Article

Advances in Machine Learning for the Behavioral Sciences

American Behavioral Scientist
1–31

DOI: 10.1177/0002764219859639
journals.sagepub.com/home/abs

Tomáš Kliegr[1], Štěpán Bahník[1], and Johannes Fürnkranz[2]

Abstract

The areas of machine learning and knowledge discovery in databases have considerably matured in recent years. In this article, we briefly review recent developments as well as classical algorithms that stood the test of time. Our goal is to provide a general introduction into different tasks such as learning from tabular data, behavioral data, or textual data, with a particular focus on actual and potential applications in behavioral sciences. The supplemental appendix to the article also provides practical guidance for using the methods by pointing the reader to proven software implementations. The focus is on R, but we also cover some libraries in other programming languages as well as systems with easy-to-use graphical interfaces.

Keywords

artificial intelligence, big data, descriptive data mining, machine learning, natural language processing, text mining

Introduction

Machine learning has considerably matured in recent years and has become a key enabling technology for many data-intensive tasks. Advances in neural network–based deep learning methodologies have yielded unexpected and unprecedented performance levels in tasks as diverse as image recognition, natural language processing, and game playing. Yet these techniques are not universally applicable, the key impediments being their hunger for data and their lack of interpretable results. These features make them less suitable for behavioral scientists where data are typically scarce and

[1]University of Economics, Prague, Czech Republic
[2]TU Darmstadt, Darmstadt, Germany

Corresponding Author:
Tomas Kliegr, Department of Information and Knowledge Engineering, Faculty of Informatics and Statistics, University of Economics, Prague, nam. W Churchilla 4, 130 67, Prague, Czech Republic.
Email: tomas.kliegr@vse.cz

results that do not yield insights into the nature of the processes underlying studied phenomena are often considered of little value.

This article presents an up-to-date curated survey of machine learning methods applicable to behavioral research. Since being able to understand a model is a prerequisite for uncovering the causes and mechanisms of the underlying phenomena, we favored methods that generate *interpretable models* from the multitude of those available. However, we also provide pointers to state-of-the art methods in terms of predictive performance, such as neural networks.

Each covered method is described in nontechnical terms. To help researchers in identifying the best tool for their research problem, we put emphasis on examples, when most methods covered are complemented with references to their existing or possible use in behavioral sciences. Each described method is supplemented with a description of software that implements it, which is provided in Supplemental Appendix B (available online). Given the predominance of R as a language for statistical programming in the behavioral sciences, we focus in particular on these packages. We also cover some libraries in other programming languages, most notably in Python, as well as systems with easy-to-use graphical interfaces.

The survey is organized by the character of input data. In the "Tabular Data" section, we cover structured, tabular data, for which we present an up-to-date list of methods used to generate classification models, as well as algorithms for exploratory and descriptive data mining. The "Behavioral Data" section covers methods and systems that can be used to collect and process behavioral data, focusing on clickstreams resulting from web usage mining, and methods developed for learning preference models from empirical data. The latter two areas can, for example, be combined for consumer choice research based on data obtained from an online retailer. Given the uptake of social media both as sources of data and objects of study, the "Textual Data" section provides an in-depth coverage of textual data, including syntactic parsing and document classification methods used to categorize content as well as new advances that allow representation of individual documents using word embeddings. The Internet also provides new machine-readable resources, which contain a wealth of information that can aid analysis of arbitrary content. Knowledge graphs and various lexical resources, covered in the "External Knowledge Sources" section, can be used, for example, for enrichment of content of small documents (*microposts*), which are an increasingly common form of online communication. The "Related Work" section discusses related work and covers also miscellaneous topics such as *machine learning as service* systems. These can provide the behavioral scientist the ability to process very large data sets with little setup costs. The conclusion summarizes methods covered in this chapter, focusing on the performance–interpretability trade-off. It also discusses emerging trends and challenges, such as the legal and ethical dimensions of machine learning.

The article comes with two supplemental appendices (available online). Due to the large number of articles covered by this review, only articles referenced from the "Applications in Behavioral Sciences" subsections are included in the main bibliography. Remaining references are available in Supplemental Appendix A. Supplemental Appendix B contains an overview of selected software packages implementing some of the methods discussed in the main text.

Table 1. A Sample Database.

Education	Marital Status	Sex	Has Children	Approve?
Primary	Single	Male	No	No
Primary	Single	Male	Yes	No
Primary	Married	Male	No	Yes
University	Divorced	Female	No	Yes
University	Married	Female	Yes	Yes
Secondary	Single	Male	No	No
University	Single	Female	No	Yes
Secondary	Divorced	Female	No	Yes
Secondary	Single	Female	Yes	Yes
Secondary	Married	Male	Yes	Yes
Primary	Married	Female	No	Yes
Secondary	Divorced	Male	Yes	No
University	Divorced	Female	Yes	No
Secondary	Divorced	Male	No	Yes

Tabular Data

The task that has received the most attention in the machine learning literature is the *supervised learning* scenario: Given a database of observations described with a fixed number of measurements or features and a designated attribute, the *class*, find a mapping that is able to compute the class value from the feature values of new, previously unseen observations. While there are statistical techniques that are able to solve particular instances of this problem, machine learning techniques provide a strong focus on the use of categorical, nonnumeric attributes, and on the immediate interpretability of the result. They also typically provide simple means for adapting the complexity of the models to the problem at hand. This, in particular, is one of the main reasons for the increasing popularity of machine learning techniques in both industry and academia.

Table 1 shows a small, artificial sample database, taken from Billari, Fürnkranz, and Prskawetz (2006). The database contains the results of a hypothetical survey with 14 respondents concerning the approval or disapproval of a certain issue. Each individual is characterized by four attributes—*Education* (with possible values *primary* school, *secondary* school, or *university*), *Marital Status* (with possible values *single, married*, or *divorced*), *Sex* (*male* or *female*), and *Has Children* (*yes* or *no*)—that encode rudimentary information about their sociodemographic background. The last column, *Approve?*, encodes whether the individual approved or disapproved the issue.

The task is to use the information in this training set to derive a model that is able to predict whether a person is likely to approve or disapprove, based on the four demographic characteristics. As most classical machine learning methods tackle a setting like this, we briefly recapitulate a few classical algorithms, while mentioning some new developments as well.

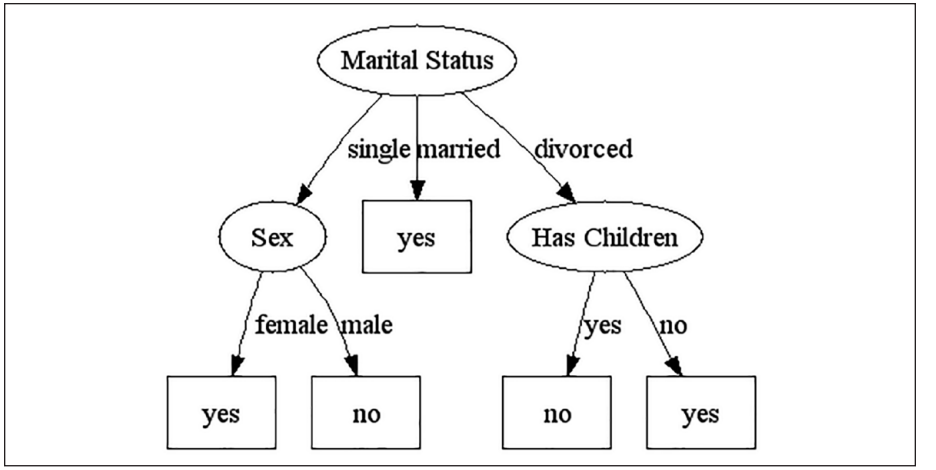

Figure 1. A decision tree describing the data set shown in Table 1.

Induction of Decision Trees

The induction of decision trees is one of the oldest and most popular techniques for learning discriminatory models, which has been developed independently in the statistical (Breiman, Friedman, Stone, & Olshen, 1984; Kass, 1980) and machine learning (Quinlan, 1986) literatures. A *decision tree* is a particular type of classification model that is fairly easy to induce and to understand. In the statistical literature (e.g., Breiman et al., 1984), decision trees are also known as *classification trees*. Related techniques for predicting numerical class values are known as *regression trees*.

Figure 1 shows a sample tree which might be induced from the data of Table 1. To classify a specific instance, the decision tree asks the question "What is the marital status for a given instance?" If the answer is "married" it assigns the class "yes." If the answer is divorced or single, additional question is sought.

In general, classification of a new example starts at the top *node*—the *root*. In our example, the root is a *decision node*, which corresponds to a test of the value of the *Marital Status* attribute. Classification then proceeds by moving down the branch that corresponds to a particular value of this attribute, arriving at a new decision node with a new attribute. This process is repeated until we arrive at a terminal node—a so-called *leaf*—which is not labeled with an attribute but with a value of the target attribute (*Approve?*). For all examples that arrive at the same leaf value, the same target value will be predicted. Figure 1 shows leaves as rectangular boxes and decision nodes as ellipses.

Decision trees are learned in a top-down fashion: The program selects the best attribute for the root of the tree, splits the set of examples into disjoint sets (one for each value of the chosen attribute, containing all training examples that have the corresponding value for this attribute), and adds corresponding nodes and branches to the tree. If there are new sets that contain only examples from the same class, a leaf node

is added for each of them and labeled with the respective class. For all other sets, a *decision node* is added and associated with the best attribute for the corresponding set as described above. Hence, the data set is successively partitioned into nonoverlapping, smaller data sets until each set only contains examples of the same class (a pure node). Eventually, a pure node can always be found via successive partitions unless the training data contain two identical but contradictory examples that have the same feature values but different class values.

