Routledge
Taylor & Francis Group

Economic Research-Ekonomska Istraživanja

ISSN: 1331-677X (Print) 1848-9664 (Online) Journal homepage: http://www.tandfonline.com/loi/rero20

Processing unstructured documents and social media using Big Data techniques

Vlad Diaconita

To cite this article: Vlad Diaconita (2015) Processing unstructured documents and social media using Big Data techniques, Economic Research-Ekonomska Istraživanja, 28:1, 981-993, DOI: 10.1080/1331677X.2015.1095110

To link to this article: http://dx.doi.org/10.1080/1331677X.2015.1095110

Economic Research-Ekonomska Istraživanja, 2015
Vol. 28, No. 1, 981–993, http://dx.doi.org/10.1080/1331677X.2015.1095110

Routledge
Taylor & Francis Group

Processing unstructured documents and social media using Big Data techniques

Vlad Diaconita*

Faculty of Economic Cybernetics, Statistics and Informatics, Bucharest University of Economic Studies, Bucharest, Romania

(*Received 5 May 2015; accepted 14 September 2015*)

Big Data technologies can be very useful when it comes to storing and processing using sophisticated algorithms, terabytes or petabytes of data. With the latest advancements, such as Hadoop YARN, processing can be done not only in batch but also in real time. In this paper, we detail a methodology followed by a case study that investigates the power of machine learning algorithms used in a Hadoop environment in classifying unstructured data. We also investigate how to capture geolocated messages from social networks and how kriging can be used to see if there is a strong relationship between two or more such datasets.

Keywords: Hadoop; MapReduce; k-NN; social media; geolocated messages; large data sets

JEL classification: L86, C88, C55

1. Introduction

There is a lot of data produced and consumed every day by people interacting with different applications and devices or to support a broad range of activities such as weather prediction, disaster aftermath evaluation, fraud detection, inefficiency detection or health evaluation, to name only a few. In some activities, the amount of data that is generated is enormous; for example, Virgin Atlantic IT director David Bulman said in an interview for Computerworld UK that a Boeing 787 generates half a terabyte of data per flight, and every piece of the plane has an internet connection helping the flight crew and the ground crew to better diagnose possible problems (Finnegan, 2013). Various businesses that use modern IT applications to track their stocks, supply options or customer's preferences, generate – but also consume – large quantities of data.

Prediction models are widely used by online merchants for targeting an advertising campaign or certain discounts to particular customers. They are also used for pushing sales by recommending similar products or services to the one in which the client has shown interest (by adding them to cart or even just by viewing them). In this case, similarity can be defined as products with characteristics similar to the reference product or as products for which other customers that considered the reference product have shown interest in.

Datasets that can be used in econometrics and statistics are therefore getting bigger, multi-dimensional and sometimes unstructured, so alternative methods for storing and analysing them are required. Added value can be generated if these data are correctly

*Email: diaconita.vlad@ie.ase.ro

processed, preferably in real-time. Some local governments are making available series of data (e.g. NYC Open Data, Madrid Open Data) that researchers can analyse and so criticise decisions that were made in relation to those data (Ho, 2012).

This paper aims to bring contributions in the field of Economic Informatics by building, refining and evaluating models that benefit from Big Data techniques. After the literature review, we show how processing data captured from social networks can take advantage of the interplay between Big Data and integrating geographical data using the kriging method. In the next part, we present a k-NN based approach to classifying documents using a Hadoop infrastructure, comparing the speed and accuracy of two different metrics. The results of the cross-validation tests are reviewed and discussed.

2. Literature review

Various methods, products and frameworks for handling and analysing large volumes of data have emerged. Hadoop, the best-known ecosystem in Big Data, was initially developed by Google and Yahoo and nowadays is growing with the help of other big IT companies such as Facebook. Different Hadoop distributions continue to appear, either open source or proprietary, such as Hortonworks, Cloudera or MapR. Social media generates a lot of data, and Big Data approaches are used and developed by networks such as Facebook or Twitter to improve their systems. As discussed in Thusoo et al. (2010), the Facebook Data Infrastructure Team was faced with daily data processing jobs that were taking more than a day to complete. Prior to 2008, they were using a processing infrastructure that was built around a data warehouse built using a commercial RDBMS. As a response to the slow processing problem, the team developed HIVE, an open-source data warehousing solution built on top of Hadoop. It implements HiveQL, an SQL-like declarative language for large-scale data analysis that allows the user to create queries that are executed using MapReduce jobs.

There are many areas where Big Data techniques proved their worth. In bioinformatics, they are employed in SNP discovery, genotyping and personal genomics. As shown in Schatz (2009), tools such as CloudBurst use MapReduce to parallelise execution using multiple compute nodes. The author concludes that CloudBurst's running time scales linearly or near linearly when the number of CPUs increases. For next-generation genome sequencing problems, in Wang et al. (2015) the authors propose a novel FPGA-based acceleration solution with MapReduce framework on multiple hardware accelerators.

