

Big Consumer Behavior Data and their Analytics: Some Challenges and Solutions



Mihai Calciu, Jean-Louis Moulins, and Francis Salerno

Abstract This chapter contributes to the still very reduced marketing literature that deals with big consumer behavior data using cloud analytics by summarizing some of the main extant academic researches and by introducing new applications, datasets, and technologies in order to complete the picture. Both internal “purchase history” and external Web-based customer reviews and social media data are discussed, organized, and analyzed. They cover volume and variety aspects that define big data and uncover analytic complexities that need to be dealt with.

Keywords Big data · MapReduce · Text mining · Sentiment analysis · Social mining

Introduction

Nowadays it is virtually impossible to open a publication without seeing some reference to big data. Big data is not just another buzzword. Over the past few years it has emerged as the new frontier of IT-enabled innovation and brings upon disruptive changes in business, science, and society as a whole (Sanders 2016). The shock wave is so extensive that Harvard Business Review even refers to data scientist in one of its article titles as the “sexiest job of the twenty-first century” (Davenport and Patil 2012). Forrester (2011, p. 4) defines big data as “techniques and technologies that make handling data at extreme scale affordable.” Firms use these technologies

M. Calciu (✉)
Université de Lille RIME-Lab, Lille, France
e-mail: michel.calciu@univ-lille1.fr

J.-L. Moulins
Aix Marseille Université Cret-Log, Marseille, France
e-mail: jean-louis.moulins@univamu.fr

F. Salerno
Université de Lille LEM, Lille, France
e-mail: francis.salerno@univ-lille1.fr

in order to capture flows of information and analyze massive volumes of data in real time (Xu et al. 2016). Marketing is at the forefront of this challenge.

The big data revolution has been started and is best controlled by a few big information companies like Google, Amazon, Facebook, and Twitter while most companies are still struggling with the dilemma of “Data, data everywhere and not a byte to use.” There is some reason for this; big data is first defined by volume which is normally huge and needs enormous computing resources often meaning “monstrous” clusters of computers which only big IT companies can afford. The good news is that some democratization is going on. More or less “huge” computing infrastructure can be accessed through cloud technologies and big data calculations use know solutions like MapReduce and its improvements that hide complexities of massive parallel computing.

While some kind of big data monopoly for a few number of information companies still exists and industry is leading marketing academia in conducting cloud analytics, as acknowledged by Xu et al. (2016), there is good hope for improvement. This chapter tries to contribute to this trend by summarizing some of the main extant academic researches and by introducing new applications, datasets, and technologies.

Big data in a narrow sense can be defined by their *volume* (V) as data too large to be dealt with by one computer. Besides volume, BD analytics must deal with variety which means besides structured data also unstructured data like clickstreams, text, and recorded eye tracking. Such unstructured or semi-structured data cannot be categorized into regular relational databases and therefore need alternative storage approaches like NoSQL (Not Only SQL). A third aspect, *velocity*, indicating that data are generated, captured, and processed rapidly, extends the definition to a 3V model. This definition has been coined by Laney (2001) as “high-volume, high-velocity and high-variety information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision-making.” More recently, Gartner updated the definition of the 3V information assets as requiring new forms of processing to enable enhanced decision-making, insight discovery, and process optimization (Beyer and Laney 2012). The 3Vs have been extended in practice to 5V, adding data *value* and *veracity* as defining elements.

Compared to traditional data practices, big data brought over unprecedented opportunities for inquiry and experimentation. Sanders (2016) enumerates three differences. The first is that it has enabled real-time digital capture of data over the Internet. This gives researchers access to data on any type of phenomenon, which is exacerbated by the ability to capture and understand the content of human dialogue. The second difference is that the nature of inquiry has changed as data can be collected for testing hypotheses that have not yet been conceived, eventually without human intervention and allowing deep machine learning. This opens the way for computers to make discoveries by creating new knowledge. The third difference is that the nature of experimentation has changed. The Internet has created the ability to conduct large-scale experiments on many phenomena.

Variety and Volume Challenges of Big Consumer Behavior Data

When thinking of big data in terms of data tables we mean a lot of rows and sometimes a lot of columns. Traditional transactional marketing data collection methods were limited to small inferential data which were sample based or aggregated from enterprise data capture systems that were not originally set up to capture customer-level identification data. Tracking technologies, including Internet, and diffusion of database and customer relationship marketing approaches have substantially increased the volume and variety of individual consumer behavior data that are and can be collected. This volume and variety are best described by the datasets we use in our analysis.

