# Towards a Natural Experiment Leveraging Big Data to Analyse and Predict Users' Behavioural Patterns Within an Online Consumption Setting

Raffaele Dell'Aversana and Edgardo Bucciarelli

**Abstract.** The authors develop a model for multi-criteria evaluation of big data within organisations concerned with the impact of an ad exposure on online consumption behaviour. The model has been structured to help organisations make decisions in order to improve the business knowledge and understanding on big data and, specifically, heterogeneous big data. The model accommodates a multilevel structure of data with a modular system that can be used both to automatically analyse data and to produce helpful insights for decision-making. This modular system and its modules, indeed, implement artificial intelligent algorithms such as neural networks and genetic algorithms. To develop the model, therefore, a prototype has been built as proof-of-concept using a marketing automation software that collects data from several sources (public social and editorial media content) and stores them into a large database so as the data can be analysed and used to implement business model innovations. In this regard, the authors are conducting a natural experiment - which has yet to be completed - to show that the model can provide useful insights as well as hints to help decision-makers take further account of the most 'satisficing' decisions among alternative courses of action.

**Keywords:** Computational behavioural economics
Online consumption setting · Natural experiments in economics
Big data · Computational intelligence

**JEL codes:** C81 · C99 · D12 · D22

## 1 Introduction

Although it is currently being experimented on real cases and opened to further experimentations in microeconomics, this paper is conceived in the framework of the theory of computation and computational complexity which identifies, largely, the foundations of computational behavioural economics (for an insightful survey, see [1])

and classical behavioural economics (for a critical discussion, see [2]). In this framework and based on previous studies, *e.g.* [3–5], the authors build a model including measurable indicators with the aim of designing and running a natural experiment, and analysing its outcome. The main research goal is to encompass a reproducible model to be used effectively in order to manage and analyse big data together with the evolution of their economic value over time. The model is made up of several integrated parts: (i) definition of the theoretical framework; (ii) data collection; (iii) data elaboration and analysis; (iv) reporting; (v) knowledge database and improvement actions (implementation and follow up).

More specifically, the authors integrate their model in a marketing automation software (MA). As is common knowledge, MA helps organisations stay connected with their audience automatically. In particular, it refers to technologies designed for marketing departments to more effectively market on multiple channels and automate repetitive tasks. In this business area, organisations regularly use electronic communications and digital information processing technology in business transactions (e-commerce) to define and redefine relationships for value creation both between or within organisations, and between organisations and final users. In doing so, organisations promote their activities through social networking and microblogging services, trying both to increase the number of users and to convert them into followers and consumers or, better yet, into customers. In addition to publishing contents on social networks and blogs, organisations customarily plan and implement direct marketing policies on proprietary and licensed databases, by which promoting their products-or-services, even focusing on specific targets, considering indistinctly all potential users or performing a stratified analysis or a meta-analysis of them. The MA automatically collects large dataset from the market (through e-commerce and social channels) including subjective information on final users, be they anonymous, simple followers, sporadic consumers, or loyal customers. Moreover, the MA provides advanced tools for data analysis. These tools enable social networking and microblogging management (*e.g.*, publishing and editing) as well as marketing automation services (*e.g.*, lead scoring, automated drip campaigns, and CRM integration). In this paper, as mentioned earlier, the authors show an integration between the MA and the model proposed by them. Through this integration, all the data collected by the MA will flow into the model for subsequent data-mining. In particular, the model provides automated analysis tools via artificial intelligent algorithms (especially based on deep learning algorithms). Through these tools the model will automatically analyse and predict users' behavioural patterns, suggesting the best marketing actions for each of them in an online microeconomic setting.

