similarity is the total distance from the values in each cluster to the centroid, where each centroid has an average cluster value. The shorter the distance, the greater the similarity, and vice versa.

K-means clustering algorithm works as follows:

1. determine the number of clusters *K*;
2. choose *K* random points from the data as centroids;
3. set all the data points to the closest cluster centroid;
4. recalculate the centroid of newly formed clusters;
5. repeat until convergence, ie, the data points stop changing clusters.

Hierarchical clustering

Hierarchical clustering methods seek to build a hierarchy of clusters, using either an agglomerative strategy or a divisive strategy (Rokach and Maimon 2005). Agglomerative clustering is a bottom-up approach, where each observation starts in its own cluster, and

pairs of clusters are merged as we move up the hierarchy. Divisive clustering is a top-down approach, where all observations start in one cluster, and splits are performed recursively as we move down the hierarchy. The results of hierarchical clustering are usually presented in a dendrogram.

DBSCAN

Density-based spatial clustering of applications with noise is a density-based clustering that is able to find arbitrarily shaped clusters and clusters with noise, ie, outliers (Ester *et al* 1996). Given a set of points in some space, this algorithm groups together points that are closely packed together (points with many nearby neighbours), marking as outliers points that lie alone in low-density regions (whose nearest neighbours are farther away).

Association rule mining

Association rule mining (Agrawal *et al* 1993; Agrawal and Srikant 1994; Larose and Larose 2014) is used for discovering interesting relationships between variables in a large database. Association rules were first introduced for discovering regularities between products in large-scale transaction data recorded by point-of-sale systems in supermarkets (Agrawal *et al* 1993). Such rules can be used in supermarket basket analysis as the basis for decisions about marketing activities such as promotional pricing or product placements. Association rules can also be used in many application areas, including Web usage mining, intrusion detection and bioinformatics.

An association rule has the form $A \rightarrow B$, where A and B are disjoint sets of items (called the antecedent and the consequent of the rule, respectively). For example, $\{\text{milk}, \text{eggs}\} \rightarrow \{\text{bread}\}$ is an association rule that says that when milk and eggs are purchased, bread is likely to be purchased as well. Mining algorithms of association rules are based on various measures of significance and interest, such as support, confidence and lift (Geng and Hamilton 2006). Algorithms apply some constraints on such significance measures in order to select interesting rules from the set of all possible ones. The best-known constraints are minimum thresholds on support and confidence.

Generally, the association rule mining problem can be decomposed into two sub-problems. First, find all combinations of items

that have a certain statistical significance (frequent itemset mining). Second, given a significant itemset, generate all rules that have a certain strength.

Dimensionality reduction

Dimensionality reduction is the process of reducing the number of random variables (features, predictors) under consideration by obtaining a set of principal variables. Dimensionality reduction techniques are used for several reasons, including

- simplification of models to make them easier to interpret by researchers and users (James *et al* 2014),
- shorter training times,
- avoiding the curse of dimensionality (Bellman 1957), and
- enhanced generalisation by reducing overfitting (reduction of variance (James *et al* 2014)).

Data analysis such as regression or classification can be done in the reduced space more accurately than in the original space (Sulayes 2017).

Approaches to dimensionality reduction can be divided into feature selection (returns a subset of the features) and feature extraction (creates new features from functions of the original features). Feature selection is the process of selecting a subset of relevant features for use in model construction (Bousquet *et al* 2004; Blum and Langley 1997).

Feature extraction (also known as feature project and feature reduction), on the other hand, aims at transforming the data from a high-dimensional space to a space of fewer dimensions. Feature extraction starts from a set of initial features (measured data) and builds derived values (features) intended to be informative and non-redundant, facilitating the subsequent learning and generalisation steps. The data transformation may be linear, as in principal component analysis (PCA), but many non-linear dimensionality reduction techniques also exist, such as Sammon's mapping (Sammon 1969), curvilinear component analysis (Demartines and Herault 1997) and kernel PCA (Schölkopf *et al* 1998).