The first file we are using can be considered as big data as it contains 343,766,402 transactions or rows (file size 9.57G) recorded during 78 weeks from 6,326,658 customers of a retail chain. It contains essentially three columns: customer identity followed by transaction date and amount. For confidentiality reasons the source of the data cannot be disclosed. We will call this dataset the *purchase history dataset*. It is used to predict future customer purchase behavior based upon aggregated customer-level variables like Recency, Frequency, and Monetary. Both the aggregation and prediction phases use cloud computing-based analytics.

Another file we use is the Amazon *customer reviews dataset* (courtesy McAuley et al. 2015, file size 58.3G) that contains 82.68 million reviews after deduplication (142.8 million reviews originally) spanning May 1996–July 2014. The two first reviews¹ separated by square brackets in json (JavaScript Object Notation) format are given in Listing 1.

Listing 1: The First Two Records in the Amazon Reviews Dataset

```
{ "reviewerID": "A00000262KYZUE4J55XGL", "asin": "B003UYU16G", "reviewerName": "Steven N Elich", "helpful": [0, 0], "reviewText": "It is and does exactly what the description said it would be and would do. Couldn't be happier with it.", "overall": 5.0, "summary": "Does what it's supposed to do", "unixReviewTime": 1353456000, "reviewTime": "11 21, 2012" }
[
  { "reviewerID": "A000008615DZQRR1946FO", "asin": "B005FYPK9C", "reviewerName": "mj waldon", "helpful": [0, 0], "reviewText": "I was sketchy at first about these but once you wear them for a couple hours they break in they fit good on my board an have little wear from skating in them. They are a little heavy but won't get eaten up as bad by your grip tape like poser dc shoes.", "overall": 5.0, "summary": "great buy", "unixReviewTime": 1357603200, "reviewTime": "01 8, 2013" }
```

¹Each review consists of the following labels: (1) reviewerID: the ID of the reviewer; (2) asin: the product ID of the item being reviewed; (3) reviewerName: the name of the reviewer; (4) Helpful: the first number is the amount of people who voted the review as being helpful and the second number is the amount of people who voted on the review; (5) reviewText: the entire review in text form; (6) overall: the rating out of 5 that the reviewer gave the product; (7) summary: a shortened version of the review; (8) unixReviewTime: time of the review; (9) reviewTime: time of the review in dd/mm/yyyy.

We will refer to this dataset as the *web reviews dataset*. It will be used to predict customer ratings from verbatim feedback.

In order to complete the data usage scenarios we present data from the only two academic marketing research papers using big data cloud analytics we have found in extant literature. The first by Liu et al. (2016) uses several unstructured Web sources of data, many of them user-generated data (UGD), from various Web platforms like Twitter,² Google,³ Wikipedia⁴, IMDB⁵, and Huffington Post⁶ in order to produce a structured prediction model for TV show ratings. Big data calculations are mobilized to produce the variables of the model by using powerful text mining techniques. We refer to this paper as the *text mining paper*. It uses three measures to extract content information from the unstructured text data, including Tweets, IMDB reviews, and Huffington Post news. They produce three datasets that will be called in order *volume*, *sentiment*, and *content* dataset. For Tweets they are described below. The *volume* dataset records how many times a TV show is discussed. Users mentioning a show are likely to watch and their social network is likely to be influenced to watch. In the *sentiment* dataset Tweets are classified by polarity, here positive and negative. Four percent of the Tweets that are used are labeled manually by experts and the rest are labeled automatically using the LingPipe⁷ linguistic analysis package. The *content* dataset uses a measure that makes inferences from the full content of the Tweets using the frequency of all n-grams of tweets in all analyzed TV shows. An n-gram is a continuous sequence of n words in the text. For example the Tweet “I love Pittsburgh Steelers” contains four 1-gram, three 2-grams, two 3-grams, and one 4-gram. Collecting the Tweets 24 h before the show produced 6,894,624 selected Tweets related to the 30 TV series with their 2339 episodes. The 2339 episodes are described by the frequency of the 28,044,202 n-grams resulting from those Tweets.