## 2 Data Structure

To describe how the model works and how it integrates with the MA, we need to introduce, briefly, the structure of the MA database and the differences with the model database proposed, formally designed in Sect. 4. To get to the point and ease the understanding, we start from a typical use case, where the MA database is equipped with the data of a hypothetical organisation having (i) an e-commerce website;

(ii) a blog or microblog website where news of interest is published for some potential final users; (iii) a few social network channels; and (iv) a mailing list to send newsletters. For each user, the MA database collects all the actions she makes. For instance, for a specific user, we have knowledge about any comments posted, *likes* added, newsletters opened, links clicked on the e-commerce platform and financing services, products/services bought on it, and so on. For each action performed by the user, furthermore, the date, time as well as usage time are shown so as to sort each action in various orders (*e.g.*, chronological order). Following the chronological order, researchers can discover, among others, that a single user might have behaved anonymously on the social networks for a certain period of time, then she might have subscribed to the newsletter service on the blog, might have acquired a certain content of interest (*e.g.*, on certain newsletters), and finally might have bought some products/services. In the meanwhile, she might have also surfed the e-commerce platform looking for specific products and/or services and visiting specific web pages. For this reason, there are a lot of data available for each user registered into the database, thus actually researchers have to deal with a huge amount of data, namely, big data. This applies to all types of organisation, not just those oriented to the sale of products/services, in order to enhance their mission readiness and familiarisation of those forces operating in a certain microeconomic and social environment.

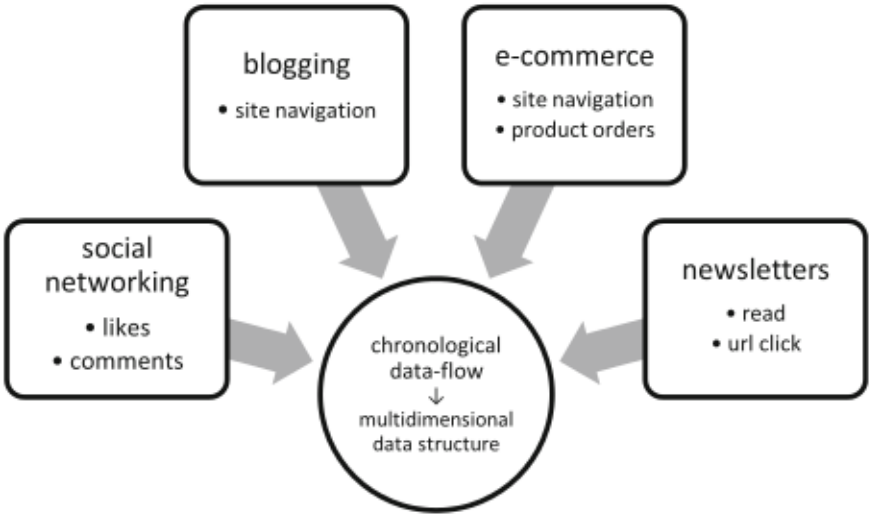


**Fig. 1.** The MA database is built by continuous automatic collection of big data from multiple sources within an online consumption setting. The multidimensional structure, discussed in Sect. 4, is built from the MA database as a single chronological data-flow.

All these data are transformed and imported as a single chronological data-flow in a large database, structured as a multidimensional data structure that can be viewed as a tensor in the way detailed in Sect. 4. The data-flow structured as multidimensional data gives us the opportunity to study users' behavioural pattern, and to predict what will be the most 'satisficing' decisions [6] among alternative course of action and, thus, the most

'satisficing' marketing actions to sort out by the organisation. If the database is big enough, that is, if it has many registered users with a lot of data for each of them, it becomes possible to better analyse their behavioural pattern, discover possible strategies, and use these patterns to predict possible targets and suggest further marketing actions.

## 3 Automated Data Analysis

With the aim to move beyond the manual analysis of final users' behavioural data mostly performed by humans, and thus implement automated analysis, note that in the MA the researcher can analyse data basically doing manual mining, looking at pre-defined reports, and then deciding what might be the most 'satisficing' decisions among alternative course of marketing action (the final objective is usually to increase sales). However, many organisations, even small to midsize businesses, tend to have multiple lines of ongoing business, each with its own marketing operations. In the course of data collection and analysis processes, therefore, what we are working on is to start with a specific research question applied to microeconomics of organisations (think, for example, of the causal inference in microeconomics and marketing [4, 5, 7]) and try to figure out what interesting themes are automated analysis of data using typical deep learning AI tools, especially neural networks and genetic algorithms (for an overview, see [8], particularly [9–13]). For example, suppose to answer to the following research question: "*Which users are likely to buy if we contact them through a newsletter?*". We know that if we contact every user without applying some filtering criteria, we might intercept some users that might buy products/services, but we might even incur possible negative effects on profitability: some users might decide to unsubscribe from the newsletter because of their frequency, while some other users might decide not to buy because the content of the newsletters might not meet their needs (maybe we promote 'wrong' products/services or maybe users might be receptive to incentives to buy, like a special offer or a discount).