The aim of PCA (Abdi and Williams 2010), also known as the Karhunen–Loeve transformation, is to perform a linear mapping

of the data to a lower-dimensional space in such a way that maximises the variance of the data in the low-dimensional representation. In other words, PCA reshapes the data along the directions of maximal variance. Simply speaking, PCA transforms data linearly into new properties that are not correlated with each other. Singular value decomposition is another factorisation method that transforms a matrix into special matrices that are easy to manipulate and to analyse.

Non-negative matrix factorisation (NMF) is a group of algorithms that factorise (decompose) a matrix into two matrices, with the property that all three matrices have no negative elements. This non-negativity makes the resulting matrices easier to inspect. NMF has many applications in astronomy, text mining and spectral data analysis (Berry *et al* 2007).

Evaluation

Evaluation metrics for clustering

The methods for evaluating the performance of a clustering algorithm are classified as either

- extrinsic, requiring ground truth labels, or
- intrinsic, not requiring ground truth labels.

Extrinsic measures are the most commonly used in clustering problems, and are based on comparisons between the output of the clustering algorithm and a gold standard usually built using human assessors. Extrinsic evaluation is based on determining the distance between both clustering solutions: the system output and the gold standard. Evaluation metrics can be grouped into four families (Amigó *et al* 2009; Meila and Heckerman 2001; Meila 2005), based on counting pairs, set matching, entropy and edit distance. Metrics that are based on set matching share the feature of assuming a one-to-one mapping between clusters and categories, and they rely on the precision and recall concepts inherited from information retrieval (Zaki and Meira 2014).

- Purity (Zhao and Karypis 2001; Manning *et al* 2008) is a measure that quantifies the extent to which a cluster contains entities from only one partition, ie, it measures how "pure" each cluster is.

- The precision, recall and F_1-measure metrics typically used for classification evaluation can also be used to evaluate the performance of clustering algorithms.
- Normalised mutual information is a measure of the mutual dependence between the system clustering and the ground truth based on the shared object membership, with a scaling factor corresponding to the number of objects in the respective clusters.

In intrinsic evaluation, the aim is to identify sets of clusters that are compact, with a small variance between members of the cluster, and well separated (where the means of different clusters are sufficiently far apart) compared with the within cluster variance. Intrinsic measures include the following.

- The Davies–Bouldin index (DBI) is a metric for evaluating clustering algorithms, where the validation of the clustering is based on quantities and features inherent to the data set, such as the scatter of points within the cluster, and the separation between different clusters (Davies and Bouldin 1979). Thus, it captures the intuition that clusters which are well-spaced from each other and are themselves very dense are likely to be "good". As the DBI shrinks, the clustering is considered to become "better".
- The Dunn index captures the same idea as the DB index, as it improves when clusters are dense and far apart from each other. But the Dunn index increases as performance improves (Dunn 1974). However, while the DBI considers the dispersion and separation of all clusters, the Dunn index only considers the worst cases in the clustering: the clusters that are closest together and the single most dispersed cluster.
- Silhouette is a method of validation of consistency within clusters (Rousseeuw 1987). The silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared with other clusters (separation). A high value indicates that the object is well matched to its own cluster and poorly matched to neighbouring clusters. The silhouette can be calculated with any distance metric, such as the Euclidean distance or the Manhattan distance.

In addition to extrinsic and intrinsic evaluation metrics, there are relative evaluation metrics that are used to compare two clusterings,

such as the Rand index and adjusted Rand index. The Rand index is a measure of the similarity between two data clusterings, which is similar to the accuracy metric of classification evaluation.

Evaluation metrics for association rules

Typically, the evaluation of association rules mining is not in terms of the performance of the mining algorithm, but rather in terms of the quality (interestingness) of the discovered rules.

Various measures are commonly used to assess the significance and interest of association rules (Geng and Hamilton 2006). For a given association rule $A \rightarrow B$, the interest measures include the following.

