Mining Consumer-Generated Product-Configuration Data

Tino Fuhrmann, Marvin Schweizer, Andreas Geyer-Schulz, and Peter Kurz

Abstract Many companies offer their customers the possibility to configure their products online and thus to tailor them to their needs and wants. Today, this capability is implemented via sophisticated product-configurators which in parallel record those consumer-generated configurations. However, while a huge volume of such configuration data is recorded, little research has been conducted in the field of mining such configuration data. This is at least in part due to the non-availability of open configuration data sets. For the workshop of the special interest group for data analysis of the German Classification Society in Karlsruhe on 20–21 November 2015, the German market research

Tino Fuhrmann
Information Services and Electronic Markets, Karlsruhe Institute of Technology (KIT)
Kaiserstraße 12, D-76131 Karlsruhe, Germany
✉ Tino.Fuhrmann@student.kit.edu

Marvin Schweizer
Information Services and Electronic Markets, Karlsruhe Institute of Technology (KIT)
Kaiserstraße 12, D-76131 Karlsruhe, Germany
✉ Marvin.Schweizer@student.kit.edu

Andreas Geyer-Schulz
Information Services and Electronic Markets, Karlsruhe Institute of Technology (KIT)
Kaiserstraße 12, D-76131 Karlsruhe, Germany
✉ Andreas.Geyer-Schulz@kit.edu

Peter Kurz
TNS Deutschland GmbH,
Landsberger Str. 284, D-80687 München, Germany
✉ Peter.Kurz@tns-infratest.com

ARCHIVES OF DATA SCIENCE (ONLINE FIRST) DOI 10.5445/IR/1000170978
KIT SCIENTIFIC PUBLISHING ISSN 2363-9881
Vol. 8, No. 1, 2024

company TNS Infratest has made available an anonymized configuration data set of a major car manufacturer. In this contribution a first report on mining this data set is given. We show that the development of data models which are specific for the analysis planned reduce the resource requirements several orders of magnitude, both for data storage and processing time. Furthermore, we use ordinal utility theory in iso-price segments of cars to operationalize the detection of interesting patterns: We detect irrational car configurations of consumers and the attribute sets for a rational choice. We show and discuss the use of such attribute combinations for sales communication in the purchase process of cars.

1 Introduction

We start this article with the following definition of a product configurator:

> A product configurator is a software-based expert system that supports the user in the creation of product specifications by restricting how predefined entities (physical or non-physical) and their properties (fixed or variable) may be combined. (Haug, 2007, p. 19)

An Internet-based end-consumer car configurator is the data source of a car configuration data set of a major car manufacturer made available by TNS Infratest, a German market research and marketing consulting company, for a workshop of the German Classification Society in Karlsruhe from November 20th to November 21st, 2015. The data set donated contains more than 900,000 configurations of more than 470,000 prospective customers and covers three consecutive days of a six month period in 2012. Since fall 2016, TNS Infratest belongs to the global Kantar group and changed its name to Kantar TNS.

Section 2 gives a short description of the development and use of product configurators in industry and of the idea of using product configurators as marketing research tools. In section 3 we discuss a generic car configuration process similar to the actual processes used by the car industry which could have been used for data collection.

The data set analyzed is introduced in section 4. A first analysis of the data set follows in the next three sections: Section 5 starts with the development of a storage-efficient physical data model for car configurations and describes the preprocessing phase implemented. In section 6 we analyze the iso-price segments in the dataset with an a-priori segmentation base consisting of the line, engine, and price attributes.

In section 7 we explain the concept of finding interesting patterns as the detection of formally irrational choice behavior defined as violation of axiomatic utility theory. Recommendations are defined as the difference between rational and formally irrational car configurations. We have deferred its axiomatic treatment to appendix A and the description of the algorithms for ordinal utility theory to appendix B. In section 8 we show how recommendations can be used for a variety of sales communication strategies during the closing phase of a car purchase. The paper ends with a short summary of our findings as well as an outlook on future research in section 9.

2 Product Configurators: The State-of-the-Art

The history of product configurators reaches back to the days of rule-based expert systems of the late 1970s. R1/XCON, a configurator for the VAX11-780 computer systems developed by John McDermott at Digital Equipment Corporation (DEC), is known for being the first expert system in daily use in industry (Barker et al (1989) and McDermott (1982)). The main motivation for the development of configurators like R1/XCON with expert systems technologies were twofold: First, the failure of applying procedural software development approaches to configuration problems and, second, the enormous potential of cost savings by delivering valid and completely configured systems: The overall net return of using R1/XCON for DEC was estimated at 40 million dollars a year (Felfernig et al, 2014, p. 9).

The generation of rule-based product configurators proved to be less flexible than expected. The if-then structure of rules assumes a direction of execution from the condition to the action part and this restricts the inference of consequences when arbitrary variables are bound: The separation between knowledge of configuration models and knowledge of configuration search is violated. Because of these maintenance problems of rule-based knowledge representations, all product configurators should be based on feature models as Sabin and Weigel (1998) point out. However, a recent study of industrial practices in variety modeling by Berger et al (2013) indicates that although feature models are used by almost three quarters of the survey participants, several other specifications are also used. E.g. about one third of the participants also use spreadsheets, key/value pairs, domain-specific languages, and UML-based representations. It is also remarkable, that almost 40 percent of the participants use home-grown domain-specific tools (Berger et al, 2013).

Benavides et al (2013, p. 165) introduce the following definition of a feature model configuration problem:

Definition 1. A *feature model configuration problem* is defined by the tuple (F,D,R) where $F = \{f_1,\ldots,f_n\}$ is a set of features, $D = \{dom(f_1), \ldots, dom(f_n)\}$ is the set of corresponding binary feature domains $dom(f) = \{\text{true (=1)}, \text{false (=0)}\}$, and the set of restrictions (or constraints) $R = UR \cup FR$. It consists of the set of user requirements $UR = \{r_1,\ldots,r_l\}$ (e.g. as the set of features configured or not configured by the user) and $FR = \{r_{l+1},\ldots,r_k\}$ a set of potentially complex feature model constraints (e.g. expressed in a feature constraint language) which can be translated into a constraint satisfaction problem.

Definition 2. A *feature model configuration* for a given (F,D,R) is a complete binding of variables $f \in F$. It is consistent, iff all contraints $r \in R$ are fulfilled by the given binding of variables.

The term *feature model configuration* (short: configuration) of a car has two possible interpretations: The individual configuration of a car selected by a consumer, but also a set of attributes common to identically configured cars. To separate these meanings we follow Soininen et al (1998, p. 360) and we refer to the first meaning as (individual) configuration and to the second as configuration type.

In the rest of this article, we adopt the following terminology: We denote the set of all possible consistent feature model configuration types of (F,D,R) as $\mathscr{C}$. A dataset of consumer configurations collected by a product configurator contains a set of configuration types $\mathbf{C} \subset \mathscr{C}$. $\mathbf{C}$ is represented as a $m \times n$ boolean matrix. The i-th row of $\mathbf{C}$ is denoted as $\mathbf{c}_i^T = \mathbf{C}_{i,\bullet}$, it represents a single configuration type as boolean feature vector of length n. The j-th column of $\mathbf{C}$ is denoted as $\mathbf{c}_j = \mathbf{C}_{\bullet,j}$. It represents the configuration of feature j of all configuration types of $\mathbf{C}$ as boolean feature vector of length m.

For recent surveys on the automated analysis of feature models and the state of the art in product configuration we refer the reader e.g. to the papers of Benavides et al (2005), Benavides et al (2010), Benavides et al (2013), and Zhang (2014). The extraction of a set of attributes common to a subset of configuration types as atomic sets (sets of attributes which can be treated as one attribute in a reduced model) used in section 9 is addressed repeatedly in the literature. See e.g. Zhang et al (2006), Zhang et al (2004), Mendonca et al (2008b), Mendonca et al (2008a), and Segura (2008).

Constraint-based problem solvers are model-based knowledge representation formalisms which can propagate consequences of assignments to other variables and for which the order of variables does not matter (Mackworth, 1977). The theoretical foundations for todays efficient constraint-based solvers are deeply rooted in algorithms for the solution of the propositional satisfiability problem (see Biere et al (2009)). Last, but not least, most configurators heavily need component-oriented knowledge representations as introduced e.g. by Mittal and Frayman (1989).

The migration from rule-based expert systems to constraint-based problem solvers with component-oriented knowledge representations opened the way for mainstream configuration environments integrated into enterprise resource planning environments at SAP, BAAN, and ORACLE in the early 1990s. For a history, see Felfernig et al (2014, pp. 11–13).

At the turn of the millenium, fierce competition in global industry markets led to the fall of mass production strategies and to the development of a system of mass customization at an industrial scale (Pine, 1999): The customer should get what he or she wants, when he or she wants it and at an attractive price. For the fulfillment of this strategic requirement, the end-consumer must be enabled to autonomously build his own product (BYO) – even, if the product is complex. To ease the burden of data entry by the consumer Jannach and Kalabis (2011) propose the dynamic selection of default values for configurator attributes based on patterns detected by association rules. In a recent literature survey Zhang (2014) discusses several approaches to integrate existing recommender technology into product configurators and their short-comings.

Product configurators turned into Internet-based mass customization tools seem to be the technological solution for this requirement and – as the data set provided by the German market research and marketing consulting company TNS Infratest shows – they are the source of a big data style source of consumer preference data. Rich Johnson, the founder of Sawtooth Inc. (the leading provider of conjoint analysis software), and his coworkers already published blueprints how product configurators and conjoint analysis can be married (Johnson et al (2006), and Rice and Bakken (2006)).