For that reason, starting from a specific research question we aim at identifying final users who responded positively to earlier newsletters (that is, bought products/services after receiving a newsletter). This is accomplished through data filtering criteria in our tensor-based multidimensional structure. The users chronological action list will then be submitted to an appropriate AI algorithm to be trained, and then we will use the trained algorithm to find which final users to contact and the most tailored newsletter. Our first results suggest that there is a positive correlation between the actions made by the users and several optimistic reactions to receiving newsletters.

In a nutshell, the algorithm and the related integration schema can be represented as follow. After identifying a specific research question, we find out – by standard mining on the multidimensional data structure – the final users within the database that satisfied similar research questions in the past. The big database is split in two datasets: the first is used as input to develop appropriate AI algorithms, while the second is used to verify the effectiveness of trained algorithms. Once we find an effective algorithm, this can be integrated into the MA in order to have a type of decision support system so that the algorithm can automatically suggest marketing actions without further manual intervention made by the researchers (Fig. 2).

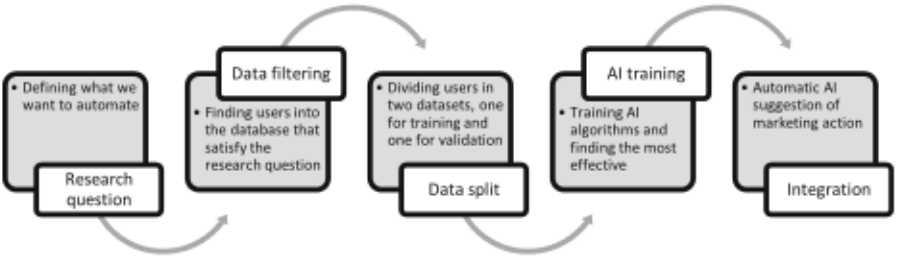


**Fig. 2.** The process to implement the AI automation into the MA software.

As in a previous framework [3], we define multilevel indicators to help find an answer to the research question. However, these indicators are built not only starting from measurable quantities but they could come from the output of AI algorithms, too. To ease the integration of our model into the MA software, we decided to let the model work as a black box: the algorithm that answers a specific research question can be written in any language (we are working, *e.g.*, with Haskell and Scala programming languages [14, 15]). The only requirement is to adhere to an application programming interface (API) according to a specification that enables the ability to communicate with our model, integrated into the MA.

## 4 Design of the Model Database

Designing more formally the architecture of our large database and how it is structured by our model data collection, we denote the representation of our database by $D$, whereby, we have:

$$D = \{C, \boldsymbol{A}\}$$

$$C = \{C_1 \ldots C_i \ldots C_n\}$$

$$\boldsymbol{A} = \{A_1 \ldots A_i \ldots A_n\}$$

where $C = \{C_1 \ldots C_i \ldots C_n\}$ is the set of final users, $n = |C|$ is the cardinality of that set, and $\boldsymbol{A} = \{A_1 \ldots A_i \ldots A_n\}$ is the tensor of all the actions registered for each user (see Fig. 1). In other words, each user $C_i$ represents a single tracked user (anonymous, simple followers, sporadic consumers, or loyal customers), while all the related actions are represented by the tensor $\boldsymbol{A}_i$ (note that $\boldsymbol{A}_i$ is a sub-tensor of $\boldsymbol{A}$ and is in turn a tensor).

