Res Eng Design
DOI 10.1007/s00163-015-0201-4

ORIGINAL PAPER

Product family architecture design with predictive, data-driven product family design method

Jungmok Ma[1] · Harrison M. Kim[2]

Received: 18 April 2014 / Revised: 25 August 2015 / Accepted: 27 August 2015

Abstract This article addresses the challenge of determining optimal product family architectures with customer preference data. The proposed model, predictive data-driven product family design (PDPFD), expands clustering-based approaches to incorporate a market-driven approach. The market-driven approach provides a profit model in the near future to determine the optimal position and number of product architectures among product architecture candidates generated by the *k*-means clustering algorithm. An extended market value prediction method is proposed to capture the trend of customer preferences and uncertainties in predictive modeling. A universal electric motors design example is used to demonstrate the implementation of the proposed framework in a hypothetical market. Finally, the comparative study with synthetic data shows that the PDPFD algorithm maximizes the expected profit, while clustering-based models do not consider market so that less profit can be achieved.

Keywords Product family design · Clustering-based approach · Market-driven approach · Prediction intervals · Predictive design analytics

✉ Harrison M. Kim
hmkim@illinois.edu

Jungmok Ma
jungmokma@kndu.ac.kr

[1] Department of National Defense Science, Korea National Defense University, Seoul, Korea

[2] Department of Industrial and Enterprise Systems Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

1 Introduction

Today's highly competitive market situation and enormous data generation environment mean companies and design engineers have to consider a wide variety of customer preferences and requirements. Massive-scale customer preference data are available from various data sources such as company databases, social networks, and clickstreams. In order to accommodate the diversity of customer preferences, designing a family of products becomes a prevailing strategy across many industries (Tseng 1998; Simpson 2004; Simpson et al. 2012).

Product family design represents designing "a set of products that share one or more common elements (e.g., components, modules, and subsystems)" in order to satisfy various market applications (Simpson et al. 2014). The product family design paradigm was successfully implemented by companies such as Sony, Hewlett Packard, Black & Decker, Volkswagen, and Rolls Royce (Simpson et al. 2012, 2014). One of the important tasks in this complex engineering design problem is the determination of optimal product family architectures (de Weck et al. 2003). The product architecture is "the arrangement of functional elements to the physical building blocks" (Ulrich and Eppinger 2012) and works as a target (e.g., performance requirements) of engineering design for product variants (de Weck et al. 2003).

The main question considered in this article is how to determine the optimal product family architectures with customer preference data. Clustering-based methodologies (Tucker et al. 2010; Chan et al. 2012) were presented to identify central points of clusters (market segments) in the customer preference space (performance requirements). The central points are *ideal points* in market segments (Chan et al. 2012) and are also product family architecture

Published online: 09 September 2015

candidates. Tucker et al. (2010) proposed that a clustering technique can enable design engineers to identify the optimal number of product architectures. This article expands these clustering-based approaches (Tucker et al. 2010; Chan et al. 2012) to incorporate a market-driven approach (de Weck et al. 2003; Kumar et al. 2009). The market-driven approach provides a profit model as an objective function to determine optimal product architectures. Unlike the previous market-driven approaches (de Weck et al. 2003; Kumar et al. 2009), this article does not assume that (1) market segments are given and (2) customer preferences are static (i.e., no change over time).

Figure 1 shows the example of a clustering-based approach in the two-dimensional (requirements 1 and 2) customer requirement (circles) space. The objective is to determine the position and number of product architectures (e.g., one rectangle and three triangles) in order to satisfy customers' requirements. Previous clustering-based approaches only consider their clustering objective functions. For example, the product architecture in the middle (rectangle) can be chosen by clustering methods but it might end up with an inferior solution from the perspective of markets. With the guidance of market-driven approaches, clustering-based approaches can produce the most economical clusters for decision makers. Once architectures are determined, then clusters can be interpreted as market segments (dotted lines).

Figure 2 shows the example of a market-driven approach, which evaluates product architecture candidate sets (one rectangle and three triangles in Fig. 1) in terms of profit. With estimated revenue and production cost, the profit and its uncertainty (dotted line) can be estimated. Note that the *X*-axis represents the number of product architectures and the *Y*-axis represents the monetary value. When the number of product architectures is increased, the fixed costs will be increased with more product variants. However, since more customers' product requirements can be satisfied, revenue can be increased too. Figures 1 and 2 together show the necessity of the market-driven approach in the clustering-based approach.

