Data-Driven, Statistical Learning Method for Inductive Confirmation of Structural Models

Wolfgang Maass
Saarland University
wolfgang.maass@iss.uni-saarland.de

Iaroslav Shcherbatyi
Saarland University
iaroslav.shcherbatyi@iss.uni-saarland.de

Abstract

Automatic extraction of structural models interferes with the deductive research method in information systems research. Nonetheless it is tempting to use a statistical learning method for assessing meaningful relations between structural variables given the underlying measurement model. In this paper, we discuss the epistemological background for this method and describe its general structure. Thereafter this method is applied in a mode of inductive confirmation to an existing data set that has been used for evaluating a deductively derived structural model. In this study, a range of machine learning model classes is used for statistical learning and results are compared with the original model.

1. Introduction

Structural equation modeling (SEM) has become a dominant statistical method for evaluating theories in behavioral sciences. At first, models and data collection study designs are developed that are subsequently evaluated by raised data sets. Later, model fitting is assessed and used as positive or negative support of an initially hypothesized model [1]. Even though that questions arise with this epistemological procedure, known as Null Hypothesis Significance Test (NHST) [2], it is quite common in information systems research that only one hypothetical model is tested even despite the existence of many potentially meaningful models supported by the same data set [3].

SEM consists of a combination of measurement and structural models [4]. With a given set of indicator and structural variables, all possible models can be enumerated, evaluated against one or many data sets in principle. In practice this is only a meaningful approach for small variable sets. Search space complexity is generally reduced by heuristics and theory-driven constraints. For instance, prohibition of bidirectional paths, focus on non-recursive structural models or search procedures looking for locally maximizing models (e.g., Maximum Likelihood). Resulting models are compared by fit indices, such as RMSEA and GFI [4]. Nonetheless, search spaces for non-trivial sets of measurement and structural variables remain extremely large. Therefore w.l.o.g, we confine our research to search spaces on structural models spanned by structural variables and keep measurement models constant.

Due to the lack of research on model search, we start with an already published model as a baseline [5]. By using a data-driven, statistical learning search procedure, we automatically extract a candidate model with various statistical learning algorithms and compare their performance. Finally we compare the best performing extracted model with the original baseline model. In the discussion we analyze general capabilities of statistical learning models for data-driven extraction of conceptual models and deliberate on the general potential of model search for information systems research. Finally limitations and an outlook are discussed.

2. Inductive, deductive and hybrid modes of research

Structural equation modeling (SEM) is used for evaluating parameters defined by a hypothesized underlying model by statistical analysis of empirical measurements [4] with an emphasis on analyzing covariance [6] or component-based [7] structures between observed and latent variables [8]. Conventionally theory drives model specification that is assessed by statistical analysis of empirically assessed data [4]. By theoretical considerations the latent variable model consisting of exogenous and endogenous variables is described in closed form as follows: eta = B η + Γ ξ + ζ with η a vector of latent endogenous variables to be explained, ξ a vector of latent exogenous variables, zeta capturing disturbances and B and Γ are coefficient matrices [8]. Similarly, measurement models

URI: http://hdl.handle.net/10125/41850
ISBN: 978-0-9981331-0-2

are described in close form as follows: $x = \Lambda_x \xi + \delta$ and $y = \Lambda_y \eta + \varepsilon$ [8].

Different structural models fit equally well to collected data and in cases with over-fitting, researchers might be even tempted to favor models that are far to specific. Also complexity-penalizing indices, such as AIC and BIC, tend to favor models with less parameters by design [3].

By reversing NHST, we ask how many potentially meaningful structural models are possible if *n* variables and *p* paths are given that exhibit equally likely interpretations of data [9] as a similarity class of models, so-called "confounds" [3].

The search model approach resembles and is even anchored in the discussion currently conducted around the term "Big Data" [10-12]. A key claim of proponents of big data research is that patterns can be extracted automatically from large data sets [13]. In contrast, it is argued that also big data is subject to sampling bias, dependent on viewpoints, tools for collecting data, and data ontologies and the need for epistemological interpretations by domain experts [12].

To consolidate the extreme research positions of inductivism and empiricism, Kitchin argues for an hybrid approach that combines inductive, data-driven and deductive methods [12]. Deductive research starts with theories and models that are evaluated by empirical studies while inductive research starts from data ontologies and data collections and results in inductively derived theories and models (cf. Fig. 1). A hybrid approach is proposed by Kitchin that starts with an inductive mode of research resulting in hypotheses that are, in turn, evaluated by deductive research (cf. Fig. 1). As with pure deductive and inductive modes, an hybrid mode is open for circular activations for further theory development.

In deductive research, models are derived from theory and empirical testing of models provides insights resulting in adaptation of theories. Inductive research provides insights derived from data. Research with big data often follows a pure inductive paradigm resulting in fragmented insights reinforcing convenient "as-if" assumptions [14]. Therefore we argue for a hybrid approach in which inductively derived insights become assumptions for models that are evaluated by deductive research (cf. Figure 1).

A hybrid mode of research enables researchers to deal with large amounts of data, (semi-) automatically derive potentially interesting hypotheses that are subsequently tested by rigorous deductive research. Hypotheses derived from data require interpretation from a domain perspective. Hypotheses do not come out of nowhere but are grounded in data ontologies that are input to inductive research. Data ontologies provide a lens by which researchers look at basic signals in a certain domain of interest, such as answers to single items in structural models. Different hypotheses will be derived from data depending on data ontologies, data collection technologies, and data analytical models. Hence, there is no objective truth in data but biased results depending on viewpoints [12].