As aforementioned, $C_i$ represents the data structure containing the data identifying the user $i$ (*e.g.*, personal data such as her email addresses, web pages visited, communities attended, etc.); the ability to identify the user is used to implement marketing actions towards her (for example, to send an email to a user we need to know her email address otherwise the marketing action cannot be performed).

$\boldsymbol{A}_i$ represents the tensor of all the actions registered for the user $i$ (see Fig. 1) and is the most interesting element of the database, because it enables the possibility of doing automated data analysis and studying the evolution of the data over time. $\boldsymbol{A}_i$ can be defined as follows:

$$\boldsymbol{A}_i = \left\{t_{i,j}, a_{i,j}, p_{i,j}\right\} j > 0 \tag{1}$$

In the formula (1) we define the actions of each user as a (potentially infinite) sequence of triplets stored as a tensor. Each triplet contains the time $t_{i,j}$ when the action happened, the type $a_{i,j}$ that characterises the action and the payload $p_{i,j}$ that is the content of the action. For each $i$ and $j$ we have that $a_{i,j} \in T$ where $T$ is a finite set of possible action types, while the payload is in general a structured content where the structure depends on the action type.

The action type is needed to distinguish between the several possible actions and to characterise the content of the payload. The most typical action types are social likes, web pages opened and visited, subscription to newsletters, products/services bought from the e-commerce website, newsletters opened and read, and so on.

Focusing on what happens when a user makes an action monitored by the MA software, let us suppose that a particular user $C_k$ adds a *like* on a post present on a social page, and suppose that $|\boldsymbol{A}_k| = m$, that is, we have $m$ triplets in $\boldsymbol{A}_k$. When the model imports the data, the model detects the new action and registers it: a triplet $\left\{t_{k,m+1}, a_{k,m+1}, p_{k,m+1}\right\}$ is added to $\boldsymbol{A}_k$ where $t_{k,m+1}$ is the time when the user added her *like*, $a_{k,m+1}$ is the action type (in this case a "social *like*"), and $p_{k,m+1}$ is the payload. The payload content is a specific structured data type, different for each type of action and contains specific details of the action. In this example of social *likes*, the typical payload will contain, among other data, the textual content of the post which received *likes* with their social hashtags. In the long run, our model accumulates a long series of actions for each tracked user. All these data for each user $i$ include a chronological order given by the $t_{i,j}$ and can be filtered by the action type. Moreover, as mentioned above, the payload content can be used for deep contextual analysis, because contains information about each action (*e.g.*, product ordered, web pages visited, content of the social post with hashtags) and it is useful to do semantic analysis.

As discussed in Sect. 3, the AI algorithms can access the multidimensional data using appropriate operators. The most common one is the *selection operator*, that is used to filter over the data to obtain the interesting actions for the specific research question. The *selection operator* is defined, as in relational algebra, as follows:

$$\boldsymbol{A}_i^{'} = \sigma_C(\boldsymbol{A}_i) \tag{2}$$

Where $\sigma$ is the selection operator, $C$ is the logical expression regarding the filtering condition on $\boldsymbol{A}_i$, and $\boldsymbol{A}_i^{'}$ is the result of the selection. The logical condition can be expressed over $\boldsymbol{A}_i$ in order to select a specific set of actions, in a specified time frame with specific action types and payloads. Several kind of analysis can be carried over this large database. The following sections will show the general outlines of the natural experiment nearing completion that we are currently conducting with these type of

data, while Fig. 3 shows the general architecture of our novel framework: the model proposed by us and the interaction with the existing MA software.