However, as far as we know, TNS Infratest's data set is the first large scale data set from an industrial production-use car configurator publicly available. We observe that the dataset provided by TNS Infratest contains a bag of consistent feature model configurations of consumers. Our knowledge of the set of constraints R is incomplete. We do not know the complete set of user requirements, because we do not know the default settings of the features and because of this which features have been actively selected by the users. We can not deduce the complete information on RF, because attributes and subsets of attributes which are not configured do not allow the inference that a certain attribute or subset of attributes can not be configured. In addition, although prices of all first car configurations configured by a consumer are available, the pricing functions for the car product lines are not disclosed.

3 Data Collection by End-Consumer Car Configurators

Modern end-consumer car configurators allow a customer to build his or her own car by following a multi-media supported navigation path through the configuration space of a model line. Car configurators check for the compatibility of options and allow only feasible configurations. All conflicts are resolved in the user dialogue.

In practice, car manufacturers have structured the configuration process for consumers in different ways: For example, BMW[1] groups its car configurator options into the following categories: Performance and efficiency; handling, ride and braking; exterior; interior seating and trim; instrumentation and controls; comfort and convenience; audio system; safety and security. In addition, different warranty options are offered. In contrast, Daimler for its Mercedes brand[2] structures the configuration process into 4 major phases, namely the configuration of the exterior, of the interior, of the entertainment and convenience options, and of the performance and safety options.

The actual car configurator used in the data collection process is unknown. However, in Fig. 1 we show a 7-step configuration process derived from the additional description of the data set for collecting all the attribute values available in the data set:

The configuration process starts with the choice of engine (engine power and type, fuel type) and line (e.g. sports line, luxury line) in steps 1 and 2 which determine the technical component system from which the car is built. In step

---

[1] http://www.bmwusa.com/Standard/Content/BYO/standardfeatures.aspx?NAModelCode=16X0 as of June 3rd, 2016

[2] http://www.mbusa.com/mercedes/vehicles/build/class-C/model-C63WS/buildId-2610975 as of June 3rd, 2016

3, the exterior colour of the car is selected by the customer. In most modern configurators, the pictures of the car displayed in the graphic configuration interface are adapted accordingly. In step 4, rims are chosen. Next, steps 5 and 6 determine the interior design: colour and material of upholsteries as well as trims used in the interior. Finally, in step 7 accessories and options provide additional possibilities for the customization of the car.

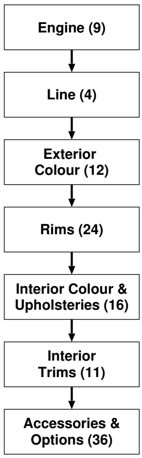


Fig. 1 The Configuration Process. (X) indicates the number of options in a step.

The end-consumer experiences configuration constraints only occasionally. If an option incompatible with the previous choices is selected, the consumer must either add additional options to make the option feasible or he must undo some of the previously selected options. Most car configurators offer recommendations for resolving such conflicts. However, the complete constraint and conflict resolution system implemented in the car configurator used for data collection is not disclosed with the data set. Nevertheless, the constraint system reduces the number of feasible configuration types considerably. Each stored configuration in the data set represents a navigation path through the car configurator and a point in configuration space.

4 The Car Configuration Data Set

The original dataset of the German market research and marketing consulting company TNS Infratest contains car configuration data collected from 473,819 prospective customers on three randomly selected days between January 2nd and July 1st, 2012 (the first 26 weeks of 2012). Each record consists of the identifier of the customer, the budget he or she is willing to spend, the price and the configuration for the first car and up to three additional car configurations with the same engine, but without price.

A car configuration type has 42 attributes which are mapped to 112 binary features. The two most important attributes – because of the a priori market segmentation of the car manufacturer – are engine (9 types) and model line

(4 lines). We call this group **segmentation attributes**. The engine attribute is described by a numerical identifier only for reasons of anonymity, the model line attribute by a short name (e.g. *sports line*) probably indicating a rough customer segment. The next group of 4 attributes (with partial information on extra charges (called add on price)) consists of colour (12 variants), trims (11 variants), cushions (16 variants), and rims (24 variants). For these six variables, only a single variant (one feature variable out of the group of feature variables representing the attribute) can be configured.

The second group of attributes (called accessories and options) consists of 36 binary attributes or options including packages (sport, comfort, light interior, storage), driving assistants (14 variants, e.g. parking assistant), security systems (alarm system), interior comfort (11 variants, e.g. seat heating), consumer electronics (5 variants, e.g. hifi-system), and trailer tow hitch. In this group of variables, one attribute maps to one feature. Combinations of features are possible. From experience with actual car configurators, we know that constraints between options exist. However, the actual constraint system of the product configurator which was used during data collection has not been disclosed. Therefore, in the worst case (the configurator has no constraints) the size of the configuration space is $9 \cdot 4 \cdot 12 \cdot 24 \cdot 16 \cdot 11 \cdot 2^{36} = 2^{47} \cdot 3^4 \cdot 11$ configuration types ($\approx 1.25 \cdot 10^{17}$).

5 Preprocessing and Improved Data Representation

An exploratory analysis of the data set revealed that the configuration space is only sparsely populated: Out of $1.25 \cdot 10^{17}$ possible configuration types (in the worst case, see section 4), a mere 943 configuration types exist in the data set of nearly a million configurations. Consequently, many consumers configured identical cars. We also found that the data set was denormalized.

Normalizing the data model considerably reduced the size of the data set, chiefly because it eliminates the large number of duplicate configurations. This normalization implicitly entails the switch from a table of configurations to a table of configuration types.

Fig. 2 illustrates the normalized data model. The information on the sequence of configurations of a respondent is stored in the table **ConfigurationSequence**, more specifically in the sequence number. The 42 attributes of each car configuration are transformed into 112 binary attributes. Note, that

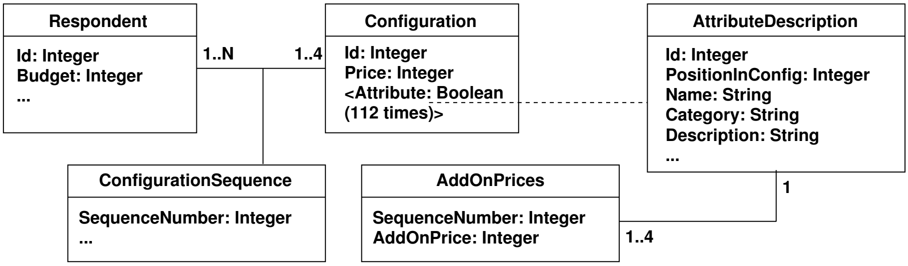


Fig. 2 Normalized Data Model as Class Diagram in the Unified Modeling Language (UML).

the table **AddOnPrices** contains the attribute prices used for the *n*-th configuration of a customer.

During the exploratory analysis, we additionally discovered that the car configurations of about 30,000 customers were broken. In most of these cases, an intact configuration was apparently split into two complementary parts. In the data set thus existed two invalid configurations for each of these customers, which had a valid attribute value exactly were its counterpart had a missing value. These complementary configurations were merged to recover the valid configuration. For a small number of cases, configurations contained missing values but no way of recovering the missing data was apparent. We discarded these incomplete configurations.

After preprocessing the data set contains a total of 962,799 configurations (469,112 first configurations and 493,687 subsequent configurations). The data set contains a total of 943 configuration types with known price. All 943 configuration types occur as a first configuration. 711 configuration types occur as a second configuration, 490 as a third and 262 as a fourth. The number of follow-up configurations drops sharply.

The configuration table in the new physical data model, which included most notably the transformation from car configurations to car configuration types, is reduced in size by the order of 10^3. At the same time, the physical data model still contains the same information as the original data set.

For practitioners, we emphasize that the development of improved, specialized data models – even if they are not generic – integrated into the data collection process lead to a reduction in storage of several orders of magnitude and additional savings in infrastructure, maintenance, and data analysis costs.

The life cycle of most major car models lasts between 8 and 10 years. The complete data set of TNS Infratest collected over a period of 6 months has a size of approximately 2TB with 60 million configurations. A rough estimate over the life cycle of this car model leads to a data volume of 40 TB with 240 million configurations at the end of the life cycle of the car models without transformation.

6 Exploratory Data Analysis: A Segmentation-Based Approach

We divide the set of 943 car configuration types in a total of 36 engine/line segments. The rationale for this decision comes from the prevalent corporate strategy of the automotive industry itself: In the 1920s, Alfred Sloan invented the new organizational structure of the multidivisional company for General Motors (Sloan, 1964). Today, this organizational structure can be considered the organizational blue-print for companies in the automotive industry.

In a multidivisional company, each division has complete autonomy to produce and sell a product (e.g. car) for its assigned market segment (Chandler, 1962). Market segmentation thus plays a strategic role for multidivisional companies. We can, therefore, expect car manufacturers to distinguish consumer segments by distinctive consumer profiles based on market research and to offer a product line for each segment. In this context, the *line* attribute in the data set seems to be designed as exactly such a segmentation base. We consequently use the attribute as a segmentation attribute.

In addition, we chose the engine type as the second segmentation attribute, because this attribute has the highest impact on the price of a car configuration of a specific line and on product performance (Fuhrmann et al, 2017, p. 67). It, therefore, reflects the usage pattern, intended by the consumer, to a large extent.

An even finer market segmentation is achieved with the price variable as further segmentation variable. Considering price as a segmentation variable is a common normative segmentation method that is based on the theory of consumer demand functions (Wedel and Kamakura, 2001, p. 26). Price reflects a mixture of a consumer's willingness to pay (Keeney and Raiffa, 1976, pp. 125–127) and his budget constraints mitigated by his financing options. However, the second motivation to look into iso-price segments is the question of formal consistency of the observed car configuration behavior to be discussed in section 7. An iso-price segment is defined as the set of all configuration types with the same price (within an engine/line segment). The fact that 175 out of the 225 iso-price segments in the sample consist of more than one configuration type, indicates that the segmentation base was not chosen too granular.