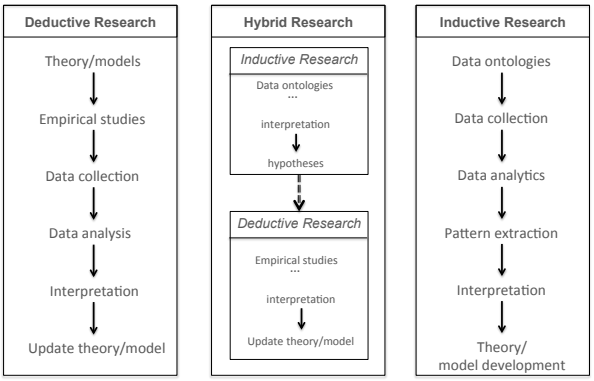

Figure 1. Hybrid mode of research

Data-driven inductive and hybrid modes of research are in an embryonic state with respect to epistemology and research results that go beyond trivial correlations (e.g., [14, 15]). Inductively extracted patterns between latent concepts grounded in compounds of data dimensions as typical for empirical research in information systems are rare. Few researchers have proposed methods for inductive modes of research [16].

In this early stage of an evolving research paradigm, the question arises whether it is still valid to assume that theoretically sound models can only come from researcher's deliberation on theoretical knowledge and cannot be derived from data.

By taking the stance of inductive research, we start with the hypotheses that statistical learning methods are able to derive structural models from measurement models alone without consideration of theoretically assumed structural relations between latent variables. This hypothesis can be tested relative to established deductive methods in two ways. First, it can be used by an ex-ante mode, i.e. structural models are derived inductively and used as hypothesis for a subsequent deductive study (inductive exploration). Second, models previously derived by deductive studies are confirmed by independent inductive studies (inductive confirmation). In this paper, we describe a method for inductive confirmation of previously evaluated models by applying various statistical learning methods.

In detail, we focus on covariance-based structural equation model (SEM) that is used as a standard tool for research in social sciences, and the information systems community in particular. Assuming measurement models with strong ties (covariance-based or PLS-based) with their latent structural variables they depend

on [8], we investigate whether structural models can be automatically extracted from data instead of deriving it from theory alone. In the following, we will discuss our research question by using an existing deductive study as a baseline. We will show how data is analyzed by various methods of statistical learning resulting in a set of structural relations as input for a structural model. Resulting structural models are compared with the original model.

3. Modeling approach and methodology

We focus on path analysis in structural models and abstain from using fit estimates in SEM for comparing models. This is done for the same reason as discussed by Fife et al. (2014), i.e. we presuppose particular measurement models but leave freedom for choosing structural models randomly [3]. Fife et al. simply generate all potential models for a set of structural variables and evaluate these structural models by RMSEA fit index. For the set of structural variable introduced in [5], this approach targets the evaluation of $1.19*10^{21}$ different structural models that is neither feasible nor reasonable with non-trivial data sets. Thus, the approach proposed by Fife et al. (2014) only works for small sets of structural variables and small structural models.

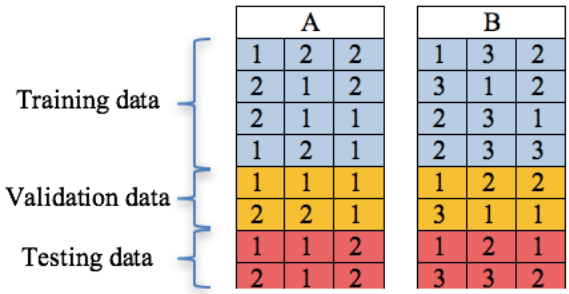

Figure 2. Example data split

Instead, we apply and compare results of different data-driven statistical learning models by introducing a targeted accuracy metric. Resulting models are tested and evaluated by separate test and evaluation data sets. We assume that we are given a set of concepts, and we want to establish whether there exist directed relational dependencies between different concepts. To do so, we assume that we are also given a set of observations, where any observation for every concept contains a vector or real values describing the concept.

We concentrate on establishing all pairwise dependencies between concepts. For every pair of concepts, we solve the supervised learning problem of predicting values describing observation of concept B given the values of corresponding observation of the concept A and vice versa. We use a value proportional to the test accuracy of obtained predictive models to obtain a value that describes the strength of a relation between two concepts. This value allows establishing an ordering of all possible relations by strength, and among them we select n strongest ones, where n characterizes how complex the model should be and is provided by user. The resulting relation set is used as input for evaluation methods for structural models, such as covariance-based SEM.

3.1. Assessment of relationship strengths

For every two structural variables A and B, we define the strength of a path relation from A to B as a value proportional to the accuracy with which indicator variables of B can be predicted given indicator variables of A. We call this value "improvement over random guess (IRG)". Let this value be denoted as I_{AB} for two structural variables A and B. Then this value is defined as a ratio

$$I_{AB} = R_{AB} / M_{AB}$$

where R_{AB} is the best error that can be obtained for prediction of values of B while neglecting corresponding values of A, and M_{AB} is the error achieved with a predictive model which takes as input values for indicators of A and tries to estimate corresponding values for indicators of B.