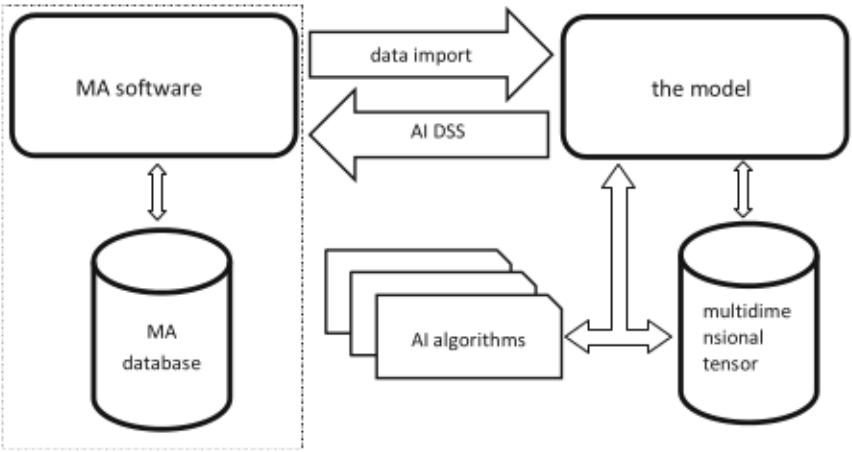


**Fig. 3.** The general architecture and interaction between the marketing automation software, our model, and the AI algorithms implementable into the model itself. The MA components are outside the model and are enclosed in the dashed rectangle (left side of the figure).

## 5 Towards a Natural Experiment

In line with Varian [4] and Rosenzweig and Wolpin [16], we started working on a first real large database originated from an e-commerce organisation that makes use of public social and editorial media content to promote its products/services with the main objective of increasing sales. The experimental subjects are the final users who buy products. The organisation sales its products/services by using an e-commerce platform. This platform is not a generalist website, rather it is specialised on a specific category of automotive products/services, so as we can assume that all the experimental subjects are similar regarding the supply they are interested in (so are quite homogeneous in this respect). All the data has been collected from the MA software and imported in our model, and users have been anonymised, so that we can distinguish between them but cannot identify them nominally. Focusing on the large database obtained, we are interested in studying the correlation between the marketing operations and the order placement.

The time-period considered in the large database is from 01 January 2017 up to 20 February 2018 (almost fourteen months). The statistical data are as follows:

- The total number of experimental subjects were 2716. Our design allows us consider as experimental subjects all the users that placed an order on the platform at least once in the aforementioned time-period.
- The number of orders placed in the considered time-period was 3861:

- Each user placed between 1 and 27 orders, and 2059 users placed only one order.
- For each user we have registered a number of actions, the mathematical average of the registered actions for each user was 36 (with a standard deviation equal to 15,75), while the median was 38.

We studied the above data so as to find possible correlations between the users' actions and the order placement, considering that the organisation sent a number of newsletters and published news and promotions on the social networks in that time-period. After our preliminary analysis, we discovered a few interesting facts.

Accounting for these facts, let us start comparing newsletter efficacy *versus* social posts efficacy:

- No. 743 orders were placed within 7 days from the date of the newsletters were sent, while only 29 orders were placed within 7 days from the date of the social posts were published. The number of the newsletters sent and the social posts published is almost equal within the considered time-period, and the users who placed the orders are a subset of the ones who read the newsletter (or added a *like* to a social post).
- Several users placed two or more orders after the newsletters have been sent and the social posts have been published. In particular, users who placed an order after opening a newsletter are 590, while only 27 placed an order after adding a *like* on a social post.

In other words, the orders placed increase significantly after sending newsletters to the users, while the order increase is very modest as a consequence of social network activities. We do not intend to argue that, generally, it is better to send newsletters than publishing posts on social networks but, in this particular case, the database supports this evidence. Therefore, we started studying more deeply the action made by the users (limited to the users who reacted positively to the newsletter) and the conversion ratio $c_N$ of each newsletter, that is, the ratio of the number of users who placed an order after receiving a newsletter *versus* the total number of users who received the newsletter:

$$c_N = \frac{o_N}{s_N} \tag{3}$$

In the formula (3), $N$ is a generic newsletter, $c_N$ is the conversion ratio of the newsletter $N$, $o_N$ is the number of orders placed and correlated to the newsletter $N$, and $s_N$ is the number of newsletters sent (thus the number of emails sent) to the users. This requires a few explanations. Particularly, we are referring to the total number of emails sent, which is different from the number of emails read/opened by the users. To cope with this task, let us call $r_N$ the number of emails effectively read by the users. In this respect, we can calculate the efficacy $e_N$ of a newsletter as the ratio of the number of emails read *versus* the total number of emails sent:

$$e_N = \frac{r_N}{s_N} \tag{4}$$

It is intuitive as well as supported by the data that the number of read emails is related to the number of orders placed. Therefore, in order to increase $c_N$, the experimenters have two possibilities: the first is to increase the number of orders, while the second, albeit apparently paradoxical, is to reduce $s_N$.