- "Support" is an indication of how frequently the rule occurs in the database, and defined as the proportion of transactions in which the itemsets A and B appear together

$$\text{supp}(A \rightarrow B) = P(A \cup B)$$

- "Confidence" is defined as the proportion of the transactions containing A that also contain B

$$\text{conf}(A \rightarrow B) = P(B \mid A) = \frac{\text{supp}(A \cup B)}{\text{supp}(A)}$$

- "Lift" is the ratio of the observed support to that expected if A and B were independent

$$\text{lift}(A \rightarrow B) = \frac{P(B \mid A)}{P(B)}$$

- "Leverage" is a symmetric measure expressing the difference between the actual probability of $A \cup B$ occurring in a transaction and the probability when A and B are statistically independent

$$\text{leverage}(A \rightarrow B) = \text{supp}(A \cup B) - \text{supp}(A)\,\text{supp}(B)$$

CASE STUDY

On July 18, 2012, the McDonalds fast-food chain started a social media campaign using the hashtag "#McDstories" to emphasise their product quality. Within hours, thousands of people turned to Twitter and used this hashtag to share their negative stories about McDonalds (Lubin 2012). A seemingly arbitrarily occurring outrage

towards people, companies, media campaigns or politicians is called an online firestorm (Pfeffer *et al* 2014). These online firestorms have the potential to seriously affect a company's reputation or stock market value. Consequently, early detection of these negative word-of-mouth events is of high significance. We will use some of the above-mentioned approaches to exemplify how machine learning methods can be used on large-scale Web-based data to better understand the dynamics in the data.

Data

To study the dynamics of this Twitter firestorm, we use historical data from the 10% sample application programming interface (API) data of Twitter: a random 10% of all tweets over a three week period around the time of the incident. While Twitter's data samples are viewed critically in academia (Pfeffer *et al* 2018), they are widely used as a data source by social media teams of companies, business consultants and government entities for real-time analysis of public opinion (Hong and Nadler 2011; Younus *et al* 2011; Cody *et al* 2016). For the purpose of this case study, we extracted 110,898 tweets including the term "mcdonalds" (case insensitive).

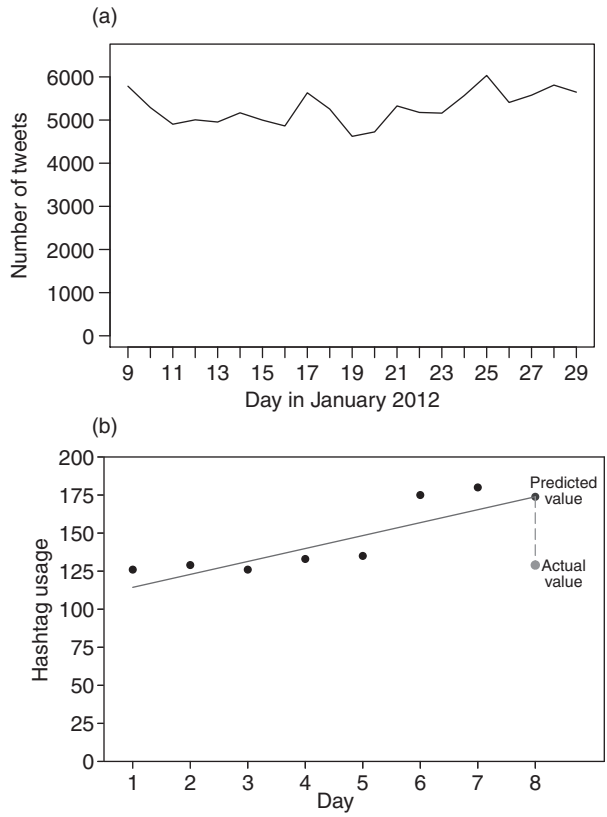
Change detection

In a first step we will identify whether a firestorm is going on. Figure 4.1(a) shows the time series plot of the overall number of tweets per day from our data source. This figure does not show suspicious changes caused by the firestorm (starting on January 18). In order to detect a change in the data, we need to apply methods that investigate the content of the tweets and that can be employed for real-time analysis. For the purpose of this case study we analysed data on a daily basis. Adapting the approaches to an hourly analysis or, in the case of more data, to a more granular temporal level is easily possible.

In order to systematically detect suspicious changes caused by the firestorm, we observe the usage of Twitter entities, such as hashtags, on a daily basis. The goal is to examine whether there is any deviation between the expected usage and the actual usage of an entity. When a significant deviation is observed, this is an indication that something unusual is going on which will need further inspection.

First, we use a supervised machine learning approach, namely linear regression, in order to predict the usage of a hashtag on a

Figure 4.1 Tweets per day and an example for regression.

