Portfolio Management: How to Find Your Standard Variants

Frank Dylla and **Daniel Jeuken** and **Thorsten Krebs** [1]

Abstract.

Product portfolio management is one of the most important tasks for companies to secure their future competitiveness. A crucial aspect for portfolio management decisions is the volume of products sold and the sales numbers development over time – one could say: What are your current or upcoming best selling products, often used as "standard products" in sales? Especially for these products it is worthwhile to take actions in reducing costs and improving revenue. Regarding discrete products the task is, simply said, looking for products with the highest quantities or profit sold or significant changes in these quantities over a certain period of time (business intelligence). In contrast, this approach does not work satisfactorily with complex multi-variant products. An aggregated view on products, i.e. ignoring the sales numbers of the variants with their individual features, does not give sufficient insights or may even lead to wrong decisions in portfolio management. The recurring combination of features across multiple types of products might be more important than the type of the product itself. In this paper we investigate differences in identifying potential standard products in comparison to identifying potential standard variants of products. Thereon we derive a high-level framework how standard variants may be deduced from a given set of variants described by characteristics and provide an algorithmic sketch and discuss resulting challenges from a pragmatic perspective.

1 Introduction

Portfolio management is a dynamic decision process evaluating, prioritizing, reorganizing, cancelling, etc. products throughout their lifecycle [6]. As managers have to deal with uncertain and changing information portfolio management is a complex task. One of the major difficulties in product portfolio management is predicting what the customers are willing to pay for. This includes knowing the market, i.e. knowing the current customer demand and knowing how it will most likely change in future. Thus, product portfolio management is complex already when considering simple products, but gets more complex when considering configurable and thus multi-variant products.

But what exactly is the challenging part of this task? Forecasts are created in order to plan the supply chain and production capacities. For simple products this is a rather straightforward task: one can assign a sales forecast to the product identifiers, e.g. material numbers, and use the bill of materials in order to get a list of components that are required. For variant-rich products such as skateboards, however this is not that easy. In general necessary components of a skateboard[2] are the deck, i.e. a plank (in general wooden, but not necessarily), two trucks, i.e. spring mounted axles, and four wheels with bearings. Optional components may be sliptape, paintings, risers, shock pads, nose/tail guards, etc. Consider that not all truck types fit to each deck and that not all wheel/truck combinations fit. An individual composition of these components is sought by the customer – leading to very few skateboards that are sold with the exact same composition of deck, axes, wheels, and so on.

From our experience, for new multi-variant products it is common that product managers guess which variants will be the top selling ones in future, i.e. the decision is based on their gut feeling. Evaluation of the quality of their initial decision is barely feasible as only standard BI techniques are available. These techniques are not sufficient for portfolio planning of multi-variant products as they ignore the structural information of the variants themselves. Standard techniques typically analyze the list of sales over a certain period of time and use product identifiers as the key to identify which one is sold the most and predict how this will change in future. But for variant-rich products that are sold in lot size 1 the product identifier cannot be used as a key criterion. It is rather important to compare characteristics and their values. For example, comparing the product ID, which identifies an individual composition, does not identify that a lot of skateboards use the same wheels. Thus we consider it is important to use the configuration model - containing product data and rule sets - as an input for a new kind of algorithm that does not compare on the level of product identifiers but on the level of a set of product characteristics, which supports better predictions of top-selling variants, i.e. what the market really is willing to pay for.

In order to support the step of evaluating past sales in comparison to original plannings on the level of characteristics and their values, we introduce the notion of *central representative* and propose a potential calculation thereof. We discriminate against the term "standard product", standard variants respectively, as this term describes products which were actually built many times. As you will see later a central representative does not need to have been built once. We are convinced that central representatives will help to recognize changes in client behavior – or the market in general – over time and whether adaptions are reasonable in order to meet the goals of portfolio management.

We start with introducing our understanding of product configuration, which is constraint-based, and introduce diverse variant spaces for later use (Sec. 2.1). We consider definitions of discrete standard and basic products (Sec. 2.2) and elaborate how this relates to standards for multi-variant products (Sec. 2.3). We sketch our approach in Section 3. To avoid misunderstandings with varying definitions of 'standard' we introduce the term *central representative* of a given variant space described by characteristics (Sec. 3.1). In order to find such a central representative a measure of dissimilarity needs to be

[1] encoway GmbH, Germany, email: {dylla,jeuken,krebs}@encoway.de

[2] see `en.wikipedia.org/wiki/Skateboard` (retrieved 8.5.2019)

defined (Sec. 3.2). In Section 3.3 we exemplify how a representative can be computed and how a deviation can be derived thereon. We summarize our considerations in an algorithm sketch (Sec. 3.4). We discuss our approach from various pragmatic perspectives (Section 4). First, we revisit the choice of the set of product vectors $\mathcal{P}$ for which the central representative should be computed (Sec. 4.1). Furthermore, in general data is not available in a well defined form in reality, i.e. not all characteristics and values are defined in a consistent manner (Sec. 4.2). Additionally, multi-variant products are subject to change such that older products may falsify the results that should reflect the current state (Sec. 4.3). Finally, we consider derivation of parameters and further prerequisites necessary in order to apply the algorithm presented to real data. (sec. 4.4).

2 Theoretical background

2.1 What is product configuration?

Felfernig et al. [8] base their understanding of configuration on a definition in [18]: *configuration is a special case of design activity where the artifact being configured is assembled from instances of a fixed set of well defined component types which can be composed conforming to a set of constraints.* A configuration task is the selection of the components and their properties to get a valid combination of the product components, the outcome is also called *product variant* [4].

As a result the component types span a space of potential configurations which are further restricted by *constraints*, which limit the possibilities of how components can be combined. Practice shows that the restrictions may arise from technical feasibility, legal requirements, product-design, or marketing purposes. In general, components or properties of a product are described by *characteristics* in formal product representations. There are additional notions to describe properties of components like attributes or features. For reasons of simplicity we will restrict to the term characteristics throughout this paper. Based on this we can define a *product characteristics vector*, product vector for short.