The larger the value of I_{AB}, the better B can be predicted given the values of A. In particular, if the value of I_{AB} is 1.0, then the model is not better than the model which simply makes random guesses for indicators of B while discarding the inputs, i.e. indicators of A. Values larger than 1.0 indicate amount of improvement achieved over the random model. Values larger than 1.0 indicate amount of improvement that can be achieved over the random model.

Normalization by random model is used in order to account for the concepts for which the distribution of their values is unbalanced and where predicting the most likely output already leads to a small error rate, compared to other concepts.

3.2. Data preprocessing

For evaluation of the values of R_{AB} and M_{AB}, firstly the following manipulations on the data are performed. Consider representation of all the data on indicators available for the variables A and B (cf. Figure 2). Every pair of rows that corresponds to the same observation is taken as a pair of inputs and outputs. The data is split into three parts: training data (50% of all pairs),

validation data (25%), and the rest of the data as testing data (cf. Figure 2).

3.3. Computing performance of a random model R_{AB}

Let A_i denote the i-th row out of n in total corresponding to the values describing variable A in i-th observation, and similarly B_i denote the i-th row describing values of variable B in i-the observation. We compute the value of the R_{AB} with a split as follows. Let A^{Tr} and B^{Tr} denote training subsets and A^{Ts} and B^{Ts} denote testing subsets of rows describing variables A and B.
For every row in B^{Ts} we obtain a random prediction by sampling uniformly a row from the training subset B^{Tr}. Let a set of such samples be denoted as $B^{Ts'}$. We measure the accuracy of such random predictions in terms of RMSE on test set:

$$R'_{AB} = \sqrt{\frac{1}{n} \sum_{i=1...n} (B_i^{Ts} - B_i^{Ts'})^2}$$

Whole rows are sampled from a training set such that the distribution of outputs of the random model corresponds closely to the actual distribution of outputs, as for example sampling rows of values where every value is sampled uniformly can lead to a distribution of outputs of random model which does not correspond to the actual distribution. Such a sampling scheme particularly helps to capture possible correlations between the values in the row, which could potentially be exploited to increase the performance of a random model.
In order to decrease the variance of the estimate of random model performance, we repeat the procedure outlined above for 100 times and average obtained values of R'_{AB}. The resulting average performance is taken as value of R_{AB}. We found experimentally that on average 100 trials allow to reduce variance to negligible values.

3.4. Computing performance of statistical learning model M_{AB}

We compute the value of the M_{AB} by applying a fitting procedure [17] to the problem of predicting the indicator values of B given the indicator values of A.
We use multiple classes of statistical learning models to estimate M_{AB} to see whether the values of IRG change significantly with different model classes. These classes are described in the following sections.

3.5. Parameter selection for statistical learning models

Every model class that we consider has a set of hyperparameters which need to be adjusted in order to maximize the performance of the trained predictive model.
For hyperparameter selection, we use the split of data as was used for the random model evaluation. Training split of data is used to train a model for a particular configuration of hyperparameters. The resulting model is evaluated on the validation set, which yields an estimate of model performance for particular setting of hyperparameters, and allows choosing the best out of a set of candidate hyperparameter configurations. Model performance is measured by RMSE of model predictions. Finally, a robust estimate of model performance with the best configuration selected on the validation set is obtained by evaluating the model on the test set. This estimate is taken as value for M_{AB}.
For models with small finite sets of possible configurations of hyperparameters, all possible configurations of hyperparameters can be enumerated in brute force manner. However, it is not uncommon that parameters of models are real numbers, which can attain arbitrary large or small values.
For enumeration of such hyperparameter values, a discrete subset of all possible values is used, taken in the wide interval of practically feasible values. This is justified by the fact that such continuous hyperparameters commonly are proportional to the complexity of the model [18]. The optimal value of such hyperparameters represents a tradeoff between under and overfitting [18] and, thus, is typically attained for a finite value of hyperparameter.

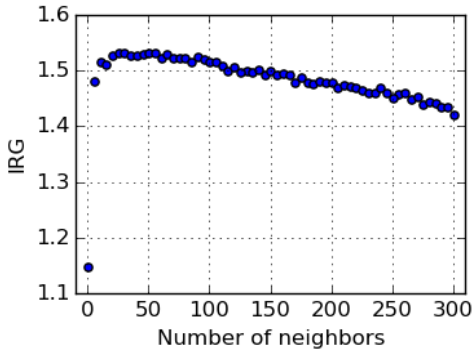

Figure 3. Example of tradeoff between complexity and overfitting of k nearest neighbor model. Smaller number of neighbors usually result in more complex decision boundaries [19].

Example of such effect is shown in Figure 3, where for varying values of parameter k of a k-nearest neighbors predictive model (cf. section 3.6.4. for details) the corresponding value of IRG is shown. When the model is too simple (right side of the graph in Figure 3) or too complex (left most side of graph in Figure 3) model does not perform well, but its performance is rather maximized for a tradeoff between model complexity and overfitting.

3.6. Classes of statistical learning models considered

Multiple classes of statistical learning models were evaluated to access effect of different classes used on IRG. All of those classes are equivalent in limit of all possible data as all of them are universally consistent [19-22]. However, we assumed that some models perform better than others if the amount of data is not large enough [23].
Some of the classes of predictive models cannot be directly used in settings where the output should be a vector. For those models, for every entry of the output vector a separate model is trained and predicted vector of outputs consists of outputs of separate models for every output entry.