The crucial step in decision tree induction is the choice of an adequate attribute. Typical attribute selection criteria use a function that measures the purity of a node, that is, the degree to which the node contains only examples of a single class. This purity measure is computed for a node and all successor nodes that result from using an attribute for splitting the data. The difference between the original purity value and the sum of the values of the successor nodes weighted by the relative sizes of theses nodes, is used to estimate the utility of this attribute, and the attribute with the largest utility is selected for expanding the tree. The algorithm C4.5 uses information-theoretic entropy as a purity measure (Quinlan, 1986), whereas CART uses the Gini index (Breiman et al., 1984). Algorithm C5.0, successor to C4.5, is noted for its best performance among all tree learning algorithms in the seminal article of Fernandez-Delgado, Cernadas, Barro, and Amorim (2014).

Overfitting refers to the use of an overly complex model that results in worse performance on new data than would be achievable with a simpler model (Mitchell, 1997). Tree models may overfit due to specialized decision nodes that refer to peculiarities of the training data. In order to receive simpler trees and to fight overfitting, most decision tree algorithms apply pruning techniques that simplify trees after learning by removing redundant decision nodes.

A general technique for improving the prediction quality of classifiers is to form an ensemble—learning multiple classifiers whose individual predictions are joined into a collective final prediction. The best-known technique is *random forests* (Breiman, 2001), which uses resampling to learn a variety of trees from different samples of the data. They also use different random subsets of all available attributes, which not only increases the variance in the resulting trees but also makes the algorithm quite fast. However, the increased predictive accuracy also comes with a substantial decrease in the interpretability of the learned concepts.

Applications in Behavioral Sciences. Given that they are not only well known in machine learning and data mining but are also firmly rooted in statistics, decision trees have seen a large number of applications in behavioral sciences, of which we can list just a few. McArdle and Ritschard (2013) provide an in-depth introduction to this family of techniques and also demonstrate their use in a number of applications in demographic, medical, and educational areas. In demography, Billari et al. (2006) have applied decision tree learning to the analysis of differences in the life courses in Austria and Italy, where the key issue was to model these events as binary temporal relations.Similar techniques have also been used in survival analysis. For example, so-called survival trees have been used in a study by De Rose and Pallara (1997).

In the political sciences, decision trees have been used for modeling international conflicts (Fürnkranz, Petrak, & Trappl, 1997) and international negotiation (Druckman, Harris, & Fürnkranz, 2006). Rosenfeld, Zuckerman, Azaria, and Kraus (2012) also used decision trees to model negotiating behavior. In psychology, Walsh, Ribeiro, and Franklin (2017) used random forests to predict future suicide attempts of patients.

Induction of Predictive Rule Sets

Another traditional machine learning technique is the induction of rule sets (Fürnkranz, Gamberger, and Lavrač, 2012). The learning of rule-based models has been a main research goal in the field of machine learning since its beginning in the early 1960s. Rule-based techniques have also received some attention in the statistical community (Friedman & Fisher, 1999).

Comparison Between Rule and Tree Models. Rule sets are typically simpler and more comprehensible than decision trees, where each leaf of the tree can be interpreted as a single rule consisting of a conjunction of all conditions in the path from the root to the leaf.

The main difference between the rules generated by a decision tree and the rules generated by a rule learning algorithm is that the former rule set consists of nonoverlapping rules that span the entire instance space—each possible combination of feature values will be covered by exactly one rule. Relaxing this constraint by allowing potentially overlapping rules that need not span the entire instance space, may often result in smaller rule sets.

However, in this case, we need mechanisms for tie breaking: Which rule to choose when more than one covers the given example. We also need mechanisms for default classifications: What classification to choose when no rule covers the given example. Typically, one prefers rules with a higher ratio of correctly classified examples from the training set.

Example of a rule model.

IF Marital Status = married	**THEN** yes
IF Sex = female	**THEN** yes
IF Sex = male	**THEN** no
DEFAULT yes	

The example above shows a particularly simple rule set for the data in Table 1. It uses two different attributes in its first two rules. Note that these two rules are overlapping, i.e. several examples will be covered by more than one rule. For instance, examples 3 and 10 are covered by both the first and the third rule. These conflicts are typically resolved by using the more accurate rule, i.e., the rule that covers a higher proportion of examples that support its prediction (the first one in our case). Also note that this rule set makes two mistakes (the last two examples). These might be resolved by resorting to a more complex rule set (such as the one corresponding to the decision tree of Figure 1) but as stated above, it is often more advisable to sacrifice accuracy in the training set for model simplicity to avoid overfitting. Finally, note the default rule at the end of the rule set. This is added for the case when certain regions of the data space are not represented in the training set.

Learning Rule Models. The key ideas for learning such rule sets are quite similar to the ideas used in decision tree induction. However, instead of recursively partitioning the data set by optimizing the purity measure over all successor nodes (in the literature, this strategy is also known as *divide-and-conquer* learning), rule learning algorithms only expand a single successor node at a time, thereby learning a complete rule that covers part of the training data. After a complete rule has been learned, all examples that are covered by this rule are removed from the training set, and the procedure is repeated with the remaining examples. This strategy is also known as *separate-and-conquer* learning. Again, pruning is a good idea for rule learning, which means that the rules only need to cover examples that are *mostly* from the same class. It turns out to be advantageous to prune rules immediately after they have been learned, before successive rules are learned (Fürnkranz, 1997).

The idea to try to prune or simplify each rule right after it has been learned has been exploited in the well-known RIPPER algorithm (Cohen, 1995). This algorithm has been frequently used in applications because it learns very simple and understandable rules. It also added a postprocessing phase for optimizing a rule set in the context of other rules. The key idea is to remove one rule out of a previously learned rule set and try to relearn the rule in the context of previous rules and subsequent rules. Another type of approach to rule learning heavily relying on effective pruning methods is Classification Based on Associations (Liu, Hsu, & Ma, 1998) and succeeding algorithms. Their key idea is to use algorithms for discovering association rules (cf. "Discovering Interesting Rules" section), and then combine a selection of the found rules into a predictive rule model.[1]

Current Trends. Current work in inductive rule learning is focused on finding simple rules via optimization (Dash, Günlük, & Wei, 2018; Malioutov & Meel, 2018; Wang et al., 2017), mostly with the goal that simple rules are more easily interpretable. However, there is also some evidence that shorter rules are not always more convincing than more complex rules (Fürnkranz, Kliegr, & Paulheim, 2018; Stecher, Janssen, & Fürnkranz, 2016). Another line of research focuses on improving accuracy of rule models, often by increasing their expressiveness through "fuzzification" by making the decision boundary between different classes softer. At the expense of lower interpretability, fuzzy rule learning algorithms such as SLAVE (García, González, & Pérez, 2014), FURIA (Hühn & Hüllermeier, 2009), and FARC-HD (Alcala-Fdez, Alcala, & Herrera, 2011) often outperform models with regular, "crisp" rules.

Applications in Behavioral Sciences. Similar to decision trees, rule learning can be generally used for prediction or classification in cases where interpretability of the model is important. Rule learning could also be useful in domains where the output of the model should be easily applicable for a practitioner, such as a physician or a psychologist, given that the resulting model can be easier to remember and apply than a logistic regression or a decision tree model.

Multiple studies used the RIPPER algorithm, which is considered to be the state-of-the-art in inductive rule learning, for learning classification rules. Classification rules

may be used for classification of documents in various categories. For example, one study used RIPPER and other algorithms to classify emails (Stumpf et al., 2009). The RIPPER algorithm outperformed Naive Bayes, another popular machine learning algorithm, in terms of classification accuracy. Furthermore, rule-based explanations were considered on average the most understandable, which might be especially useful when interpretation of the output of the algorithm or further work with the algorithm's results is necessary.

Other uses of RIPPER include classifying the strengths of opinions in nested clauses (Wilson, Wiebe, & Hwa, 2004) and predicting students' performance (Kotsiantis, Pierrakeas, & Pintelas, 2002). Some of the studies using decision trees are also used for rule learning (Billari et al., 2006; Fürnkranz et al., 1997).

Rule learning is suggested as a possible computational model in developmental psychology (Shultz, 2013). These algorithms, or decision tree models convertible to rules, could therefore be used in psychology to simulate human reasoning.

Discovering Interesting Rules

The previous section focused on the use of rules for prediction, but rule learning can be also adapted for exploratory analysis, where only rules corresponding to interesting patterns in data are generated.

A commonly used approach for this task is association rule learning. Algorithms belonging to this family are characterized by outputting all rules that match user defined constraints on interestingness. These constraints are called interest measures and are typically defined by two parameters: minimum confidence threshold and minimum support threshold.

If we consider rule r: **IF** Antecedent **THEN** Consequent, then *rule confidence* is the proportion of objects correctly classified by the rule to all objects matched by the antecedent of the rule. Object is correctly classified when it matches the entire rule (its antecedent and consequent), and incorrectly classified if it matches only the antecedent, but not consequent. *Rule support* is typically defined as the proportion of objects correctly classified by the rule to all objects in the training data.

Example. Let us consider the following object $o = \{income = low, district = London, savings = high, risk = low\}$
and rule r: **IF** income=low **AND** district=London **THEN** *risk=high*. Object o matches rule r, because o meets all conditions in the antecedent of r. Rule r will incorrectly classify o, because the class assigned by rule consequent does not match the value of the target attribute *risk* of o.

Apriori (Agrawal, Imielinski, & Swami, 1993) is the most well-known algorithm for mining association rules. There are also newer algorithms, such as FP-Growth, which can provide faster performance. While association rule mining is commonly used for discovering interesting patterns in data, the simplicity of the generated rules as well as restricted options for constraining the search space may become a limitation.

One common problem with application of association rule mining stems from the fact that all rules matching user-defined interestingness thresholds are returned. There may be millions of such rules even for small data sets, resulting in impeded interpretability of the resulting list of rules. A possible solution is to apply pruning, which will remove redundant rules. Another limitation of association rule mining is lack of direct support for numeric attributes.