Definition 1. *Given a set of characteristics $k_i \in \mathcal{K}$ with $i \in \{0, \ldots, N-1\}$ with values from domain $D_i \in \mathcal{D}$ each, we define $[k_0, k_1, \ldots, k_{N-1}]$ as the product (characteristics) vector $\vec{p}$.*

We note that N denotes the maximum number of possible characteristics. Especially, if a characteristic is optional, a specific domain value must be available defining that this characteristic is not chosen, evaluated respectively. As combinations of domain values are not restricted the product vector may reflect a product which is technically not feasible.

Naturally, a configuration task can be considered as a *constraint satisfaction problem* (CSP), see e.g. [8].

Definition 2. *Constraint Satisfaction Problem (CSP): $\langle \mathcal{K}, \mathcal{D}, \mathcal{C} \rangle$: A CSP is defined as a set of variables $k_i \in \mathcal{K}$ with $i \in \{0, \ldots, N-1\}$ with values from domain $D_i \in \mathcal{D}$ together with a set of constraints $c_j \in \mathcal{C}$ and $j \in \{0, \ldots, M-1\}$ defining which combinations of values are allowed or not. A solution of a CSP is a consistent evaluation to all variables (value assignment to all k_i), i.e. no constraint is violated. Otherwise the assignment is called inconsistent. Furthermore, within an assignment values of k_i do not need to be unique, i.e. that k_i may contain multiple valid values which can be considered as alternatives. Given a solution with a unique value per k_i, it is called an atomic solution or according to variant management a variant.*

In more detail, if an assignment contains multiple values for one or more characteristics k_i it contains at least two different atomic solutions. For example, given an assignment where a characteristic contains a value, e.g. for deck A and deck B, this means that the customer at some configuration front-end can still decide for either deck A or B resulting in a valid assignment definitely. In the remainder of this paper we will use k_i synonymously for referring to the characteristic itself as well as for its evaluation, i.e. value assignment, as the meaning becomes clear from the context in most cases. In case of ambiguities we clarify the meaning.

In order to discuss notions of *standard variant*, we need to define several solution spaces based on the CSP definition.

Definition 3. *Theoretical Configuration Space ($\mathcal{S}^{\varnothing}$): This space contains all combinations of characteristics which are possible from a mereological perspective, i.e. from all minimalist configurations to all maximum configurations containing all optional components, but ignoring further constraints. In terms of CSP this is reflected by $\langle \mathcal{K}, \mathcal{D}, \varnothing \rangle$.*

Consider the skateboard example. The minimalist configurations consist of a deck, two trucks, and four wheels as these components are necessary to obtain a functional skateboard from a mereological perspective. A configuration with two decks is not part of $\mathcal{S}^{\varnothing}$, whereas configurations with different truck or wheel sizes are part of $\mathcal{S}^{\varnothing}$, although this may make the skateboard unusable. Maximum configurations consist of the above components plus all optional components which can be installed in parallel. As risers and shock pads are installed in the same place[3], there is no maximum configuration containing both components. As we are interested in valid configurations in the end, we need to define a second configuration space.

Definition 4. *Valid Configuration Space or variant space ($\mathcal{S}$): This space contains only valid configurations, i.e. configurations that satisfy all constraints of the underlying configuration model. Therefore it is given $\mathcal{S} \subseteq \mathcal{S}^{\varnothing}$. As valid configurations are also called* variants*, we will talk of 'variant space' in the remainder of this paper. The variant space directly relates to the space of atomic solutions of a CSP.*

Taking the skateboard example again, configurations with different wheel sizes are not part of $\mathcal{S}$, whereas configurations with different wheel colors may be, depending whether such configurations are permissible with reference to the configuration model.

Other variant spaces may be defined on the 'trading status' of each $\vec{p}$ contained, for example:

Definition 5. *Offered variant space ($\mathcal{S}^{O}$) and sold variant space ($\mathcal{S}^{\$}$): $\mathcal{S}^{O}$ is defined as the space of all variants which have been quoted to customers. $\mathcal{S}^{\$}$ contains only those variants which have been sold.*

As in mass customization the variant space is rather large, in general it can be assumed that not all variants were sold or offered. Nevertheless, in the very extreme case all possible variants have been offered and sold and thus $\mathcal{S}^{\$} \subseteq \mathcal{S}^{O} \subseteq \mathcal{S}$. Based on the presented definition of variant space, an arbitrary number of variant spaces based on relevant criteria can be defined for investigation and comparison.

[3] between trucks and deck

2.2 Discrete standard and basic products

In the context of discrete products a central term for entrepreneurial considerations and decisions is *standard product*.

According to the Lexico dictionary (Oxford)[4] on a general level a standard is (a) *a certain quality or attainment level reached* or (b) *something considered exemplary or as a measure or model according to which others assess to (cf. benchmark, scale, guideline)*.

Following information given by Wikipedia *a technical standard is an established norm or requirement in regard to technical systems. It is usually a formal document that establishes uniform engineering or technical criteria, methods, processes, and practices. In contrast, a custom, convention, company product, corporate standard, and so forth that becomes generally accepted and dominant is often called a de facto standard*.[5]

Specifically considering discrete products a wide variety of definitions is available which take different aspects into account. For example, in the Gabler Wirtschaftslexikon standard product is defined with a focus on quality: *Products that have a generally agreed (standardized) minimum quality. Product changes focus on quantities, prices and times. Standard products can be traded on the stock exchange.*[6] Other definitions base on the criteria whether they are *ready for batch production*.[7]

From our experience the term *standard product* is mainly used in two different ways in manufacturing industry:

1) Either as a label of a product which should be presented as a standard (defined before product is sold at all)
2) or as a product which is established on the basis of different criteria e.g. it is sold the most within a given context, e.g. a region or a specific type of customer.

In order to dissolve this ambiguity we speak of a *predefined standard* in case of 1) and a *derived standard* in case of 2).