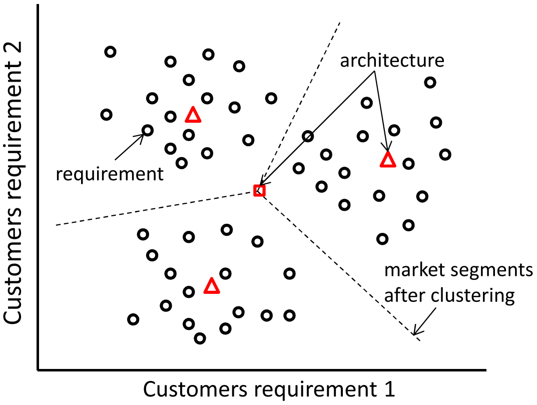


Fig. 1 Example of clustering-based approach

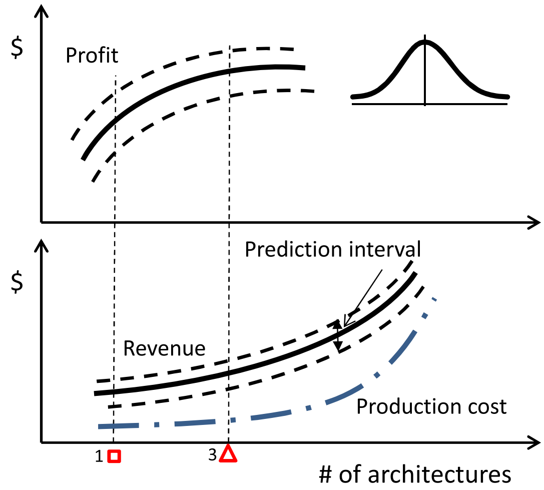


Fig. 2 Example of market-driven approach

The product family design scenario that this article focuses on is presented as follows. The products of interest are products or parts that can be highly shared by many other products, including universal motors in power tools and home appliances, engines in on and off-road vehicles, and batteries in electronics. These products should satisfy a wide variety of different customers' requirements. A company wants to analyze historical transactional data in order to support its next product family architecture decision for new orders.

Predictive, data-driven product family design (PDPFD) proposed in this article aims to merge clustering-based and market-driven approaches based on the predictive design analytics (Ma et al. 2014; Ma and Kim 2014), which enables design engineers to extract knowledge from large-scale, multidimensional, unstructured, volatile data, and transform that knowledge and trend into design decision making. The proposed framework introduces predictive profit modeling in a clustering-based model so that it can support complex product family architecture decisions. For predictive profit modeling, an updated market value prediction method is proposed with time series analysis. The proposed framework is demonstrated using a universal motor design problem (Simpson et al. 2001) with a larger volume of customer preference data than previous clustering-based approaches (Tucker et al. 2010; Chan et al. 2012). Finally, a previous clustering-based method (Tucker et al. 2010) is compared to the PDPFD method in order to show the benefits of the proposed method.

The rest of the paper is organized as follows: Sect. 2 provides related work in the area of clustering-based and market-driven product family design. The proposed approach is presented in Sect. 3 followed by a case study in Sect. 4. The benefits and limitations of the proposed framework along with future work are discussed in Sect. 5.

2 Related work

2.1 Product family design

Recent advances in product family design were discussed in Pirmoradi et al. (2014) from customer needs, functional requirements, design parameters, process variables to logistics variables. Basically, there are two approaches in product family design to utilize a product platform ("the set of parameters and/or features that remain constant" (Simpson et al. 2001): module-based and scale-based product family design (Simpson 2004). Module-based product family design represents building-related products using functional modules from the platform, while scale-based product family design represents designing related products by varying (e.g., stretch or shrink) scaling variables while making common parameters constant. Examples of both approaches can be found in Simpson (2004).

This article focuses on multiple-platform scale-based product family design with known common parameters. Multiple-platform design was studied using a heuristic approach with clustering analysis based on sensitivity analysis (Dai and Scott 2007) and an information theoretical approach (Chen and Wang 2008). Some previous works (Nayak et al. 2002; Messac et al. 2002; Chen and Wang 2008) discussed product family design with unknown common parameters.