3.6.1. Neural networks

Particular strength of neural networks is that they are easy to extend to multiple outputs [24] and recently demonstrate state of the art performance across many domains [25]. Furthermore, it is proven that neural networks are universally consistent [26] in contrast to commonly used linear models whose modeling capability is only limited to linear dependencies between inputs and outputs.
In this work a class of fully connected neural networks is used, due to their simplicity [27]. Such networks consist of input layer, output layer, and a number of hidden layers. These layers are arranged in a sequence, starting from the input layer, then proceeding with the hidden layers, and ending with the output layer. For all layers except for the input layers, the output of every neuron of the layer depends on the outputs of all neurons of the previous layer. Computing outputs y of the feed forward fully connected neural network can be expressed as follows for input x:

$$y = b_0 + W_n^T a(...a(b_1 + W_1^T a(b_0 + W_0^T x)))$$

where W_i is a matrix and b_i is a vector whose size is determined by number of neurons in the neural network, n denotes a number of layers, and function a is a vector valued function which applies an activation function [27] to every value in the input vector. In this work the rectified linear activation function was used [27].
Training of the neural networks was done using the stochastic gradient descent with momentum training procedure [17]. We posed the training procedure as direct optimization of the mean squared deviations of the predictions of the neural network from the desired target values.
Complexity of fully connected feed forward neural networks is determined by the *number of layers* used in the architrecture and *number of neurons* in every layer.

3.6.2. Kernel support vector machines

Linear support vector machines represent a class of linear models which is well-studied theoretically [28]. An extension of linear support vector machines are kernel support vector regression (SVR) [29], which can approximate any function where the output values are real numbers, given sufficient amount of data about the function.
Computing the outputs of SVR requires a kernel function [29], which takes two input vectors as parameters and outputs real number which characterizes how similar or dissimilar inputs are. One particularly popular choice for a kernel function is Radial Basis Kernel, for which it is proven that SVR is universally consistent [30].
In this paper, we use SVR with Radial Basis Kernel. Such kernel has one hyperparameter (*gamma*) which requires tuning. Additionally, SVR itself has two hyperparameters, one of which is a positive real number that controls the complexity of the model, and *epsilon* value which in original SVR formulation controls maximum error of the regression model [29].

3.6.3. Boosted models

Boosted models work by combining multiple simple models into one complex model. The output of such complex models are a weighted sum of outputs of simple models. Such model class is similar to neural network with a single hidden layer, where the output is also a weighted sum of outputs of neurons (simple models).
The difference between boosted models and neural networks is that they are trained differently. While training of the neural networks constitutes adjustment of parameters of all neurons at once, boosted models consist of simple models which are added sequentially one by one [31].
Similar to neural networks, complexity of boosted models is adjusted by the number of *nodes* used in the final model. Additionally, a *learning rate* value can be

adjusted, which tunes how much a new node added to the boosting model changes its outputs.

3.6.4. K nearest neighbors

K-nearest neighbors model (kNN) represents one of the simpler classes of statistical learning models [19]. The output of a kNN model for some input is the mean of k outputs which corresponds to inputs in a training set being most similar to a selected input value. Such similarity is defined by a similarity function, which is typically Euclidian distance.
The only hyperparameter of a kNN model is the number *k* of closest data points to be considered. This number is limited by the number of data points that are available in the training dataset.

3.7. Direction dependency of IRG values

The strength of a relation as defined by the value of the improvement over the random prediction in the previous sections depends on the direction of the relation between variables.

A
1
2
2
1
2
1
2

B	
1	2
3	2
3	3
1	2
2	3
1	3
2	2

Figure 4. Example pair of data for variables A and B

In the Figure 4, the value of the variable A can be determined exactly given the indicator values of the row of indicators for B, however this does not hold true for the other direction. This is due to the fact that relation between B and A is many to one, and so given the value of A, the exact value of B cannot be determined.

3.8. Selection of structural model from IRG values

In order to come up with a model of relations between latent constructs, called variables for brevity, we firstly compute the IRG value for every pair of variables that describes the strength of the relation between both variables.

Obtained IRG values allow to come up with ordering of all possible binary relations starting from the strongest to the weakest. In order to come up with a structural model, n strongest relations are selected, where n is provided by the user.

3.9. Implementation details

The statistical learning tool was implemented in Python programming language, using Tensor Flow [32] python package to implement neural networks training, and sklearn [33] for the other classes of predictive models.[1]

4. Inductive confirmation

The starting point for our analysis is a paper published by Meseguer et al. that uses the Technology Acceptance Model (TAM) for perception of Wikipedia as a teaching resource [5] and its data set[2]. We have chosen a TAM study because it is one of the most established models in IS research derived by using a deductive mode of research [34]. It relates latent variables of Perceived Usefulness (PU), Perceived Ease of Use (PEU), and User Acceptance (UA) with one another. Davis found a strong relation between PU and UA and a weaker relation between PEU and UA but a stronger path from PEU to PU to UA. TAM has been replicated in many deductive studies. Meseguer et al. extended TAM by adding six constructs, i.e. quality, perceived enjoyment, image, sharing attitude, job relevance, and profile 2.0 [5]. In their study, they found support for 15 causal relationships between these 15 latent structural variables. Each structural variable is defined by two to four indicator variables. All proposed relationships were tested being significant by a SEM analysis with path coefficients ranging from 0.105 to 0.683.
Starting with the data set for the measurement model, we tested abovementioned statistical learning models without prior knowledge on relationships between latent variables, i.e. the structural model.