Furthermore, a *basic product* – also called generic product – is defined to realize *the core benefit of the product*. This implies that a basic product cannot be further reduced without losing the possibility of intended product usage.[8] In case of a skateboard this is the ability to ride on such a board with pushing oneself forward by foot. A basic product may not be saleable, e.g. due to legal restrictions. An extended product is one which offers additional benefit to customers. In the context of manufacturing companies ... *a basic product might be a rather simple good that experiences relatively consistent consumer demand ...*.[9] Sometimes a core product is differentiated from the product: *The core product of a book is information. It is not the book itself.*[10] The book itself is then the basic product.

2.3 Multi-variant products and standard variants

The term *mass customization* defines the challenge of anticipating individualized products to be manufactured simultaneously with the efficiency of mass production or as stated in [8]: ... *is based on the idea of the customer-individual production of highly variant products under near mass production pricing conditions*. In general, in this context products are multi-variant, i.e. there is more than one option available. One important question for variant management is how the variants can be compared in a reasonable manner. Buchholz states that all variants need to be considered with respect to their product type and that relevant characteristics need to be selected for a reasonable comparison [2]. Buchholz also discusses the relationship between variants and standard. It is critically scrutinised whether a standard variant is the one with maximum quantity, some sort of average or a yardstick for other variants. Nevertheless, it is specifically emphasized that a standard variant is something special compared to other variants. For comparison a measure of discrimination between variants is necessary, but not all characteristics are important such that relevant characteristics need to be selected. In our notation this means, that the product vector $\mathcal{K} = [k_0, k_1, \ldots, k_{N-1}]$ is abstracted to a reduced product vector $\mathcal{K}' \subset \mathcal{K}$ with $N' < N$.

Buchholz also presents different views from literature whether such a standard variant needs to be part of the variant space itself or not. For example, according to Boysen a basic or standard product may be a theoretical construct that has never been physically manufactured [1]. Whether it needs to be manufacturable at all remains unclear. For further details we refer to [2].

On the one hand, to define a standard variant based on aggregated sales numbers over all variants of a variant space is unreasonable from our perspective as it exactly ignores the possible differences of the available variants. Such an approach could be rather considered as a 'standard variant space'. On the other hand to only consider the sales numbers of each variant individually bears problems as well, it even may lead to wrong interpretations. In general, the exact same variant is not sold more than 'a few times'. For example, consider 100 skateboards of 96 different variants sold. This means that most variants were sold once and two may have been sold three times each. This also means that the standard variants may change within a few new sales. Therefore, from our perspective it would not be useful to define these "top selling" variants as standard variants.

From the perspective of the product management and with the aim of an efficient portfolio handling, it is also useful for multi-variant products on the one hand to offer and place a standard variant in the market and on the other hand to analyze which product variant is sold most or is never sold at all.

From our point of view the notion of a basic product can be directly transferred to a basic variant: to cover the basic functionality necessary characteristics must be set with corresponding values reflecting a "basic" quality. In case of a skateboard a deck, two trucks, and four wheels each of rather low quality. In case only one component (characteristic) is missing, it is no variant of a skateboard anymore as it is non-functional. In addition *top-level variants* can be given: variants with a maximum number of characteristics evaluated with corresponding values reflecting a high level of quality, i.e. based on the configuration model no further feature can be selected without deselecting at least one other feature. In some cases, depending on the context, it might appear that not more options are chosen in case of a professional board compared to a basic one, but components of better quality, e.g. the material types of the deck or the wheels. In the end this must be reflected in the underlying metrics.

In Figure 1 we depict relations between basic (b_i), top-level (t_i), and 'regular' (p_i) product variants. Furthermore, each variant may also be computed or defined as a standard variant (marked with *).

The level, i.e. the number of selected characteristics and 'rank' of

[4] www.lexico.com/en/definition/standard (retrieved 2.8.2019)
[5] en.wikipedia.org/wiki/Technical_standard (retrieved 2.5.2019)
[6] wirtschaftslexikon.gabler.de/definition/standardprodukte-42877 (retrieved 6.5.2019, in German)
[7] e.g. www.lawinsider.com/dictionary/standard-products (retrieved 2.5.2019)
[8] wirtschaftslexikon.gabler.de/definition/produkt-42902 (retrieved 2.5.2019, in German)
[9] www.businessdictionary.com/definition/basic-product.html (retrieved 8.5.2019)
[10] www.marketing91.com/five-product-levels/ (retrieved 8.5.2019)

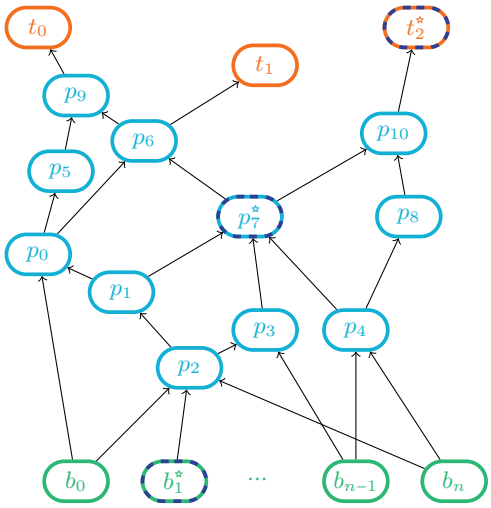

Figure 1. Schematic structure of basic (b_i), top-level (t_i), and standard (p_i) variants inbetween. Variants of any of these levels can be designated as a standard variant (*).

corresponding values is reflected by height. The edges depict that the variants differentiate in a single characteristic.[11] Naturally, basic variants are rather at the bottom and top-level variants at the top of the figure. Nevertheless, it is possible to have feature combinations that aren't separable and so basic as well as top-level variants can exist on different levels. But irrevocably basic variants must not have another connected variant 'below' them, top-level variants 'above' respectively. All other variants inbetween have 'smaller' predecessors and 'larger' successors. Standard variants can be defined on any of these levels. Consider our skateboard example. We define a basic variant as standard skateboard for beginners, a mid-range skateboard as a standard for trained half pipe skaters and a top-level variant as a standard for skate competitions.