In multiple-platform scale-based product family design, product family architecture design is a target setting problem for product variants (de Weck et al. 2003). It is also a positioning problem of a product family into different market segments or clusters of customer preferences (Pirmoradi et al. 2014). Clustering can be used to find the number of product variants which encompass the maximum possible customer preferences (Pirmoradi et al. 2014). Two approaches in product family architecture design will be reviewed in the following sections.

2.2 Clustering-based product family design

With the emergence of large database management system and significant improvements in storage devices, corporations are now able to utilize large-scale data for decision making. To this end, clustering-based or data mining models were proposed to support the product family architecture design problem.

Agard and Kusiak (2004) introduced the possible usage of data mining-based methodologies for product family design. Given that customer demographics and functional requirements are available, clustering methods can be applied to group similar customers so that a representative customer can be identified. Also, functional requirements can be associated with each other in order to find dependencies using association rule mining techniques (Witten and Frank 2005). Moon et al. (2006) proposed that data mining techniques can identify a platform with variants and unique modules. Association rule mining captured associated rules from product functions, and these rules were clustered as modules using fuzzy c-means clustering (Bezdek 1981). Tucker et al. (2010) developed a product family optimization model with ReliefF attribute weighting (Kira and Rendell 1992) and X-means clustering (Pelleg and Moore 2000) techniques. The X-means clustering gave the number and specifications of architectures, and the ReliefF provided the importance of each design attribute in the optimization model. Chan et al. (2012) proposed fuzzy clustering to group customer requirements as market segments. The center points of market segments were used for the development of product variants.

Tucker et al. (2010) and Chan et al. (2012) showed that market segmentation can be realized automatically by clustering methods instead of being assumed to be given or resorting to experts' opinions. However, they did not consider market so it is possible to have sub-optimal solutions in terms of profit as shown in Figs. 1 and 2. Moreover, previous studies involved small data sets [e.g., 50 in Chan et al. (2012) and 1000 in Tucker et al. (2010)].

2.3 Market-driven product family design

A market-driven approach in product family design aims to integrate market considerations with product family architecture design (Pirmoradi et al. 2014). In order to translate customer requirements into design requirements (including functional requirements), quality function deployment and its variant techniques were used (Simpson et al. 2012; Pirmoradi et al. 2014). Discrete choice analysis (Train 2003; Wassenaar and Chen 2003; Wassenaar et al. 2005) is a popular model in engineering design problems to map design attributes into market share estimation.

de Weck et al. (2003) proposed a methodology that determines the optimum number of product platforms to maximize product family profitability with simplifying assumptions. The methodology is divided into family-level (platform architecting) and variant-level (product optimization) design. First, market segments and corresponding market leaders should be identified. The number of market segments is set to be the maximum number of product platforms. Second, the design variable set, objective function, and demand equation for a single market segment needs to be established. Since each market segment is assumed to have a unique performance requirement, each segment represents each platform. Third, product architectures should be optimized for a given performance requirement, and the profit of the product family can be estimated. de Weck et al. (2003) assumed that all the necessary information of the first and second step is given so that the determination of number of platforms is the only decision variable in the family level (i.e., no architecture specification).

Kumar et al. (2009) developed market-driven product family design, which expands the demand modeling part of de Weck et al. (2003). First, the methodology starts from the creation of market segments. All the necessary information such as required performance, price, customer demographics, and competitors are identified. After that, a nested logit demand model (Train 2003) is built. The role of the demand model is to determine the market share of each market segment with specified product performance, customer demographics, and price. Second, models for product performance and costs need to be built. These models make trade-offs between cost and performance in the demand model. Third, optimal product specifications and number of platforms are identified to maximize the overall profits. Similar to the work of de Weck et al. (2003), product specifications for each segment were given as different constraints.

These market-driven approaches extend the scope of product family design by introducing a profit model. The number of product family architectures was considered as one of the design variables to maximize the profit function. However, information about market segments was assumed to be given instead of derived. Moreover, the profit model based on discrete choice analysis is static, which means a built model in the past can be used anytime in the future. This article relaxes the stationarity of profit modeling. Ma et al. (2014) and Ma and Kim (2014) showed that future trends could be captured from historical data using trend mining techniques, and incorporated in design problems.