An alternative approach to pruning is to better focus the generation of association rules. This approach is provided by the GUHA method (Hájek, Holeňa, & Rauch, 2010), which was initially developed with the intent to automatically search for all statistical hypotheses supported by data. The method enables many fine-grained settings for expressing what should be considered as an interesting hypothesis. The trade-off is that GUHA has slower performance on larger data sets compared with association rule mining performed with Apriori (Rauch & Simunek, 2017).

Another related task applicable to descriptive and explorative data mining is *subgroup discovery*, which finds groups of instances in data, which exhibit "distributional unusualness with respect to a certain property of interest" (Wrobel, 1997). A number of quality measures were developed for subgroup discovery, but interest measures applied in association rule mining can be used as well. By choosing a suitable quality measure, the subgroup discovery task can thus be adapted for a range of diverse goals, such as mining for unexpected patterns. A subgroup can be considered as unexpected when it significantly deviates from the total population in terms of the selected quality measure (Atzmüller, 2015).

Subgroup discovery approaches are algorithmically diverse, with both association rule mining and predictive rule learning algorithms used as a base approach (Herrera, Carmona, González, & del Jesús, 2011; Kralj Novak, Lavrac, & Webb, 2009). The use of subgroup discovery can be considered over association rule mining when the task at hand involves a numeric target attribute. Some subgroup discovery algorithms also address the problem of too many rules generated by the convenient *top-k* approach, which returns only k top subgroups according to the selected quality metric.

Applications in Behavioral Sciences. Association rule mining has been extensively used to find interesting patterns in data in a number of disciplines. Selected recent applications include exploration of mathematics anxiety among engineering students (Herawan, Vitasari, & Abdullah, 2011) or discovering color–emotion relationships (Feng, Lesot, & Detyniecki, 2010). More recently, subgroup discovery was used to study relationships between technology acceptance and various personas presented by users (Behrenbruch et al., 2012). Goh and Ang (2007) provides an accessible introduction to association rule mining aimed at behavioral researchers.

Neural Networks and Deep Learning

Neural networks have a long history in artificial intelligence and machine learning. First works were motivated by the attempt to model neurophysiological insights,

which resulted in mathematical models of neurons, so-called perceptrons (Rosenblatt, 1962). Soon, their limitations were recognized (Minsky & Papert, 1969) and interest in them subsided until Rumelhart, Hinton, and Williams (1986) introduced back-propagation, which allowed to effectively train multilayer networks. While a perceptron can essentially only model a linear function connecting various input signals x_i to an output signal $f(x) = \sum_i w_i \cdot x_i$ by weighting them with weights w_i, multilayer networks put the linear output through nonlinear activation functions, which allow one to model arbitrary functions via complex neural networks (Hornik, 1991). This insight led to a large body of research in the 1990s, resulting in a wide variety of applications in industry, business, and science (Widrow, Rumelhart, & Lehr, 1994) before the attention in machine learning moved to alternative methods such as support vector machines.

Recently, however, neural networks have surfaced again in the form of so-called deep learning, which often leads to better performance (Goodfellow, Bengio, & Courville, 2016; Lecun, Bengio, & Hinton, 2015; Schmidhuber, 2015). Interestingly, the success of these methods is not so much based on new insights—the key methods have essentially been proposed in the 1990s—but on the availability of huge labeled data sets and powerful computer hardware that allows their use for training large networks.

The basic network structure consists of multiple *layers* of fully connected nodes. Each node in layer L_{i+1} takes the outputs of all nodes in layer L_i as input. For training such networks, the input signals are fed into the input layer L_0, and the output signal at the last layer L_l is compared to the desired output. The difference between the output signal and the desired output is propagated backward through the network, and each node adapts the weights that it puts on its input signals so that the error is reduced. For this adaptation, error gradients are estimated, which indicate the direction into which the weights have to be changed in order to minimize the error. These estimates are typically not computed from single examples, but from small subsets of the available data, so-called mini-batches. Several variants of this stochastic gradient descent algorithm have been proposed with AdaGrad being one of the most popular ones (Duchi, Hazan, & Singer, 2011). Overfitting the data has to be avoided with techniques such as dropout learning, which in each optimization step randomly exempts a fraction of the network nodes from training (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014).

Multiple network layers allow the network to develop data abstractions, which is the main feature that distinguishes deep learning from alternative learning algorithms. This is most apparent when auto-encoders are trained, where a network is trained to map the input data upon itself but is forced to project them into a lower-dimensional embedding space on the way (Vincent, Larochelle, Lajoie, Bengio, & Manzagol, 2010).

In addition to the conventional fully connected layers, there are various special types of network connections. For example, in computer vision, convolutional layers are commonly used, which train multiple sliding windows that move over the image data and process just a part of the image at a time, thereby learning to recognize local

features. These layers are subsequently abstracted into more and more complex visual patterns (Krizhevsky, Sutskever, & Hinton, 2017). For temporal data, one can use recurrent neural networks, which do not make predictions for individual input vectors, but for a sequence of input vectors. To do so, they allow feeding abstracted information from previous data points forward to the next layers. A particularly successful architecture are LSTM networks, which allow the learner to control the amount of information flow between successive data points (Hochreiter & Schmidhuber, 1997).

The main drawback of these powerful learning machines is the lack of interpretability of their results. Understanding the meaning of the generated variables is crucial for transparent and justifiable decisions. Consequently, the interest in methods that make learned models more interpretable has increased with the success of deep learning. Some research has been devoted to trying to convert such arcane models to more interpretable rule-based (Andrews, Diederich, and Tickle, 1995) or tree-based models (Frosst & Hinton, 2017), which may be facilitated with appropriate neural network training techniques (González, Loza Mencía, & Fürnkranz, 2017). Instead of making the entire model interpretable, methods like LIME (Ribeiro, Singh, & Guestrin, 2016) are able to provide local explanations for inscrutable models, allowing a trade-off between fidelity to the original model with interpretability and complexity of the local model. There is also research on developing alternative deep learning methods, most notably sum–product networks (Peharz, Gens, Pernkopf, & Domingos, 2017). These methods are firmly rooted in probability theory and graphical models and are therefore easier to interpret than neural networks.

Applications in Behavioral Science. Neural networks are studied and applied in psychological research within the scope of connectionist models of human cognition since about 1980s (Houghton, 2004). The study of artificial neural networks in this context has intensified in recent years in response to algorithmic advances. McKay, Abramowitz, and Storch (2017) review approaches involving artificial neural networks for studying psychological problems and disorders. For example, schizophrenic thinking is studied by purposefully damaging artificial neural networks. Neural networks have also been used to study nonpathological aspects of human decision making, such as consumer behavior (Greene, Morgan, & Foxall, 2017).

Deep neural networks have enjoyed considerable success in areas such as computer vision (Krizhevsky et al., 2017), natural language understanding (Deng & Liu, 2018), and game playing (Silver et al., 2016). However, these success stories are based on the availability of large amounts of training data, which may be an obstacle to wide use in behavioral sciences.

Behavioral Data

Machine learning and data mining have developed a variety of methods for analyzing behavioral data, ranging from mimicking behavioral traces of human experts, and area also known as behavioral cloning (Sammut, 1996), to the analysis of consumer behavior in the form of recommender systems (Jannach, Zanker, Felfernig, & Friedrich,

2010). In this section, we will look at two key enabling technologies, the analysis of log data, and the analysis of preferential data.

Web Log and Mobile Usage Mining

Logs of user interactions with web pages and mobile applications can serve as a trove of data for psychological research seeking to understand, for example, consumer behavior and information foraging strategies. The scientific discipline providing the tools and means for studying this user data in the form of click streams is called web usage mining (Liu, 2011). Many web usage mining approaches focus on the acquisition and preprocessing of data. These two steps are also the main focus of this section.

Data Collection. For web usage mining, there are principally two ways of collecting user interactions. Historically, the administrators of servers where the web site is hosted were configuring the server in such a way that each request for a web page was logged and stored in a text file. Each record in this web log contains information such as name of the page requested, time stamp, the IP address of the visitor, name of the browser, and resolution of the screen, providing input for web usage mining. An alternative way is to use Javascript trackers embedded in all web pages of the monitored web site instead of web logs. When a user requests the web page, the script is executed in the user's browser. It can collect similar types of information as web logs, but the script can also interact with the content of the page, acquiring, the price and category of the product displayed. The script can be extended to track user behavior within the web page, including mouse movements. This information is then typically sent to a remote server, providing web analytics *as a service*. In general, Javascript trackers provide mostly advantages over web logs as they can collect more information and are easier to set up and operate. Figure 2A presents an example of a clickstream collected from a travel agency website, and Figure 2B shows the additional information about the content of the page, which can be sent by the Javascript tracker.

Data Enrichment. In addition to user interactions, data collection may involve obtaining semantic description of data being interacted with, like price and category of a product. This information can be sent by the tracked web page. When this is not possible, one can resort to using web crawlers and scrapers. Web crawler is software which downloads web pages and other content from a given list of web sites and stores them in a database. Web scrapers provide means of subsequent processing of the content of web pages. This software provides a description of information to look for, such as prices or product categories, finds the information on the provided web page, and saves it in a structured way to a database.