3 Approach

We believe that the availability of a standard variant in the sense of an average product of the most selling variants is very helpful in portfolio management. In order to prevent misunderstandings with other definitions (see Sec. 2.2 and 2.3) we will talk of a *central representative of a variant space* instead. One possibility to exploit the central representative in portfolio management is to compare it with predefined standards and adapt them accordingly. In order to discuss the challenges in defining such a central representative in the context of multi-variant products, we need to give some formal definitions regarding configuration spaces (Section 3.1). We define a measure $\mathcal{M}$ (Sec. 3.2) for computation of a central representative (Sec. 3.3). We close this section with an algorithmic sketch, integrating definitions from preceding subsections (Sec. 3.4).

3.1 Definition of a central representative of a variant space

In Section 2.1 we introduced the notion of a product (configuration) vector $\vec{p}$, which holds all characteristics which define a certain product. Let $\mathcal{P} = \{\vec{p}_0, \vec{p}_1, \ldots, \vec{p}_{P-1}\}$ be a set of P product vectors. With $\mathcal{S}^\varnothing$, $\mathcal{S}$, $\mathcal{S}^O$ and $\mathcal{S}^\$$ (see Sec. 2.1) we already defined specific $\mathcal{P}$, i.e. sets where all $\vec{p}_j$ fulfill certain properties. As we are interested in the "best representative" of $\mathcal{P}$ we define a *central representative of* $\mathcal{P}$ based on a measure of similarity or dissimilarity.

Definition 6. *Central representative $\vec{r}_\mathcal{P}$ and deviation $\vec{\nu}_\mathcal{P}$: $\vec{r}_\mathcal{P}$ is the product vector of a product space $\mathcal{P}$ which has the overall minimal dissimilarity to all $\vec{p}_j \in \mathcal{P}$ considering a measure $\mathcal{M}$. Furthermore, we define the deviation $\vec{\nu}_\mathcal{P}$ to be the vector of the individual deviations ν_i of assigned values per characteristic k_i (see Figure 2).*

Simplified, one could say $\vec{r}_\mathcal{P}$ is the average product of $\mathcal{P}$ regarding the measure $\mathcal{M}$ or more specific, the one that minimizes the dissimilarity to all $p_i \in \mathcal{P}$. The deviations ν_i can be defined in multiple ways. We detail this in Section 3.3. We note that, based on this definition, it is not necessary, that $\vec{r}_\mathcal{P} \in \mathcal{P}$. Furthermore, as several solutions may have the same aggregated distance regarding $p_i \in \mathcal{P}$ based on $\mathcal{M}$, there may be no unique central representative $\vec{r}_\mathcal{P}$. We sketch how a measure $\mathcal{M}$ can be defined below.

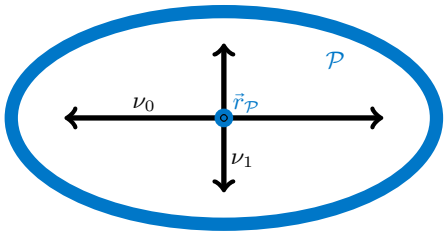

Figure 2. Schematic depiction of a variant space $\mathcal{P}$ with its central representative $\vec{r}_\mathcal{P}$ and its deviation $\vec{\nu} = [\nu_0, \cdots, \nu_{n-1}]$ with $n = 2$

3.2 Measure $\mathcal{M}$: Dissimilarity of variants

$\mathcal{M}$ could be either a measure of similarity or dissimilarity. Although $\mathcal{M}$ can be defined arbitrarily, e.g. based on $\sum$, $\prod$, min, max or some complex aggregation function, we stick to a specific distance based measure, and thus dissimilarity, for reasons of simplicity. For future research a promising link is given by case-based reasoning (CBR) as the notion of similarity is central to this approach [9, 16, e.g.]. Nevertheless, although CBR has been applied to product configuration, to our knowledge specific product similarities have not been extensively investigated in the literature; exceptions are [12, 21, 20]. Aspects of similarity have been studied in the context of CSP [7, 5, e.g.] resulting in the need of Euclidian distance measures from a practical perspective. In the following of this section we summarize aspects of similarity measures relevant to our approach.

A Euclidian distance measure δ for some entities o, p and q is reflexive: $\delta(p,p) = 0$, symmetric: $\delta(p,q) = \delta(q,p)$, and transitive: $\delta(o,q) \leq \delta(o,p) + \delta(p,q)$. For reasons of simplicity, we will talk of *distance* in the remainder of this paper.

In order to define a distance measure $\mathcal{M}$ consider a variant space, e.g. $\mathcal{S}$, and a subset thereof, e.g. $\mathcal{S}^\$$ ($\mathcal{S}^\$ \subseteq \mathcal{S}$). This implies that $\vec{p} \in \mathcal{S}$ and $\vec{q} \in \mathcal{S}^\$$ contain the same characteristics k_x with $x \in \{0, \cdots, N-1\}$ in the same order. First, we need a distance between values from the same characteristic δ_x for all $x \in \{0, \cdots, N-1\}$, for example:

$$\delta_x(k_x^p, k_x^q) = |k_x^p - k_x^q| \tag{1}$$

with k_x^p denoting the value of the x-th characteristic of product vector $\vec{p}$, k_x^q of $\vec{q}$ respectively. Depending on the type of scale of

[11] For reasons of simplicity we neglect that connected variants may differ in more than one characteristic as they are inseparable due to the rule set.

the characteristic (i.e. nominal, ordinal, interval or ratio scale) certain calculations may not be possible, e.g. subtraction or addition on nominal scale is not reasonable. On nominal scale only the equality between values can be determined, i.e. are two values the same or not. If a level of similarity is required at least an ordinal scale for the values must be available, i.e. a linear order for the values for the definition of a median. For interval or ratio scale a mean can be defined.