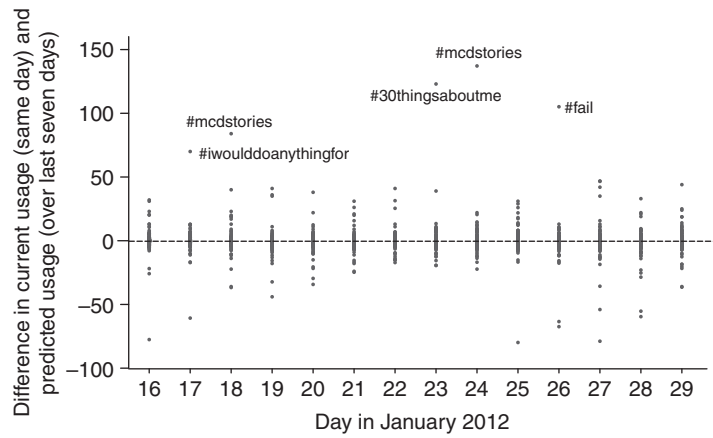


Part (a) shows the number of tweets per day in January 2012 that include the term "mcdonalds" in the 10% sample data source from Twitter. In part (b), using linear regression, we can predict the usage of a hashtag on a given day by extrapolating the usage over the last seven days. The dashed grey line shows the deviation.

given day d by extrapolating its usage over the previous seven days: $[d-7, d-1]$. As shown in Figure 4.1(b), we compare the predicated value of usage with the actual value to obtain the deviation. In this case, we can see a general upwards trend that is captured by the seven-day regression model. The actual data point on day eight deviates negatively from this trend.

We then repeated the operation of predicting the next data point with linear regression models for each hashtag that was used at least five times on every day (starting from January 16, since we need seven days of previous usage). The results are shown in Figure 4.2.

Figure 4.2 Deviation of actual hashtag usage from expected usage, on daily basis.



The hashtag "#mcdstories" appears on January 18 and January 24.

We observe that most hashtags form a cluster centred around zero, where the deviation, either positive or negative, is not significant. However, several outliers to that cluster can be observed on different days, which indicates significant positive or negative deviations. While a negative deviation of a hashtag indicates its decay, a significant positive deviation indicates a rise in new trending hashtags. In particular, we observe that our hashtag of interest, "#mcdstories", appears way above expectation on January 18, and then reappears again on January 24, followed by "#fail", which was used to describe McDonalds' response to the online firestorm.

Understanding what is going on

In order to understand what is going on, we extracted a subset of the tweets that contain the hashtag #mcdstories and similar terms, such as #mcdonaldstories. On this subset, for each tweet (text) as a document, we applied a preprocessing pipeline, typical for text analysis methods, including tokenisation, lower-case conversion and stop-words removal. Hence, each document became a list of tokens (terms). Then, we applied the unsupervised machine learning approach described above, namely association rules mining, using

Table 4.2 Association rules.

Rule	Support	Confidence	Lift
{“backfires”} → {“#mcdstories”}	0.142	1.0	1.021
{“horror”} → {“#mcdstories”}	0.107	1.0	1.021

the Apriori algorithm (Agrawal and Srikant 1994). A few of the interesting association rules among the many that we can identify in this data set are shown in Table 4.2.

Many tweets associated the hashtag “#mcdstories” with keywords such as “horror” and “backfires”. Examples of such tweets are the following:

- “McDonalds’ Twitter promotion fail: Users hijack #McDstories hashtag to share fast food horror stories”;
- “The @McDonalds social media campaign backfired. Now people are using #McDstories to share McDonalds horror stories”.

Text classification task

In the previous step, we identified possible story lines about tweets related to specific topics. Generalising this idea to identify newly emerging topics with negative stories about the brand brings us to topic modelling and text classification methods. Using latent Dirichlet allocation, the standard topic modelling approach, we could identify topics in the tweets about McDonald’s. Here, we focus on a classical machine learning challenge, namely classification. The underlying idea is to have a set of tweets precoded as being positive or negative towards the brand or unrelated to the brand. With these codes, a machine learning model is trained and applied to newly incoming tweets in order to automatically classify them.