By providing an a-priori structure for the data set, the segmentation reduces the complexity and thus supports subsequent exploratory data analysis and the data analysis of the following sections.

Fig. 3 shows the structure of the iso-price market segments for engine 1 and the *sports line*: 26 configuration types in 7 iso-price segments with 1, 1, 3, 3, 10, 4, and 4 configuration types. Each horizontal line of identical symbols in the configuration type/price plot of Fig. 3 forms an iso-price segment.

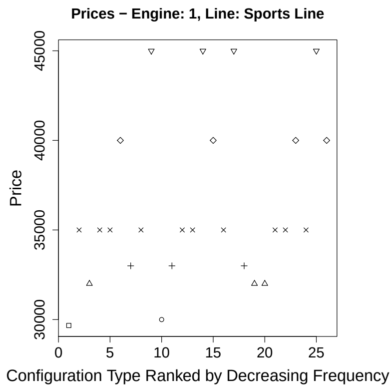


Fig. 3 Segment **Engine 1/Sports Line**: Iso-Price Lines of Car Configuration Types (Decreasing frequency). Symbols of data points show configuration types with the same price.

Table 1 shows the engine/model line segments covered by the car configurator. Each segment is characterized by two groups of attributes:

- The first attribute group measures the variety in the segment (and the computational complexity needed for further analysis). *No. of. Conf. Types* is the number of configuration types, whereas *(Max)* specifies the maximum number of configuration types with the same price, i.e. the maximum size of an iso-price line in this segment. *Price range* and *(Levels)* indicate the number of iso-price lines in the respective segment and their price range.
- The second group, consisting of *No. of Cars Configured*, *Price Average* and *Value* relates to the overall size and value of the segment.

Model Line	**No Line**	**Sports Line**	**Luxury Line**	**Modern Line**
Engine 1: No. of Conf. Types (Max)	50 (12)	26 (10)	4 (3)	24 (7)
No. of Cars Configured	53,773	28,908	7,156	27,982
Price Average (Euro)	33,870	35,269	33,604	35,864
Price Range (in 1,000 Euro) (Levels)	20 - 55 (10)	29 - 45 (7)	31 - 35 (2)	28 - 55 (9)
Value (in Mio Euro)	854	546	143	445
Engine 2: No. of Conf. Types (Max)	49 (9)	87 (23)	43 (11)	44 (9)
No. of Cars Configured	48,999	89,675	40,532	41,566
Price Average (Euro)	37,633	39,134	37,672	38,122
Price Range (in 1,000 Euro) (Levels)	28 - 47 (14)	35 - 45 (10)	28 - 45 (9)	34 - 45 (11)
Value (in Mio Euro)	770	1,885	781	714
Engine 3: No. of Conf. Types (Max)	15 (6)	61 (7)	9 (4)	9 (4)
No. of Cars Configured	13,479	60,977	9,252	10,910
Price Average (Euro)	42,556	47,426	46,452	45,154
Price Range (in 1,000 Euro) (Levels)	40 - 46 (4)	38 - 83 (13)	40 - 55 (3)	40 - 50 (5)
Value (in Mio Euro)	219	1443	232	238
Engine 4: No. of Conf. Types (Max)	- (-)	2 (2)	1 (1)	- (-)
No. of Cars Configured	-	1,865	3,639	-
Price Average (Euro)	-	45,000	45,500	-
Price Range (in 1,000 Euro) (Levels)	- (-)	45 - 45 (1)	45 - 45 (1)	- (-)
Value (in Mio Euro)	-	64	80	-
Engine 5: No. of Conf. Types (Max)	37 (16)	51 (11)	11 (5)	9 (6)
No. of Cars Configured	37,097	52,270	11,153	8,033
Price Average (Euro)	35,337	37,558	36,481	38,019
Price Range (in 1,000 Euro) (Levels)	28 - 40 (7)	34 - 45 (11)	34 - 40 (4)	32 - 40 (4)
Value (in Mio Euro)	597	1,000	215	135
Engine 6: No. of Conf. Types (Max)	56 (15)	99 (31)	24 (8)	39 (13)
No. of Cars Configured	55,179	101,272	24,991	37,783
Price Average (Euro)	38,716	40,687	42,093	39,525
Price Range (in 1,000 Euro) (Levels)	34 - 45 (11)	30 - 52 (14)	35 - 50 (9)	35 - 48 (9)
Value (in Mio Euro)	873	2,041	468	884
Engine 7: No. of Conf. Types (Max)	24 (10)	39 (11)	18 (16)	17 (9)
No. of Cars Configured	23,476	42,913	18,020	17,456
Price Average (Euro)	43,583	42,805	43,944	45,168
Price Range (in 1,000 Euro) (Levels)	39 - 50 (5)	38 - 50 (9)	40 - 45 (2)	40 - 50 (4)
Value (in Mio Euro)	464	867	362	460
Engine 8: No. of Conf. Types (Max)	12 (3)	34 (7)	2 (1)	12 (4)
No. of Cars Configured	10,510	34,578	1,721	11,054
Price Average (Euro)	50,192	52,582	54,371	63,145
Price Range (in 1,000 Euro) (Levels)	48 - 55 (6)	47 - 62 (11)	53 - 55 (2)	50 - 94 (5)
Value (in Mio Euro)	146	938	17	440
Engine 9: No. of Conf. Types (Max)	3 (2)	17 (5)	8 (4)	7 (4)
No. of Cars Configured	3,662	18,359	8,172	6,387
Price Average (Euro)	57,433	63,094	57,061	50,682
Price Range (in 1,000 Euro) (Levels)	55 - 65 (2)	47 - 96 (5)	50 - 65 (3)	45 - 96 (3)
Value (in Mio Euro)	58	632	202	158

Table 1 Engine/Model Line Segments and Their Price Structure. Prices and values truncated. (Max) is the maximal number of different configuration types with the same price. (Levels) is the number of iso-price segments in an engine-model line segment with a total of 225 iso-price segments.

Table 1 shows that for about 2/3 of the engine/model line segments less than 30 different configuration types have been configured over the whole price range. 50 iso-price segments consist of one configuration, 35 of two configurations. Just a single segment (*Sports Line* with engine 6) has an iso-price segment with more than 30 configuration types. Since the number of configuration types per segment is a major factor in the computational complexity needed for further analysis, this is generally good news for the subsequent analysis.

7 Detecting Patterns of Interest: Preferences and Deviations from Rationality

Configuration data reveals the economic choices of consumers (Ben-Akiva and Gershenfeld, 1998; Liechty et al, 2001). A rational consumer always chooses a non-dominated car configuration. Non-dominated means, that no car configuration exists in the set considered which – all other attributes equal – is strictly better in at least one attribute. The set considered is operationalized by the set of configurations in an iso-price segment.

$Only(A) := A \cap (\neg B)$ and means "set of features configured in A but not in B". Consider a segment with $\{A, B, C, D\}$, then $Only(A \cap B) := A \cap B \cap (\neg C) \cap (\neg D)$ serves as a short notation to express the set of common attributes of A and B which are not configured in the rest of the segment, namely C and D.

In an iso-price segment with two configuration types A and B, non-dominance means,

1. if we do not know the utility function of the consumer (ordinal case) that $Only(A)$ and $Only(B)$ are both not empty,
2. if we know the utility function of the consumer (cardinal case) that $U(Only(A)) = U(Only(B))$.

We call consumers who configure dominated car configurations formally irrational and this only means that they could have configured a car with more attributes at the same price. An axiomatic analysis of this situation and its interpretation is given in appendix A. For us, interesting patterns are dominated car configurations, because they identify consumers who could have done better.

In the rest of this section we illustrate the concepts of non-dominance (case 1) and dominance (cases 2 and 3) of three iso-price segments with 2 configuration types A and B.

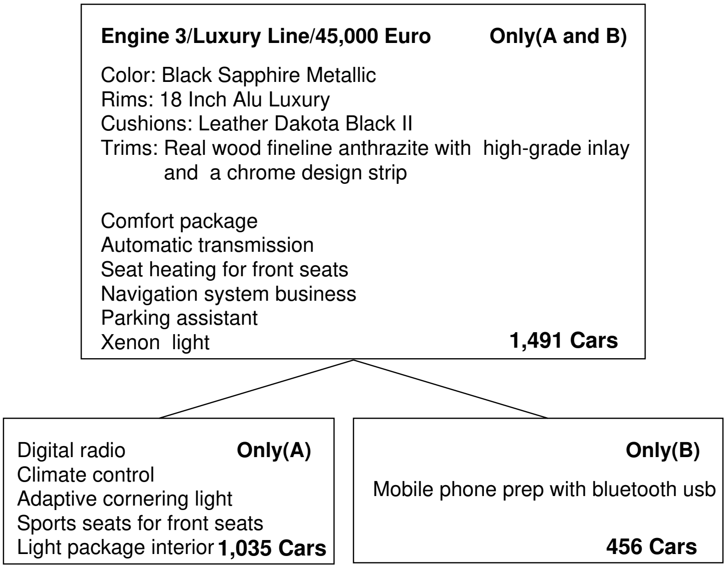


Fig. 4 Partially Ordered Attribute Sets for the Segment *Engine 3/Luxury Line* at a Price of 45,000 Euro

Case 1. For the *Engine 3/Luxury Line/ 45,000 Euro* segment (see Fig. 4) consumers choose either the attribute set $Only(A)$ with *digital radio*, *climate control*, *adaptive cornering light*, *sports seats for front seats*, and *light package interior* or the attribute set $Only(B)$ *mobile phone prep with bluetooth usb*. The assumption of rational consumers combined with unknown part-worths of attributes allows us to deduce that the attribute sets $Only(A)$ and $Only(B)$ are of equal value (indicated by the price of the configuration) to the consumers of this segment. The observed choice behavior in this segment does not violate the axioms of utility theory (see appendix A).