The first experimenter's strategy is related to the increase of $r_N$: if the experimenters were able to increase the number of users that read the newsletters, the experimenters should have an effect on $o_N$ (more final users reading the newsletter should reflect on more orders placed). The second experimenter's strategy is only apparently paradoxical, because if we were able to reduce $s_N$ without reducing $r_N$, we will obtain a more effective newsletter, addressing only users willing to open it and reducing the risk of being considered like spam from users not interested in the newsletters.

Arguably, the first strategy has an emergent tangible economic value: in fact, more orders placed by the users means more revenues for the organisation. The second strategy, conversely, has an intangible value consisting in reducing the risk of being considered unsolicited or undesired (*e.g.*, electronic spamming), and thus no emergent tangible value. With regard to the first strategy, we designed the following experimental treatment: the users are divided in two groups, $U_1$ and $U_2$, and they are random assigned to $U_1$ and $U_2$, where $U_1$ is the treatment group. For each user $u \in U_1$ the MA software sends newsletters at a specific time $t_u$, calculated as the time when the user is presumed to be active on internet (based on reading time of earlier newsletters or activity time on social networks and e-commerce website recovered from the action list). We are working in order to find one or more algorithms to assign the sending time $t_u$ for each user. The second group $U_2$ is our control group: the newsletters are sent with the usual strategy (the date and time of day decided by the e-commerce firm). We will compare the conversion ratio of the first group with the conversion ratio of the baseline group so as to measure the effectiveness of this strategy. With regard to the second strategy, we have several ideas to find the users that should not receive the newsletter (for example, using Recurrent Neural Networks, as outlined in the next section). The natural experiment is designed to compare the result of each newsletter sent and opened with the forecast made by an algorithm on which we are working on: basically, the efficacy of our algorithm is given by comparing the set of users that opened each newsletter with the set of users forecasted by our algorithm. For each newsletter $N$, ideally, the algorithm should be able to reduce to zero both the wrong positive forecasts $W_N^+$ (defined as the set of users not included in the list that in the real case opened the newsletter) and the wrong negative forecasts $W_N^-$ (the set of users forecasted as newsletter openers that in the real case did not open the newsletter). In formula, let $O_N$ be the set of users that opened the newsletters and $F_N$ the set forecasted by the second strategy. Using a mathematical set notation, the wrong forecasts can be calculated as a set difference, where $\setminus$ is the notation for set subtraction:

$$W_N^+ = O_N \setminus F_N \tag{5}$$

$$W_N^- = F_N \setminus O_N \tag{6}$$

Our objective is to minimise the size of both forecasts:

$$\min\left(\left|W_N^+\right| + \left|W_N^-\right|\right) \tag{7}$$

By using the first strategy, we cannot know in advance if and how much the strategy will be effective: only when we will complete the natural experiment with future newsletters we will be able to know the efficacy and eventually tune it to be more effective. Pursuing the second strategy, we are able to test the second strategy on existing data to tune the algorithm – and then experiment – on future newsletters, having the calculated efficacy of historical data as a baseline.