The second paper by Culotta and Cutler (2016) develops a fully automated method for inferring attribute-specific brand perception ratings by mining the brand’s social connections on Twitter. We refer to this paper as the “social mining” paper. It matches followers of exemplar accounts representing a perceptual attribute

²Selecting relevant tweets demands the use of four identifiers: (1) name of the show (e.g., Breaking Bad); (2) official Twitter account of the show (e.g., @TwoHalfMen_CBS); (3) a list of hashtags associated with the show (e.g., #AskGreys); and (4) the characters’ names on the show (e.g., Sheldon Cooper)

³Google Trends provides total search volume for a particular search item. For the TV series data, one can use the name of the show (e.g., Two and a Half Men) and character names on the show (e.g., Walden Schmidt) as the keywords.

⁴Many of the Wikipedia editors are committed followers of TV and edit-related articles earlier than the show’s release date. Wikipedia edits or views may be good predictors of TV ratings.

⁵Consumers also post reviews on discussion forums such as the IMDB, chosen here because it has the highest Web traffic ranking (according to Alexa) among all TV show-related sites.

⁶Consumers may also be driven to watch TV series by news articles. Huffington Post is a site that offers news, blogs covering entertainment, politics, etc. It ran 26th on Alexa as of January 29, 2015.

⁷<http://alias-i.com/lingpipe/demos/tutorial/sentiment/read-me.html>

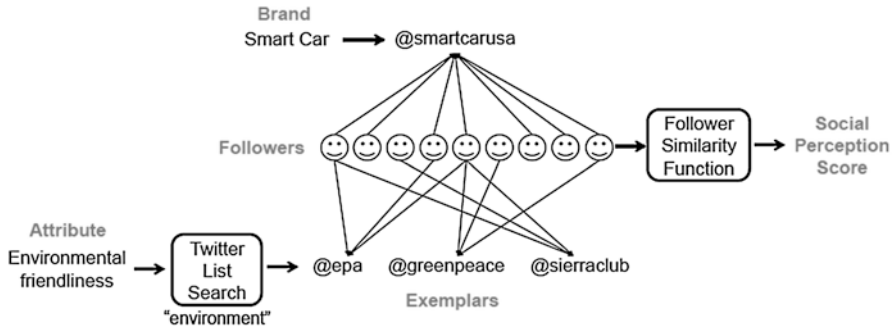


Fig. 1 Matching eco-friendly exemplars’ followers with brand followers. Source: Cullota and Cutler (2016)

with the followers of brands. GreenPeace for example is an exemplar for the eco-friendliness attribute of a brand. The *brand-followers dataset* used Twitter’s API to collect up to 500,000 followers for each brand. It consists of 239 brand lines containing their names and the IDs of their followers (30.6M followers, 14.6M unique, 314M file size). The distribution of these brands by sector is Apparel 70, Cars 37, Food and Beverages 70, and Personal Care 62. The *exemplar-followers datasets* collect for each of the exemplar accounts the IDs of up to 50,000 of their Twitter followers. They consist of 74 eco-friendly exemplars (2.0M followers, $\lt; 1.0\text{M}$ unique, 25M file size), 110 luxury exemplars (4.4M followers, 2.3M unique, 46M file size), and 405 nutrition exemplars (4.7M followers, 2.7M unique, 48M file size).

Figure 1 illustrates the core methodology that was used in order to match brand and exemplar followers in order to adapt a similarity function between the brand and the attribute represented by exemplars.⁸ The privileged measure was the Jaccard index that defines the similarity of two sets as the cardinality of their intersection divided by the cardinality of their union:

$$J(X,Y) = \frac{|X \cap Y|}{|X \cup Y|} \tag{1}$$

In order to keep brands with different numbers of followers comparable, in the Social Perception Score (SPS) the Jaccard scores are normalized by weighting each exemplar inversely proportional to its number of followers:⁹

⁸For the example illustrated in figure 1, the brand smartcar had 11,052 followers, out of which 953 (8.6%) were also followers of environmental friendliness exemplars.

⁹This is analogous to the “inverse document frequency” adjustment used in information retrieval to encourage documents containing rare query terms to be ranked higher than documents containing common query terms (Manning et al. 2008).

$$\text{SPS}(B,E) = \frac{\sum_{E_i} \frac{1}{|F_{E_i}|} J(F_B, F_{E_i})}{\sum_{E_i \in E} \frac{1}{|F_{E_i}|}} \quad (2)$$

The higher this affinity score, the more strongly consumers associate the brand B with the attribute represented by the set of exemplars (E).