This results in a distance vector of distances per characteristic

$$\vec{\delta}(\vec{p},\vec{q}) = \begin{bmatrix} \delta_0(k_0^p, k_0^q) \\ \vdots \\ \delta_{N-1}(k_{N-1}^p, k_{N-1}^q) \end{bmatrix} = \begin{bmatrix} d_0 \\ \vdots \\ d_{N-1} \end{bmatrix} \tag{2}$$

The next step is to aggregate these individual distances into a single distance value describing the distance between two product vectors. It needs to be reflected that not all characteristics are equally important. Therefore a weighting factor w_i needs to be integrated for each characteristic. If characteristic k_i should not be considered, the corresponding w_i needs to be set to zero. Furthermore, not all distances for individual characteristics may have the same range and thus, one characteristic may dominate others, therefore a normalizing factor v_i is necessary. For example, consider a distance vector with $N = 3$ where d_0 represents a binary distance ($d_0 \in \{0,1\}$), d_1 represents a distance between zero and five ($d_1 \in [0,5]$), and d_2 represents a distance between zero and thousand ($d_2 \in [0,1000]$). In most cases d_2 would dominate or overrule d_1, which in turn also dominates d_0. Therefore, it is import that all value ranges of the k_i are normalized, e.g. to values between zero and one. This results in a distance between two product vectors $\vec{p}$ and $\vec{q}$.

$$\Delta(\vec{p},\vec{q}) = \frac{1}{N}\sum_{i=0}^{N-1} w_i v_i d_i \tag{3}$$

We give a schematic impression of a distance Δ between two product vectors $\vec{r}_{\mathcal{S}}$ and $\vec{r}_{\mathcal{S}^\$}$ in Figure 3. Nevertheless, it still remains open how central representatives like $\vec{r}_{\mathcal{S}}$ and $\vec{r}_{\mathcal{S}^\$}$ can be determined based on Δ.

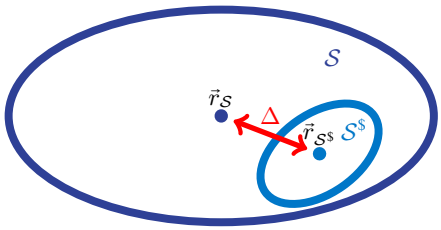

Figure 3. Schematic depiction of a general variant space ($\mathcal{S}$, dark blue) and its central representative ($\vec{r}_{\mathcal{S}}$) and sales variant space ($\mathcal{S}^\$$, light blue) also with its central representative ($\vec{r}_{\mathcal{S}^\$}$). The Δ depicts the difference between $\vec{r}_{\mathcal{S}}$ and $\vec{r}_{\mathcal{S}^\$}$.

3.3 Calculation of central representatives

We defined the central representative $\vec{r}_{\mathcal{P}}$ as a variant which *minimizes the overall dissimilarity* (cf. Definition 6). Furthermore, it is not a requirement that $\vec{r}_{\mathcal{P}}$ is itself an element of $\mathcal{P}$. Consider these two definitions of central representatives of $\mathcal{S}^\$$.

$$\vec{r}_{\mathcal{S}^\$} = \underset{\vec{r}\in\mathcal{S}^\$}{\operatorname{argmin}} \sum_{i=0}^{N-1} \Delta(\vec{r},\vec{p}_i) \text{ with } \vec{p}_i \in \mathcal{S}^\$ \tag{4}$$

$$\vec{r}_{\mathcal{S}^\$} = \underset{\vec{r}\in\mathcal{S}}{\operatorname{argmin}} \sum_{i=0}^{N-1} \Delta(\vec{r},\vec{p}_i) \text{ with } \vec{p}_i \in \mathcal{S}^\$ \tag{5}$$

In the first case (Eq. 4) $\vec{r}$ has been sold itself as $\vec{r} \in \mathcal{S}^\$$, whereas in the second case (Eq. 5) $\vec{r}$ is a general technically feasible variant ($\vec{r} \in \mathcal{S}$). One could even relax that the representative not even needs to be technically feasible, and thus select $\vec{r} \in \mathcal{S}^\varnothing$ (cf. 2.3).

In conjunction with the central representative it is also of interest 'how large' or 'how widespread' the set is, which it represents. For this we need a notion of deviation, diameter, or variance. For now, we stick with the notion of average deviation per characteristic (ν_i) for all $\vec{p}_j \in \mathcal{P}$ as it suffices our needs.

$$\nu_i = \frac{1}{P}\sum_{j=0}^{P-1} \delta_i(r_i, k_i^j) \tag{6}$$

Then $\vec{\nu} = [\nu_0, \ldots, \nu_{N-1}]$ denotes a vector of all deviations per characteristic.

It is not beneficial if a central representative covers a 'too wide range' of variants, i.e. one or several ν_i are rather high for some characteristics k_i, as it would not give much help for portfolio optimization, especially if members of the set of product vectors are not distributed uniformly. Consider the case depicted in Figure 4. Products were sold in two rather distant regions of the variant space. Considering them as one set would lead to a representative which does not reflect the situation at hand (orange space). We need to look for separate subsets, i.e. clusters, instead, to come to a result depicted by the two separate regions $\mathcal{S}_0^\$$ and $\mathcal{S}_1^\$$ (light blue). As we have defined a central representative and a deviation thereof, various cluster analysis methods are applicable, e.g. centroid-based or density based clustering. For an overview of existing clustering methods we refer to [13, 23, 17, e.g.]. The adequate selection of a clustering method will be a crucial task for the successful application of the approach proposed.

For pragmatic reasons we restrict our considerations to clustering parameters (assuming a clustering method given) to *maximum deviation per characteristic* and a *minimum number of members per cluster*. Therefore, a vector of thresholds $\vec{\theta} = [\theta_0, \ldots, \theta_{N-1}]$ for the corresponding characteristics k_i and $\theta_\#$ for the minimum number needs to be given.