Further enrichment of data can be performed, for example, through mapping IP addresses to regions via dedicated databases and software services. Their outputs include, among other information, zip codes, which might need to be further resolved to variables meaningful for psychological studies. This can be achieved using various openly accessible data sets. For example, for the United States there is the income tax

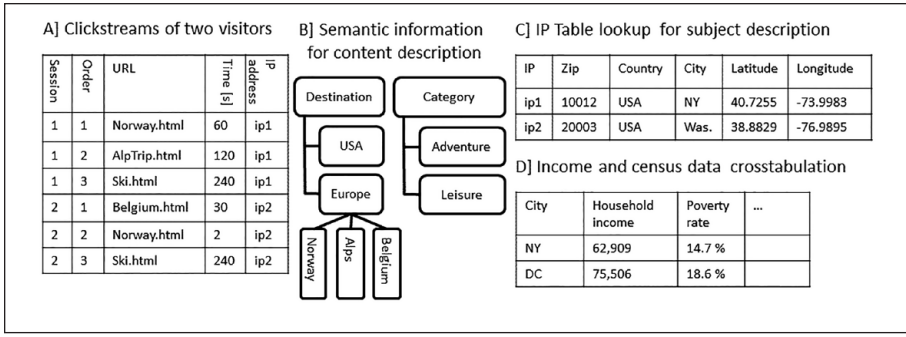

Figure 2. Data collection for web usage mining.

statistics data set,[2] which maps zip codes to several dozen income-related attributes. Other sources of data include https://datausa.io/ and https://data.world. This enrichment is exemplified in Figure 2C-D.

Data Preprocessing and Mining. The output of the data collection phase for web usage mining can be loosely viewed as a set of *n user interactions*. User interactions that take place within a given time frame (such as 30 minutes) are organized into sessions. Each user interaction is also associated with a unique user identifier. When web logs are used, individual records may need to be grouped into sessions by a heuristic algorithm, possibly resulting in some errors. On the other hand, records are naturally grouped into sessions when Javascript-based trackers are used.

Clickstream data are in a sequential format, in which, for example, sequential patterns or rules (Agrawal & Srikant, 1995) can be discovered.

Example.

Considering the input presented in Fig. 3A and a minimum support threshold of 30%, the maximum gap between two sequences = 2 and minimum confidence of 50%, the list of discovered sequential rules includes:

IF Norway.html, AlpTrip.html **THEN** Ski.html, conf = 100%, supp =50%.

This rule says that in all (100%) sessions where the user visited Norway.html and later AlpTrip.html, the user later also visited Ski.html.

The number of sessions complying to this rule amounted to 50% of all sessions.

Note that the elements in the consequent of a sequential rule occur at a later time than the elements of the antecedent. As shown in Liu (2011), the sequential representation can also be transformed to a tabular format, which allows for application of many standard implementations of machine learning algorithms.

Applications in Behavioral Sciences. The use of clickstreams has a direct application in the study of consumer behavior. For example, Senecal, Kalczynski, and Nantel (2005) examined the use of product recommendations in online shopping. Other related research involves using various cognitive phenomena to explain the effects of online advertisements (Rodgers & Thorson, 2000), determine the visitor's intent (Moe, 2003), or analyze reasons for impulse buying on the Internet (Koski, 2004). However, the use of data from web sites does not have to be limited to the study of consumer behavior. For example, primacy and recency effects were used to explain the effect of link position on the probability of user clicking on the link (Murphy, Hofacker, & Mizerski, 2006). Process tracing methods have a rich history in the study of decision making and some methods, for example, mouse tracking analysis (Stillman, Shen, & Ferguson, 2018), can be easily employed with data from Javascript trackers.

Preference Learning

Preference learning is a recent addition to the suite of learning tasks in machine learning (Fürnkranz & Hüllermeier, 2010). Roughly speaking, preference learning is about inducing predictive preference models from empirical data, thereby establishing a link between machine learning and research fields related to preference modeling and decision making. The key difference to conventional supervised machine learning settings is that the training information is typically not given in the form of single target values, like in classification and regression, but instead in the form of pairwise comparisons expressing preferences between different objects or labels.

In general, the task of preference learning is to rank a set of objects based on observed preferences. The ranking may also depend on a given context. For example, the preference between red wine or white wine for dinner often depends on the meal one has ordered. Maybe the best-known instantiation of preference learning are *recommender systems* (Gemmis et al., 2010; Jannach et al., 2010), which solve the task of ranking a set of products based on their interest for a given user. In many cases, neither the products nor the user is characterized with features, in which case the ranking is based on similarities between the recommendations across users (user-to-user correlation) or items (item-to-item correlations) (Breese, Heckerman, & Kadie, 1998). In many cases, we can observe features of the context, but the objects are only designated with unique labels. This task is also known as label ranking (Vembu & Gärtner, 2010). In object ranking, on the other hand, the objects are described with features, but there is no context information available (Kamishima, Kazawa, & Akaho, 2010). Finally, if both the contexts and the objects are characterized with features, we have the most general ranking problem, dyad ranking (Schäfer & Hüllermeier, 2018), where a set of objects is ranked over a set of different contexts. The best-known example is

the problem of learning to rank in Web search where the objects are web pages, the contexts are search queries, and the task is to learn to rank Web pages according to their relevance to a query.

Preferences are typically given in the form of pairwise comparisons between objects. Alternatively, the training information may also be given in the form of (ordinal) preference degrees attached to the objects, indicating an absolute (as opposed to a relative/comparative) assessment.

There are two main approaches to learning representations of preferences, namely utility functions, which evaluate individual alternatives, and preference relations, which compare pairs of competing alternatives. From a machine learning point of view, the two approaches give rise to two different kinds of learning. The latter, learning a preference relation, deviates more strongly from conventional problems like classification and regression, as it involves prediction of complex structures, such as rankings or partial order relations, rather than prediction of single values. Moreover, training input in preference learning will not be offered in the form of complete examples, as is usually the case in supervised learning, but it may comprise more general types of information, such as relative preferences or different kinds of indirect feedback and implicit preference information. On the other hand, the learning of a utility function, where the preference information is used to learn a function that assigns a numerical score to a given object, is often easier to apply because it enforces transitivity on the predicted rankings.

Applications in Behavioral Sciences. For many problems in the behavioral sciences, people are required to make judgments about the quality of certain courses of actions or solutions. However, humans are often not able to determine a precise utility value of an option, but they are typically able to compare the quality of two options. Thurstone's *Law of Comparative Judgment* essentially states that such pairwise comparisons correspond to an internal, unknown utility scale (Thurstone, 1927). Recovering this hidden information from such qualitative preference is studied in various areas such as ranking theory (Marden, 1995), social choice theory (Rossi, Venable, & Walsh, 2011), voting theory (Coughlin, 2008), sports (Langville & Meyer, 2012), negotiation theory (Druckman, 1993), decision theory (Bouyssou et al., 2002), democratic peace theory (Cuhadar & Druckman, 2014), and marketing research (Rao, Green, & Wind, 2007). Thus, many results in preference learning are based on established statistical models for ranking data, such as the Plackett–Luce (Luce, 1959; Plackett, 1975) or Bradley–Terry (Bradley & Terry, 1952) models, which allow an analyst to model probability distributions over rankings.

Given that preference and ranking problems are ubiquitous, computational models for solving such problems can improve prediction and lead to new insights. For example, in voting theory and social choice, Bredereck, Chen, Niedermeier, and Walsh (2017) use computational methods to analyze several parliamentary voting procedures.

Textual Data

Much data analyzed in the behavioral sciences take the form of text. The rise of online communication has dramatically increased the volume of textual data available to

behavioral scientists. In this section, we will review methods developed in computational linguistics and machine learning that can help the researcher to sift through textual data in an automated way. These methods increase the scale at which data can be processed and improve reproducibility of analyses, since subjective evaluation of a piece of text can be replaced by automated processing, which produces the same results given the same inputs.

We review various methods for representing text with vectors, providing a gateway for further processing with machine learning algorithms. This is followed by methods for text annotation, including additional information, such as parts of speech for individual words or the political orientation of people mentioned in the text. The section concludes with machine learning algorithms for document classification, which operates on top of the vector-based representation of text.

Word Vectors and Word Embeddings

A vector space model was developed to represent a document in the given collection as a point in a space (Turney & Pantel, 2010). The position of the document is specified by a vector, which is typically derived from frequency of occurrence of individual words in the collection. The notion of vector space models was further extended to other uses, including representation of words using their context.

Vector-based representation has important psychological foundations (Hinton, McClelland, & Rumelhart, 1986; Turney & Pantel, 2010). Word vectors closely relate to a distributed representation; that is, using multiple reusable features to represent a word. Landauer, McNamara, Dennis, and Kintsch (2013) provide further empirical and theoretical justification for the psychological plausibility of selected vector space models.

There are multiple algorithms that can be applied to finding word vectors. They have a common input of an unlabeled collection of documents and their output can be used to represent each word as a list or vector of weights. Depending on the algorithm, the degree to which the individual weights can be interpreted varies substantially. Also, the algorithms differ in terms of how much the quality of the resulting vectors depends on the size of the provided collection of documents. Table 2 is aimed at helping the practitioner to find the right method for the task at hand.[3] All of the methods covered in Table 2 are briefly described in the following text.

Bag of Words (BoW). One of the most commonly applied type of vector space model is based on a *term–document matrix*, where rows correspond to terms (typically words) and columns to documents. For each term, the matrix expresses the number of times it appears in the given document. This representation also called a *bag of words*. The term frequencies (TFs) act as weights that represent the degree to which the given word describes the document. To improve results, these weights are further adjusted through normalization or through computing inverse document frequencies (IDFs) in the complete collection. IDF reflects the observation that rarer terms—those that appear only in a small number of documents—are more useful in discriminating documents in the collection from each other than terms that tend to appear in all or most

Table 2. Methods Generating Word Vectors.

Method	Required data size	Features	Algorithmic approach
BoW	Small	Explicit (terms)	Term–document matrix
ESA	Medium	Explicit (documents)	Inverted index
LDA	Smaller	Latent topics	Generative model
LSA	Smaller	Latent concepts	Matrix factorization
word2vec	Large	Uninterpretable	Neural network
Glove	Large	Uninterpretable	Regression model

Note. BoW = bag of words; ESA = explicit semantic analysis; LDA = latent Dirichlet allocation; LSA = latent semantic analysis.

documents. Bag-of-words representation incorporating IDF scores is commonly referred to as TF-IDF.