As stated in Einav and Levin (2014), Big Data methods can be used successfully in economics. The authors don't see a clear distinction between predictive modelling and causal inference so they argue that statistical data-mining techniques should be increasingly used by economists. As showed in Varian (2014), data analysis in economics can be valuable for finding patterns in data, doing predictions, estimations, and hypothesis testing. Also, in econometrics, the authors show that column-oriented databases such as Cassandra are especially suitable for economic time series or event data processing.

There are also problems and concerns regarding analysing digital traces left by people. On one hand, it may help create better tools and services while, on the other, it encourages a new wave of privacy incursions and invasive marketing.

Data captured from social networks can be valuable in order to observe, among other things, how social connections and the information feed that a person is getting shapes his consuming preferences. Traces left by people on social media, on the websites they browsed or on the applications that they've used can by looked into by using powerful machine learning algorithms or with simpler statistical correlations. The

purpose is to draw conclusions about unknown facts such as anticipating consuming trends or even opinion trends. These are useful for firms but also for political parties or governmental agencies. As argued in Einav and Levin (2014), if proper information and statistics were available, people might better restrain themselves from making bad financial decisions, such as buying a house when their economic situation is very likely to lead them to default.

When trying to draw conclusions starting from data generated by people – for example as a result of their web searching patterns, or when analysing posts written on social media – ensuring acuity can be complicated. In such cases, as shown in studies such as Boyd and Crawford (2012), Big Data methods can be subjective. For example, during the data cleaning process, decisions that can affect the outcome are made regarding which attributes and variables are included or which time span is chosen. Starting from a message corpus there have been attempts to predict the success of a product launch, to assess the political affiliation of active social media users, or to see if online buzz translates into electoral success. In Mustafaraj and Metaxas (2010) the authors collected more than 185,000 messages related to the 2010 Massachusetts special senatorial election using the Streaming Twitter API. They claim that graph-theoretic techniques guessed 98% of users' political orientation from the top 200 group. This group contained users who posted more than 100 messages, and the results of the algorithms were compared with a manual assessment of political affiliation done by reviewing user's self-description and their posts. They also discussed attempts to influence undecided voters, for example by gaming search engine ranking results to push forward negative news about the other party. Ratkiewicz et al. (2011) present results that show a 96% acuity in using supervised learning to detect social media political campaigns that seem to spread naturally when in fact they are orchestrated by a single person or organisation.

One apparently promising idea is to try to predict actual election vote shares based on the number of tweets mentioning a party during the campaign (Tumasjan, Sprenger, Sandner, & Welpe, 2010). The accuracy of such predictions is questioned, for example, in Jungherr (2013) where the author concludes that for Germany's 2009 elections, analysing the hashtag mentions of a party (or its leading candidate) isn't a valid indicator of electoral success. He also observes that the share of hashtag mentions varies considerably from day to day, usually in connection with an offline event, so the chosen time span of a study greatly influences the results. In addition, it is very unlikely that the people who post on Twitter are a representative statistical sample of a country's voting population.

Lazer, Kennedy, King, and Vespignani (2014) discuss the 2013 situation when Google Flu Trends (GFT) predicted more than double the proportion of doctor visits related to influenza-type viruses (such as AH1N1 or AH5N1) than the American Centres for Disease Control and Prevention (CDC). This was happening despite the fact that GFT was especially built for this type of predictions using a methodology initially described in Ginsberg (2009), updated in September 2009 and evaluated in Cook, Conrad, Fowlkes, and Mohebbi (2011). In Lazer et al. (2014) it is argued that the prediction errors are a result of the Google algorithm's dynamics and of the so-called Big Data Hubris. They observe that some studies lack transparency and replicability and start from the assumption that Big Data techniques are more likely a substitute rather than a companion of traditional data collecting and analysing. As also observed in Boyd and Crawford (2012), having access to a large quantity of data does not mean that one should ignore the basic problems regarding measuring and constructing valid and trustworthy dependencies among data.

As shown in Mantelero and Vaciago (2014), despite the weakness of some Big Data approaches, more focused on correlation than on statistical evidence, such approaches can be suitable for predicting and perceiving the birth and evolution of macro-trends that can be later analysed in a more traditional statistical way in order to identify their causes.

Polato, Ré, Goldman, and Kon (2014) conducted a systematic literature review of the Hadoop research. From IEEEXplore, ACM, and ScienceDirect the authors selected 106 studies from over 1500 papers that contain the search terms 'Hadoop' and 'MapReduce' to assess research contributions to Apache Hadoop and to spot promising areas and propose topics for future research within the framework. It's detailed how some authors put forward ideas to alter the Hadoop MapReduce data flow in order to improve performance. For example, Wang, Que, Yu, Goldenberg, and Sehgal (2011) offer a solution that overlaps the Shuffle, Merge and Reduce phases. In Vernica, Balmin, Beyer, and Ercegovac (2012) the focus is on the interaction of Mappers, introducing an asynchronous communication channel between Mappers.