From the choice behavior of the consumers in this segment, we can construct an empirical conditional (on the segment) preference distribution: We infer that almost 70% of the consumers prefer $Only(A)$ to $Only(B)$:

$$P(U(Only(A)) \succ U(Only(B)) \mid A,B) = \frac{1,035}{1,491} = 0.69$$

where $P(\cdot)$ denotes the probability of an event and $U(\cdot)$ the utility function of the customers in the segment. This is information can be used for extracting probabilistic preferences between attribute sets conditioned on the segment.

Case 2. For the *Engine 4/Sports Line/ 45,000 Euro* segment (see Fig. 5) the set of attributes of configuration type B is a proper subset of the set of attributes of configuration type A. In this segment we observe a violation of the axioms of utility theory by the 426 consumers choosing configuration type B: They could have added a navigation system and climate control for the same price. The set difference $A - B$ is the set of attributes which can be offered to a formally irrational consumer at the same price. In addition, these attributes are potential recommendations in the sales process, because empirical evidence for their choice exists.

In 12 iso-price segments with two configuration types (more than one third) we observe violations of the axioms of utility theory in this way. Note, that to detect a violation, we do not need to know the part-worth utility function of the consumers.

Fig. 5 Partially Ordered Attribute Sets for the Segment *Engine 4/Sports Line* at a Price of 45,000 Euro

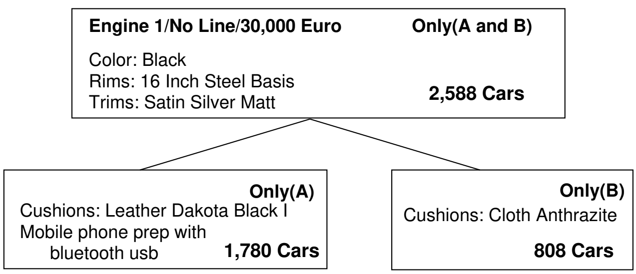


Fig. 6 Partially Ordered Attribute Sets for the Segment *Engine 1/No Line* at a Price of 30,000 Euro

For the iso-price segments contained in the engine/model line segments of table 1 a large number of deviations from rationality which violate the *no proper subset condition* for rationality of this case have been detected. We have relegated the description of the algorithms for this analysis to appendix B. A complete analysis of all 225 iso-price segments shows that 81,947 consumers (17.5%) of the 469,112 consumers in the dataset showed formally irrational behavior by choosing configuration types which violate the no proper subset condition. 50 iso-price segments consist of a single configuration type. 68 iso-price segments of the 175 segments with more than one configuration type do not contain irrational configuration types, when using ordinal utility theory only.

Case 3. For the *Engine 1/No Line/ 30,000 Euro* segment (see Fig. 6) our capability to detect a a violation of rationality depends on our knowledge of the part-worths of the attributes: We know that *leather cushions* add 1,750 Euro to the configuration price whereas *cloth anthrazite* is the standard configuration for cushions and, therefore, does not change the configuration price. In addition, the *mobile phone preparation* provides additional value to the customer, so configuration type A offers more value than configuration type B. Customers which prefer configuration type B violate the axioms of utility theory.

This case shows that knowledge of the part-worth function of the attributes configurations of a consumer segment allows the detection of additional violations of the axioms utility theory which can not be detected by making use of ordinal information only. However, in this article we neither estimate the cardinal utility (part-worth) functions of consumers nor attempt to reverse engineer the pricing function of the manufacturer embedded in the product configurator.

A first finding is that the car configurator allows the configuration of cars which do not maximize the utility of the consumers. To remedy the situation, we could recommend that the car configurator should be improved by a recommendation function which in the last step suggests improvements so that the configuration is not an inferior choice in the iso-price segment.

A second finding is that the detection of configuration choices deviating from rationality is an implementable and scalable operationalization of the extraction of patterns of interest (or of e.g. anomalies as in Fayyad et al (1996)) from car configuration data, a topic on which the literature on data mining is remarkable silent (Geyer-Schulz, 2016). Deviation from rationality is used in this article as a purely formal (and testable) concept without additional semantic interpretation. The strengths of this concept is that it is:

- Formally testable on the ordinal level (in this article) as well as on the cardinal level.
- A minimal model. Instead of finding a model for each consumer group, one single model, namely that of the rational consumer, is used.
- Directly operationally useful by extracting attribute sets as recommendations for sales communications as described in section 8.

To summarize, we use axiomatic utility theory as a purely descriptive theory which helps in the process of detecting deviations from rational behavior. Irrational behavior is formally defined as a violation of the axioms of utility theory. It is well known from behavioral economic and psychological experiments that homo economicus is a rather weak and unrealistic model for the explanation of the actual behavior of consumers (see e.g. Kahneman and Tversky (2009), Thaler (2009), and Koehler and Harvey (2004)). However, in the context of detecting interesting patterns in car configurations axiomatic utility theory is a strong theory for detecting anomalies in an economic way – we have to formalize only a single model – as long as we do not insist on finding causes or interpretations of formally irrational behavior.

8 Recommendations for Hard-Sell

How we proceed from the discovery of formally irrational car configurations in section 7 depends on the philosophy of marketing management used by the car manufacturer who owns the car configurator: Marketing or selling?

For a car manufacturer following the marketing concept (Kotler, 1980, pp. 31-33) which respects the time honoured concept of customer sovereignty (for the origins of this concept, see Schwarzkopf (2011)) the recommendation is obvious: We suggest an improvement of the product configurator which we consider as being defective, so that the customer is offered his/her configuration and, in addition, a non-dominated car (*best*) configuration which contains the attributes of his/her configuration for the same price.

For car manufacturers which want to establish a direct Internet sales channel, this additional offer can be offered as a service guarantee: The best car configuration available at a certain price is always offered in addition to the car configuration found by the consumer. This reduces the consumer's risk and uncertainty in using the product configurator and helps in establishing a reputation of fairness for the manufacturer.

Next, selling: Instead of interpreting the existence of formally irrational car configurations as evidence of a defect in the product configurator, we consider the existence of formally irrational car configurations as evidence of the implementation of the selling concept (Kotler, 1980, pp. 29-30) by the car manufacturer. For such a manufacturer, “goods are sold, not bought.” (Kotler, 1980, p. 29): Sales volume drives profitability because of economies of scale and scope. That the remuneration throughout the sales organisation (dealers and sales persons) heavily depends on sales volume is an important part of an integrated selling strategy.

Kotler claims that car dealers “often are prime practitioners of the selling concept” (Kotler, 1980, p. 30). By 2015 all major car manufacturers featured Internet-based car configurators of their main brands for end consumers (Blazek et al, 2016). However, a full integration of the configurators into the sales process is still missing. After configuring a car, the consumer is guided to a near-by dealer for arranging a test-drive with his/her preferred car and for conducting the bargaining process.

Rein (1972b) presents a vivid introduction to several sales communication strategies used in bargaining. He devotes a whole chapter to sales communication strategies used by car dealers (Rein, 1972a), most noteably hard-sell and up-sell strategies. A hard-sell sales communication strategy combines a logical appeal to a rational consumer’s self-interest with a psychological pressure aiming at a fast close of the contract. Psychological pressure and exploitation of information advantages as well as of behavioral biases (e.g. risk aversion) are often part of the strategies of sellers and have contributed to the bad reputation of these strategies. As one of the reviewers of a previous version of this paper has put it: “I do not think that research works should suggest what can be considered as short-sighted or even unethical behavior”. However, a deeper economic analysis of hard-sell reveals (1) that hard-sell can be a profitable strategy in a competitive environment, (2) that it is detrimental and annoying to all consumers including those consumers who avoid direct exposure to the practice, and, (3) that moderate hard-sell increases the well-fare of society as a whole, because it is not a zero-sum game (sellers gain more than consumers lose) (Chu et al, 1995). For the car industry, hard-sell leads to a reduction of configuration types produced and as a consequence to a considerable reduction of unit cost. Another often neglected effect is a faster dissemination of information on innovative add-ons which, when properly packaged, have a positive impact on consumer satisfaction.

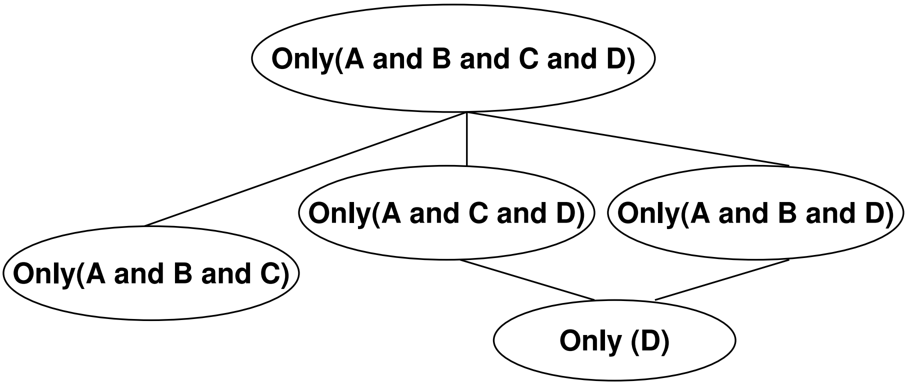
In the following, we restrict our efforts to the support of a hard-sell communication strategy of the dealer. We also think that the true reasons for formally irrational configurations are best uncovered in the sales encounter at the dealer. For example, preference of a cheap synthetic cloth for cushions over natural leather cushions can be explained e.g. by an allergy to leather or to a vegan life style (Fig. 6). Note that, support of the up-sell strategy can be done by choosing an engine/line segment spanning several price levels, and of cross-sell by segments spanning lines or engines or both.