## 6 Next Steps to Be Implemented

As we argued in Sect. 5, we started to conduct a natural experiment, which is nearing completion, by using real input data of final users with the MA software. Our first step is about investigating the different conversion ratio of newsletters (where the conversion ratio is the ratio between the number of users that placed an order *versus* the number of users that received the newsletter) in order to find a way to increase the conversion ratio. In the near future, our research agenda will focus (i) on developing a microeconomic theoretical framework related to the consumption setting investigated in this paper, and (ii) on completing the natural experiment started with two different strategies: the first is to send the newsletter at different times during the day, because our data analysis over the activities extracted from the multidimensional data shows that different users read the e-mail at different times. In this respect, we are collecting further data to verify the effectiveness of this strategy, that is based on a simple statistical correlation. Following this strategy, we do not reduce the number of email sent but try to adapt to the preferred email opening time of users. The second strategy is about reducing the number of emails sent by selecting the most likely users that will open the newsletter. We are experimenting it also with Recurrent Neural Networks (RNN: for an overview see [13]) that seems the most effective to model along the time dimension and arbitrary sequence of events and inputs. Even in this case, we are collecting big data to have a statistical significance about the results, that seems quite promising. Our main objective with the model presented in this paper is to provide an automatic tool that meets the typical research questions within organisations using marketing automation software. As long as we find good strategies, the model will be able both to help decisions and provide automatic intelligent strategies that improve the effectiveness of the marketing actions. We will continue to study different algorithms and strategies as well as the natural experimentation. On a long term range, finally, our objective will be to promote our model as a decision support system for marketing automation backed by artificial intelligent algorithms equipped with a friendly and intuitive graphical user interface. To reach this long term goal, we will continue to study several algorithms and we plan to add semantic analysis of the content so as to improve the effectiveness of deep learning algorithms and broaden the possibilities of the MA software.

## References


1. Chen, S.-H., Kao, Y.-F., Venkatachalam, R.: Computational behavioral economics. In: Frantz, R., Chen, S.-H., Dopfer, K., Heukelom, F., Mousavi, S. (eds.) Routledge Handbook of Behavioral Economics, pp. 297–315. Routledge, Abingdon, Oxon (2017)
2. Kao, Y.-F., Velupillai, V.K.: Behavioural economics: classical and modern. Eur. J. Hist. Econ. Thought **22**(2), 236–271 (2015)
3. Dell'Aversana, R.: A unified framework for multicriteria evaluation of intangible capital assets inside organizations. In: Bucciarelli, E., Chen, S.-H., Corchado, J.M., (eds.) Decision Economics: In the Tradition of Herbert A. Simon's Heritage, pp. 114–121. Springer, Cham (2018)
4. Varian, H.R.: Causal inference in economics and marketing. Proc. Nat. Acad. Sci. U.S.A. (PNAS) **113**(27), 7310–7315 (2016)
5. Einav, L., Levin, J.: Economics in the age of big data. Science **346**(6210), 1243089 (2014)
6. Simon, H.A.: Administrative Behavior: A Study of Decision-Making Processes in Administrative Organizations, 4 th edn. Free Press, New York (1957, 1976, 1997) [1945]
7. Varian, H.R.: Big data: new tricks for econometrics. J. Econ. Perspect. **28**(2), 3–27 (2014)
8. Chen, S.-H. (ed.): Genetic Algorithms and Genetic Programming in Computational Finance. Kluwer, New York (2002)
9. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading (1989)
10. Mitchell, M.: An Introduction to Genetic Algorithms. M.I.T. Press, Cambridge (1996)
11. Chen, S.-H., Kuo, T.-W., Shien, Y.-P.: Genetic programming: a tutorial with the software simple GP. In: Chen, S.-H. (ed.) Genetic Algorithms and Genetic Programming in Computational Finance, pp. 55–77. Kluwer, New York (2002)
12. Deng, L.: A tutorial survey of architectures, algorithms, and applications for deep learning. APSIPA Trans. Signal Inf. Process. **3**, E2 (2014)
13. Schmidhuber, J.: Deep learning in neural networks: an overview. Neural Netw. **61**, 85–117 (2015)
14. Marlow, S. (ed.): Haskell 2010 language report (2010). https://www.haskell.org/onlinereport/haskell2010. Accessed 2 Feb 2018
15. Odersky, M., Spoon, L., Venners, B.: Programming in Scala: Updated for Scala 2.12, 3rd edn. Artima Press, Walnut (2016)
16. Rosenzweig, M.R., Wolpin, K.I.: Natural "natural experiments" in economics. J. Econ. Lit. **38**(4), 827–874 (2000)