Using data mining techniques in economics is described in works such as Hastie, Tibshirani, and Friedman (2008), which describes nearest-neighbour methods as those methods that find the observations in the training set closest in input space to x to form Y. The method finds the k observations with x_i closest to x in input space based on a metric. Kašćelan, Kašćelan, and Jovanović (2014) used a data mining method based on logistic regression, clustering, and decision trees. Such resource-demanding algorithms (k-NN is called a lazy classifier) can benefit from being implemented in a Hadoop environment that can be easily scaled by adding new nodes.

3. Methodology and data

3.1. Capturing and processing social media data

For developers, the social networks provide powerful APIs. There are also unofficial Java libraries for the Twitter API and other tools that can be used to retrieve social media data. For example, *Facepager* was designed for fetching data from Facebook, Twitter, and other JSON-based APIs. The data are stored in a local SQLite database and may be exported to CSV or JSON. An integration tier can be built using PL/SQL and Java procedures to make the applications communicate or to manipulate relational or semi-structured XML/JSON data (Diaconita, Lungu, & Bara, 2009).

Social media can also be used to study the timeframe of natural disasters or even to help build a better map of the damages. From the number of tweets, we can see what area was worse hit by an earthquake, fire, flood or another event. Official data coming from the police, firefighters or satellites can be supplemented, integrated and cross-referenced with data mined from social media for a better assessment of the damage. For constructing maps, crowdsourcing projects such as OpenStreetMap (OSM) can be useful because they offer for free an editable map of the whole world. By comparison, using Google Maps is limited by the API or by the terms of service (ToS). In OSM, users maintain data about roads, trails or other points of interests from around the world in different ways, for example by uploading GPS track logs. Data can be exported in XML format and freely used in any project. There are many research projects from around the world that use OSM data, e.g. *FEMroute* or *OpenStreetSLAM*.

On 22 November 2014, at about 21:15, there was a medium-sized earthquake (5.7 M) in Romania near Marasesti, Vrancea. The official location was at 45.865°N 27.158°E depth=39.1 km (24.3 miles).

We retrieved data from Twitter that contain the #cutremur and #earthquake hashtags using Facepager developed by Keyling and Jünger (2013). Using the API, we can search for tweets from a particular geographical area. For example, to search for tweets in a 250 km radius from the quake event we can use the following string as a query: https://api.twitter.com/1.1/search/tweets.json?q=%23cutremur&geocode=45.865%2C27.158%2C250 km. For geolocated data to be available, the user has to have location enabled on his or her device or a location set in his or her profile. If someone tweets from their phone or tablet with location enabled, a bounding box for the place will be available, but not necessarily showing the real originating place, e.g. 'coordinates': [[[20.2428259690001, 43.6500499480002], [20.2428259690001, 48.2748322560001], [29.6995548840001, 48.2748322560001], [29.6995548840001, 43.6500499480002]]]. If location is not enabled on the device, Twitter will try to tie the messages with the town in the user's profile (if provided) and return geolocated data. If not even this is available, we can loosely tie a tweet to a country by using the time zone info that is usually available and most likely shows the country's capital city: 'time_zone': 'Bucharest'. In this case, the GPS coordinates bounding box will have to be constructed.

In Romania, Twitter isn't very popular so data can be limited. To supplement data, we can retrieve posts from Facebook but there we are faced with the privacy settings of the user and hashtags are less used. On Facebook, pictures with location set return geolocated data in this format: 'location': {'latitude': 43.7333, 'longitude': 7.41667}. However, the uncertainty of the geodata captured from social networks makes this type of information more likely to be used to study the timeframe of an event and less likely to study the consequences. In Figure 1 we used data binning to summarize the tweets on 10 minutes intervals for the 22 November 2014 quake. For this event, we observed a peak in tweets with the #cutremur (earthquake) hashtag between 21:20 and 21:49. About 49% of the studied messages were retweets (RT @) and 3% were replies directed to a specific user (@). We didn't count replies that didn't contain the observed hashtags. We haven't noticed any signs of information manipulation. Unfortunately, for this event,