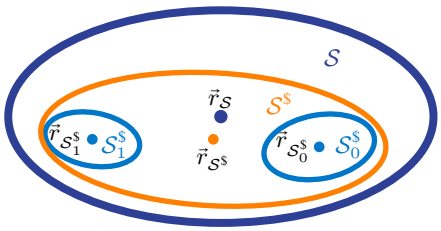

Figure 4. Schematic depiction of cluster splitting due to high variance in single cluster consideration.

3.4 An algorithm sketch

We summarize the parts of how to find adequate representatives for a given set of product vectors $\mathcal{P}$ (e.g. sold variants $\mathcal{S}^\$$) out of another given set of product vectors $\mathcal{Q}$ (e.g. the overall variant space $\mathcal{S}$) in Algorithm 1. In the beginning only the single cluster P exists for which central representative $\vec{r}_{\mathcal{P}}$ and deviation $\vec{\nu}_{\mathcal{P}}$ is calculated.

If there is any deviation ν_i which is above its defined threshold θ_i $\mathcal{P}$ needs to be splitted in two clusters.[12] In the next iteration at least two clusters need to be considered. At some point clusters with only few members are computed ($< \theta_\#$). We ignore these clusters from further consideration in this iteration. We continue with increasing the number of clusters until we obtain a set of clusters with each containing a central representative with each deviation per characteristic below the given threshold ($\forall i \;\; \nu_i \leq \theta_i$). We note, that we increase the number of clusters iteratively and start the cluster splitting from the original set $\mathcal{P}$ on purpose. If not doing so the order of considering $p_i \in P$ might have an effect and thus, would lead to different results if p_i are represented in a different order.

Input: $\mathcal{P}$, $\mathcal{Q}$, $\vec{\theta}$, $\mathcal{M}$, $\theta_\#$
Result: $\mathcal{S}$:= set of central representatives for $\mathcal{P}$ out of $\mathcal{Q}$
no_of_clusters := 1 ;
$\mathcal{S} = \{\mathcal{P}\}$;
$\mathcal{R}$:= calculate list of representatives from Q for all $s_j \in \mathcal{S}$ based on $\mathcal{M}$;
Θ := calculate list of all deviations for corresponding r_j and s_j based on $\mathcal{M}$;
while $\exists i, j$ *with* $\nu_i^j \in \Theta > \theta_i$ *for any* $s_j \in \mathcal{S}$ **do**
    no_of_clusters := *no_of_clusters* +1 ;
    $\mathcal{S}$:= $clusterSplitting(\mathcal{P}, \mathcal{M}, no\_of\_clusters)$;
    delete all $s_j \in \mathcal{S}$ from $\mathcal{S}$ where $|s_j| < \theta_\#$;
    $\mathcal{R}$:= calculate list of representatives from $\mathcal{Q}$ for all $s_j \in \mathcal{S}$ based on $\mathcal{M}$;
    Θ := calculate list of all deviations for corresponding r_j and s_j based on $\mathcal{M}$;
end

Algorithm 1: Algorithmic sketch for deducing central representatives out of the variant space $\mathcal{Q}$ based on the variants given by the variant space $\mathcal{P}$.

4 Pragmatic considerations

Not all characteristics of product vectors must be considered as relevant information might be covered by other characteristics (Sec. 4.1). In general, data provided by companies needs some preparation as this data is often not consistent concerning characteristics' and values' denomination (Sec. 4.2). We consider temporal restriction of data and how observations over time can be derived (Sec. 4.3). Before Algorithm 1 can be applied value ordering and weighting factors for each characteristic must be available 4.4.

4.1 Contentual evaluation

In order to support a business question a contentual focus on data is necessary. Simplified, two levels of contentual constraints can be differentiated. First, the context of each variant ($\vec{p}_i$) can be considered. Context can be defined on different perspectives, e.g. in which shop or region the variant has been generated, by whom, whether it has been sold, only offered, or never even offered (cf. $\mathcal{S}$, $\mathcal{S}^O$, $\mathcal{S}^\$$ in Sec. 2.1), or for which application, domain respectively, it was bought if this information is available. Second, the relevance of each characteristic should be checked as consideration of all characteristics may block the view on relevant information, for example, the color of the trucks or some non-visible strings on some component. Chizi and Maimon state that a focus on relevant characteristics has several advantages [3]. For example, removal of irrelevant characteristics improves efficiency as well results are more conclusive and easier to interpret due to the focus on key features. Nevertheless, a too limited choice of characteristics leads to information loss and reduces the quality of the results. For further information on feature selection methods we refer to [22]. If a characteristic is considered irrelevant for an evaluation at hand w_i (cf. Eq. 3) should be set to zero in the calculations. For all characteristics with $w_i > 0$ the relative relevance needs to considered very carefully as slight changes may lead to significant changes in the classification of the data. For example, if the results are designed for adapting standard products a slight change in the parameters might lead to a different variant.

[12] How this is actually done depends on the clustering algorithm chosen.

4.2 Data preparation

Practice shows that within companies often master data is not coordinated. In general, this leads to multiple characteristics containing the same information, potentially represented differently, e.g. using different text strings, numbers, or different units. As products are subject to permanent change, the inconsistency of data increases over time. In order to ease and automatize analysis in the long run, data synchronization is inevitable. Nevertheless, considering given data, data cleansing is essential to prevent bad decisions based on bad analysis results [24]. Maletic described the data preparation as a multistep procedure comprising (1) definition of error types, (2) finding instances of these errors, and (3) correction of them [11]. He emphasizes that each of these steps is a complex task in itself.