Beard (2004) studied the first century of the debate between hard-sell *killers and soft-sell* poets in advertising. He emphasizes the fact that for over a century, hard-sell always has favored logical appeals to a rational consumer's self-interest. The most influential message strategists in the U.S. of the second half of the 20th century, namely Leo Burnett and John Caples, pragmatically combined rational arguments with persuasive, emotional, and subconscious argument which stirs the consumer into action. The importance of actually selling cars by sales communication campaigns for car dealers is best illustrated by the forced resignation of Nissan USA President Bob Thomas in 1997 because of a highly lauded advertising campaign which failed to move inventory (Vagnoni, 1997).

Violations of the *no proper subset* condition in iso-price segments of configurations can be exploited for generating data-based hard sell configurations for those consumers who selected a configuration which is a proper subset. One rationale for this is that other consumers of the same segment have configured additional valuable attributes of a car with the same price and that there is a quantified empirical support that these additional attributes are preferred by a certain number of consumers. Another rationale comes from a self-selection argument: Consumers who have configured a subset of attributes might also have the same preferences with regard to the attributes which have been configured by consumers who have configured a superset of attributes at the same price. If a car dealer knows the preference structure on such attribute sets, a set of hard sell strategies can be formulated by constructing time-limited offers either with the whole set of attributes as an add-on of a car with the same price or with subsets.

Fig. 7 illustrates this analysis for the segment *Engine 8/Sports Line* at a price of 47,000 Euro which contains a set of 4 configuration types $\{A,B,C,D\}$ configured by 1759 consumers. We detect that the 639 consumers choosing configuration types B and C show formally irrational behavior.

For these customers such offerings of a seller might look like this:



Only$(\mathbf{A} \cap \mathbf{B} \cap \mathbf{C} \cap \mathbf{D})$ **(1,759 Cars)**
Trims: Aluminum with Fine Longitudinal Grain with Black Accent Strip
Rims: 18 Inch Alu Sport III
Cushions: Fabric Imola Anthracite with Grey Contrasting Seam
light package interieur
variable sports steering
xenon light
performance leather steering wheel
seat heating for front seats
lumbar support for front seats
navigation system business
mobile phone preparation with bluetooth usb

Only$(\mathbf{A} \cap \mathbf{C} \cap \mathbf{D})$ **(1,339 Cars)**
climate control
arm rest for front seats
Only$(\mathbf{A} \cap \mathbf{B} \cap \mathbf{D})$ **(1,540 Cars)**
parking assistant
Only$(\mathbf{A} \cap \mathbf{B} \cap \mathbf{C})$ **(1,646 Cars)**
Colour: Black Sapphire Metallic
automatic transmission
glass sunroof
comfort access
Only(D) (113 Cars)
Colour: Black

Fig. 7 Partial Order of Irreducible Attribute Sets for the Segment *Engine 8/Sports Line* at a Price of 47,000 Euro. The frequency of configuration type A is 1007, of B 420, of C 219, and of *D* 113. A is a superset of B and of C.

- For consumers who configured a car of type *C*, because configuration type *A* is a superset of configuration type *C*: *If you sign the contract today, I can offer you the car you configured with climate control and arm rests for front seats for free …* (The dealer knows that 1,540 consumers have selected this option).
- For consumers who configured a car of type *B*, because configuration type *A* is a superset of configuration type *B*: *If you sign the contract today, I can offer you the car you configured with a parking assistant for free …* (The dealer knows that 1,339 consumers have selected this option).

A marketer takes out the psychological pressure:

- For consumers who configured a car of type C, because configuration type A is a superset of configuration type C: *Are you aware that you can add climate control and arm rests for front seats to your car configuration and still buy it at the same price* ... (1540 other consumers have preferred this configuration).
- For consumers who configured a car of type B, because configuration type A is a superset of configuration type B: *Are you aware that you can add a parking assistant to your car configuration and still buy it at the same price* ... (1339 other consumers have preferred this configuration).

Formally, if a customer has configured a car with a configuration type A which is a subset of an other configuration type B, the recommendation of additional attributes contains the set difference $A \setminus B = \cup_{R_i \in R} R_i$. R is the set of all irreducible attribute sets which contain attributes of A but not of B.

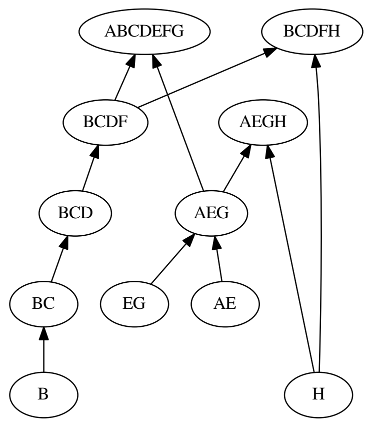
Subset $X \subset Y$	**Cars** $\lvert Y \rvert$	$Y \setminus X = \cup_{i \in R} I_i$	**Recommendations** $Y \setminus X$	for	$n = \lvert X \rvert$
$A \subset E$	223	EG	storage package	A	1,479
$C \subset B$	968	B	rear view camera	C	438
$D \subset C$	438	BC	hitch	D	232
$D \subset B$	968	$B \cup BC$	hitch, rear view camera	D	232
$F \subset D$	232	BCD	climate control	F	116
$F \subset C$	438	$BCD \cup BC$	climate control, hitch	F	116
$F \subset B$	968	$BCD \cup BC \cup B$	climate control, hitch, rear view camera	F	116
$G \subset E$	223	AE	comfort package	G	112

Table 2 Recommentations for the Segment *Engine 1/No Line* at a Price of 33,000 Euro. *BCD* is short for Only$(B \cap C \cap D)$.

Fig. 8 is an example of a larger iso-price segment with 8 configuration types. It illustrates how configuration types, namely A, C, D, F, and G which are subsets of other configuration types (5 of 8) can be exploited for generating recommendations for consumers. Fig. 8 depicts two different results: The set of irreducible attribute sets for this segment (left column) (see section 9) and the visualization of the partial order of irreducible attribute sets (top, right column) (see section 9).

The partial order induced by the subset relation for this segment which is shown in Fig. 9 has 3 components: the subsets A and G of E, the subset chain $F - D - C - B$ with B as superset and the singleton H.

Irrecudible Attribute Sets (I_i)
Only($\mathbf{A \cap B \cap C \cap D \cap E \cap F \cap G}$) **(3,568 Cars)**
Cushions: Fabric Anthracite
Only($\mathbf{B \cap C \cap D \cap F \cap H}$) **(1,858 Cars)**
Color: Alpine White
parking assistant
digital radio
Only($\mathbf{B \cap C \cap D \cap F}$) **(1,754 Cars)**
Rims: 16 Inch Alu Basis I
Trim: Fine-Wood Burr Walnut with Black Accent Strip
seat heating for front seats
alarm system
Only($\mathbf{A \cap E \cap G \cap H}$) **(1,918 Cars)**
Rims: 16 Inch Steel Basis
Trims: Matt Satin Silver
adaptive cornering light
Only($\mathbf{A \cap E \cap G}$) **(1,814 Cars)**
Color: Glacier Silver Metallic
xenon light
glass sunsroof
Only($\mathbf{B \cap C \cap D}$) **(1,638 Cars)**
climate control
Only($\mathbf{B \cap C}$) **(1,406 Cars)**
hitch
Only($\mathbf{E \cap G}$) **(335 Cars)**
storage package
Only($\mathbf{A \cap E}$) **(1,702 Cars)**
comfort package



Only($\mathbf{B}$) **(968 Cars)**
rear view camera
Only($\mathbf{H}$) **(104 Cars)**
Cushions: Leather Dakota Black I
adaptive chassis with lowering
sport leather steering wheel
sport seats for front seats

Fig. 8 Irreducible Attribute Sets and Their Partial Order for the Segment *Engine 1/No Line* at a Price of 33,000 Euro with 3,672 cars. *BCD* is short for Only$(B \cap C \cap D)$. Cars: $A = 1,479$, $B = 968$, $C = 438$, $D = 232$, $E = 223$, $F = 116$, $G = 112$, $H = 104$.

The recommendations derived from the subset relation of the partial order of configuration types (Fig. 9) and computed from the irreducible subsets (Fig. 8) are shown in table 2. This table lists recommended attributes for the subset configurations A, C, D, F, and G. Note, that for consumers of a car of configuration type F at least three sets of recommendations are detected which are potentially useful in subsequent sales communications. Again, there is a difference, how this information is used by a marketer and a seller:

- The marketer makes an unconditional offer of upgrading the user's car configuration with climate control, hitch, and rear view camera.

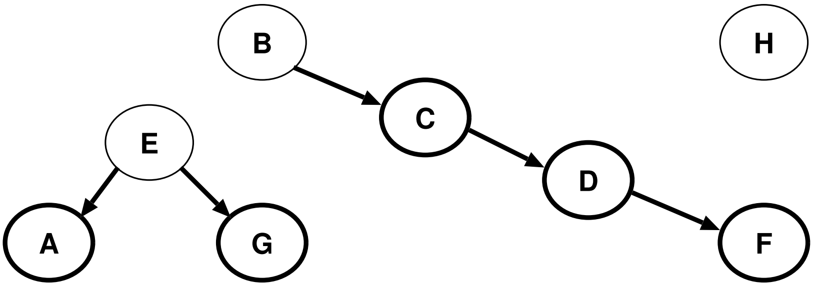


Fig. 9 Partial Order induced by Subset Relation of Configurations for the Segment *Engine 1/No Line* at a Price of 33,000 Euro with 3,672 cars. Subsets in bold, arrows point to subsets.

- The seller judges the psychological profile of the consumer and decides either on a frontal attack by offering the upgrade free, but only if you sign today. Or he can choose a protracted haggling in which he adds one add-on after the other until the set of recommendations is exhausted.

The last column of table 2. indicates for how many consumers a recommendation has been found.