What all these datasets have in common is that they are or can virtually become big data. They illustrate the first two V s that define big data, volume, and variety. The third V , velocity, is fundamentally linked to the techniques and technologies that make handling data at extreme scale affordable. These are essentially cloud computing and parallel computing techniques on computer clusters. Cloud computing provides easy and relatively inexpensive access to huge computing capacities. With clouds there is potentially no limit for the size of data and calculations. Parallel computing techniques on computer clusters that have revolutionized big data calculations are essentially based on the MapReduce approach and its evolutions. These need to be understood and dealt with as they impact all algorithms and models used to analyze data at a larger scale.

Big Data Calculation Velocity with MapReduce and Spark

MapReduce Calculations on the Analyzed Datasets

MapReduce is a high-level programming model and an associated implementation for large-scale parallel data processing. It has the merit to hide all complexities of parallel computing on distributed servers from users and to have contributed massively to democratize big data processing. The name MapReduce originally referred to the proprietary Google technology (Dean and Ghemawat 2004), but has since become a generic trademark. Its most popular implementation is part of Apache's Hadoop,¹⁰ an open-source software framework, written in Java, for distributed storage and distributed processing of very large datasets on computer clusters built from commodity hardware. MapReduce is based on the observation that most computations can be expressed in terms of a Map() procedure that performs filtering and sorting and of a Reduce() procedure that performs aggregating operations. The map and reduce steps can be parallelized on a cluster of computers. Between the map and

¹⁰Hadoop is an open-source software framework that allows the distributed processing of large datasets across clusters of computers. It contains (1) the Hadoop Common package, which provides file system and OS-level abstraction; (2) Yarn, a MapReduce engine; and (3) the Hadoop Distributed File System. These mechanisms automatically break down jobs into distributed tasks, schedule jobs, and tasks efficiently at participating cluster nodes, and tolerate data and task failures.

Table 1 MapReduce for calculation tasks on the analyzed datasets

Dataset	Task	Key	Value	Reduce
Purchase history	Frequency	Customer ID	1	Summation
Purchase history	Monetary	Customer ID	Amount	Summation
Purchase history	Recency	Customer ID	Date	Maximum
Text mining volume model	Select relevant tweets/wiki	Keyword	1, text	Summation
Text mining content model	N-gram frequency count	N-gram	1	Summation
Text mining PCA	SSVD-Matrix multiplication	Matrix row index	Matrix row vector	Null
Text mining PCA	SSVD-Orthogonalization	Sub-matrix	QR matrix	Summation

reduce, the process involves shuffling and sorting the keys so that all key-value pairs of the same key go to the same reduce for the next step.

Table 1 summarizes the implementation of the MapReduce approach to the various computing tasks implied by the datasets described before. MapReduce can be best understood when analyzing aggregation tasks in the purchase history dataset like computing transaction Frequency, Monetary value, and Recency per customer. The map phase implies sorting customer transactions by name, date, or amount into queues, one queue for each name. The reduce phase then performs aggregating operations such as counting the number of transactions in each queue, yielding customer purchase Frequency, or retaining the maximum date, yielding customer Recency or summing transactions amount, yielding customer Monetary amount. Some of the *text mining* tasks like selecting relevant tweets in the *volume dataset* or counting the n-gram frequency in the *content dataset* mentioned above are very similar to the purchase Frequency calculation.

While computing customer RFM variables in marketing are quite straightforward applications of the MapReduce approach, more sophisticated marketing science models need to and many can be adapted for MapReduce. Many models and calculations used in marketing science and data analysis use linear algebra calculations. One very important calculation that needs to be adapted to the MapReduce approach is matrix multiplication. Multivariate models like linear regression, principal components analysis (PCA), or discriminant analysis use computationally more sophisticated algorithms over a summary, often symmetric matrix of rather small dimensions given by the number of variables. This small matrix is obtained by applying matrix multiplication to so-called tall-and-skinny big data data matrices resulting from customer recordings, observations, or declarations. A matrix X with m rows and n columns is tall and skinny if it has many more rows than columns ($m \gg n$). A special case of matrix multiplication that is central to the abovementioned data analysis methods is the multiplication of the transposed matrix with itself. This matrix multiplication has a rather straightforward solution with MapReduce.¹¹ The multiplication can be done chunk wise in memory in the map