To give an idea of the effort that needs to be taken, we present a non-exhaustive list of different error types in (master) data below. A common error type is conditioned by different notions or representations, i.e. characteristics and values holding the same information, but represented with different spellings. These errors often arise from inconsistent usage of blanks, hyphens, prefixes, suffixes or abbreviations. Different units may also be used, e.g. due to different intended usage. Characteristics holding complex information, i.e. connected information, are problematic as well as further processing might be limited. A common example is a combined string representation of length, width, and height (sometimes without a given unit) instead of having individual numerical characteristics for each of them. A tricky type of errors comprises misleading value specifications, e.g. frame sizes termed with numerical values which have to be interpreted in a specific manner so that naive calculation is not possible. Consider frame sizes 5, 8, and 12 which reflect three consecutive frame sizes. The physical difference in size cannot be calculated from these values, instead other data like length, width, and height of certain components need to be considered. Furthermore, the conceptual distance cannot be calculated from these 'values': as the categories are consecutive the distance is 1 and not 3 and 4. In order to prevent trimming of leading zeros, such terms may be even stored as strings. Elimination of errors of this type requires very specific semantic knowledge, which makes it not only hard to spot these errors, but also to correct them. For further information on data cleansing and data quality we refer to [14, 15].

As a result of data preparation we get a set $\mathcal{P}$ of product vectors $\vec{p}_i$ with consistent $[k_0, k_1, \ldots, k_{N-1}]$, i.e. with comparable information stored in the same characteristic with the same value for every product variant.

4.3 Temporal evaluation

Products are subject to permanent change. They are designed, developed, sold, and refined, potentially several times. Such refinements and changes in expectations of the market may result in changes of central representatives. Therefore, regardless whether from technical or sales perspective, it is not reasonable to consider outdated data, which leads to the application of methods from time series analysis. Furthermore, as sales numbers for the products of interest may vary significantly over time, consideration of single time points (or rather small time intervals only) may show varying results for each of these time points.

One applicable method in order to generate smoothed results is the sliding window approach (SWA), see for example [10]. The basic idea is to evaluate overlapping intervals, so called windows, to get smoother and more consistent results. We depict relevant parameters for the SWA in Figure 5. Let d denote the overall period under review (one year in the given example). The window size is denoted by w (three month) with $w \ll d$ and the corresponding step size by s (1 month) with $s \leq w$. Analysis is then performed for data in each window separately.

The choice of specific values for d, w and s is very crucial and must be considered carefully, especially if conclusions on future developments are drawn. For example, if d is chosen too small the corresponding data set may be too small to generate significant results. Statistical or learning methods support a reasonable choice, [19, e.g.].

Algorithm 1 can be extended in such a way that not only a single time point is considered ($\mathcal{P}$), but subsequent sets, i.e. subsequent windows. On this basis developments of the central representatives and their corresponding deviations can be observed: how they 'wonder around' and how the number of clusters increases or decreases.

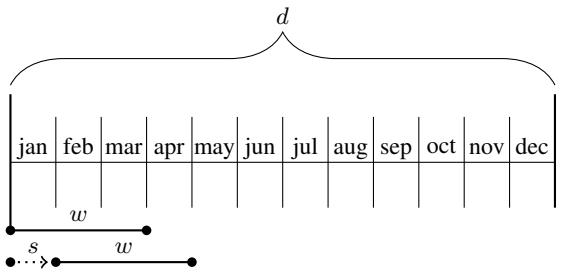

Figure 5. Sliding window approach with d denoting the overall period considered, w denoting the window size and s the step size.

4.4 Weighting factors and value ordering

The approach is significantly based on the definition of the measure $\mathcal{M}$ containing the distances δ and Δ, which in turn contains weighting factors w_i for each characteristic. First experiments have shown that distance measures on nominal data very much influences the results significantly as the distance can be only either zero or one. A rather low weighting factor for these characteristics compared to the other ones may be a solution, but must be evaluated further in future. For now we tend to ignore these characteristics as dissimilarity is in most cases reflected in other characteristics as well. Our gut feeling, but without proof, tells us that similar effects may be the case for integrating ordinal scale data with interval and ratio scale data. For interval and ratio scale data naturally a distance is given – assuming the characteristic is not misinterpreted as such and is 'only' on ordinal scale (cf. Sec. 4.2). For ordinal data this is not the case, a linear ordering has to be defined manually. Although, an ordering of terms like "basic", "advanced", "expert", and "professional" might be considered trivial in the first place, it is a tricky, currently manual and time consuming task and thus, also error prone. Looking at the terms "expert" and "professional" the question is whether "expert" is before or after "professional" or equal in the end as they relate to completely different aspects of the product. It may be possible that a reasonable distance between terms like "basic" and "advanced" is definable, i.e. how far is "basic" from "advanced", "advanced" from "expert" and so forth. We refrain from this as the resulting costs would not be in a reasonable cost-benefit relation for an industrial company. For a start an equidistant conceptual distance measure should suffice, i.e. all preceding and succeeding terms in a linear order have the same distance.

In business intelligence it is common to not only consider the number of sold units, but also profit or the number of sold units per quote is part of the analysis for example. On the one hand a pragmatic way without changing the algorithm is to modify the original set by reducing or multiplying the number of equal product vectors in $\mathcal{P}$. On the other hand an additional weighting factor per p_i could be introduced, which would be much more efficient regarding run-time of the algorithm.

5 Summary and Outlook

To support portfolio management for multi-variant products we examined definitions of 'standard' for discrete and multi-variant products. To differentiate from these definitions we introduced the term central representative of a variant space. We derived an algorithmic sketch based on a measure $\mathcal{M}$ to calculate representatives for clusters with reasonable size. Finally, we discussed tasks necessary before the algorithm can be applied to real data.

As the work on central representatives for a variant space is in an early stage many tasks and questions remain open. The straightforward next step is to experiment with large scale real data instead of few small toy examples. Furthermore, the determination of weighting factors w_i is a challenging task. We need to investigate to what extent learning methods, either supervised or unsupervised, may ease the task. Once real data is available it will be a worthwhile task to reconsider alternative definitions of distance functions, e.g. investigating the impacts of choosing $\prod$, $\min$, $\max$ or some other function as aggregation operators. In theory it is possible that multiple central representatives are available. If this case also appears with real data, we need to investigate how to deal with it.