For training and testing purposes, we use only tweets related to the #McDStories hashtag; we establish a ground truth by the manual labelling of tweets as good, bad or unrelated. The allocation to those classes is as follows: bad (10%), good (52%), unrelated (38%). Given the tweet text as a document, after preprocessing (tokenisation, etc) and computing of TF–IDF scores, each data entry is a vector of TF–IDF scores.

Then, we split the annotated tweets into two subsets: training data (80%) to train the classifier, and test data (the remaining 20%) to

Table 4.3 Classification results.

	Accuracy	Precision	Recall	F_1-score
Random forest	0.85	0.77	0.85	0.80
Support-vector machine	0.53	0.28	0.53	0.37
Logistic regression	**0.89**	0.90	0.89	**0.86**

Notes: Boldface denotes the highest values (see text).

evaluate the accuracy of the classifier. For comparison, we use three classification approaches: random forests, SVMs and logistic regression. Each of those classifiers is trained using the training data set and evaluated using the test data set.

The classification results are shown in Table 4.3, where the evaluation metrics used are accuracy, recall, precision and F_1-score (see p. 55).

We find that the random forest classifier has a reasonable accuracy (with 85% of instances being correctly classified). It also has a good enough recall and precision; hence the F_1-score is good (0.80). In contrast, the SVM classifier has a moderate accuracy (53% correctly classified instances). This classifier has a moderate recall but a low precision, hence the F_1-score is low too (0.37). In fact, in this case, SVM classifier performed badly in identifying the bad and unrelated classes. Hence, the precision and recall for those classes were very low, which is why the overall performance of this classifier was not good. Finally, the logistic regression outperforms the other classifiers, with an accuracy of 89% and an F_1-score of 0.86.

DISCUSSION

In the previous section, we showcased and discussed the machine learning methods used to analyse data characteristics from millions of tweets. The use of large-scale Web data can be a very good complement to gain insight into real-time dynamics, ie, knowledge and information for anticipating and controlling processes. The combination of traditional data sets (data from information service providers, financial reports, institutions, etc) and big data can be an information advantage here.

While these methods are relatively easy to use, some of them are algorithmically very complex and almost impossible to comprehend in detail for researchers from most fields. This leads to the

biggest issue related to computational methods: researchers deploying methods without considering their limitations or preconditions for the data.

Humans and machines: possible pitfalls of big data and machine learning

As Kolanovic and Krishnamachari (2017) discussed, the methods provided cannot entirely replace human intuition. Machine learning models can, if not properly guided, overfit or uncover spurious relationships and patterns. Data scientists who lack subject matter expertise may not achieve the desired investment results. When using big data, it is still necessary to understand the economics behind the data and signals. The role of humans and machines is twofold: machines have the ability to rapidly collect and analyse news feeds and tweets, scrape websites and trade on these continuously, but they are unlikely to be able to compete with strong macroanalysis and the refined intuition of human investors (Kolanovic and Krishnamachari 2017). Regarding the validity of data sets, biases and inaccuracies not only occur at the source of the data, but also are introduced during processing. The rigour with which these issues are addressed by different researchers is known to vary widely. In practice, a variety of dangers regarding social media data have been identified and studied (Pfeffer *et al* 2018; Olteanu *et al* 2019). As Lazer *et al* (2014) demonstrate, research on whether search data or social media "can predict x" has become commonplace and is often presented in sharp contrast with traditional methods and hypotheses. Although these studies have shown the value of such data, it is far from supplanting more traditional methods or theories. Perspectives on these challenges address the scientific infrastructure supporting data sharing, data management, informatics and statistical methodology. Research ethics and policy are discussed in the literature, and suggestions on how to tackle these challenges need to be discussed as well (Lazer *et al* 2020; King 2011; Vespignani 2009).

Because of these challenges, we recommend decision makers are sensitive to the following issues when applying machine learning methods, especially when using social media big data as alternative data sets (Ruths and Pfeffer 2014; Olteanu *et al* 2019).