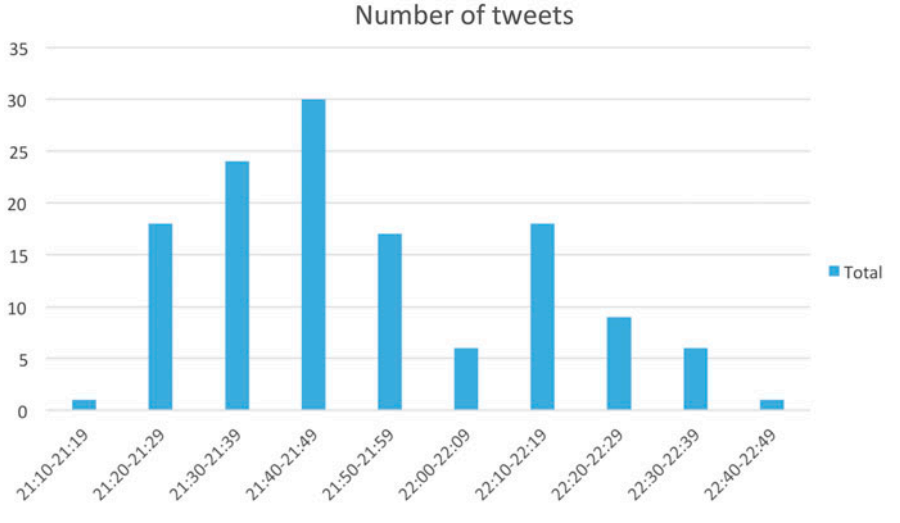

Figure 1. Number of tweets on 10 min intervals for the earthquake event.
Source: Constructed by the author.

we lacked enough geolocated elements to build a model to assess the damage, which was limited. If the earthquake had been stronger, we would probably have acquired enough data to create a damage evaluation map. For an earthquake, data from the Seismological Institute are instantly available. For other events, such as an accident or a fire, social data can show the authorities that there is a significant chance that such an event or a consequence of an event occurred.

Even when official data exist, a small number of good tweets or posts with geolocated pictures or video can help assess the damage (Schnebele, Cervone, & Waters, 2014). The severity of the damage shown by images or videos can be evaluated, and the geolocation cross-referenced. Crowdsourcing can be useful; someone who knows the area can confirm that a picture is taken where the coordinates say it was taken. We can use kriging to integrate different data, a method similar to regression analysis and IDW. As described in Johnston, Ver Hoef, Krivoruchko, and Lucas (2003), this technique is built on the assumption that things that are close to each other are more similar than those farther away. It weights the surrounding measured values to derive a prediction for each location. The weights are based on the general spatial arrangement among the measured points and the distance between them:

$$\hat{Z}(s_0) = \sum_{i=1}^{n} \lambda_i Z(s_i) \tag{1}$$

where:

- $Z(s_i)$ is the measured value at the ith location, in our case, for example, the severity of the event as shown in a geolocated item;
- λ_i is an unknown weight for the measured value;
- s_0 is the prediction location;
- n is the number of measured values.

The spatial variation of $Z(s_i)$ can be measured using a semivariogram. This can be built starting from (a) Euclidean distance calculated for every two locations, as in equation (4) later, with p and q being the coordinates of two sites (e.g. p=[20.24, 43.65], q=[21.34, 48.27]); (b) the squared measured values at those locations; and (c) the empirical semivariance (ES) calculated as:

$$\mathrm{ES} = 0.5 * [\mathrm{Z(s_i)} - \mathrm{Z(s_j)}]^2 \tag{2}$$

If we have many observed values, we can group values together by intervals of distances. After the average distances and semivariances are calculated for every interval, we can fit them in a linear regression model with the constant omitted from the equation. The semivariance will be estimated as $a * X$ where X is the distance between two points. After that, we can construct the gamma matrix Γ and the inverse matrix Γ^{-1}. Now we can compute the weights vector λ:

$$\lambda = \Gamma^{-1} * g \tag{3}$$

In equation (3), the g vector is constructed for the location we want to predict (or cross-validate) by applying the calculated semivariance to the Euclidean distance between the coordinates of the new point and the existing points. After that, we can calculate λ, the weights for each measured value. Now we have all the necessary data to compute the predicted value $\hat{Z}(s_0)$ from equation (1). The kriging variance can be calculated as

$g^*\lambda$ to measure the uncertainty of each prediction, the standard error being the square root of the variance.

Kriging is usually used to predict the unknown measured value of a new location in connection with the existing known values. In this case, we can use it to see if there is a strong relationship between two or more datasets. We have measured values from two sources so we can use cross-validation to see if the model is robust.

3.2. Storing unstructured data method

Using a vector space model (VSM) can be a solution for representing unstructured text. It can also be an appropriate solution for representing time series that can be used in forecasting in areas such as statistics or econometrics.

For storing unstructured text, starting from the corpus, we build a dictionary, where each word is given a position in the vector, as shown in Table 1, usually after eliminating case and stopover words (Diaconita, 2014).

Next, we can represent every document in sparse arrays using the dictionary:

[document 1: [[a_1,b_1],….,[a_n,b_n]]… document z: [[a_1,b_1],….,[a_n,b_n]]]

where a_i is the position of a word in the dictionary from Table 1 and b_i is the number of occurrences in the document we represent in VSM.