4.1. Hyperparameter selection

For every class of models considered in the previous section, we state the set of hyperparameters that is optimized, as well as a set of values from an interval used if a particular hyperparameter is unbound (cf. tables below). We verified that larger intervals of hyperparameter values do not improve the performance of predictive models.
While it is likely that for other practical problems the intervals presented in this work will also be sufficient, it

[1] Code repository http://goo.gl/r35rzv

[2] https://archive.ics.uci.edu/ml/datasets/wiki4HE

is generally advised to verify this by extending the intervals and observing whether this leads to large change in obtained results. For every class of models, a grid search was performed to determine the best configuration of hyperparameters. Comparison of different hyperparameter settings was done using the validation split of data as previously described.

Table 1. Hyperparameters of ANN and ranges over which grid search was performed.

Name	*Values tried*
Number of layers	{1, 2, … 5}
Num. of neurons in layer	{2, 4, 8, … 512}

Table 2. Hyperparameters of Kernel SVM and ranges over which grid search was performed

Name	*Values tried*
Complexity parameter	$\{2^{-10}, 2^{-9}, 2^{-8}, \dots, 2^{10}\}$
Gamma of RBF kernel	$\{2^{-10}, 2^{-8}, 2^{-6}, \dots, 2^{0}\}$
Epsilon	$\{2^{-10}, 2^{-8}, 2^{-6}, \dots, 2^{0}\}$

Table 3. Hyperparameters of Boosted models and ranges over which grid search was performed.

Name	*Values tried*
Num. of weak learners	{2, 4, 8, … 1024}
Learning rate	$\{2^{-10}, 2^{-8}, 2^{-6}, \dots, 2^{0}\}$

Table 4. Hyperparameters of kNN models and ranges over which grid search was performed. Here, N is the size of the training data set.

Name	*Range*
Num. of neighbors (k)	{1, 2, 3, … N}

4.2. Experimental results and discussion

Obtained results indicate that for the type of data considered different classes of learning models perform rather similarly. In particular, deviation between different values of IRG for different model classes is 0.02 on average (cf. Table 5). This is supported by the fact that every model class considered in this research is universally consistent (see previous section) and, thus, for arbitrary large dataset available should perform similarly. This suggests that in practice any proposed single class of universally consistent model is already sufficient for our proposed method. This is likely to not hold true when the amount of training data is much smaller than used in this work (300 training records) or if the number of items per latent structural variable is large, e.g., much larger than 5 as is in our experiments. [18].

4.3. Structural model selection

Directed binary relations between any structural variable A and B (A→B) of the model [5] with high IRG are selected that indicate a strong deviation from predictions based on random guesses. Higher IRG stand for better prediction of B given A (cf. Table 5). Derived from test data analysis, we determined an IRG of 1.40 as a lower threshold for relations between latent structural variables of interest, resulting in a matrix I with IRG values for all directed binary relations between structural variables.

Next a binary trigger map τ is applied that obtains a positive value if a relation is supported by previous research and a neutral value if unsupported (cf. Table 6). A positive value indicates a path that directly or indirectly connects two variables. An operator $\Psi(I, \tau)$ is defined that first deletes all weaker IRG values if relations between two variables are supported in both directions by an IRG value above an assumed threshold and second interchanges IRG values of both relations between A and B if $\tau(B, A)$ is positive but $\tau(A, B)$ is not. Thus higher IRG values are exchanged according to theoretically stronger directed relations.

Table 5. Average (first number in cell) and deviation of values of IRG across different model classes. Average deviation of IRG values is 0.02, which shows that selection of relations with statistical learning is stable to choice of particular model class among those considered. Bold: selected relation; italics: deselected relation; Grey cell: relation reversed due to τ.

	Use	Qu	PU	PEU	ENJ	BI	JR	Pf	IM	SA
Use		*1.41* 0.02	**1.73** 0.02	1.39 0.01	**1.49** 0.02	**1.95** 0.04	**1.48** 0.02	**1.5** 0.01	*1.44* 0.01	1.32 0.03
Qu	**1.53** 0.03		**1.55** 0.03	1.39 0.03	**1.52** 0.05	**1.62** 0.03	**1.44** 0.06	**1.44** 0.01	1.36 0.04	1.32 0.02
PU	*1.62* 0.02	*1.41* 0.03		1.38 0.03	**1.55** 0.04	**1.7** 0.05	**1.48** 0.02	**1.46** 0.01	**1.42** 0.02	1.32 0.02
PEU	**1.4** 0	1.36 0.02	**1.4** 0.01		**1.58** 0.06	**1.45** 0.03	**1.47** 0.04	**1.47** 0.02	1.38 0.01	1.3 0.01
ENJ	*1.43* 0.01	1.35 0.02	*1.48* 0.01	*1.45* 0.04		**1.48** 0.03	**1.48** 0.03	**1.43** 0.01	1.38 0.01	1.35 0.01
BI	*1.72* 0.01	1.39 0.02	*1.58* 0.01	1.39 0.01	*1.45* 0.02		1.37 0.05	**1.49** 0.01	*1.41* 0.02	1.32 0.01
JR	1.39 0.02	1.28 0.01	1.33 0.01	1.38 0	*1.41* 0.01	**1.43** 0.01		*1.46* 0.01	*1.43* 0.01	1.33 0.04
Pf	**1.41** 0.01	1.29 0.01	1.35 0.02	1.38 0	*1.42* 0.02	*1.46* 0.01	**1.52** 0.01		1.38 0.01	1.33 0.02
IM	**1.46** 0.02	1.31 0.01	*1.42* 0.01	1.38 0.01	**1.45** 0.01	**1.46** 0.04	**1.55** 0.02	*1.43* 0.01		*1.4* 0.03
SA	**1.42** 0.02	1.3 0.01	1.36 0.02	**1.4** 0.03	**1.47** 0.03	**1.43** 0.02	**1.46** 0.04	**1.49** 0.03	**1.44** 0.03	