- In designing your method, consider carefully what representation to use for your data, which algorithm to use, how to optimise it and how to evaluate its performance.
- Test the validity of both internal and external data.
- Data might lack quality due to sparsity, noise or bias effects.
- Population biases may affect the representativeness of a data sample.
- Data acquisition involves a query specifying a set of criteria for selecting, ranking and returning the data being requested, but different APIs may support different types of queries.
- Data filtering entails the removal of irrelevant portions of the data; sometimes this cannot be done during data acquisition due to the limited expressiveness of an API or query language.
- Biases introduced by data processing operations such as cleaning, enrichment and aggregation are likely to compromise the internal validity.

Therefore, we encourage decision makers to combine data science expertise with high levels of competence in the given field (eg, economics, behavioural economics, statistics, methodology). Applied properly, machine learning methods have the ability to complement established techniques and, when applied to very large data sets, have the potential to capture complex facts and dynamics.

REFERENCES

Abdi, H., and L. J. Williams, 2010, "Principal Component Analysis", *WIREs Computational Statistics* 2(4), pp. 433–59.

Agrawal, R., T. Imieliński, and A. Swami, 1993, "Mining Association Rules between Sets of Items in Large Databases", in *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pp. 207–16 (New York, NY: Association for Computing Machinery).

Agrawal, R., and R. Srikant, 1994, "Fast Algorithms for Mining Association Rules in Large Databases", in *Proceedings of the 20th International Conference on Very Large Data Bases*, pp. 487–99 (San Francisco, CA: Morgan Kaufmann).

Altman, N. S., 1992, "An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression", *American Statistician* 46(3), pp. 175–85.

Amigó, E., J. Gonzalo, J. Artiles and F. Verdejo, 2009, "A Comparison of Extrinsic Clustering Evaluation Metrics Based on Formal Constraints", *Information Retrieval* 12(4), pp. 461–86.

Bańbura, M., D. Giannone, M. Modugno and L. Reichlin, 2013, "Now-Casting and the Real-Time Data Flow", in *Handbook of Economic Forecasting*, pp. 195–237 (Elsevier).

Bellman, R., 1957, *Dynamic Programming*, Rand Corporation Research Study (Princeton University Press).

Berry, M. W., M. Browne, A. N. Langville, V. P. Pauca and R. J. Plemmons, 2007, "Algorithms and Applications for Approximate Nonnegative Matrix Factorization", *Computational Statistics and Data Analysis* 52(1), pp. 155–73.

Blum, A. L., and P. Langley, 1997, "Selection of Relevant Features and Examples in Machine Learning", *Artificial Intelligence* 97(1), pp. 245–71.

Bollen, J., H. Mao and X.-J. Zeng, 2011, "Twitter Mood Predicts the Stock Market", *Journal of Computational Science* 2(1), pp. 1–8.

Bousquet, O., U. von Luxburg and G. Ratsch, 2004, *Advanced Lectures on Machine Learning: ML Summer Schools 2003* (Berlin: Springer).

Ciulla, F., D. Mocanu, A. Baronchelli, B. Gonçalves, N. Perra and A. Vespignani, 2012, "Beating the News Using Social Media: The Case Study of American Idol", *EPJ Data Science* 1, pp. 1–11.

Cody, E. M., A. J. Reagan, P. S. Dodds and C. M. Danforth, 2016, "Public Opinion Polling with Twitter", e-print, arXiv:1608.02024 [physics.soc-ph].

Cook, S., C. Conrad, A. L. Fowlkes and M. H. Mohebbi, 2011, "Assessing Google Flu Trends Performance in the United States during the 2009 Influenza Virus A (H1N1) Pandemic", PLoS ONE 6(8), e23610.

Cover, T. M., and P. E. Hart, 1967, "Nearest Neighbor Pattern Classification" *IEEE Transactions on Information Theory* 13, pp. 21–7.

Davies, D. L., and D. W. Bouldin, 1979, "A Cluster Separation Measure", *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1(2), pp. 224–7.

Demartines, P., and J. Herault, 1997, "Curvilinear Component Analysis: A Self-Organizing Neural Network for Nonlinear Mapping of Data Sets", *Transactions on Neural Networks* 8(1), pp. 148–54.

Domingos, P., 2012, "A Few Useful Things to Know about Machine Learning", *Communications of the ACM* 55(10), pp. 78–87.

Dunn, J. C., 1974, "Well-Separated Clusters and Optimal Fuzzy Partitions", *Journal of Cybernetics* 4(1), pp. 95–104.