Semantic Analysis. The explicit semantic analysis (ESA) approach (Gabrilovich & Markovitch, 2007) represents a particular word using a weighted list of documents (typically Wikipedia articles). ESA represents words based on an inverted index, which it builds from documents in the provided knowledge base.[4] Each dimension in a word vector generated by ESA corresponds to a document in the training corpus and the specific weight indicates to what extent that document represents the given word. Latent semantic analysis (LSA; Landauer & Dumais, 1997) and latent Dirichlet allocation (LDA; Blei, Ng, & Jordan, 2003) are two older, well-established algorithms, which are often used for topic modeling, namely, the identification of topics or concepts best describing a given document in the collection. The concepts and topics produced by these methods are latent. That is, LDA topics are not given an explicit label by the method (such as "finances"), but instead can be interpreted through weights of associated words (such as "money" or "dollars"; Chen & Wojcik, 2016).

Semantic Embeddings. *Word2vec* is a state-of-the-art approach to generating word vectors (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013). The previously covered algorithms generate interpretable word vectors essentially based on analyzing counts of occurrences of words. A more recent approach is based on predictive models. These use a predictive algorithm—word2vec uses a neural network—to forecast a word given a particular context or vice versa. Word vectors created by word2vec are sometimes called *word* embeddings: an individual word is represented by a list of weights or real numbers.

Glove (Global Vectors for Word Representation) is an algorithm inspired by word2vec, which uses a weighted least squares model trained on global word-word co-occurrence counts (Pennington, Socher, & Manning, 2014). Word embeddings trained by the Glove algorithm do particularly well on the word analogy tasks, where the goal is to answer questions such as "Athens is to Greece as Berlin is to ?"

Quality of Results Versus Interpretability of Word Vectors. Predictive algorithms such as word2vec have been shown to provide better results than models based on analyzing counts of co-occurrence of words across a range of lexical semantic tasks, including word similarity computation (Baroni, Dinu, & Kruszewski, 2014). While the individual dimensions in word2vec or Glove models do not directly correspond to explicit words or concepts as in ESA, distance between word vectors can be computed to find analogies and compute word similarities (see Figure 3).

Applications in Behavioral Sciences. Caliskan, Bryson, and Narayanan (2017) have shown that semantic association of words measured using the distance of their embeddings, generated by the Glove algorithm, can reproduce results obtained with human subjects using the implicit association test. The results suggest that implicit associations might be partly influenced by similarities of words which co-occur with concepts measured by implicit association test. The method could also be fruitful in predicting implicit associations and examining possible associations of people in the past.

Word embeddings might also be useful for preparation of stimuli in tasks where semantic similarity of words is important, such as in semantic priming or memory research. The method provides a means of creating stimuli and also can be used to measure semantic similarity in models of performance on tasks depending on semantic similarity of words. For example, Howard and Kahana (2002) used LSA to examine how semantically similar words are recalled in sequence in a memory study. Similarly, the DeeseRoediger–McDermott paradigm (Roediger & McDermott, 1995) uses semantically related words to elicit false memories. The described methods could then

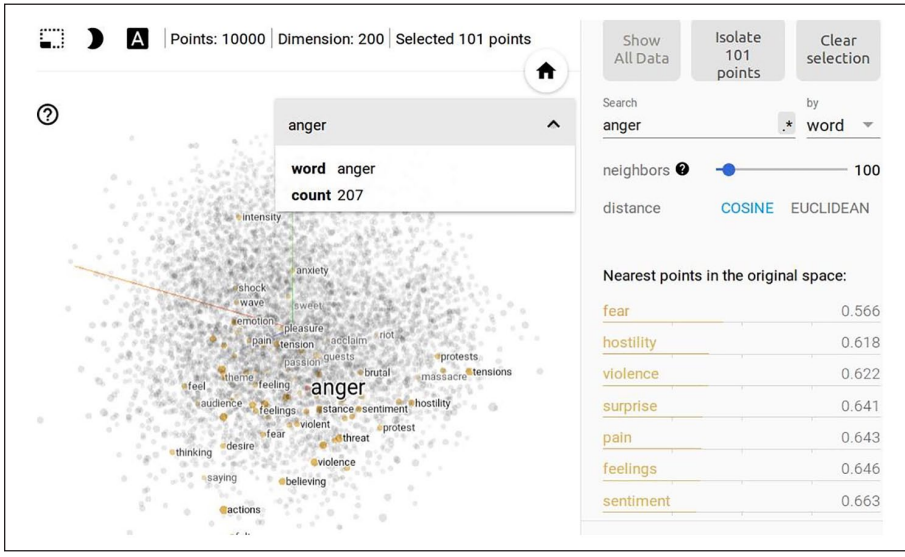

Figure 3. Nearest words to word "anger" (Embeddings Projector, Word2Vec 10K model).

be used to measure semantic similarity of words which could influence the probability or strength of the false memories.

The LDA algorithm is typically used for topic modeling. Based on analysis of input documents, these algorithms generate a list of topics. Each document is assigned a list of scores that expresses to what degree the document corresponds to each of the topics. A recent use of LDA and word2vec include detection of fake news on Twitter (Helmstetter & Paulheim, 2018). For other examples of uses of the LSA and LDA algorithms in a psychological context, we refer the reader to Chen and Wojcik (2016) and Altszyler, Ribeiro, Sigman, and Slezak (2017).

Text Annotation

Textual documents can be extended with additional structure using a variety of algorithms developed for natural language processing.

Syntactic Parsing. Analysis of a textual document often starts with syntactic tagging. This breaks the words in the input text into tokens and associates tokens with tags, such as parts of speech and punctuation. Syntactic parsing may also group tokens into larger structures, such as noun chunks or sentences. Other types of processing include lemmatization—reducing the different forms of a word to one single form—which is important particularly for inflectional languages, such as Czech.

The result of syntactic parsing is typically used in further linguistic processing but it also serves as a source of insights on the writing style of a particular group of subjects (O'Dea, Larson, Batterham, Calear, & Christensen, 2017).

Named Entity Recognition (NER). Syntactic parsing can already output noun chunks, such as names consisting of multiword sequences ("New York"). Named entity recognition goes one step further, by associating each of these noun chunks with an *entity type*. The commonly recognized types of entities are persons, locations, organizations, and miscellaneous entities that do not belong to the previous three groups (Tjong Kim Sang & De Meulder, 2003).

NER systems are pretrained on large tagged textual corpora and are thus generally language dependent. Adjusting them to a different set of target classes requires a substantial amount of resources, particularly of tagged training data.

Wikification: Linking Text to Knowledge Graphs. The NER results are somewhat limited in terms of the small number of types recognized and lack of additional information on the entity. A process popularly known as wikification addresses these limitations by linking entities to external knowledge bases. The reason why this process is sometimes called wikification is that multiple commonly used knowledge bases are derived from Wikipedia (Mihalcea & Csomai, 2007).

The first step in entity linking is called mention detection. The algorithm identifies parts of the input text, which can be linked to an entity in the domain of interest. For

Table 3. Example Wikification result for input string: "Late Apple Inc. Co-Founder Steve Jobs 'Testifies' In iTunes Case" generated by DBpedia Spotlight

URI	support	types	surfaceForm	offset	sim	Percentage of second rank
Apple Inc.	14402	Organisation, Company, Agent	Apple Inc.	5	1.00	2.87E-06
Steve Jobs	1944	Person, Agent	Steve Jobs	27	1.00	8.66E-11
ITunes	13634	Work, Software	iTunes	53	0.98	2.12E-02

Note. The column names have the following meaning. URI: values were stripped of the leading http://dbpedia.org/resource/, support: indicates how prominent is the entity by the number of inlinks in Wikipedia, types: were stripped of the leading http://dbpedia.org/ontology/, surfaceForm: the entity as it appears in the input tweet, offset: the starting position of the text in the input tweet in characters, sim: similarity between context vectors and the context surrounding the surface form, perc (percentageOfSecondRank): indicates confidence in disambiguation (the lower this score, the further the first ranked entity was "in the lead").

example, for input text "Diego Maradona scored a goal," mention detection will output "Diego Maradona" or the corresponding positions in the input text.

When mentions have been identified, the next step is their linking to the knowledge base. One of the computational challenges in this process is the existence of multiple matching entries in the knowledge base for a given mention. For example, the word "Apple" appearing in an analyzed Twitter message can be *disambiguated* in Wikipedia to Apple Inc or Apple (fruit).

Always assigning the most frequent meaning of the given word has been widely adopted as a base line in word sense disambiguation research (Navigli, 2009). When entity linking is performed, the knowledge base typically provides a machine-readable entity type, which might be more fine-grained than the type assigned by NER systems. An example of a Wikification output is shown in Table 3.

Entity Salience and Text Summarization. When text is represented by entities, an optional processing step is to determine the level of salience of the entity in the text. Entities with high salience can help to summarize content of longer documents, but the output of entity salience algorithms can also serve as input for subsequent processing such as document classification.

Supervised entity salience algorithms, such as the one described by Gamon, Yano, Song, Apacible, and Pantel (2013), are trained on a number of features derived from the entity mention (whether the word starts with an upper-case or lower-case letter), from the local context (how many characters the entity is from the beginning of the document), and global context (how frequently does the entity occur in inlinks and outlinks). Knowledge bases can be used as a complementary source of information (Dojchinovski, Reddy, Kliegr, Vitvar, & Sack, 2016).

Sentiment Analysis. With the proliferation of applications in social media, the analysis of sentiment and related psychological properties of text gained in importance.

Sentiment analysis encompasses multiple tasks, such as determining valence and intensity of sentiment, determination of subjectivity, and detection of irony (Serrano-Guerrero, Angel Olivas, Romero, & Herrera-Viedma, 2015).