3 Proposed approach: predictive, data-driven product family design

3.1 Overview

Figure 3 outlines the framework of PDPFD. There are two stages: individual product design stage and product family design stage. The individual product design stage involves the enterprise level and engineering level (Wassenaar and Chen 2003; Wassenaar et al. 2005; Chen et al. 2012). The enterprise level represents managerial level decision making for maximizing the expected profit with respect to the number and specifications of architectures as targets. The engineering level represents physical design with respect to engineering-level design variables (e.g., thickness and length of parts). The objective function consists of local objective functions (e.g., minimizing product's weight) and the deviation term for target matching (e.g., satisfying performance requirements). If the enterprise-level target is infeasible, then a new target should be explored. Once the individual product design stage decisions are made, the next step is to determine product family design. Based on the determined product variants, a decision-making process for scale-based product family design is explored. Scaling variables (i.e., the reduced design variables) of the architectures can be stretched or shrunk to satisfy the same objective function in engineering level, while common parameters remain constant. The common parameters constitute the product platform.

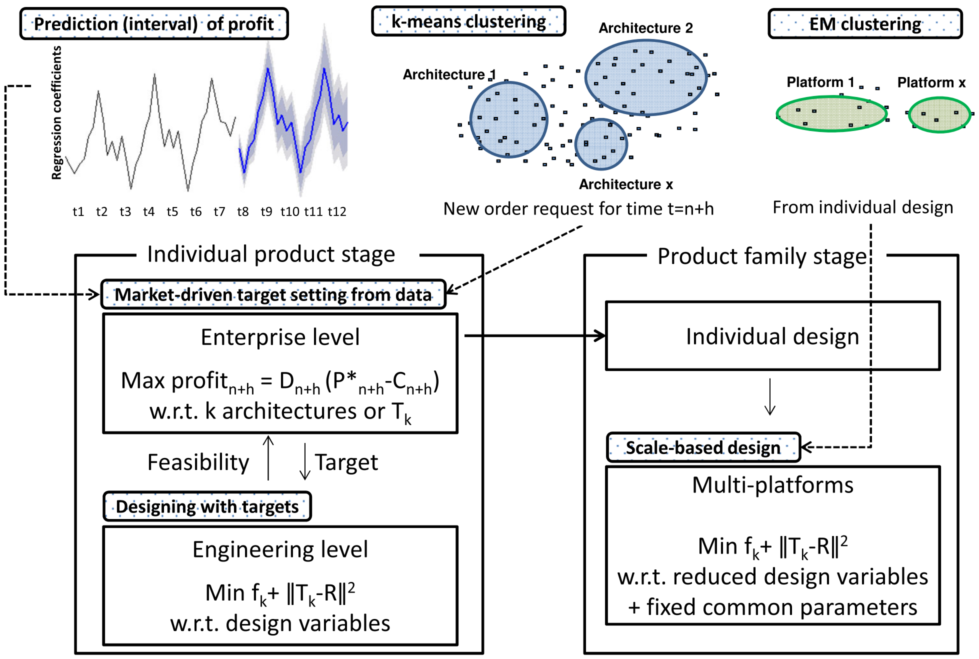
Three important tools are a market value prediction model with exponential smoothing for market considerations (Sect. 3.3), *k*-means clustering (MacQueen 1967; Witten and Frank 2005) for product family architecture candidates (Sect. 3.4.1), and expectation maximization clustering (Dempster et al. 1977) for multiple-platform design (Sect. 3.5). The first tool will capture a trend of customer preferences and uncertainties, the second tool will find the optimal number of architectures to minimize deviations between customer requirements and performance of architectures, and the last tool will figure out the possibility of multiple platforms.