For instance, Use → PU has an IRG value of 1.73 while PU → Use has an IRG value of 1.62. Thus, application of Ψ(I, τ) deletes the IRG value for PU → Use (1.62) and subsequently moves the IRG value of Use → PU (1.73) to become the IRG value of PU → Use as supported by previous research (cf. Table 6). No changes are made if τ(A, B) and τ(B, A) are neutral, i.e. previous research is neutral towards a particular relation between A and B. Thus Ψ recognizes known directions but is also susceptible for new relations. Application of Ψ(I, τ) results in a transformed IRG matrix I' from which a structural model is derived.

In our example, we simply derived τ directly from [5] while changes to τ might also be derived from a researcher's hypotheses. In our example, statistical learning methods determined 40 relations with an IRG above 1.40. In 19 cases, Ψ made a selection on relations (cf. Table 5) and 14 relations were reversed (cf. Table 5).

4.4. Evaluation of the extracted structural model

The output of the proposed data-driven approach for extracting structural relations by statistical learning methods is used as input for a covariance-based SEM. This is done in compliance with the original model [5] while other analytical methods for structural models are feasible.

Table 6. Trigger matrix τ derived from [5]

	Use	Qu	PU	PEU	ENJ	BI	JR	Pf	IM	SA
Use	0	0	0	0	0	0	0	0	0	0
Qu	1	0	1	0	1	1	0	0	0	0
PU	1	0	0	0	0	1	0	0	0	0
PEU	1	0	1	0	0	1	0	0	0	0
ENJ	1	0	1	1	0	1	0	0	0	0
BI	1	0	0	0	0	0	0	0	0	0
JR	1	1	1	1	1	1	1	1	1	1
Pf	1	0	1	0	0	1	0	0	0	0
IM	1	1	1	1	1	1	0	1	1	1
SA	1	1	1	1	1	1	0	1	0	1

Potential higher order concepts internally derived by statistical learning methods, e.g., created by multi-layer ANN, are decoupled from resulting structural models. This resembles how researchers deliberate on theoretical knowledge when they finally come up with a hypothetical, linear structural model. Therefore, we anticipate that some statistical learning models obtain the potential for using higher-order concepts internally but we only focus on results that are expressed by linear structural models without higher-order concept.

In out work, resulting relations were integrated into a SEM that was evaluated by SPSS Amos 24 with a test data set with 300 cases that has not been used for training and validating the extracted model. Due to the fact that the measurement was re-used from the original model, we discuss results for the structural variable model alone.

After deletion of non-significant relations, the model consists of 19 relations, i.e. three relations more than the original model on which SEM analysis was applied (cf. Table 7). Compared to the original model, all relations of the original model were found except the relation between Quality (QU) and Perceived enjoyment (ENJ). Instead we found a weakly significant relation between Profile 2.0 (PRF) and Perceived Ease-of-use (PEU).

Table 7. Regression weights (bold: relation compliant with [5]; italics: relation supported by a path in [5]; *: new relation; Est: estimates for regression weights, S.E.: standard error; C.R.: critical ratio, p: p-value).

			Est.	S.E.	C.R.	p
IMG	←	**JR**	.291	.066	4.382	***
SA	←	**JR**	.149	.053	2.806	.005
SA	←	**IMG**	.210	.069	3.037	.002
PRF	←	**SA**	.391	.092	4.258	***
PRF	←	*JR*	.115	.058	1.979	.048
PEU	←*	PRF	.087	.048	1.800	.072
QU	←	*JR*	.123	.045	2.734	.006
PEU	←	**ENJ**	.578	.091	6.324	***
PU	←	**QU**	.399	.060	6.696	***
PU	←	**IMG**	.303	.056	5.394	***
PU	←	**PEU**	.625	.122	5.135	***
BI	←	**PU**	.715	.087	8.260	***
BI	←	*PRF*	.179	.064	2.787	.005
BI	←	*QU*	.213	.073	2.922	.003
BI	←	*JR*	.328	.055	5.992	***
USE	←	**BI**	.689	.051	13.480	***

5. Summary and Outlook

We have presented a novel approach for extracting structural relationships by data-driven, statistical learning methods. Therefore we described a method consisting of seven steps: (1) splitting data set into training, evaluation, and test subsets, (2) determination of a random model, (3) training statistical learning models incl. adjustment of hyperparameters so that under- and overfitting is minimized, (4) extraction of relevant relationships by IRG values, (5) theoretical adjustment (i.e., application of Ψ(I, τ)), (6) evaluation of the resulting model, (7) using results for assessment of the original model (inductive confirmation) or

derivation of hypotheses that are afterwards tested by deductive studies (inductive exploration).