¹¹The same applies to that is also needed when estimating linear regression

phase and the reduce phase will simply sum up the output of all those multiplications, exploiting the fact that the sum of matrices is commutative while multiplication is not. Formally in the map stage, each task collects a chunk of rows to form a local matrix and then computes. The reduce stage then sums up the results of these local multiplications. Further, it can be shown (see Benson et al. 2013) that QR factorization together with single vector decomposition (SVD) is a fundamental matrix from which decomposition can be obtained. The upper triangular matrix R can be obtained by applying Cholesky decomposition to and the orthogonal matrix Q results from. Both QR factorization and SVD are used in the MapReduce implementation of PCA mentioned in Table 1.

In order to incorporate all of the content information, the text mining model uses the frequency of all n -grams as features. This generates a significant feature space with 28,044,202 n -grams resulting from Tweets describing the 2339 TV episodes mentioned above. This produces a “short-and-fat” matrix that is too large to be stored in memory. Applying the PCA dimension reduction technique on such a “fat” matrix required the use of SSVD (stochastic SVD) method developed by Halko (2012). Both SSVD and the related SVD methods are available in the open-source Mahout machine learning library that applies MapReduce on Hadoop. SVD adapted for MapReduce breaks down into two basic operations, which are matrix multiplication and orthogonalization, as indicated in Table 1. As a result four principal components (PC) from the 28,044,202 n -gram features could be selected using the “elbow” rule. Phrases such as “tonight,” “can’t wait,” and “watch” have the largest projection on the first PC. Overall, the first four PCs cover consumers’ intention to watch the shows and are the most important independent variables in the final regression model explaining TV show ratings (R-squared 0.756) which is comparable with the R-squared of the model with only the lagged rating included. Both the volume and the sentiment model have little explanatory power. This is also the main contribution of this *text mining paper*, showing that easily accessible online content such as Twitter Tweets carefully extracted, sorted, and reduced using big data techniques can provide timely representations of consumer intentions.

As we could see computing solutions need to be adapted to MapReduce because conventional analytics would not work as the data is split across nodes. MapReduce is not suitable for all problems; new programming models and frameworks are still being created that build upon these ideas.

Apache Spark, an Enhanced MapReduce

As acknowledged by some authors and reviewers in the latest special issue on Big Data of Marketing Science (Liu et al. 2016), academia lags behind industry in conducting cloud analytics. They state that cloud tools, like Spark, used by the industry go beyond the simple MapReduce programming model and suggest that future marketing research that requires large-scale data analytics should consider adopting these tools and explicitly. Our paper introduces Apache Spark which is today’s most

powerful and popular enhancement of MapReduce. Spark implements directed acyclic graphs (DAGs) and resilient distributed datasets (RDDs). DAG is a programming style for distributed systems. It can be seen as an enhanced alternative to MapReduce. While MapReduce has just two stages (map and reduce), DAG can have multiple stages that can form a lineage or a tree structure and is therefore more flexible due to more high-level functions like map, filter, union, and join. Also DAG execution is faster due to intermediate results not being written to disk. RDDs are "... parallel data structures that let users explicitly persist intermediate results in memory, control their partitioning to optimize data placement, and manipulate them using a rich set of operators" (Zaharia et al. 2012). Spark takes MapReduce to the next level with less expensive shuffles in the data processing. With capabilities like in-memory data storage and near-real-time processing, the performance can be several times faster than other big data technologies. Spark is also very concise in terms of lines of code. Its first published version was only 14,000 lines of code written in Scala, while Hadoop's MapReduce that was written in Java was almost ten times bigger (Zaharia 2014). While Spark has replaced MapReduce which is closely associated to Hadoop, the two solutions are complementary. Hadoop remains essentially a distributed data infrastructure and a format for storing data that can be processed by Spark or other programs.

In order to use the Spark big data engine on our datasets we had to choose among the four available languages that can be used. It is a real dilemma whether to choose specialized statistical languages like R or general-purpose languages like Scala, Java, or Python. We preferred the Scala language as it is the native language in which Spark was written. For more details about scalability, elegance, and conciseness of Scala one could read Odersky et al. (2011). Also in order to better understand what recommends Scala as a platform for statistical computing and data science one could refer to D. Wilkinson's research blog articles (Wilkinson 2013).