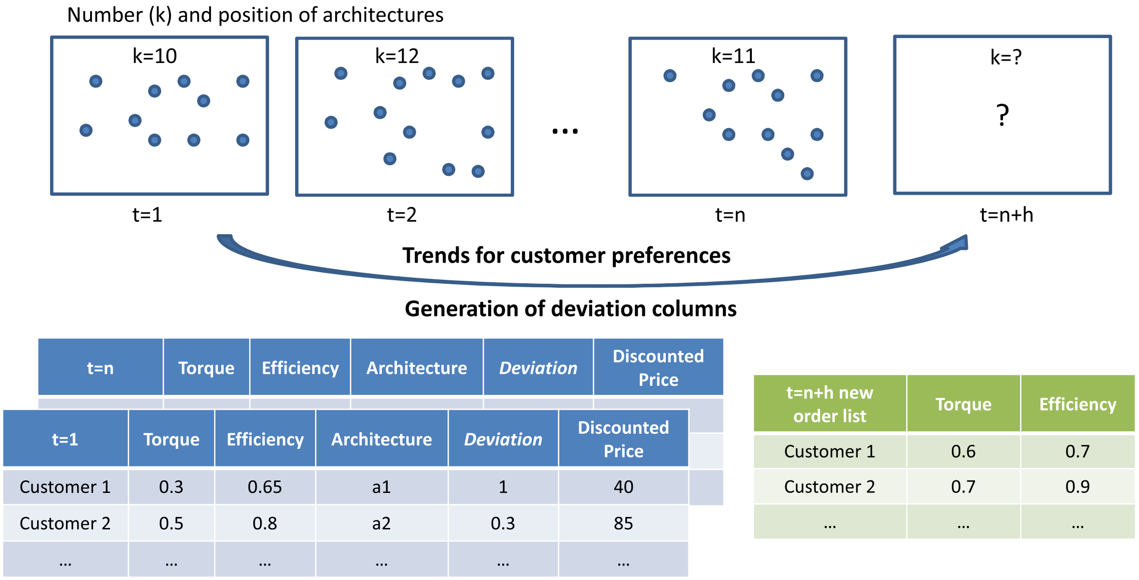
3.2 Data structure and assumptions

The main question in a data-driven model is how to represent data. Figure 4 shows the basic data structure. The index t represents discrete time, and data at $t = n$ indicate the current data. In the historical data set from $t = 1$ to $t = n$, transactional information is available, which is the set of data on product requirements (e.g., torque and efficiency), chosen product architectures (e.g., a1 and a2), and the discounted price that customers paid based on their utility for the chosen product architecture. Note that discounts can be applied if the product requirements cannot be matched. The goal is how to determine the position and number of product architectures at h time ahead (i.e., at $t = n + h$). Furthermore, the trend in customer preferences in historical data is captured and reflected in a profit function.

The transaction tables in Fig. 4 also show the generation of the deviation between what customers want and what products provide. By generating the deviation columns from product requirements and product architectures, the impact of increasing or decreasing product architectures can be investigated in terms of discounts.

Fig. 3 Overall framework of PDPFD




Number (k) and position of architectures
k=10
k=12
…
k=11
k=?
?
t=1
t=2
t=n
t=n+h
Trends for customer preferences
Generation of deviation columns

t=n	Torque	Efficiency	Architecture	*Deviation*	Discounted Price

t=1	Torque	Efficiency	Architecture	*Deviation*	Discounted Price
Customer 1	0.3	0.65	a1	1	40
Customer 2	0.5	0.8	a2	0.3	85
…	…	…	…	…	…

t=n+h new order list	Torque	Efficiency
Customer 1	0.6	0.7
Customer 2	0.7	0.9
…	…	…

Fig. 4 Data structure example

The availability and quality of data are critical in data-driven models. While public data (e.g., product specifications) can be used for product positioning problems (Lei and Moon 2015), the data set utilized in this study is transactional information, which can be found in company databases. Instead of directly analyzing real data sets, randomly generated data sets will be used to test the proposed model. Regardless of the type of data—real or synthetic, the proposed model would provide the most profitable clusters based on a profit model. Since the quality of data-driven models can be hugely affected by the quality of data, great efforts should be made for the preparation of input data sets. To improve the quality of data, data cleaning methods were adopted such as removing abnormality values and handling missing values (Witten and Frank 2005).