By application of our method to a data set previously used for evaluating a deductively developed structural model, a model was derived with 24 relations more than the original model. Previous knowledge represented by τ caused reversal of the direction of 19 relations that might be considered as being rather high. After deselecting non-significant relations by SEM analysis, the resulting model captured all relations of the original model except for one relation, while three new relations were found (complexity increase by 18%). Therefore, we conclude that our method provides strong support for the original model while recommending a re-assessment of the direction of causal relationships of 19 relations. Additionally, the newly proposed relation and one unsupported relation are recommended for re-examination as well.

A limitation is that the proposed method has been applied to one original model only. Studies for applying this method to a set of structural models including cross-evaluation is under investigation. Furthermore, a statistical method for comparing structural models is needed as stressed in [3]. Recent work presented some initial methods for comparing models based on statistically sound similarity measures [1].

Finally, the proposed hybrid mode of research needs further refinement, in particular, with respect to its epistemological underpinning. Several researchers have recently discussed how data analytics and big data changes research in various fields [15, 35, 36], such as information systems. Nonetheless, an epistemologically sound method is in its infancies. An extension of the proposed hybrid mode of research is required that guides researchers in their search for deriving model candidates (inductive exploration) from data but also data-driven evaluation of existing models (inductive confirmation).

6. References

[1] Lai, K., Green, S.B., Levy, R., Reichenberg, R.E., Xu, Y., Thompson, M.S., Yel, N., Eggum-Wilkens, N.D., Kunze, K.L., Iida, M.: Assessing Model Similarity in Structural Equation Modeling. Structural Equation Modeling: A Multidisciplinary Journal 1-16 (2016)

[2] Rodgers, J.L.: The epistemology of mathematical and statistical modeling: a quiet methodological revolution. American Psychologist 65, 1 (2010)

[3] Fife, D.A., Rodgers, J.L., Mendoza, J.L.: Model Conditioned Data Elasticity in Path Analysis: Assessing the "Confoundability" of Model/Data Characteristics. Multivariate behavioral research 49, 597-613 (2014)

[4] Kaplan, D.: Structural equation modeling : foundations and extensions. Sage Publications, Thousand Oaks, Calif. (2000)

[5] Meseguer Artola, A., Aibar Puentes, E., Lladós Masllorens, J., Minguillón Alfonso, J., Lerga Felip, M.: Factors that influence the teaching use of Wikipedia in Higher Education. (2014)

[6] Jöreskog, K.G., Sörbom, D.: LISREL 8: User's reference guide. Scientific Software International (1996)

[7] Hair, J.F., Ringle, C.M., Sarstedt, M.: PLS-SEM: Indeed a silver bullet. Journal of Marketing theory and Practice 19, 139-152 (2011)

[8] Bollen, K.A.: Structural equations with latent variables. John Wiley & Sons (2014)

[9] MacCallum, R.C., Wegener, D.T., Uchino, B.N., Fabrigar, L.R.: The problem of equivalent models in applications of covariance structure analysis. Psychological bulletin 114, 185 (1993)

[10] Anderson, C.: The end of theory. Wired magazine 16, (2008)

[11] Floridi, L.: Big data and their epistemological challenge. Philosophy & Technology 1-3 (2012)

[12] Kitchin, R.: Big Data, new epistemologies and paradigm shifts. Big Data & Society 1, 2053951714528481 (2014)

[13] Prensky, M.: H. sapiens digital: From digital immigrants and digital natives to digital wisdom. Innovate: journal of online education 5, 1 (2009)

[14] Bentley, R.A., O'Brien, M.J., Brock, W.A.: Mapping collective behavior in the big-data era. Behavioral and Brain Sciences 37, 63-76 (2014)

[15] Rai, A.: Synergies Between Big Data and Theory. MIS quarterly 40, iii-ix (2016)

[16] Shmueli, G., Koppius, O.R.: Predictive analytics in information systems research. MIS quarterly 35, 553-572 (2011)

[17] Bottou, L.: Stochastic learning. Advanced lectures on machine learning, pp. 146-168. Springer (2004)

[18] Vapnik, V.N., Vapnik, V.: Statistical learning theory. Wiley New York (1998)

[19] Devroye, L., Gyorfi, L., Krzyzak, A., Lugosi, G.: On the strong universal consistency of nearest neighbor regression function estimates. The Annals of Statistics 1371-1385 (1994)

[20] Lugosi, G., Vayatis, N.: On the Bayes-risk consistency of regularized boosting methods. Annals of Statistics 30-55 (2004)

[21] Steinwart, I.: Support vector machines are universally consistent. Journal of Complexity 18, 768-791 (2002)

[22] Hornik, K., Stinchcombe, M., White, H.: Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. Neural networks 3, 551-560 (1990)

[23] Wolpert, D.H.: The supervised learning no-free-lunch theorems. Soft Computing and Industry, pp. 25-42. Springer (2002)

[24] Zhang, X.-S.: Introduction to artificial neural network. Neural Networks in Optimization, pp. 83-93. Springer (2000)

[25] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp. 1097-1105. (Year)

[26] Hornik, K.: Approximation capabilities of multilayer feedforward networks. Neural networks 4, 251-257 (1991)