In this iso-price segment (with a total of 3,672 consumers) recommendations for 2,377 consumers (64.7%) have been found.

Last but not least this segment contains several asymmetrically dominated alternatives (A, G, C, D, and F, bold in Fig. 9). An asymmetrically dominated alternative is both dominated by at least one alternative and not dominated by at least one alternative of the choice set. Huber et al (1982) provide experimental evidence that the presence of an asymmetrically dominated alternative in a choice set increases the probabiblity of choosing the alternative that dominates it. The asymmetrically dominated alternative, although seldom chosen, acts as a decoy which attracts sales to the dominating alternative.

A complete analysis of all 225 iso-price segments shows that for 81,947 consumers (17.5%) of the 469,112 consumers in the dataset one or more recommendations have been found. 50 iso-price segments consist of a single configuration type. 68 iso-price segments of the 175 segments with more than one configuration type do not contain irrational configuration types.

For the consumers in these iso-price segments, no recommendations can be given by the approach proposed in this section. However, because the dataset made available by the German market reseach and marketing consulting company TNS Infratest is a sample of only 2% of the complete dataset, the coverage of 17.5 % of consumers is a pessimistic lower bound.

9 Conclusion

In this article we have reported on an exploratory data-driven analysis of a large car configuration data set provided by TNS Infratest. The main contributions of this article are

1. Two information preserving transformations of the original data set: First, from individual car configurations to weighted configuration types, and second from car configuration types to irreducible attribute sets for sets of car configuration types (segments). The first transformation reduces the size of the data set by at least three orders of magnitude. This allows the development of fast analysis and visualization algorithms for irreducible attribute sets whose complexity – for naive implementations – is of the order of $O(n^2)$ subset tests with n the number configuration types in a segment. The second transformation to irreducible attribute sets reduces the cost of a subset test by one order of magnitude. The number of elements of irreducible attribute data sets is bounded by the number of attributes and independent of the number of configuration types.
2. We emphasize the role of background knowledge in identifiying meaningful contexts for further analysis: The segmentation base of line, engine, and price (although almost obvious after ist detecttion) provides the context for an efficient extraction of sales recommendations and for the visualization of meaningful car configuration lattices.
   Several data mining and machine learning methods (association rule mining and gradient boosted neural networks) could not automatically extract useful information.
3. By applying utility theory to iso-price segments we have formalized a way of automatically identifying interesting patterns in sets of configuration types: Sets of configuration types with the same price form iso-price segments. From such segments, we show how partial group preferences can be extracted and we analyze two types of deviation from rationality: The first is the identification of configurations which are proper subsets of other configurations, the second is the identification of configurations of lower value

in iso-price segments by exploiting known markup prices for attributes. In this paper we have implemented the first approach.

4. Last, but not least, we exploit the identification of configurations which are proper subsets of other configurations in iso-price segments for the extraction of recommendations a dealer could use e.g. for hard sell offers in the closing phase of the sales process.

Automatic detection of lower value configurations in iso-price segments as well as systematic preference extraction between attribute sets is left for further research. It requires the proper estimation of part-worth utility functions for homogeneous consumer segments. A second challenge is reengineering the pricing function of the product-configurator. The decomposition of the configuration types of an iso-price segment might help with the model selection problem for bundle pricing. The implementation of recommendation support for cross- and up-selling is an other topic for future extension.

Appendix A. Utility Theory and Car Configurations

Preferences, Weak Order, Utility, Rationality ...

Let X a finite set of choices, e.g. 2 cars.

A binary relation $R \subseteq X \times X$ is (see Fishburn (1970, p. 10))

1. *reflexive*, if $xRx, \forall x \in X$;
2. *symmetric*, if $xRy \Rightarrow yRx, \forall x,y \in X$;
3. *asymmetric*, if $xRy \Rightarrow \neg(yRx), \forall x,y \in X$;
4. *transitive*, if $xRy, yRz \Rightarrow xRz, \forall x,y,z \in X$; and
5. *negatively transitive*, if $\neg(xRy), \neg(yRz) \Rightarrow \neg(xRz), \forall x,y,z \in X$.

A weak order $(X, \prec)$ is asymmetric and negatively transitive.

$x \prec y$ means *x is less preferred than y*, e.g. the consumer has configured car x and not car y.

$x \sim y \Leftrightarrow \neg(x \prec y) \wedge \neg(y \prec x)$.

If $(X, \prec)$ is a weak order, then

1. (i) exactly one of $x \prec y$, $y \prec x$, $x \sim y$ holds, $\forall x,y \in X$,
2. (ii) $\prec$ is transitive, and
3. (iii) $(X, \sim)$ is an equivalence relation (reflexive, symmetric, transitive).

For a proof, see Fishburn (1970, p. 13).

If $(X, \prec)$ is a weak order and $X/\sim$ is countable, then there exists a utility function $U : X \to \mathbb{R}$, such that $x \prec y \Leftrightarrow U(x) < U(y)$. For a proof, see (Fishburn, 1970, p. 13).

An ordinal utility function U is unique up to monotone transformations. Utility differences can not be compared. A cardinal utility function U is unique up to linear transformations. Utility differences can be compared.

Given X, a rational consumer always chooses the best x, this means $\nexists y \in X : x \prec y$, respectively $\arg\max_{x \in X} U(x)$.

Sets of Attributes: Preferences, Weak Order, Utility, . . .

The configuration space of cars with n binary attributes is $\mathscr{A} = \{0,1\}^n$. We denote a single car configuration $\mathbf{a} \in \mathscr{A}$ by $\mathbf{a} = (a_i)$ with $a_i \in \{0,1\}$.

$(\mathscr{A}, <)$ is a partial order with $<$ defined as follows: Let $\mathbf{a}, \mathbf{b} \in \mathscr{A}$. $\mathbf{a} < \mathbf{b}$ ($\mathbf{a}$ is dominated by $\mathbf{b}$), if for all a_i, b_i $b_i \geq a_i$, and $b_i > a_i$ for some a_i, b_i.

A car configuration $\mathbf{a}$ is efficient, if it is not dominated. This means $\nexists \mathbf{b} \in \mathscr{A} : \mathbf{a} < \mathbf{b}$.

The attributes in A are *mutually preferentially independent*, if every subset of these attributes is preferentially independent of its complementary set of attributes. This simplifies to: If every pair of attributes is preferentially independent of its complementary set, then the attributes are mutually preferentially independent. For a proof, see Gorman (1968)).

For a complete and transitive weak order $(\mathscr{A}, <)$ a linear part-worth utility function $P : \mathscr{A} \to \mathbb{R}$ exists. For a proof see Debreu (1960).

Rational consumers

A rational consumer always prefers more money to less money. His preferences form a weak order $(X, \prec)$. Therefore, his valuation function for cars and money $U : X \to \mathbb{R}$ is a utility function. In addition, a rational consumer also has a valuation function for car configuration functions $P : \mathscr{A} \to \mathbb{R}$ which is also a utility function. Both functions are isomorphic under linear transformations.

Product configurators for a product line are closed world models, this means measuring the part-worth utility function $P(A)$ of rational consumers is possible only relative to a default configuration of a car. Let C the set of all binary attributes configurable in a product line. Let $D \subset C$ the set of binary attributes of the default configuration. Let $M \subset C$ the set of binary attributes not in the default configuration.

The simplest rational relative pricing model for a product line is a linear pricing function $P(A)$:

$$P(A) = \underbrace{\beta_0}_{\substack{\text{Default Configuration} \\ D = \{a_i \mid a_i \notin M\}}} + \underbrace{\sum_{a_j \in M} \beta_j \cdot a_j}_{\text{Contribution of change to } a_j} \tag{1}$$

We set up the price equations for $|C|$ configurations: $A = (C_1, \ldots, C_{|C|})^T$ and $\mathbf{b}$ the vector of configuration prices of the car producer. The general solution of the linear least squares problem (GLS) $A\beta \approx \mathbf{b}$ is

$$\beta = A^+\mathbf{b} + (I - A^+A)\mathbf{w}, \quad \mathbf{w} \quad \text{arbitrary.} \tag{2}$$

with A^+ the Moore-Penrose inverse of A. We compute the coefficients of the GLS solution and get a family of infinitely many isomorphic pricing/utility functions:

$$\beta_{GLS} = \beta_{GLS|\mathbf{w}=0} + \underbrace{\mathcal{N}(A) \cdot \mathbf{w}}_{\text{Affine linear Transformation}}$$

For the variables in D to be 0 in order to get an OLS solution, the following constraint system must hold:

$$\mathbf{0} = \beta_{GLS,D} + \mathcal{N}(A)_D \cdot \mathbf{w}_D \tag{3}$$

To compute the weights $\mathbf{w}_D$ for which the constraint system (3) holds, we solve the following linear equation for the default configuration D:

$$\mathbf{w}_D = \mathcal{N}(A)_D^{-1} \cdot (-1) \cdot \beta_{GLS,D} \tag{4}$$

And we get $\beta_{OLS} = \beta_{GLS} + \mathcal{N}(A)\mathbf{w}_D$. By an appropriate choice of D, we can always find at least one default configuration for which $\beta_{OLS} \geq 0$ holds. If $\beta_{OLS,M} > 0$ holds, then $R = M$ with R the set of all attributes relevant for price/value increases. Else, the set of attributes E for which $\beta_{OLS,M} = 0$ is irrelevant for price changes, and $R = M \setminus E$.

As consequence of this relative pricing model a linear part-worth utility function with $\beta_{OLS,R} > 0$ exists for a rational consumer. This implies that a configured attribute ($a_i = 1$, $a_i \in R$) is always preferable to an unconfigured attribute ($a_i = 0$), because $\beta_i > 0$.

The pricing function of the producer

In practice, the producer's price configurator $PC(A)$ is not necessarily a utility function. E.g., bundle pricing functions which are often used in the car industry violate the axioms of utility theory given above.