The basic assumptions are depicted in Fig. 5. The circles represent customers' requirements in terms of performance of products, and the rectangle shows the centroid of the cluster or the architecture. In the extreme case, seven

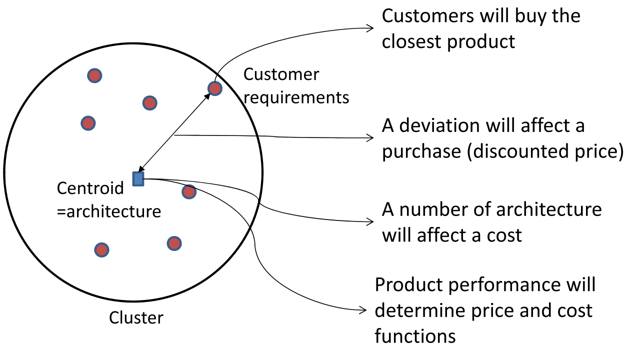

Customers will buy the closest product
Customer requirements
A deviation will affect a purchase (discounted price)
Centroid =architecture
A number of architecture will affect a cost
Product performance will determine price and cost functions
Cluster

Fig. 5 Basic assumptions of PDPFD

product architectures can be developed to satisfy all customers. Or, only one product architecture (the current figure) can serve as a single medium to embrace all the requirements if the customers can ignore the differences. It is assumed that customers will buy the product that is closer to their requirements in terms of the Euclidean distance. Basically, the performance of the product will determine price and cost functions. For example, key performances of notebook computers (e.g., memory, processors, screen size) determine notebook computer price and cost. In addition, the deviation or distance between a product architecture and customer requirements will affect a purchase in terms of the discounted price, and the increasing the number of architectures will increase the fixed costs.

Under the aforementioned assumptions, the result of the PDPFD framework can be used in a product design decision support system. No competing product is considered so that the impact of product brand is not investigated in this study. Well-defined market segments are not given so customer preferences data should be clustered. If market segments information is available, the market-driven approach in Sect. 2.3 can be used with either a static profit model (Kumar et al. 2009) or a predictive profit model (Sects. 3.3, 3.4). Also, product performance in the customer requirement space is limited to continuous variables. The proposed model attempts to model the trend of customer preferences in the market and use the trend and prediction intervals for the product design decision support system. Since the predicted model is designed to be used for a short forecasting horizon (e.g., one-step-ahead short prediction such as 3 and 6 months later), the evolution of a product family and technology shifts are not considered.

3.3 Market value prediction for a profit model

In order to build a predictive profit model (Sect. 3.4.1), market value prediction will be discussed with prediction intervals (i.e., lower and upper bounds). Most of all, significant factors for prices and costs should be identified. Subject matter experts are helpful to manage the list of candidate factors, and stepwise regression procedures can be applied to find the significant factors in a stepwise manner.

3.3.1 Market value prediction with regression coefficients

Prediction of product prices with regression coefficients was proposed by Rutherford and Wilhelm (1999) for a notebook computer (hereinafter RW model). Recently, this model was revalidated with a more mature notebook market (DesAutels and Berthon 2011). Though the RW model was validated with a notebook computer, it was also used to relate demand, price, and the features that comprise a general product (Wilhelm et al. 2003; Damodaran and Wilhelm 2005) and suggested as a possible prediction method of product design (Kwak and Kim 2013).

The RW model consists of two phases. Phase 1 fits a linear regression model to each time series (e.g., regression model for each month for monthly data). Phase 2 uses linear trend analysis of regression coefficients to capture a trend over time. Then, future market values of target products can be predicted with given features. From publicly available data (notebook price data), the model predicted the rate of price erosion of a notebook computer up to 7 months ahead within 10 % error. The RW model is used for the base case of price prediction.

The main difference between the RW model and the predictive model in PDPFD (hereinafter PDPFD model) is that the PDPFD model uses exponential smoothing models (Hyndman et al. 2008) at Phase 2, which is more flexible (e.g., linear trend model can be considered one of exponential smoothing models) and provides prediction intervals for prediction uncertainty. The general form of the regression model in this study is given in Eq. (1):

$$P_t = \beta_{0t} + \sum_{i \in A} \beta_{it} a_{it} + \theta_t, \quad \text{for} \quad t = 1, \ldots, n \tag{1}$$

where P_t is the price or market value of a product at discrete time t, β_{0t} is the intercept, i is the index for levels or alternatives of factors (product features), A is the set of factors, β_{it} is the regression coefficients of factor i, a_{it} is the measurement of factor i, and θ_t is the random error. Note that the price is determined by product features, but the discounted price considers one more factor, deviation in Sect. 3.4. It does not need to be linear, but homogeneous forms of regression models are required over time (i.e., linear, squared, and cubic). Linear regression is usually adopted as a general model with the following assumptions: linear relationship between factors and response, independent factors and random errors, and random error with constant variance.