After we have enough training data, we can apply k-NN using the MapReduce approach, to find, given a new document (a query), what the nearest k items are so we can classify it. We can use different metrics (similarity measures), to see which are the most adjacent documents to the query q_i from d_i, such as:

Euclidean distance:

$$\mathrm{D}(\mathrm{d},\mathrm{q}) = \sqrt{\sum_{i+1}^{n}(q_i - d_i)^2} \tag{4}$$

Cosine similarity:

$$\mathrm{Cos}(\theta) = \sum_{i=1}^{n} d_i * q_i / \sqrt{\sum_{i=1}^{n}(d_i)^2} * \sqrt{\sum_{i=1}^{n}(q_i)^2} \tag{5}$$

We can also find the nearest neighbours of a set of query documents from a data-set of every point in another data-set by finding nearest neighbours and performing joins (k-NN-join). These algorithms can be implemented using, for example, a PL/SQL user-defined function in a relational database (Oracle) or by using Java, Python or Ruby in a Hadoop environment. For example, given the following documents, each consisting of one sentence (partly adapted after Project Gutenberg's *The Adventures of Sherlock Holmes*, by Arthur Conan Doyle):

Table 1. Dictionary represented as a vector.

V[1]	V[2]	V[3]	….	V[*N*]
Word1	Word2	Word3		Word*N*

Source: Created by the author.

D1: *I have seldom heard him mention her under any other name.*
D2: *I heard him mention the suspect before.*
D3: *I know without reading it that it is all perfectly familiar to me.*
D4: *Rarely I heard him mention her under any other name.*

The query is Q: *I know him under other familiar name.*

The dictionary represented as a vector is:

D: [1, I], [2, have], [3, seldom], [4, heard], [5, mention], [6, her], [7, under], [8, any], [9, other], [10, name], [11, him], [12, the], [13, suspect], [14, before], [15, know], [16, without], [17, reading], [18, it], [19, that], [20, is], [21, all], [22, perfectly], [23, familiar], [24, to], [25, me], [26, rarely]

If we use the number of occurrences of a word as weights we describe the documents as following:

D1: [1, 1], [2, 1], [3, 1], [4, 1], [5, 1], [6, 1], [7, 1], [8, 1], [9, 1], [10, 1], [11, 1]
D2: [1, 1], [4, 1], [5, 1], [11, 1], [12, 1], [13, 1], [14, 1]
D3: [1,1], [15,1], [16,1], [17,1], [18,2], [19,1], [20,1], [21,1], [22,1], [23,1], [24,1], [25, 1]
D4: [1, 1], [4, 1], [5, 1], [7, 1], [8, 1], [9, 1], [10, 1], [11, 1], [26, 1]
Q: [1, 1], [7, 1], [9, 1], [10, 1], [11, 1], [15, 1]

When using Euclidean distance we compare positive numbers, the smaller the value, the closer are the documents. In the case of cosine similarity, the values are in the interval [–1, 1], –1 being opposite, 1 exactly the same and 0 meaning independence. The found neighbours can vary depending on the used metric and the weights being used.

When we calculate the distances, we use 0 if there is not a value for a position in one vector and there is one in the other vector. Using Euclidean distance, the distance between Q and D1 is $D(Q,D1)= ((1–1)^2+(1–0)^2+(1–0)^2+(1–0)^2+(1–0)^2+(1–1)^2+(0–1)^2+(1–1)^2+(0–1)^2)^{1/2}=7^{1/2}=2.6457$. We calculate the distances from Q to the other documents. Using Euclidean distance they are D(Q, D2) = 3; D(Q, D3)=4.1231; D(Q, D4)=2.2360. Using cosine similarity the distances are COS(Q,D1)=0,6,154; COS (Q,D2)=0.3086; COS (Q,D3)=0.2108; COS (Q,D4)=0.6804.

So, if we search for the two nearest neighbours (2-NN) for Q we get them in this order: D4 and D1 using both the Euclidean distance and the Cosine similarity as metrics.

Depending on the training data, to further refine the algorithm, we can use term frequency–inverse document frequency (TF-IDF) to better weigh the importance of a word in a paper. By doing so, we can better order the found neighbours in terms of similarity to the query. Also, we can reduce the search space of the algorithm or we can use a non-supervised machine learning technique such as Self-Organising Maps (SOM) for non-parametric clustering problems to see what documents form similar groups.

3.3. Data description and classification method

For predictions, economists tend to rely on linear or logistic regression. When a lot of multi-dimensional data are involved, a k-NN approach, such as the one discussed below, can provide excellent results and the *k*-fold cross-validation can be a useful tool to rate the model.

For testing the implemented model, we used 30,100 documents containing a title, body, and categories. One document could belong to one or more categories that aren't

known upfront. All the documents are written in English and came from heterogeneous sources: forum dumps, wiki dumps, and a few conference proceedings. The records were merged into one big file that was stored in HDFS.