Most systems rely on lexicon-based analysis, machine learning, or a combination of both approaches. Lexicon-based approaches rely on the availability of lists of words, terms, or complete documents which are preclassified into different categories of sentiment. A well-known example developed for psychometric purposes is the LIWC2015 Dictionary, which assigns 6,400 words into several dozen nuanced classes such as swear words, netspeak, or religion (Pennebaker, Boyd, Jordan, & Blackburn, 2015).

Applications in Behavioral Sciences. Entities linked to knowledge graphs can be used to improve results of many natural language processing tasks. Troisi, Grimaldi, Loia, and Maione (2018), for example, studied variables influencing the choice of a university by using wikification to find topics discussed in the context of writing about universities in various online sources. External information can be particularly useful in domains where the available documents are short and do not thus contain much information. To this end, Varga, Cano Basave, Rowe, Ciravegna, and He (2014) report significant improvement in performance when the content of tweets is linked to knowledge graphs as opposed to lexical-only content contained in the input tweets.

The LIWC system has been widely used in the behavioral sciences (see the article by Donohue et al., 2014). Among other topics, it has been used to study close relationships, group processes, deception, and thinking styles (Tausczik & Pennebaker, 2010). In general, it can be easily used to study differences in communication of various groups. For example, it was used to analyze psychological differences between Democrats and Republicans by Sylwester and Purver (2015). This research focused on general linguistic features, such as part of speech tags and sentiment analysis. The study found, for example, that those who identified as Democrats more commonly used first-person singular pronouns and that the expression of positive emotions was correlated with following Democrats, but not Republicans.

Many uses of sentiment analysis deal with microposts such as Twitter messages. Examples of this research include characterization of debate performance (Diakopoulos & Shamma, 2010) or analysis of polarity of posts (Speriosu, Sudan, Upadhyay, & Baldridge, 2011).

Document Classification

Document classification is a common task performed on top of a vector space representation of text, such as bag of words, but document classification algorithms can also take advantage of entity-annotated text (Varga et al., 2014). The goal of document classification is to assign documents in a given corpus to one of the document categories. The training data consist of documents for which the target class is already known and specified in the input data.

In the following, we describe centroid-based classifier, a well-performing algorithm. Next, we cover a few additional algorithms and tasks.

Centroid Classifier. The centroid classifier is one of the simplest classifiers working on top of the BOW representation (Han & Karypis, 2000). Input for the training phase is a set of documents for each target class and the output is a centroid for each category. Centroid is a word vector, which is intended to represent the documents in the category. It is computed as an average of word vectors of documents belonging to the category.

Application of the model works as follows. For each test document with an unknown class, its similarity to all target classes is computed using a selected similarity measure. The class with the highest similarity is selected. There are several design choices when implementing this algorithm such as the word weighting method, document length normalization, and the similarity measure. The common approach to the first two choices is TF-IDF, covered in the "Word Vectors and Word Embeddings" subsection, and L1 normalization. L1 normalization is performed by dividing each element in the given vector by the sum of absolute values of all elements in the vector. The similarity measure used for document classification is typically cosine similarity.

Other Tasks and Approaches. Centroid classifier is a simple approach, which has the advantage of good interpretability. The simplicity of the algorithm can make it a good choice for large data sets. Centroid-based classifiers are noted to have excellent performance on multiple different collections of documents but are not suitable for representing classes that contain fine-grained subclasses (Pang, Jin, & Jiang, 2015).

Support vector machines (SVMs) is a frequently used algorithm for text classification, which can be adapted for some types of problems where centroid-based classification cannot be reasonably used (Boser, Guyon, & Vapnik, 1992). According to experiments reported by Pang et al. (2015), SVM is a recommended algorithm for large balanced corpora. Balanced corpora have a similar proportion of documents belonging to individual classes. SVMs can also be adapted to hierarchical classification, where target classes can be further subdivided in subclasses (Dumais & Chen, 2000). Another adaptation of the text classification problem is multilabel text classification, where a document is assigned multiple categories.

Applications in Behavioral Sciences. Document classification methods have varied uses. One possible use is in predicting a feature of a person based on a text they wrote. For example, using a training set of documents, it is possible to train a model to distinguish between documents written by men and women. Given a document for which an author is not known, the algorithm may be able to say whether the document was more likely to be written by a man or a woman. Similarly, Komisin and Guinn, (2012), used SVM and Bayes classifier to identify persona types based on word choice. Profiling using SVMs was also successfully applied for distinguishing among fictional characters (Flekova & Gurevych, 2015).

The use of document classification can be further extended. Once the model is trained to classify documents using a list of features, it is possible to study and interpret the distinguishing features themselves. That is, it might be of interest not only to

be able to predict gender of the author of a document but also to say what aspects of the documents written by males and females differ.

External Knowledge Sources

Enrichment with external knowledge can be used to improve results of machine learning tasks, but the additional information can also help to gain new insights about the studied problem (Paulheim, 2018).

Two major types of knowledge sources for the machine learning tasks covered in this article are knowledge graphs and lexical databases. In this section, we cover DBpedia and Wikidata, prime examples of knowledge graphs which are semi-automatically generated from Wikipedia. For lexical databases, we cover WordNet, expert-created thesaurus with thousands of applications across many disciplines.

Knowledge Graphs

Resources providing a mix of information in structured and unstructured format are called knowledge bases. A knowledge base can be called a knowledge graph when information contained in it has a network structure and can be obtained with structured queries.[5] There is no universal graph query language used to obtain information from knowledge graphs, but the openly available knowledge graphs covered in this section support SPARQL (Harris, Seaborne, & Prud'hommeaux, 2013). The goal of a typical query is to retrieve a list of entities along with their selected properties given a set of conditions. Entity roughly corresponds to a thing in human knowledge described by the knowledge graph.

DBpedia[6] is one of the largest and oldest openly available knowledge graphs (Lehmann et al., 2015). The English version of DBpedia covers more than 6 million entities, but it is also available for multiple other languages. For a knowledge base to contain the information on an entity, it must have been previously populated. DBpedia is populated mostly by algorithms analyzing semistructured documents (Wikipedia articles).

Wikidata[7] is another widely used knowledge graph, which is available since 2012 (Vrandečic & Krötzsch, 2014). Wikidata currently contains information on 45 million items or entities. Similar to DBpedia, Wikidata is partly populated by robots extracting data from Wikipedia, but it also allows the general public to contribute. Information from DBpedia and Wikidata can be obtained either through a web interface, with a SPARQL query, or by downloading the entire knowledge graph.

Other Knowledge Graphs. Thanks to the use of global identifiers for entities and their properties, many knowledge graphs are connected to the Linked Open Data Cloud. A list of more than 1,000 knowledge graphs catalogued by domain, such as life sciences, linguistics, or media, is maintained at https://lod-cloud.net/. In addition to open initiatives, there are proprietary knowledge graphs, which can be accessed via various APIs. These include Google Knowledge Graph Search API, Microsoft's Bing Entity Search API, and Watson Discovery Knowledge Graph.

Applications in Behavioral Sciences. One of the main uses of Knowledge graphs in the behavioral sciences is in the study of spread of disinformation (Ciampaglia et al., 2015; Fernandez & Alani, 2018). DBpedia is used for computational fact checking in several systems, including DeFacto (Gerber et al., 2015). Knowledge graphs are also used to enhance understanding of text by linking keywords and entities appearing in text to more general concepts. DBpedia has been also used to analyze the discourse of extremism-related content, including detection of offensive posts (O'Halloran et al., 2019; Saif, Dickinson, Kastler, Fernandez, & Alani, 2017; Soler-Company & Wanner, 2019).

WordNet and Related Lexical Resources

WordNet is a large English thesaurus that was created at Princeton University (Fellbaum, 2010). It covers nouns, verbs, adjectives, and adverbs. Synonyms are grouped together into synsets, that is, sets of synonyms. In WordNet 3.0, there are about 150,000 words grouped into more than 100,000 synsets. For each synset, there is a short dictionary explanation available called a *gloss*. There are several types of relations captured between synsets depending on the type of synset such as hypo-hypernymy, antonymy, or holonymy-meronymy. For example, for the noun "happiness" WordNet returns the synonym "felicity" and for "sad" the antonym "glad."

Use for Word Similarity Computation. WordNet is also an acclaimed lexical resource that is widely used in the literature for word similarity and word disambiguation computations. With Wordnet, one can algorithmically compute semantic similarity between a word and one or more other words. There are many algorithms or formulas for this purpose, which differ predominantly in the way they use the paths between the two words in the WordNet thesaurus as well as in the way they use external information—such as how rare the given word is in some large collection of documents. Well-established algorithms include the Resnik (1995) and Lin (1998) measures. A notable example in the behavioral context is the Pirro and Seco (2008) measure, which is inspired by the feature-based theory of similarity proposed by Tversky (1977).

Use for Sentiment Analysis. Further elaborating on the variety of possible uses of WordNet, recent research has provided an extension called "Wordnet-feelings" (Siddharthan et al., 2018), which assigns more than 3,000 WordNet synsets into nine categories of feeling. A related resource used for sentiment classification is SentiWordNet (Baccianella, Esuli, & Sebastiani, 2010).

Applications in Behavioral Sciences. WordNet is often used in the behavioral sciences to complement free association norms, which are costly and time-consuming to develop (Maki, Krimsky, & Munoz, 2006). Maki, McKinley, and Thompson (2004) showed that semantic distance computed from WordNet is related to participants' judgment of similarity.

Specific uses of WordNet in behavioral research include studies of perceptual inference (Johns & Jones, 2012), access to memory (Buchanan, 2010), and predicting survey responses (Arnulf, Larsen, Martinsen, & How Bong, 2014). For example, Arnulf

et al. (2014) showed that semantic similarity of items computed with an algorithm using WordNet predicted observed reliabilities of scales as well as associations between different scales.