[27] Auer, P., Burgsteiner, H., Maass, W.: A learning rule for very simple universal approximators consisting of a single layer of perceptrons. Neural Networks 21, 786-795 (2008)
[28] Hearst, M.A., Dumais, S.T., Osman, E., Platt, J., Scholkopf, B.: Support vector machines. IEEE Intelligent Systems and their Applications 13, 18-28 (1998)
[29] Smola, A.J., Schölkopf, B.: A tutorial on support vector regression. Statistics and computing 14, 199-222 (2004)
[30] Park, J., Sandberg, I.W.: Universal approximation using radial-basis-function networks. Neural computation 3, 246-257 (1991)
[31] Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: Icml, pp. 148-156. (Year)
[32] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M.: Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467 (2016)
[33] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825-2830 (2011)
[34] Davis, F.D.: Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. MIS Ouarterly 13, 319-339 (1989)
[35] Abbasi, A., Sarker, S., Chiang, R.: Big data research in information systems: Toward an inclusive research agenda. Journal of the Association for Information Systems 17, 3 (2016)
[36] Agarwal, R., Dhar, V.: Editorial—Big data, data science, and analytics: The opportunity and challenge for IS research. Information Systems Research 25, 443-448 (2014)

Appendix

Table 8. IRG derived using ANN model class.

	Use	Qu	PU	PEU	ENJ	BI	JR	Pf	Im	SA
Use		1.4	1.76	1.38	1.5	2.02	1.47	1.49	1.44	1.34
Qu	1.49		1.51	1.39	1.56	1.63	1.31	1.42	1.29	1.31
PU	1.65	1.45		1.41	1.55	1.74	1.48	1.46	1.43	1.34
PEU	1.4	1.37	1.41		1.65	1.42	1.47	1.47	1.36	1.31
ENJ	1.44	1.35	1.49	1.49		1.43	1.46	1.42	1.38	1.35
BI	1.73	1.39	1.59	1.39	1.43		1.43	1.46	1.42	1.31
JR	1.4	1.29	1.34	1.38	1.42	1.43		1.47	1.42	1.36
Pf	1.41	1.28	1.36	1.37	1.45	1.47	1.52		1.38	1.35
Im	1.5	1.3	1.42	1.39	1.48	1.45	1.56	1.43		1.43
SA	1.42	1.29	1.37	1.42	1.48	1.4	1.49	1.52	1.41	

Table 9. IRG derived using SVR model class.

	Use	Qu	PU	PEU	ENJ	BI	JR	Pf	Im	SA
Use		1.43	1.72	1.38	1.51	1.96	1.45	1.49	1.43	1.27
Qu	1.53		1.53	1.33	1.49	1.55	1.51	1.45	1.33	1.34
PU	1.6	1.42		1.32	1.58	1.75	1.51	1.45	1.41	1.28
PEU	1.4	1.38	1.41		1.62	1.5	1.51	1.49	1.39	1.32
ENJ	1.44	1.38	1.47	1.38		1.47	1.45	1.41	1.4	1.36
BI	1.7	1.42	1.6	1.38	1.49		1.41	1.49	1.4	1.32
JR	1.41	1.3	1.31	1.38	1.4	1.41		1.47	1.45	1.26
Pf	1.42	1.31	1.36	1.38	1.4	1.46	1.51		1.37	1.34
Im	1.42	1.31	1.4	1.36	1.45	1.47	1.58	1.43		1.35
SA	1.41	1.3	1.31	1.36	1.47	1.44	1.49	1.45	1.47	

Table 10. IRG derived using AdaBoost model class.

	Use	Qu	PU	PEU	ENJ	BI	JR	Pf	Im	SA
Use		1.42	1.7	1.4	1.49	1.91	1.47	1.51	1.44	1.34
Qu	1.58		1.61	1.43	1.56	1.66	1.46	1.44	1.38	1.28
PU	1.63	1.43		1.38	1.6	1.6	1.44	1.44	1.38	1.31
PEU	1.4	1.35	1.38		1.58	1.43	1.39	1.44	1.36	1.28
ENJ	1.42	1.34	1.48	1.48		1.5	1.5	1.42	1.37	1.35
BI	1.73	1.39	1.58	1.38	1.42		1.32	1.49	1.39	1.34
JR	1.38	1.26	1.32	1.37	1.42	1.43		1.43	1.43	1.38
Pf	1.4	1.29	1.32	1.38	1.43	1.43	1.5		1.38	1.32
Im	1.45	1.31	1.43	1.37	1.45	1.4	1.52	1.41		1.43
SA	1.41	1.3	1.36	1.43	1.44	1.42	1.43	1.47	1.49	

Table 11. IRG derived using kNN model class.

	Use	Qu	PU	PEU	ENJ	BI	JR	Pf	Im	SA
Use		1.39	1.72	1.38	1.46	1.91	1.51	1.52	1.45	1.33
Qu	1.52		1.54	1.41	1.45	1.64	1.47	1.44	1.42	1.33
PU	1.59	1.36		1.4	1.47	1.72	1.47	1.47	1.45	1.34
PEU	1.4	1.32	1.39		1.47	1.45	1.52	1.48	1.39	1.3
ENJ	1.41	1.33	1.49	1.45		1.5	1.51	1.45	1.37	1.34
BI	1.73	1.36	1.56	1.39	1.45		1.33	1.5	1.43	1.3
JR	1.36	1.28	1.34	1.38	1.41	1.43		1.46	1.42	1.31
Pf	1.42	1.29	1.34	1.38	1.4	1.47	1.53		1.4	1.31
Im	1.46	1.3	1.43	1.39	1.43	1.53	1.54	1.46		1.37
SA	1.4	1.3	1.38	1.39	1.45	1.47	1.5	1.51	1.44	