As infrastructure, we used a single node cluster, Hadoop 2.2 installed on Linux CentOS. In Hadoop, data is stored in a distributed file system (HDFS) and the main framework that can be used for distributed data processing is MapReduce. We can consider the MapReduce paradigm as similar to the divide and conquer technique. Google promoted it as a method of solving large volumes of data in analysing problems by using clusters of commodity machines. The MapReduce consists of three main phases: map, sort/shuffle and reduce. The Map phase transforms the input records into (key, value) pairs based on a given user-defined function. The reducer performs an aggregation or summarisation step in which all records from a partition are processed together, and the needed output is produced (which is usually written directly to the HDFS output file system). Between the Map and the Reduce, there is a Sort and Shuffle phase where the system performs the sorting and then transfers the Map outputs to the Reducers as inputs. It ensures that there will be one partition for each reduce task. As shown in White (2012) the shuffle is an area where refinements and improvements are continually being made, and its details are mostly invisible to the programmer.

Starting with Hadoop-2 YARN, the resource manager is more general and can be used to run other computing and processing models, such as BSP, besides legacy MapReduce.

The workflow of the proposed classification algorithm is:

(1) Using labelled data train a model based on the nearest-neighbour method;
(2) Test the model on holdout data;
(3) Refine the model based on the results;
(4) Make a prediction for a new instance.

We constructed – using a mapper class and a reducer class – the dictionary containing all the used words starting from the corpus. We sorted the dictionary and kept only the first 150,000 words in order of their frequency. We extracted the documents and represented them in sparse arrays using the dictionary. For the big file, this whole process took less than 1 min. The file containing all the documents had 126 MB and the resulting text file with the sparse array representation had 32 MB. A line containing a VSM representation of the 29,700th document is shown below:

29700 – [[131,703, 2], [137,202, 1], [142,035, 3], [144,246, 1], [144,267, 2], [144,770, 1], [144,803, 1] …]

All the documents were represented in small caps, and the punctuation was removed. We implemented a k-NN type approach using two different metrics, Euclidean distance, and cosine similarity.

4. Results and discussions

To test the Hadoop implemented k-NN model, we used cross-validation (also implemented in Hadoop), a method that can be more efficient than residual validation in order to validate a model. It tests predictions on data not previously seen. We used a 5-fold cross-validation, where the data are divided into test and training sets five different times. Every time we have 20% holdout data that are tested against the function estimated from the remaining data. In this approach, every data-point is used once in a test

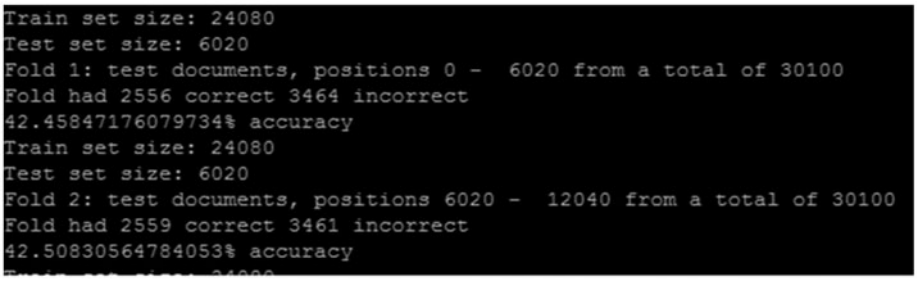

Figure 2. Cross-validation results on the whole data-set.
Source: Research results.

set and four times in a training set. The validation of a test set (subsample) against the remaining data is known as a fold. Because we calculate the distance from every point to every other point at every step, this approach is very time-consuming, especially if no search space reduction method is used. It took about 12 h for just one fold to be completed (test set: 6,020 records, training set 24,080) so we only let it run for two folds (Figure 2). The accuracy of the folds was about 42% with cosine similarity as a metric.

For faster testing, we randomly sampled 2000 and then 200 records and applied the 5-fold cross-validation to see the difference in acuity. The results for the 5-fold cross-validation of the model for both samples are shown in Tables 2 and 3.

We can observe that the Cosine similarity performs better than the Euclidean distance. As the number of records increases, the model acuity is slightly improved but the running time increases exponentially, as expected from a lazy-classifier. This approach can benefit from the scalable infrastructure of Hadoop; more nodes, and more mappers/reducers will help reduce the execution time.

The accuracy may seem a bit low, but this classification problem isn't a simple one. The corpus is heterogeneous, there isn't a fixed number of categories, and every document can belong to more than one classification. For example, if an article or a post had three keywords or tags, we considered that it belonged to all three categories and regarded it as a match if the model correctly predicted at least one category. Also, in this case, we are dealing with unstructured text that has associated context that we are losing when we reduce the data so they can be stored in VSM to enable use to calculate and compare numbers.

Table 2. The results of the *k*-fold cross-validation on a 2000 records sample using two metrics.