Engelberg, J. E., and C. A. Parsons, 2011, "The Causal Impact of Media in Financial Markets", *Journal of Finance* 66(1), pp. 67–97.

Ester, M., H.-P. Kriegel, J. Sander and X. Xu, 1996, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pp. 226–31 (Palo Alto, CA: AAAI Press).

Geng, L., and H. J. Hamilton, 2006, "Interestingness Measures for Data Mining: A Survey", *ACM Computing Surveys* 38(3), 9-es.

Gentzkow, M., B. Kelly and M. Taddy, 2019, "Text as Data", *Journal of Economic Literature* 57, pp. 535–74.

Gentzkow, M., and J. M. Shapiro, 2010, What Drives Media Slant? Evidence from US Daily Newspapers", *Econometrica* 78(1), pp. 35–71.

Ginsberg, J., M. H. Mohebbi, R. S. Patel, L. Brammer, M. S. Smolinski and L. Brilliant, 2009, "Detecting Influenza Epidemics Using Search Engine Query Data", *Nature* 457(7232), pp. 1012–14.

Goel, S., J. M. Hofman, S. Lahaie, D. M. Pennock and D. J. Watts, 2010, Predicting consumer behavior with Web search. *Proceedings of the National Academy of Sciences* 107(41), pp. 17486–90.

Hastie, T., R. Tibshirani and J. H. Friedman, 2009, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Second Edition, Springer Series in Statistics (Berlin: Springer).

Ho, T. K., 1995, "Random Decision Forests", in *Proceedings of the Third International Conference on Document Analysis and Recognition*, Volume 1, pp. 278–82 (Hoboken, NJ: IEEE Press).

Hoerl, A. E., and R. W. Kennard, 2000, "Ridge Regression: Biased Estimation for Nonorthogonal Problems", *Technometrics* 42(1), pp. 80–86.

Hong, S., and D. Nadler, 2011, "Does the Early Bird Move the Polls? The Use of the Social Media Tool 'Twitter' by US Politicians and Its Impact on Public Opinion", in *Proceedings of the 12th Annual International Digital Government Research Conference: Digital Government Innovation in Challenging Times*, pp. 182—6 (New York, NY: Association for Computing Machinery).

Huang, A., 2008, "Similarity Measures for Text Document Clustering", in *Proceedings of the Sixth New Zealand Computer Science Research Student Conference*, pp. 49–56 (Wellington: New Zealand Computer Society).

James, G., D. Witten, T. Hastie and Tibshirani, R., 2014, *An Introduction to Statistical Learning* (Berlin: Springer).

King, G., 2011, "Ensuring the Data-Rich Future of the Social Sciences", *Science* 331(6018), pp. 719–21.

Kolanovic, M., and R. T. Krishnamachari, 2017, "Big Data and AI Strategies, Machine Learning and Alternative Data Approach to Investing", Technical Report, JP Morgan.

Laney, D., 2001, "3D Data Management: Controlling Data Volume, Velocity, and Variety", Blog Post, February 21, Application Delivery Strategies.

Larose, D. T., and C. D. Larose, 2014, *Discovering Knowledge in Data: An Introduction to Data Mining*, Second Edition (Hoboken, NJ: John Wiley & Sons).

Lazer, D., R. Kennedy, G. King and A. Vespignani, 2014, "The Parable of Google Flu: Traps in Big Data Analysis", *Science* 343(6176), pp. 1203–5.

Lazer, D., A. Pentland, L. Adamic, S. Aral, A.-L. Barabási, D. Brewer, N. Christakis, N. Contractor, J. Fowler, M. Gutmann, T. Jebara, G. King, M. Macy, D. Roy and M. V. Alstyne, 2009, "Computational Social Science", *Science* 323(5915), pp. 721–3.

Lazer, D. M. J., A. Pentland, D. J. Watts, S. Aral, S. Athey, N. Contractor, D. Freelon, S. Gonzalez-Bailon, G. King, H. Margetts, A. Nelson, M. J. Salganik, M. Strohmaier, A. Vespignani and C. Wagner, 2020, "Computational Social Science: Obstacles and Opportunities", *Science* 369(6507), pp. 1060–62.