Predicting Consumer Ratings from Amazon Reviews Using Spark

Finally we use the *web reviews* dataset mentioned earlier in order to predict consumer ratings. Every day millions of users leave reviews and ratings on products using Amazon. Besides influencing sales, helping other potential buyers figure out whether or not they should purchase the specific product, they provide valuable feedback to the companies that sell those products. The Amazon Reviews dataset has widely been used in machine learning research (McAuley et al. 2015; Martin and Pu 2014) and various models have been tested as to their ability to predict consumer ratings from features extracted from the reviews and from the other non-text elements that are included. The lasso regression, applied here, uses a form of regularized least squares that like Ridge regression is suited when the number of independent variables is big, and has the advantage over the latter to automatically select

more relevant features and discard the others. It presents a somehow complementary exercise to the text mining content model presented earlier. While the latter was predicting TV show ratings from Tweets' n-grams reduced with PCA, here we predict Amazon consumer ratings from reviews' unigrams by selecting a reduced number of more relevant features. It also illustrates the flexibility of Spark compared to the classical MapReduce approach.

The calculations that are listed in the Appendix are organized in a pipeline that chains several operations: tokenizing, hashing with term frequency and lasso regression (see line 10). The tokenizer extracts and counts the individual words (line 11). For the review text a bag of words (BOW) representation is constructed. Each distinct word (token, unigram) defines a feature (independent variable) of each of the reviews in both the training and test sets. HashingTF (line 12) is a transformer which takes BOWs and converts them into fixed-length feature vectors. After this stage a "featurized" short review text would look like this:

```
Array([0,Hi I heard about Spark, WrappedArray(hi, i, heard, about, spark),(20,[0,5,9,17],[1.0,1.0,1.0,2.0]))).
```

It is an array containing a vector with four positions. The first is the identifier of the customer, the second the review text, the third a wrapped array with the tokenized words, and the last a sparse matrix with 2 rows and 20 columns corresponding to the number of distinct words used in a small set of mock reviews. The first row indicates the position of the word in the total word list and the second shows how frequently the word was used. After defining the pipeline that chains transformations, an initial parameter grid is prepared (line 15) and also an evaluator that uses R2 as a goodness-of-fit metric (line 18). Finally the cross-validation procedure combines the pipeline as an estimator, the evaluator, and the parameter grid (line 19) while real calculations occur much later when cross-validation is launched to fit over the training data (line 22). Once the model has been estimated over the reviews training set its performance can be evaluated over the test set (line 25). The R2 score (line 26) and rating predictions can be given by using the calibrated model with customer reviews from the test set (line 29).

Finding suitable models and methods like the lasso (least absolute shrinkage and selection operator) capable of reducing the feature space generated by text mining approaches in order to predict consumer ratings or sentiments should remain an important research direction for the future. Computationally more sophisticated models might need to be adapted to the enhanced MapReduce approaches that make big data calculations possible.

Adapting the content model (Liu et al. 2016) from Tweets to the customer reviews context besides potentially opening new research direction can also contribute to the external validity of that model.

Conclusion and Discussions

Marketing relies more and more on information technology. From channel choice to personalization and recommendation systems, user-generated content, online reviews, and social influence in social networks (Goes 2014, p 3), marketing is now considered to be the driver of big data technologies, just like accounting was for the databases in the 1980s (Albescu and Pugna 2014). Important market transformations have been generated by leading e-commerce enterprises such as Amazon and eBay through their innovative and highly scalable e-commerce platforms and recommender systems (Bello-Orgaza et al. 2016). Amazon is also a forerunner in cloud technology with its Amazon Web Services (AWS) and especially its Elastic Cloud Computing Services (EC2).

Technology has always been transforming marketing science following a rather systematic and predictable trend. By paraphrasing Rust and Huang (2014) we would say that, by enabling big ubiquitous customer communication, technology pushed marketing science to resemble to a greater degree formerly specialized areas like direct marketing and more recently service marketing. This implies changes in both the topics and methods to be employed and increased emphasis on marketing analytics. Academic marketing scientists, while having good knowledge in statistics, econometrics, and operations research, seem to have poor knowledge in modern programming and IT. They tend to ignore features that take advantage of the distributed and data-rich context provided by the Internet. The cloud becomes more and more the place where big data calculations are done. Under these circumstances marketing models can no longer remain “small and beautiful.” They must be capable to grow to become scalable and be able to deal with big data. Neglecting factors that enhance usability of models risks to make the latter irrelevant and limit their use. “To develop models that do get used, modelers must pay attention to the IT infrastructure under which their models will be used” (Lilien and Rangaswamy 2000, p. 232).