	Cosine similarity accuracy	Euclidean distance accuracy
First fold	44.5%	36.5%
Second fold	39.25%	34.25%
Third fold	42%	39.5%
Fourth fold	37.7%	37.5%
Fifth fold	40.0%	33.25%
Total runtime	10m12.691s	7m54.286s
Accuracy average	40.69%	36.19%

Source: Authors' calculation using the algorithm executed in a Hadoop environment.

Table 3. The results of the *k*-fold cross-validation on a 200 records sample using two metrics.

	Cosine similarity accuracy	Euclidean distance accuracy
First fold	35%	27.5%
Second fold	35%	35%
Third fold	32.5%	37.5%
Fourth fold	52.5%	35%
Fifth fold	37.5%	30%
Total runtime	0m7.238s	0m6.545s
Accuracy average	38.5%	33%

Source: Authors' calculation using the algorithm executed in a Hadoop environment.

This case study confirms one of the conclusions from Boyd and Crawford (2012), that ontologies are not obsolete and numbers don't speak for themselves, especially when used in connection with text written by people. Even though this method can't be trusted for automated classification, it can be used to provide hints in the form of a ranked categories list to which a new document might belong. This method can also prove useful, in connection with techniques from text mining, in assessing how similar two or more documents are in the sense of plagiarism detection. Depending on how the data are stored, the assessment can be done, at different levels: sentence, paragraph or by comparing the whole document. For example, we could detect the paragraphs with the most common words from a series of documents. One of the advantages is that a rephrasing that only changes the order of words can be easily detected. Given the linear speedup of the MapReduce model, the processing speed of this approach can be improved by adding more CPUs or nodes. For improving the prediction accuracy, as a future development, we can advance to an ensemble method that combines two or more metrics with the same algorithm or multiple machine learning methods.

Conclusions

Using Big Data on unstructured data can be challenging (choosing the right metric, weight, space reduction approach, and so on) but rewarding. In order to use machine-learning algorithms on text posted by people on blogs, forums or social networks, data are often concentrated into what can fit into a mathematical model, so the context is hard to interpret and maintain. Data lose meaning and value if taken out of context, or data might be used to support a point that is far from the original author's intention.

Using Hadoop, testing, refining and validating machine learning approaches can lead to obtaining and providing real value (such as identifying patterns and correlations that otherwise might be ignored) when analysing large volumes of data. Especially when trying to understand human interactions or to predict behaviours, we shouldn't look for answers only in large volumes of data. Big Data, small data and more traditional data handling approaches can become trustworthy companions.

In economics, a k-NN approach similar to the one described here can be used in time-series classifications. We can predict to which category a new series belongs based on a similarity measure and a decision combination method. Depending on the time series type, different approaches and similarity measures can be appropriate, as discussed in Lee, Wei, Cheng, and Yang (2012): i.e. shape search, Discrete Fourier transform (DFT) and subsequence matching. Methods such as DFT can use the Euclidean distance to measure the similarity between time-series sequences. We assume that time series that

are close to each other belong to the same group. For example, if a new unclassified time-series related to product sales has as the nearest neighbours, series for products where steep decreases in orders were observed, we can classify it in a group that requires special attention. Also, given the fact that in trading, profits can be made if the trend for a stock is correctly identified, we can classify stock series to isolate those where the probability of a particular direction is high. Going further, an interesting approach that capitalises on the power of a Big Data architecture might use time series with multiple variables (trade prices, trade volumes, number of sell/buy orders at different price intervals, to name a few) and try to classify new series according to a weighted distance to each variable. If we have long time series and enough distributed computing power, we can search (e.g. by using a brute force approach or maybe numerical optimisation algorithms) for a combination of distances to each variable that signals a particular trend.

Even though many articles are being published, we can consider Big Data as a field in an emerging phase, the value and veracity of the results are still to be proven and improved with the help of case studies.

Acknowledgement

This paper was co-financed from the European Social Fund, through the Sectoral Operational Programme Human Resources Development 2007–2013, project number POSDRU/159/1.5/S/138907 'Excellence in scientific interdisciplinary research, doctoral and postdoctoral, in the economic, social and medical fields – EXCELIS', coordinator: The Bucharest University of Economic Studies.

Disclosure statement

No potential conflict of interest was reported by the author.

ORCID

Vlad Diaconita http://orcid.org/0000-0002-5169-9232

References

Boyd, D., & Crawford, K. (2012). Critical questions for big data, provocations for a cultural, technological, and scholarly phenomenon. *Information, Communication & Society, 15* (5), 662–679.

Cook, S., Conrad, C., Fowlkes, A. L., & Mohebbi, M. H. (2011). Assessing Google flu trends performance in the United States during the 2009 influenza virus A (H1N1) pandemic. *PLoS ONE, 6*, e23610.

Diaconita, V. (2014). *Big data and machine learning for knowledge management*. 9th International Conference on Business Excellence, 9–10 November 2014.