Let $\mathscr{P}(A)$ denote the set of all subsets of the attributes of a configuration and let $B \in \mathscr{P}(A)$ denote a specific subset of attributes (a bundle). The function $I(A,B)$ is a boolean function which is true if at least one attribute of B is configured in A.

$$I(A,B) = \begin{cases} 1 & \text{if} \quad \exists a_k \in A : a_k \in B \\ 0 & \quad\quad \text{else} \end{cases} \tag{5}$$

$$PC(A) = \beta_0 + \sum_{B \in \mathscr{P}(A)} \beta_B \cdot I(A,B) \tag{6}$$

The economic rational for such bundle pricing functions is either based on common infrastructure necessary for a set attributes in a bundle (e.g. all Internet services of the car (e.g. digital radio, media streaming, navigation) require a common bus system) or on the aim of increasing economies of scale and scope by a reduction of the number of configuration types.

The analysis of exchange in an iso-price segment

An iso-price segment contains a set of configurations S at a price p. For example, $S = \{A,B\}$ with A and B two car configurations. A consumer buys a car, only if $p \preceq A$ or $p \preceq B$. The producer set the prices of A and B to $p = PC(A)$ and $p = PC(B)$.

For the consumer, $\neg(A \prec p)$ or $\neg(B \prec p)$ holds. The producer is indifferent which car he sells: $A \sim B$, because $p = PC(A)$ and $p = PC(B)$ (transitivity).

Next, let us consider the level of car configurations $A = \mathbf{a}$ and $B = \mathbf{b}$ for the set of attributes R.

Theorem 1. *$b \subset a$ violates the axiom for preferences that exactly one of $x \prec y$, $y \prec x$, $x \sim y$ holds, $\forall x, y \in X$.*

Proof. Assume that $b \subset a$. Then $a = (a \cap b) \cup (a \setminus b)$ and $b = (a \cap b) \cup \emptyset$.

Let $\beta = \beta_{OLS,R}$. Next, compute: $P(A) = \sum_{i \in (a \cap b)} \beta_i a_i + \sum_{j \in (a \setminus b)} \beta_j a_j$ and $P(B) = \sum_{i \in (a \cap b)} \beta_i a_i + 0$. Because for the first sum $P(A) = P(B)$ holds, and for the second sum of $P(A)$ is greater than 0, $P(A) > P(B)$. This implies $B \prec A$. However, this contradicts $B \sim A$.

The existence of dominated car configurations in an iso-price segment implies that the pricing function PC of the producer is inconsistent and violates the axioms of utility theory. If $E = \emptyset$, this holds independent of the concrete form of $P(A)$, we do not have to estimate P.

However, this does not mean that using PC is irrational behavior of the producer, because this depends on the unknown profit function of the producer.

Even a consumer who has chosen the dominated car configuration has not shown irrational behavior, because he may not have realized that he could configure a better configuration. The information that the consumer perceives during the configuration process is not revealed by the dataset and not described. And, last but not least, the actual purchase decisions of consumers are not contained in the dataset. Therefore, we call dominated car configurations *formally irrational* car configuration of the iso-price segment S and we label consumers who configured dominated car configurations as *formally irrational.*

Dominated car configurations in an iso-price segment ($b \subset a$) operationalize the concept of finding interesting patterns and $a \setminus b$ is the set of attributes which should be recommended to formally irrational consumers.

Appendix B. The Lattice of Irreducible Atomic Sets and Its Visualization

The generalization of the detection of configuration types with proper subsets of attributes in a set of configuration types of a market segment (case 2 of section 7) represented as a $s \times n$ matrix $\mathbf{S}$ which contains a subset of s rows of $\mathbf{C}$ is straightforward: A naive operationalization of this idea requires pairwise subset tests of all configuration types in in $\mathbf{S}$ as shown in Algorithm 1. This implies s^2 subset tests. However, this has essentially two drawbacks: First, for larger sets of configuration types, it is costly. Second, it does not reveal the structure of common attribute sets in the segment **S**.

Algorithm 1 Algorithm for finding subsets in sets of configuration types

Require: **S** a boolean matrix where each row represents a configuration type

function FINDSUBSETS(**S**)
  $s \leftarrow nrow(\mathbf{S})$ ▷ number of rows of **S**
  $\vec{sub} \leftarrow replicate(0, s)$ ▷ generate a vector of zeros of length s
  for $i \in 1, \ldots, s$ **do**
    for $h \in 1, \ldots, s$ **do**
      if $\mathbf{S}[i,] \subset \mathbf{S}[h,]$ **then**
        $\vec{sub}[i] \leftarrow 1$
  return $\vec{sub}$

Whenever a subset of configuration types in a market segment **S** has a set of common features J, then all columns $\mathbf{s}_j, j \in J$ are equal. To improve the computational efficiency, we transform the boolean matrix **S** with duplicate columns into a boolean matrix $\mathbf{S}_{IAS}$ without duplicate columns and we build a vector of index sets IAS with $ncol(\mathbf{S}_{IAS})$ sets as elements. The g-th set of IAS contains the indices of the feature columns in **S** which are duplicates of the g-th column of $\mathbf{S}_{IAS}$. Algorithm 2 converts a set of configuration types **S** into its irreducible atomic set configuration $(\mathbf{S}_{IAS}, IAS)$. It returns $\mathbf{S}_{IAS}$ and the vector IAS of index sets.

Example (Fig. 7): The representation of **S** (with 19 of 112 features configured) in terms of irreducible attribute sets is given by the labelled boolean matrix $\mathbf{S}_{IAS}$ and by the 5 attribute tables shown in Fig. 7.

$$\mathbf{S}_{IAS} = \begin{array}{c} \begin{array}{ccccc} \{A,B,C,D\} & \{A,B,C\} & \{A,C,D\} & \{A,B,D\} & \{D\} \end{array} \\ \left(\begin{array}{ccccc} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \end{array}\right) \end{array} \begin{array}{c} \\ (A) \\ (B) \\ (C) \\ (D) \end{array} \tag{7}$$

With regard to the subset-relation, the matrix $\mathbf{S}_{IAS}$ allows the extraction of the partial order of configuration types and of the partial order of irreducible subsets. The idea of representing the lattice of car configuration types as a lattice of irreducible subsets of car configuration attributes comes from Gusfield's irreducible partially ordered representation of the set of stable matchings in

Algorithm 2 Algorithm for computing irreducible atomic set representation of a set of configuration types

Require: $\mathbf{S}$ a $s \times n$ boolean matrix

```
function COMPUTEIARS(S)
    s ← nrow(S)
    n ← ncol(S)
    S_IAS ← matrix(s, 0)                          ▷ Reduced matrix
    IAS→ ← ∅                                      ▷ Vector of attribute index sets
    for j ∈ 1, ..., n do
        if S[, j] ∉ columns(S_IAS) then
            S_IAS ← cbind(S_IAS, S[, j])
            IAS→ ← IAS→ ∪ {j}
        else
            h ← columnIndex(S_IAS, S[, j])
            IAS→[h] ← IAS→[h] ∪ j
    return S_IAS, IAS→
```

the basic stable marriage problem (Gusfield and Irving, 1989, Chapter 2, pp. 67-102). Note that the representation of a set of car configurations as a partial order of irreducible attribute sets is a concrete formalization of the idea of atomic sets as informally introduced by Zhang et al (2006), Zhang et al (2004), Mendonca et al (2008b), Mendonca et al (2008a), and Segura (2008).

Algorithm 3 Algorithm for extracting edge list of partial order and set difference from a set of configuration types represented as irreducible atomic sets

Require: $\mathbf{S} \leftarrow \mathbf{S}_{IAS}$ for partial order of configuration types or
Require: $\mathbf{S} \leftarrow \mathbf{S}_{IAS}^{T}$ for partial order of irreducible atomic sets

```
function EDGELIST(S)
    s ← nrow(S)
    EdgeLst→ ← ∅
    SetDiffLst→ ← ∅
    for i ∈ 1, ..., s do
        for h ∈ 1, ..., s do
            if S[i, ] ⊂ S[h, ] then
                EdgeLst→ ← EdgeLst→ ∪ (i, h)
                SetDiffLst→ ← SetDiffLst→ ∪ {setDifference(S[h, ], S[i, ])}
    return EdgeLst→, SetDiffLst→
```

We adapt the naive algorithm 1 to extract the edge list of the two partial orders and the set difference for each successful subset test. We use algorithm 3 to extract the partial order of

1. configuration types and the set differences (e.g. table 2) : $\textsc{edgeList}(\mathbf{S}_{IAS})$. The nodes of the graph are labelled by the configuration types $A, B, \cdots \in \mathbf{S}$. Note, that, if the edge list is empty, the graph is disconnected.
2. irreducible atomic sets of attributes (e.g. Fig. 7): $\textsc{edgeList}(\mathbf{S}^T_{IAS})$
   The nodes of the graph are labelled with the names of the irreducible attribute sets. The edge list contains arcs for all subset relations between atomic sets. In the graphs shown in this paper, the edge list is pruned, so that only a minimal graph is shown.

Note, that we draw only the partial order between the irreducible attribute sets and that we represent each irreducible attribute set as a table of attributes common to all configuration types in the irreducible atomic set.

The algorithms described above are implemented in Python. However, in contrast to the pseudo-codes presented, the dataset and the results of the analysis are stored in a PostgreSQL data base with the help of the `psycopg2` Python-PostgreSQL adapter. For the visualization of graphs, the Python package `graphviz` is used.