By trying to demystify big data approaches our paper invites marketing scientists to pay more attention to technologic evolutions, to become more involved in developing specific analytics, and not to leave the battlefield completely to computer or data scientists. The same invitation goes towards managers who according to an IBM study (2011) of 1700 CMOs from 19 industries and 64 countries revealed that 71% feel their organizations are unprepared to handle the explosion of big data. Demystifying big data approaches and technologies does not mean banalizing them but on the contrary we insist upon the high importance and groundbreaking changes they generate for the human society in general and for marketing science in particular. This chapter presents a hands-on approach to big data in marketing. To our knowledge it is the first attempt to apply the newer enhanced MapReduce technologies like Spark to marketing science problems. Explaining their importance, relative simplicity, and applicability, based on a variety of marketing datasets, can contribute to the adoption of big data computational techniques among marketing scientists.

It becomes clear that decision support modeling and data analytics are undergoing an industrialization process during which academia will continue to build “big models for small data” and the industry will build “small models for big data.” This gap between academia and industry has some objective grounding due to limited access to real big data and the fact that not all models can be transposed easily for high-performance computing (HPC) and particularly for MapReduce frameworks. The latter will become part of the marketing analyst ecosystem and the quick developments will privilege simpler modeling approaches, speed improvements in algorithms, and applications that reduce the dimensionality of data through aggregation, selection, sampling, etc. For a thorough discussion on this subject one could refer to Wedel and Kannan (2016). However academic marketing scientists cannot isolate themselves in an ivory tower.¹² Marketing scientists are deemed to develop models that solve managerial problems and can be used by an industry where the big data paradigm is gaining ground.

Appendix

Listing 2 Measuring customer sentiment on the Amazon Reviews Dataset*

```

1. import org.apache.spark.ml.feature.{HashingTF,
   IDF, Tokenizer}
2. import org.apache.spark.ml.regression._
3. import org.apache.spark.ml.{Pipeline,
   PipelineModel}
4. import org.apache.spark.ml.tuning.
   {ParamGridBuilder, CrossValidator}
5. import org.apache.spark.ml.evaluation.
   RegressionEvaluator
6. //Load dataset and cache it
7. val data = spark.read.json(/media/storage1/
   reviews-train.json).cache()
8. //Define a pipeline combining text feature extrac-
   tors + linear regression
9. val tokenizer = new Tokenizer().
   setInputCol("reviewText").setOutputCol("words")
10. val hashingTF=new HashingTF().setInputCol("words").
   setOutputCol("features")
11. val lasso = new LinearRegression().
   setLabelCol("overall").setElasticNetParam(1.0).
   setMaxIter(100)

```

¹²For quantitative methods and model builders this privilege in our opinion is only reserved to pure “creators of mathematics.”

```

12. val pipeline = new Pipeline().
    setStages(Array(tokenizer, hashingTF, lasso))
13. val paramGrid = new ParamGridBuilder().
    addGrid(lasso.regParam, Array(0.005, 0.01, 0.05)).
    build()
14. //Define evaluation metric
15. val evaluator = new RegressionEvaluator().
    setLabelCol("overall").setMetricName("r2")
16. val cv=newCrossValidator().setEstimator(pipeline).
    setEvaluator(evaluator).setEstimatorParamMaps(paramGrid)
17. //Run everything!
18. val cvModel = cv.fit(data)
19. //Evaluate on test data:
20. val test = spark.read.json("/media/storage1/
    reviews-test.json")
21. var r2 = evaluator.evaluate(cvModel.
    transform(test))
22. println("Test data R^2 score:", r2)
23. val sparkPredictions = cvModel.transform(test)
24. sparkPredictions.write.format("json").
    mode("overwrite").save(/media/storage1/predictions.json)

```

*The listing is adapted by us to Scala from a Python version (Bradley 2016)