Lubin, G., 2012, "McDonald's Twitter Campaign Goes Horribly Wrong #McDStories", Blog Post, January 24, Business Insider.

Manning, C. D., P. Raghavan and H. Schütze, 2008, *Introduction to Information Retrieval* (Cambridge University Press).

Mao, H., S. Counts and J. Bollen, 2011, "Predicting Financial Markets: Comparing Survey, News, Twitter and Search Engine Data", e-print, arXiv:1112.1051 [physics, q-fin].

Meila, M., 2005, "Comparing Clusterings", in *Proceedings of the 22nd International Conference on Machine Learning*, pp. 577–84 (New York, NY: Association for Computing Machinery).

Meila, M., and D. Heckerman, 2001, "An Experimental Comparison of Model-Based Clustering Methods", *Machine Learning* 42, pp. 9–29.

Moon, S. J., and P. Hadley, 2014, "Routinizing a New Technology in the Newsroom: Twitter as a News Source in Mainstream Media", *Journal of Broadcasting and Electronic Media* 58(2), pp. 289–305.

Olteanu, A., C. Castillo, F. Diaz and, E. Kıcıman, 2019, "Social Data: Biases, Methodological Pitfalls, and Ethical Boundaries", *Frontiers in Big Data* 2, Article 13.

Pfeffer, J., K. Mayer and F. Morstatter, 2018, "Tampering with Twitter's Sample API", *EPJ Data Science* 7, Article 50, pp. 1–21.

Pfeffer, J., T. Zorbach and K. M.Carley, 2014, "Understanding Online Firestorms: Negative Word-Of-Mouth Dynamics in Social Media Networks. *Journal of Marketing Communications* 20(1), pp. 117–28.

Rokach, L., and O. Maimon, 2005, "Clustering Methods", in O. Maimon and L. Rokach (eds), *Data Mining and Knowledge Discovery Handbook*, pp. 321–352 (Boston, MA: Springer).

Rousseeuw, P. J., 1987, "Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis", *Journal of Computational and Applied Mathematics* 20, pp. 53–65.

Ruths, D., and J. Pfeffer, 2014, "Social Media for Large Studies of Behavior", *Science* 346(6213), pp. 1063–64.

Sammon, J. W., 1969, "A Nonlinear Mapping for Data Structure Analysis", *IEEE Transactions on Computers* 18(5), pp. 401–9.

Schölkopf, B., A. Smola and K.-R. Müller, 1998, "Nonlinear Component Analysis as a Kernel Eigenvalue Problem", *Neural Computation* 10(5), pp. 1299–1319.

Sulayes, A. R., 2017, "Reducing Vector Space Dimensionality in Automatic Classification for Authorship Attribution", *Revista Ingeniería Electrónica, Automática y Comunicaciones* 38(3), pp. 26–35.

Tibshirani, R., 1996, "Regression Shrinkage and Selection via the Lasso", *Journal of the Royal Statistical Society (Series B)* 58, pp. 267–88.

Vermeer, S. A. M., T. Araujo, S. F. Bernritter and G. van Noort, 2019, "Seeing the Wood for the Trees: How Machine Learning Can Help Firms in Identifying Relevant Electronic Word-of-Mouth in Social Media", *International Journal of Research in Marketing* 36(3), pp. 492–508.

Vespignani, A., 2009, "Predicting the Behavior of Techno-Social Systems", *Science* 325, pp. 425–8.

Younus, A., M. A. Qureshi, F. F. Asar, M. Azam, M. Saeed and N. Touheed, 2011, "What Do the Average Twitterers Say: A Twitter Model for Public Opinion Analysis in the Face of Major Political Events", in *Proceedings of the 2011 International Conference on Advances in Social Networks Analysis and Mining*, pp. 618–23 (Hoboken, NJ: IEEE Press).

Zaki, M. J., and W. Meira, Jr, 2014, *Data Mining and Analysis: Fundamental Concepts and Algorithms* (Cambridge University Press).

Zhao, Y., and G. Karypis, 2001, "Criterion Functions for Document Clustering: Experiments and Analysis", Technical Report 01-40, Army HPC Research Center, Minneapolis, MN.