Related Work

In this section, we point readers to several works which also aimed at communicating recent advances in machine learning algorithms and software to researchers in behavioral science. McArdle and Ritschard (2013) provide an edited volume exploring many topics and applications at the intersection of exploratory data mining and the behavioral sciences. Methodologically, the book has a strong focus on decision tree learning, exploring its use in areas as diverse as life-course analysis, the identification of academic risks, and clinical prediction, to name but a few.

Tonidandel, King, and Cortina (2018) provide a discussion of "big data" methods applicable to organizational science, which is complemented by a list of various software systems across different programming languages (Python, R, . . .), environments (cloud, desktop), and tasks (visualization, parallel computing, . . .). Varian (2014) reviews selected "big data" methods in the context of econometrics, focusing on random forests and trees.

Chen and Wojcik (2016) give a practical introduction to "big data" research in psychology, providing an end-to-end guide covering topics such as selection of a suitable database and options for data acquisition and preprocessing, focusing on web-based APIs and processing HTML data. Their article focuses on methods suitable for text analysis, giving a detailed discussion including worked examples for selected methods (LSA, LDA). There is also a brief overview of the main subtasks in data mining, such as classification or clustering. The article also contains advice on processing large data sets, referring to the MapReduce framework.

Machine Learning Versus Big Data

While many articles use the term *big data*, most data sets in behavioral science would not qualify. According to Kitchin (2017) and Gandomi and Haider (2015), big data consist of terabytes or more of data. Consequently, "big data" requires adaptation of existing algorithms, so that they can be executed in a parallel fashion in a cloud or in grid-based computational environments. R users have the option to use some of the R packages for high performance computing.[8] Examples of dedicated big data architectures include Apache Spark or cloud-based machine learning services (Hashem et al., 2015).

Machine Learning as a Service (MLaaS). In this article, we focused on packages available in the R ecosystem.[9] The R data frame, used usually to store research data, is principally limited to processing data that do not exceed the size of available memory (Lantz, 2015), which puts constraints on the size of analyzed data for packages that use this structure. As noted above, there are several options for scaling to larger data, but

the behavioral scientist may find it most convenient to use a cloud-based machine learning system, such as Bigml.[10]

MLaaS systems provide comfortable web-based user interface, do not require installation or programming skills, and can process very large data sets. The disadvantage of using API-based or web tools such as MLaaS include impeded reproducibility of studies which used them for analysis. The researcher reproducing the analysis may not be able to employ the specific release of the system that was used to generate the results. The reason is that these systems are often updated.

Conclusion

The continuing shift of communication and interaction channels to online media provides a new set of challenges and opportunities for the behavioral scientist. The fact that much interaction is performed online also allows for evolution in research methods. For example, certain research problems may no longer require costly laboratory studies as suitable data can be obtained from logs of interactions automatically created by social networking applications and web sites. This article aimed to introduce a set of methods that allow for analyses of such data in a transparent and reproducible way. Where available, we therefore suggested software available under an open source license.

We put emphasis on selecting proven algorithms, favoring those that generate interpretable models that can be easily understood by a wide range of users. When easy-to-interpret models lead to worse results then more complex models, it is possible to use the latter to improve the former. For example, Agrawal, Peterson, and Griffiths (2019) used neural networks to predict moral judgments. Because the neural network model was itself not easily interpretable, they looked at situations where the neural network model fared particularly well in comparison to a simpler, but more easily interpretable, choice model. They then iteratively updated the choice model to better predict judgments in the situations where the neural network model predicted better. A similar strategy can be used generally by behavioral scientists if interpretability of the models is considered valuable.

There are several other noteworthy areas of machine learning that could be highly relevant to particular subdomains of behavioral science, we left them uncovered due to space constraints. These include reinforcement learning, image processing, and the discovery of interesting patterns in data. Another interesting technological trend in terms of how data are collected and processed is the connection between crowdsourcing services, and Machine Learning as a Service offering. Crowdsourcing may decrease the costs by outsourcing some parts of research such as finding and recruiting participants and can also aid replicability by engaging large and varied participant samples. See article by Crump, 2019 which is in this issue on the challenges of recruiting participants. Employment of MLaaS systems may have benefits in terms of setup costs, ease of processing and the security of the stored data. On the other hand, experimenters relying on crowdsourcing lose control of the laboratory environment. MLaaS

may impede reproducibility and accountability of the analysis since results of these systems may vary in time as they are often updated.

Overall, we expect that the largest challenge for the behavioral scientist in the future will not be the choice or availability of suitable machine learning methods. More likely, it will be ensuring compliance with external constraints and requirements concerning ethical, legal, and reproducible aspects of the research.

Authors' Note

Only articles referenced from the "Applications in Behavioral Sciences" subsections are included in this bibliography. Remaining references can be found in Supplemental Appendix A.

Declaration of Conflicting Interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The authors disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: TK was supported by Faculty of Informatics and Statistics, University of Economics, Prague by grant IGA 33/2018 and by institutional support for research projects. TK would like to thank BigML Inc. for providing subscription that allowed to test processing of large datasets in BigML.com free of charge. The work of ŠB was supported by the Internal Grant Agency of the Faculty of Business Administration, University of Economics, Prague (Grant No. IP300040).

Supplemental Material

Supplemental material for this article is available online.
Due to the large number of papers covered by this review, only papers referenced from the "Applications in behavioral sciences" subsections are included in the main bibliography. Remaining references were put into online Appendix A. Online Appendix B contains an overview of selected software packages implementing some of the methods discussed in the main text.

Notes

1. The Classification Based on Associations algorithm does not generate a rule set but a rule *list*. The difference is that in a predictive rule list, the order of rules is important as it signifies precedence.
2. https://catalog.data.gov/dataset/zip-code-data
3. It should be emphasized that this comparison is only illustrative (cf. Altszyler et al., 2017; Cimiano, Schultz, Sizov, Sorg, & Staab, 2009).
4. ESA assumes that documents in the collection form a knowledge base such that each document covers a different topic.
5. https://www.ontotext.com/knowledgehub/fundamentals/what-is-a-knowledge-graph/
6. https://dbpedia.org
7. https://wikidata.org
8. https://cran.r-project.org/web/views/HighPerformanceComputing.html
9. For a general introductory reference to R, we refer readers to Torgo (2010).

10. https://bigml.com

References

NOTE: Due to the large number of papers covered by this review, only papers referenced from the "Applications in behavioral sciences" subsections are included in the main bibliography. Remaining references were put into online Appendix A.

Altszyler, E., Ribeiro, S., Sigman, M., & Fernandez Slezak, D. (2017). The interpretation of dream meaning: Resolving ambiguity using Latent Semantic Indexing in a small corpus of text. *Consciousness and Cognition*, *56*, 178-187.

Behrenbruch, K., Atzmüller, M., Evers, C., Schmidt, L., Stumme, G., & Geihs, K. (2012). A personality based design approach using subgroup discovery. In M. Winckler, P. Forbrig, & R. Bernhaupt (Eds.), *Human-centered software engineering* (pp. 259-266). Berlin, Germany: Springer.

Billari, F. C., Fürnkranz, J., & Prskawetz, A. (2006). Timing, sequencing, and quantum of life course events: A machine learning approach. *European Journal of Population*, *22*, 37-65.

Bouyssou, D., Jacquet-Lagrèze, E., Perny, P., Slowiński, R., Vanderpooten, D., & Vincke, P. (2002). *Aiding decisions with multiple criteria—Essays in honor of Bernard Roy*. Boston, MA: Kluwer Academic.

Bradley, R. A., & Terry, M. E. (1952). The rank analysis of incomplete block designs—I. The method of paired comparisons. *Biometrika*, *39*, 324-345.

Bredereck, R., Chen, J., Niedermeier, R., & Walsh, T. (2017). Parliamentary voting procedures: Agenda control, manipulation, and uncertainty. *Journal of Artificial Intelligence Research*, *59*, 133-173.

Caliskan, A., Bryson, J. J., & Narayanan, A. (2017). Semantics derived automatically from language corpora contain human-like biases. *Science*, *356*(6334), 183-186.

Chen, E. E., & Wojcik, S. P. (2016). A practical guide to big data research in psychology. *Psychological Methods*, *21*, 458.

Ciampaglia, G. L., Shiralkar, P., Rocha, L. M., Bollen, J., Menczer, F., & Flammini, A. (2015). Computational fact checking from knowledge networks. *PLoS One*, *10*, e0141938.

Coughlin, P. J. (2008). *Probabilistic voting theory*. Cambridge, England: Cambridge University Press.

Cuhadar, E., & Druckman, D. (2014). Representative decision making: Challenges to democratic peace theory. In M. Galluccio (Ed.), *Handbook of international negotiation: Interpersonal, intercultural, and diplomatic perspectives* (pp. 4-14). Dordrecht, Netherlands: Springer.

De Rose, A., & Pallara, A. (1997). Survival trees: An alternative non-parametric multivariate technique for life history analysis. *European Journal of Population*, *13*, 223-241.

Deng, L., & Liu, Y. (2018). *Deep learning in natural language processing*. New York, NY: Springer Verlag.

Diakopoulos, N. A., & Shamma, D. A. (2010). Characterizing debate performance via aggregated twitter sentiment. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '10; pp. 1195-1198). Atlanta, GA: ACM.

Donohue, W. A., Yuhua, L., & Daniel, D. (2014). "Validating LIWC dictionaries: The Oslo I accords." *Journal of Language and Social Psychology, 33*, 282-301.

Druckman, D. (1993). The situational levers of negotiating flexibility. *Journal of Conflict Resolution*, *37*, 236-276.