References

- Albescu, F., & Pugna, I. B. (2014). Marketing intelligence—The last frontier of business information technologies. *Romanian Journal of Marketing*, 3, 55–68.
- Bello-Orgaza, G., Jungb, J. J., & Camacho, D. (2016). Social big data: Recent achievements and new challenges. *Information Fusion*, 28, 45–59.
- Benson, A. R., Gleich D. F. & Demmel J. (2013). Direct QR factorizations for tall-and-skinny matrices in MapReduce architectures, *2013 IEEE International Conference on Big Data*, October 6–9, TBD Silicon Valley.
- Beyer, M. A., & Laney, D. (2012). *The importance of 'big data': A definition*. Stamford, CT: Gartner.
- Bradley, J. (2016). *Apache® Spark™ MLlib: From Quick Start to Scikit-Learn*. Retrieved October, 2017, from <http://go.databricks.com/spark-mlib-from-quick-start-to-scikit-learn>.
- Culotta, A., & Cutler, J. (2016). Mining brand perceptions from twitter social networks. *Marketing Science*, 35(3), 343–362.
- Davenport, T., & Patil, D. (2012). Data scientist: The sexiest job of the 21st century. *Harvard Business Review*, 90(10), 70–76.
- Dean, J. & Ghemawat, S. (2004, December). MapReduce: Simplified data processing on large clusters, *OSDI'04: Sixth symposium on operating system design and implementation*, San Francisco, CA.

- Forrester, (2011). *Expand your digital horizon with big data*. Forrester. Retrieved May 27 from http://www.asterdata.com/newsletter-images/30-04-2012/resources/Forrester_Expand_Your_Digital_Horiz.pdf Accessed July 7, 2017.
- Goes, P. (2014). Big data and IS research. *MIS Quarterly*, 38(3), III–VIII.
- Halko, N. P. (2012). *Randomized methods for computing low-rank approximations of matrices*. Unpublished doctoral dissertation, University of Colorado, Boulder.
- IBM. (2011) *From stretched to strengthened—Insights from a global CMO study*. Retrieved September 17, 2015, from <http://www.ibm.com/services/us/cmo/cmstudy2011/downloads.html>.
- Laney, D. (2001). *3D data management: Controlling data volume, velocity, and variety*, technical report. Retrieved October, 2017, from <https://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf>.
- Lilien, G. L., & Rangaswamy, A. (2000). Modeled to bits: Decision models for the digital, networked economy. *International Journal of Research in Marketing*, 17, 227–235.
- Liu, X., Singh, P. V., & Srinivasan, K. (2016). A structured analysis of unstructured big data by leveraging cloud computing. *Marketing Science*, 35(3), 363–388.
- Martin, L. & Pu, P. (2014). *Prediction of helpful reviews using emotions extraction*. AAAI Publications.
- McAuley, J., Pandey, R. & Leskovec J. (2015). Inferring networks of substitutable and complementary products, *KDD '15 Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794.
- Odersky, M., Spoon L., Venners B. (2011), Programming in Scala. In *A comprehensive step-by-step guide* (2nd ed) (January 4, 2011), Artima Inc.
- Rust, R. T., & Huang, M. H. (2014). The service revolution and the transformation of marketing science. *Marketing Science*, 33(2), 206–221.
- Sanders, N. R. (2016). How to use big data to drive your supply chain. *California Management Review*, 58(3), 26–48.
- Wedel, M., & Kannan, P. K. (2016). Marketing analytics for data-rich environments. *Journal of Marketing*, 80(6), 97–121.
- Wilkinson, D. (2013). *Scala as a platform for statistical computing and data science*. Retrieved October, 2017, from <https://darrenjw.wordpress.com/2013/12/23/scala-as-a-platform-for-statistical-computing-and-data-science/>
- Xu, Z., Frankwick, G. L., & Ramirez, E. (2016). Effects of big data analytics and traditional marketing analytics on new product success: A knowledge fusion perspective. *Journal of Business Research*, 69(5), 1562–1566.
- Zaharia, M. (2014). *An architecture for fast and general data processing on large clusters*, University of California at Berkeley, Technical Report No. UCB/EECS-2014-12.
- Zaharia, M., Chowdhury M., Das T., Dave A., Ma J., McCauley M., Franklin M. J., Shenker S., Stoica I. (2012, April). *Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing*, NSDI 2012.