References

Barker VE, O'Connor DE, Bachant J, Soloway E (1989) Expert systems for configuration at Digital: XCON and beyond. Communications of the ACM 32(3):298 – 318, DOI 10.1145/62065.62067

Beard FK (2004) Hard-sell "killers" and soft-sell "poets": Modern advertising's enduring message strategy debate. Journalism History 30(3):141 – 149

Ben-Akiva M, Gershenfeld S (1998) Multi-featured products and services: Analysing pricing and bundling strategies. Journal of Forecasting 17(3-4):175–196, DOI 10.1002/(SICI)1099-131X(199806/07)17:3/4<175::AID-FOR690>3.0.CO;2-N

Benavides D, Trinidad P, Ruiz-Cortés A (2005) Automated reasoning on feature models. In: Pastor O, Falcão e Cunha J (eds) Advanced Information Systems Engineering: 17th International Conference, CAiSE 2005, Porto, Portugal, June 13-17, 2005. Proceedings, Lecture Notes in Computer Science, vol 3520, Springer, Berlin, pp 491 – 503, DOI 10.1007/11431855_34

Benavides D, Segura S, Ruiz-Cortés A (2010) Automated analysis of feature models 20 years later: A literature review. Information Systems 35(6):615 – 636, DOI 10.1016/j.is.2010.01.001

Benavides D, Felfernig A, Galindo JA, Reinfrank F (2013) Automated analysis in feature modelling and product configuration. In: Favaro J, Morisio M (eds) Safe and Secure Software Reuse: 13th International Conference on Software Reuse, ICSR 2013, Pisa, June 18-20. Proceedings, Lecture Notes in Computer Science, vol 7925, Springer Berlin Heidelberg, Berlin, Heidelberg, pp 160 – 175, DOI 10.1007/978-3-642-38977-1_11

Berger T, Rublack R, Nair D, Atlee JM, Becker M, Czarnecki K, Wąsowski A (2013) A survey of variability modeling in industrial practice. In: Proceedings of the Seventh International Workshop on Variability Modelling of Software-intensive Systems, ACM, New York, NY, USA, VaMoS '13, pp 71 – 78, DOI 10.1145/2430502.2430513

Biere A, Heule M, van Maaren H, Walsh T (eds) (2009) Handbook of Satisfiability, Frontiers in Artificial Intelligence and Applications, vol 185. IOS Press, Amsterdam

Blazek P, Partl M, Streichsbier C (2016) Configurator database. URL `https://www.configurator-database.com/`

Chandler AD (1962) General motors – creating the general office. In: Strategy and Structure: Chapters in the History of the American Industrial Enterprise, MIT Press, Cambridge, chap 3, pp 114 – 162

Chu W, Gerstner E, Hess JD (1995) Costs and benefits of hard-sell. Journal of Marketing Research 32(1):97 – 102, DOI 10.1177/002224379503200

Debreu G (1960) Topological methods in cardinal utility theory. In: Arrow KJ, Karlin S, Suppes P (eds) Mathematical Methods in the Social Sciences, 1959. Proceedings of the First Stanford Symposium, Stanford University Press, Stanford, California, chap 2, pp 16 – 26

Fayyad U, Piatetsky-Shapiro G, Smyth P (1996) From data mining to knowledge discovery in databases. AI Magazine 17(3):37–54, DOI 10.1609/AIMAG.V17I3.1230

Felfernig A, Hotz L, Bagley C, Tiihonen J (eds) (2014) Knowledge-Based Configuration: From Research to Business Cases. Morgan Kaufman, Waltham

Fishburn PC (1970) Utility Theory for Decision Making. John Wiley & Sons, New York

Fuhrmann T, Schweizer M, Geyer-Schulz A, Kurz P (2017) On estimating pricing models from end-consumer internet car-configuration data. In: Mucha HJ (ed) Big Data Clustering: Data Preprocessing, Variable Selection, and Dimension Reduction, Report of the Weierstraß-Institut für Angewandte Analysis und Stochastik, vol 29, Weierstraß-Institute (WIAS), Berlin, chap 4, pp 55 – 69

Geyer-Schulz A (2016) The problem of finding interesting patterns. In: Krolak-Schwerdt S, Bömer M (eds) European Conference on Data Science (Program, Abstracts, and Draft Notes), University of Luxembourg, Luxembourg, p 42

Gorman WM (1968) Conditions for additive separability. Econometrica 36(3/4):605 – 609, DOI 10.2307/1909527

Gusfield D, Irving R (1989) The Stable Marriage Problem: Structure and Algorithms. Foundations of Computer Science, MIT Press, Cambridge

Haug A (2007) Representation of industrial knowledge – as a basis for developing and maintaining product configurators. PhD thesis, Department of Manufacturing Engineering & Management, Technical University of Denmark, Lyngby

Huber J, Payne JW, Puto C (1982) Adding asymmetrically dominated alternatives: Violations of regularity and the similarity hypothesis. Journal of Consumer Research 9(1):90 – 98, DOI 10.1086/208899

Jannach D, Kalabis L (2011) Incremental prediction of configurator input values based on association rules – a case study. In: Shchekotykhin K, Jannach D, Zanker M (eds) Proceedings of the International Workshop on Configuration (ConfWS 2011 at IJCAI 2011), Barcelona, Spain, pp 32 – 35

Johnson R, Orme B, Pinnell J (2006) Simulating market preference with build your own data. In: Sawtooth Software (ed) Proceedings of the Sawtooth Software Conference 2006, Sawtooth Software, Inc., Sequim, Washington, pp 245 – 259

Kahneman D, Tversky A (eds) (2009) Choices, Values, and Frames. Cambridge University Press, Cambridge

Keeney RL, Raiffa H (1976) Decisions with Multiple Objectives: Preferences and Value Tradeoffs. John Wiley & Sons, New York

Koehler DJ, Harvey N (2004) Blackwell Handbook of Judgment and Decision Making. Blackwell Publishing, Malden

Kotler P (1980) Tasks and Philosophies of Marketing Management, 4th edn, Prentice Hall, Englewood Cliffs, chap 2, pp 18 – 37

Liechty J, Ramaswamy V, Cohen SH (2001) Choice menus for mass customization: An experimental approach for analyzing customer demand with an application to a web-based information service. Journal of Marketing Research 38(2):183 – 196, DOI 10.1509/jmkr.38.2.183.18849

Mackworth AK (1977) Consistency in networks of relations. Artificial Intelligence 8(1):99 – 118, DOI 10.1016/0004-3702(77)90007-8

McDermott J (1982) R1: A rule-based configurer of computer systems. Artificial Intelligence 19(1):39 – 88, DOI 10.1016/0004-3702(82)90021-2

Mendonca M, Cowan D, Malyk W, Oliveira T (2008a) Collaborative product configuration: Formalization and efficient algorithms for dependency analysis. Journal of Software 3(2):69 – 82, DOI 10.4304/jsw.3.2.69-82

Mendonca M, Wasowski A, Czarnecki K, Cowan D (2008b) Efficient compilation techniques for large scale feature models. In: Proceedings of the 7th International Conference on Generative Programming and Component Engineering (GPCE ’08), ACM, New York, NY, USA, pp 13 – 22, DOI 10.1145/1449913.1449918

Mittal S, Frayman F (1989) Towards a generic model of configuraton tasks. In: Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI-89), AAAI, vol 2, pp 1395 – 1401

Pine BJ (1999) Mass Customization: The New Frontier in Business Competition. Harvard Business School Press, Harvard

Rein IJ (1972a) The Rhetoric of the Car Lot, Scott, Foresman & Company, Glenview, chap 12, pp 118 – 127

Rein IJ (1972b) Rudy’s Red Wagon: Communication Strategies in Contemporary Society. Scott, Foresman & Company, Glenview

Rice J, Bakken DG (2006) Estimating attribute level utilities from design your own product data. In: Sawtooth Software (ed) Proceedings of the Sawtooth Software Conference 2006, Sawtooth Software, Inc., Sequim, Washington, pp 229 – 253

Sabin D, Weigel R (1998) Product configuration frameworks – a survey. IEEE Intelligent Systems and their Applications 13(4):42 – 49, DOI 10.1109/5254.708432

Schwarzkopf S (2011) The consumer as “voter," “judge," and “jury": Historical origins and political consequences of a marketing myth. Journal of Macromarketing 31(1):8 – 18, DOI 10.1177/0276146710378168

Segura S (2008) Automated analysis of feature models using atomic sets. In: First Workshop on Analyses of Software Product Lines (ASPL08) of 12th

International Software Product Lines Conference (SPL08), Limerick, Ireland, pp 201 – 207
Sloan A (1964) My Years with General Motors. Doubleday, Garden City, New York
Soininen T, Tiihonen J, Männistö T, Sulonen R (1998) Towards a general ontology of configuration. Artificial Intelligence for Engineering Design, Analysis and Manufacturing 12(4):357 – 372, DOI 10.1017/S0890060498124083
Thaler RH (2009) Nudge: Improving Decisions about Health, Wealth and Happiness. Penguin Books, London
Vagnoni A (1997) Creative differences. Advertising Age 68(46):1 – 30
Wedel M, Kamakura W (2001) Market Segmentation: Conceptual and Methodological Foundations, International Series in Quantitative Marketing, vol 8. Kluwer Academic Publishers, Boston
Zhang LL (2014) Product configuration: A review of the state-of-the-art and future research. International Journal of Production Research 52(21):6381 – 6398, DOI 10.1080/00207543.2014.942012
Zhang W, Zhao H, Mei H (2004) A propositional logic-based method for verification of feature models. In: Davies J, Schulte W, Barnett M (eds) Formal Methods and Software Engineering: 6th International Conference on Formal Engineering Methods, ICFEM 2004, Seattle, WA, USA, November 8-12, 2004. Proceedings, Lecture Notes in Computer Science, vol 3308, Springer, Berlin, pp 115 – 130, DOI 10.1007/978-3-540-30482-1_16
Zhang W, Mei H, Zhao H (2006) Feature-driven requirement dependency analysis and high-level software design. Requirements Engineering 11(3):205 – 220, DOI 10.1007/s00766-006-0033-x