Druckman, D., Harris, R., & Fürnkranz, J. (2006). Modeling international negotiation: Statistical and machine learning approaches. In R. Trappl (Ed.), *Programming for peace: Computer-*

aided methods for international conflict resolution and prevention. Vol. 2: Advances in group decision and negotiation (pp. 227-250). Dordrecht, Netherlands: Kluwer Academic.

Feng, H., Lesot, M.-J., & Detyniecki, M. (2010). Using association rules to discover color-emotion relationships based on social tagging. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems* (pp. 544-553). New York, NY: Springer.

Fernandez, M., & Alani, H. (2018). Online misinformation: Challenges and future directions. In *Companion Proceedings of the Web Conference 2018* (WWW '18, pp. 595-602). Lyon, France: International World Wide Web Conferences Steering Committee.

Flekova, L., & Gurevych, I. (2015). Personality profiling of fictional characters using sense-level links between lexical resources. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (pp. 1805-1816). Lisbon, Portugal: Association for Computational Linguistics.

Fürnkranz, J., Petrak, J., & Trappl, R. (1997). Knowledge discovery in international conflict databases. *Applied Artificial Intelligence*, *11*, 91-118.

Gerber, D., Esteves, D., Lehmann, J., Bühmann, L., Usbeck, R., Ngonga Ngomo, A. C., & Speck, R. (2015). Defacto-temporal and multilingual deep fact validation. *Journal of Web Semantics*, *35*(Pt. 2), 85-101.

Goh, D. H., & Ang, R. P. (2007). An introduction to association rule mining: An application in counseling and help-seeking behavior of adolescents. *Behavior Research Methods*, *39*, 259-266.

Greene, M. N., Morgan, P. H., & Foxall, G. R. (2017). Neural networks and consumer behavior: Neural models, logistic regression, and the behavioral perspective model. *Behavior Analyst*, *40*, 393-418.

Helmstetter, S., & Paulheim, H. (2018, August 28-31). *Weakly supervised learning for fake news detection on Twitter*. *Paper published in 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, *Barcelona, Spain*.

Herawan, T., Vitasari, P., & Abdullah, Z. (2011). Mining interesting association rules of student suffering mathematics anxiety. In J. Mohamad Zain, W. Maseri bt Wan Mohd, & E. El-Qawasmeh (Eds.), *Software engineering and computer systems* (pp. 495-508). Berlin, Germany: Springer.

Houghton, G. (2004). Introduction to connectionist models in cognitive psychology: Basic structures, processes, and algorithms. In *Connectionist models in cognitive psychology* (pp. 11-19). New York, NY: Psychology Press.

Howard, M. W., & Kahana, M. J. (2002). When does semantic similarity help episodic retrieval? *Journal of Memory and Language*, *46*, 85-98.

Komisin, M., & Guinn, C. (2012). Identifying personality types using document classification methods. In *Proceedings of Twenty-Fifth International Florida Artificial Intelligence Research Society Conference* (pp. 232-237). Palo Alto, CA: Association for the Advancement of Artificial Intelligence.

Koski, N. (2004). Impulse buying on the Internet: Encouraging and discouraging factors. *Frontiers of E-business Research*, *4*, 23-35.

Kotsiantis, S., Pierrakeas, C., & Pintelas, P. (2002). *Efficiency of machine learning techniques in predicting students' performance in distance learning systems*. Agrinio, Greece: University of Patras.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84-90.

Langville, A. M., & Meyer, C. D. (2012). *Who's #1? The science of rating and ranking*. Princeton, NJ: Princeton University Press.

Luce, R. D. (1959). *Individual choice behavior: A theoretical analysis*. New York, NY: Wiley.
Maki, W. S., Krimsky, M., & Munoz, S. (2006). An efficient method for estimating semantic similarity based on feature overlap: Reliability and validity of semantic feature ratings. *Behavior Research Methods*, *38*, 153-157.
Maki, W. S., McKinley, L. N., & Thompson, A. G. (2004). Semantic distance norms computed from an electronic dictionary (WordNet). In *Behavior Research Methods, Instruments, & Computers*, *36*, 421-431.
Marden, J. I. (1995). *Analyzing and modeling rank data*. Boca Raton, FL: Chapman & Hall.
McArdle, J. J., & Ritschard, G. (Eds.). (2013). *Contemporary issues in exploratory data mining in behavioral sciences*. New York, NY: Routledge.
Moe, W. W. (2003). Buying, searching, or browsing: Differentiating between online shoppers using in-store navigational clickstream. *Journal of Consumer Psychology*, *13*, 29-39.
Murphy, J., Hofacker, C., & Mizerski, R. (2006). Primacy and recency effects on clicking behavior. *Journal of Computer-Mediated Communication*, *11*, 522-535.
O'Halloran, K. L., Tan, S., Wignell, P., Bateman, J., Pham, D.-S., Grossman, M., & Vande Moere, A. (2019). Interpreting text and image relations in violent extremist discourse: A mixed methods approach for big data analytics. *Terrorism and Political Violence*, *31*, 454-474.
Plackett, R. (1975). The analysis of permutations. *Applied Statistics*, *24*, 193-202.
Rao, V. R., Green, P. E., & Wind, J. (2007). *Applied conjoint analysis*. Thousand Oaks, CA: Sage.
Rodgers, S., & Thorson, E. (2000). The interactive advertising model: How users perceive and process online ads. *Journal of Interactive Advertising*, *1*(1), 41-60.
Roediger, H. L., & McDermott, K. B. (1995). Creating false memories: Remembering words not presented in lists. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *21*, 803.
Rosenfeld, A., Zuckerman, I., Azaria, A., & Kraus, S. (2012). Combining psychological models with machine learning to better predict people's decisions. *Synthese*, *189*(Suppl. 1), 81-93.
Rossi, F., Brent Venable, K., & Walsh, T. (2011). *A short introduction to preferences: Between artificial intelligence and social choice* (Synthesis Lectures on Artificial Intelligence and Machine Learning). London, England: Morgan & Claypool.
Saif, H., Dickinson, T., Kastler, L., Fernandez, M., & Alani, H. (2017). A semantic graph-based approach for radicalisation detection on social media. In *Proceedings of the European Semantic Web Conference* (pp. 571-587). New York, NY: Springer.
Senecal, S., Kalczynski, P. J., & Nantel, J. (2005). Consumers' decision-making process and their online shopping behavior: A clickstream analysis. *Journal of Business Research*, *58*, 1599-1608.
Shultz, T. R. (2013). Computational models in developmental psychology. In P. D. Delazo (Ed.), *The Oxford handbook of developmental psychology. Vol. 1: Body and mind* (p. 477). Oxford, England: Oxford University Press.
Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., . . . Hassabis, D. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, *529*, 484-489.
Soler-Company, J., & Wanner, L. (2019). Automatic classification and linguistic analysis of extremist online material. In I. Kompatsiaris, B. Huet, V. Mezaris, C. Gurrin, W. H. Cheng, & S. Vrochidis (Eds.), *Multimedia modeling* (pp. 577-582). Cham, Switzerland: Springer International.

Speriosu, M., Sudan, N., Upadhyay, S., & Baldridge, J. (2011). Twitter polarity classification with label propagation over lexical links and the follower graph. In *Proceedings of the First Workshop on Unsupervised Learning in NLP* (EMNLP '11, pp. 53-63). Edinburgh, Scotland: Association for Computational Linguistics.

Stillman, P. E., Shen, X., & Ferguson, M. J. (2018). How mouse-tracking can advance social cognitive theory. *Trends in Cognitive Sciences*, *22*, 531-543.

Stumpf, S., Rajaram, V., Li, L., Wong, W.-K., Burnett, M. M., Dietterich, T. G., . . . Herlocker, J. L. (2009). Interacting meaningfully with machine learning systems: Three experiments. *International Journal of Human-Computer Studies*, *67*, 639-662.

Sylwester, K., & Purver, M. (2015). Twitter language use reflects psychological differences between democrats and republicans. *PloS One*, *10*, e0137422.

Tausczik, Y. R., & Pennebaker, J. W. (2010). The psychological meaning of words: LIWC and computerized text analysis methods. *Journal of Language and Social Psychology*, *29*, 24-54.

Thurstone, L. L. (1927). A law of comparative judgement. *Psychological Review*, *34*, 278-286.

Troisi, O., Grimaldi, M., Loia, F., & Maione, G. (2018). Big data and sentiment analysis to highlight decision behaviours: A case study for student population. *Behaviour & Information Technology*, *37*, 1111-1128.

Varga, A., Cano Basave, A. E., Rowe, M., Ciravegna, F., & He, Y. (2014). Linked knowledge sources for topic classification of microposts: A semantic graph-based approach. *Journal of Web Semantics*, *26*, 36-57.

Walsh, C. G., Ribeiro, J. D., & Franklin, J. C. (2017). Predicting risk of suicide attempts over time through machine learning. *Clinical Psychological Science*, *1*, 12.

Wilson, T., Wiebe, J., & Hwa, R. (2004). Just how mad are you? Finding strong and weak opinion clauses. *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI-04)*, *4*, 761-769.

Author Biographies

Tomáš Kliegr is an associate professor at the Department of Information and Knowledge Engineering, Faculty of Informatics and Statistics, University of Economics, Prague. His research interests include rule learning, knowledge graphs, and natural language processing. He is currently working on the analysis of interactions between psychological phenomena and machine learning models.

Štěpán Bahník is a researcher at the Faculty of Business Administration of the University of Economics, Prague. He is mainly interested in judgment and decision making. In particular, he does research on processing fluency, anchoring, moral judgment, and choice blindness. He is also interested in psychological methodology, statistics, programming, and intersections between psychology and computer science.

Johannes Fürnkranz is a professor for Knowledge Engineering at TU Darmstadt, Germany. His research focuses on symbolic and logical techniques for learning qualitative, interpretable models from data, most notably inductive rule learning and preference learning, with applications in engineering, game playing, and the social sciences.