Diaconita, V., Lungu, I., & Bara, A. (2009). Technical solutions for integrated trading on Spot, futures and bonds stock markets. *WSEAS Transactions on Information Science and Applications, 6*, 798–808.

Einav, L., & Levin, J. D. (2014, January). The data revolution and economic analysis. *Innovation Policy and the Economy, 14*, 1–24.

Finnegan, M. (2013). Boeing 787s to create half a terabyte of data per flight, Computerworld UK, published 06 March 2013.

Ginsberg, J., Mohebbi, M. H., Patel, R. S., Brammer, L., Smolinski, M. S., & Brilliant, L. (2009). Detecting influenza epidemics using search engine query data. *Nature*, *457*(7232), 1012–1014.

Hastie, T., Tibshirani, R., & Friedman, J. (2008). *The elements of statistical learning data mining, inference, and prediction* (2nd ed.). New York: Springer.

Ho, D. E. (2012). Fudging the nudge: Information disclosure and restaurant grading. *Yale Law Journal, 122*, 574–688.

Johnston, K., Ver Hoef, J., Krivoruchko, K., & Lucas, N. (2003). ArcGIS 9 Using ArcGIS Geostatistical Analyst, ESRI.

Jungherr, A. (2013). Tweets and votes, a special relationship: The 2009 federal election in Germany. In *Proceedings of the 2nd workshop on Politics, elections and data* (pp. 5–14). New York: ACM.

Kašćelan, L., Kašćelan, V., & Jovanović, M. (2014). Analysis of investors' preferences in the Montenegro stock market using data mining techniques. *Economic Research-Ekonomska Istraživanja, 27*, 463–482.

Keyling, R., & Jünger, J. (2013). Facepager (Version, f.e. 3.3). *An application for generic data retrieval through APIs*. Source code. Retrieved from https://github.com/strohne/Facepager

Lazer, D., Kennedy, R., King, G., & Vespignani, A. (2014, 14 March). The parable of Google Flu: Traps in big data analysis. *Science,* 343, 1203–1205.

Lee, Y. H., Wei, C. P., Cheng, T. H., & Yang, C. T. (2012). Nearest-neighbor-based approach to time-series classification. *Decision Support Systems, 53*, 207–217.

Mantelero, A., & Vaciago G. (2014). *Social media and big data, cyber crime and cyber terrorism investigator's handbook* (1st ed). Waltham, MA: Syngress (Elsevier).

Mustafaraj, E., & Metaxas, P. T. (2010, April 26-27th). From obscurity to prominence in minutes: Political speech and real-time search. In *Proceedings of the WebSci10: Extending the Frontiers of Society On-Line*, Raleigh, NC: US.

Polato, I., Ré, R., Goldman, A., & Kon, F. (2014). A comprehensive view of Hadoop research – A systematic literature review. *Journal of Network and Computer Applications, 46*, 1–25.

Ratkiewicz, J., Conover, M., Meiss, M., Gonçalves, B., Flammini, A., & Menczer, F. (2011, July). Detecting and Tracking Political Abuse in Social Media. In Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media (ICWSM-2011, pp 297–304), AAAI Press, Menlo Park, CA.

Schatz, M. C. (2009). CloudBurst: Highly sensitive read mapping with MapReduce. *Bioinformatics, 25*, 1363–1369.

Schnebele, E., Cervone, G., & Waters, N. (2014). Road assessment after flood events using non-authoritative data. *Natural Hazards Earth System Science, 14*, 1007–1015.

Thusoo, A., Sen, Sarma J., Jain, N., Shao Z., Chakka, P., … Murthy, R. (2010). Hive – A Petabyte Scale Data Warehouse Using Hadoop, *26th IEEE International Conference on Data Engineering*, Long Beach, CA, 996–1005.

Tumasjan, A., Sprenger, T. O., Sandner, P. G., & Welpe, I. M. (2010). Predicting elections with Twitter: What 140 characters reveal about political sentiment. *ICWSM, 10*, 178–185.

Varian, H.R. (2014). Big data: New tricks for econometrics. *Journal of Economic Perspectives, 28* (Spring 2014), 3–28.

Vernica, R., Balmin, A., Beyer, K.S., & Ercegovac, V. (2012). Adaptive MapReduce using situation- aware mappers, *Proceedings of the 15thinternational conference on extending database technology* (pp. 420–431). New York, NY: ACM.

Wang, C., Li, X., Chen, P., Wang, A., Zhou, X., & Yu, H. (2015). Heterogeneous cloud framework for big data genome sequencing. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on, 12*, 166–178.

Wang, Y., Que, X., Yu, W., Goldenberg, D., & Sehgal, D. (2011). Hadoop acceleration through network levitated merge, *Proceedings of international conference for high performance computing, networking, storage and analysis* (Vol. 57, pp 1–10). New York, NY, USA.

White, T. (2012). *Hadoop: The definitive guide* (3rd ed.). O'Reilly, 630 p. ISBN 978-1-4493-1152-0