Acknowledgement

We thank the anonymous reviewers for critically reading the manuscript and providing helpful comments for clarification and improvement of the manuscript.

REFERENCES

[1] Nils Boysen, *Variantenfließfertigung*, volume 49, Deutscher Universitätsverlag, 2005.

[2] M. Buchholz, *Theorie der Variantenvielfalt: Ein produktions- und absatzwirtschaftliches Erklärungsmodell*, SpringerLink : Bücher, Gabler Verlag, 2012.

[3] Barak Chizi and Oded Maimon, 'Dimension reduction and feature selection', in *Data Mining and Knowledge Discovery Handbook, 2nd ed.*, eds., Oded Maimon and Lior Rokach, 83–100, Springer, (2010).

[4] Bjørn Christensen and Thomas D. Brunoe, 'Product configuration in the eto and capital goods industry: A literature review and challenges', in *Customization 4.0*, eds., Stephan Hankammer, Kjeld Nielsen, Frank T. Piller, Günther Schuh, and Ning Wang, pp. 423–438, Cham, (2018). Springer International Publishing.

[5] Jean-François Condotta, Souhila Kaci, Pierre Marquis, and Nicolas Schwind, 'A syntactical approach to qualitative constraint networks merging', in *Logic for Programming, Artificial Intelligence, and Reasoning - 17th International Conference, LPAR-17, Yogyakarta, Indonesia, October 10-15, 2010. Proceedings*, eds., Christian G. Fermüller and Andrei Voronkov, volume 6397 of *Lecture Notes in Computer Science*, pp. 233–247. Springer, (2010).

[6] Robert Cooper, Scott Edgett, and Elko Kleinschmidt, 'Portfolio management - fundamental to new product success', *The PDMA Toolbook for New Product Development*, (01 2002).

[7] Frank Dylla, Jan Oliver Wallgrün, and Jasper van de Ven, 'Merging qualitative information: Rationality and complexity', in *QUAC2015: Workshop on Qualitative Spatial and Temporal Reasoning: Computational Complexity and Algorithms*, (September 2015).

[8] Alexander Felfernig, Lothar Hotz, Claire Bagley, and Juha Tiihonen, *Knowledge-based Configuration: From Research to Business Cases*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1 edn., 2014.

[9] Gavin Finnie and Zhaohao Sun, 'Similarity and metrics in case-based reasoning', *Information Technology papers*, **17**, (03 2002).

[10] Yupeng Hu, Cun Ji, Ming Jing, Yiming Ding, Shuo Kuai, and Xueqing Li, 'A continuous segmentation algorithm for streaming time series', in *Collaborate Computing: Networking, Applications and Worksharing - 12th International Conference, CollaborateCom 2016, Beijing, China, November 10-11, 2016, Proceedings*, eds., Shangguang Wang and Ao Zhou, volume 201 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp. 140–151. Springer, (2016).

[11] Jonathan I. Maletic and Andrian Marcus, *Data Cleansing: A Prelude to Knowledge Discovery*, 19–32, Springer US, 07 2010.

[12] Hiroya Inakoshi, Seishi Okamoto, Yuiko Ohta, and Nobuhiro Yugami, 'Effective decision support for product configuration by using CBR', in *International Conference on Case-Based Reasoning*, (01 2001).

[13] Leonard Kaufman and Peter J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, John Wiley & Sons, 1990.

[14] Lukasz A. Kurgan and Petr Musilek, 'A survey of knowledge discovery and data mining process models', *Knowl. Eng. Rev.*, **21**(1), 1–24, (March 2006).

[15] Ohbyung Kwon, Namyeon Lee, and Bongsik Shin, 'Data quality management, data usage experience and acquisition intention of big data analytics', *International Journal of Information Management*, **34**(3), 387 – 394, (2014).

[16] Michael M. Richter and Rosina O. Weber, *Case-Based Reasoning - A Textbook*, Springer, 2013.

[17] Lior Rokach, 'A survey of clustering algorithms', in *Data Mining and Knowledge Discovery Handbook, 2nd ed.*, eds., Oded Maimon and Lior Rokach, 269–298, Springer, (2010).

[18] D Sabin and R Weigel, 'Product configuration frameworks-a survey', *Intelligent Systems and their Applications, IEEE*, **13**, 42 – 49, (08 1998).

[19] Hela Sfar and Amel Bouzeghoub, 'Dynamic streaming sensor data segmentation for smart environment applications', in *Neural Information Processing - 25th International Conference, ICONIP 2018, Siem Reap, Cambodia, December 13-16, 2018, Proceedings, Part VI*, eds., Long Cheng, Andrew Chi-Sing Leung, and Seiichi Ozawa, volume 11306 of *Lecture Notes in Computer Science*, pp. 67–77. Springer, (2018).

[20] Sara Shafiee, Katrin Kristjansdottir, and Lars Hvam, 'Automatic identification of similarities across products to improve the configuration process in eto companies', *International Journal of Industrial Engineering and Management*, **8**(3), 167–176, (2017).

[21] Hwai-En Tseng, Chien-Chen Chang, and Shu-Hsuan Chang, 'Applying case-based reasoning for product configuration in mass customization environments', *Expert Syst. Appl.*, **29**(4), 913–925, (2005).

[22] Cen Wan, *Hierarchical Feature Selection for Knowledge Discovery*, Advanced Information and Knowledge Processing, Springer International Publishing, 2019.

[23] Rui Xu and Donald C. Wunsch II, 'Survey of clustering algorithms', *IEEE Trans. Neural Networks*, **16**(3), 645–678, (2005).

[24] Marcus Zwirner, 'Datenbereinigung zielgerichtet eingesetzt zur permanenten Datenqualitätssteigerung', in *Daten- und Informationsqualität: Auf dem Weg zur Information Excellence*, chapter 6, 101–120, Springer Fachmedien Wiesbaden, (06 2018). (in German).