The next step is to trace the trend of regression coefficients β_{it}, which is considered as customer preferences for product features over time. Exponential smoothing based on innovations state space models (Hyndman et al. 2008) is proposed to model the time series. Equations (2) and (3) show generalized state space equations for β_{it}:

$$\beta_{it} = w(x_{i(t-1)}) + r(x_{i(t-1)})\epsilon_{it} \tag{2}$$

$$x_{it} = f(x_{i(t-1)}) + g(x_{i(t-1)})\epsilon_{it} \tag{3}$$

where β_{it} is the observed value at time t, x_{it} is the state vector which contains unobserved components such as the level, trend, and seasonality of a time series, $w()$ and $r()$ are scalar functions, $f()$ and $g()$ are the vector functions, and ϵ_{it} is the white noise process with variance σ^2. The white noise process has zero mean, constant and finite variance, and uncorrelated values. For a succinct notation, index $i \in A \cup \{0\}$ is used in Eqs. (2) and (3).

By combining Eqs. (1), (2), and (3), the following state space-based price equations are formulated:

$$\begin{aligned} P_t = & \left[w(x_{0(t-1)}) + r(x_{0(t-1)})\epsilon_{0t}\right] \\ & + \sum_{i \in A} \left[w(x_{i(t-1)}) + r(x_{i(t-1)})\epsilon_{it}\right] a_{it} + \theta_t \end{aligned} \tag{4}$$

$$x_{it} = f(x_{i(t-1)}) + g(x_{i(t-1)})\epsilon_{it} \tag{5}$$

Finally, estimation of the price at h time ahead is formulated as follows:

$$\hat{P}_{t+h} = \hat{\beta}_{0(t+h|t)} + \sum_{i \in A} \hat{\beta}_{i(t+h|t)} a_{i(t+h)} \tag{6}$$

where $\hat{\beta}_{t+h|t}$ represents the forecast of $\hat{\beta}_{t+h}$ based on all the data up to time t.

There are a total of 30 exponential smoothing models classified based on trend, seasonality, and error in additive, multiplicative or mixed ways. Hyndman et al. (2008) provided details of the classifications. The automatic forecasting method (Hyndman and Khandakar 2008) is adopted to determine all the necessary parameters and the best model. The first step is to apply all the 30 exponential smoothing models, and estimate initial states and parameters using maximum likelihood estimation based on the innovations representation of the probability density function [refer to Eq. (8)]. The next step is to choose the best model according to an information criterion: Akaike's information criterion (AIC), corrected Akaike's information criterion (AICc), or Bayesian information criterion (BIC).

3.3.2 *Prediction interval of market value*

In the previous section, point forecasting of the time series β_{it} was discussed, which provides an average market value of products. In order to consider the uncertainty in market trends, prediction intervals in time series prediction are used as well.

Three sources of uncertainty were identified in forecasting a future value (Hyndman et al. 2008): 1. selected model, 2. estimated parameters and initial states, 3. future innovations: $\epsilon_{i(n+1)}, \ldots, \epsilon_{i(n+h)}$. If it is assumed that the uncertainties from the first and second sources can be minimized by applying the automatic forecasting method in Sect. 3.3.1, the uncertainty in the future innovations is the only source that needs to be considered for prediction intervals.

If the initial state value x_{i0} is known, the innovation ϵ_{it} is a one-step-ahead prediction error. The conditional expectation (Hyndman et al. 2008), which is also the one-step-ahead point forecast $\hat{\beta}_{it|(t-1)}$, is given by:

$$\begin{aligned} E\left(\beta_{it}|\beta_{i(t-1)}, \ldots, \beta_{i1}, \beta_{i0}\right) & = E\left(\beta_{it}|x_{i(t-1)}\right) = \hat{\beta}_{it|(t-1)} \\ & = w(x_{i(t-1)}) \end{aligned} \tag{7}$$

The probability density function (Hyndman et al. 2008) for β_i is also given as a function of innovations ϵ_{it} in Eq. (8):

$$P(\beta_i|x_{i0}) = \prod_{t=1}^{n} P\left(\beta_{it}|x_{i(t-1)}\right) = \prod_{t=1}^{n} P(\epsilon_{it})/r(x_{i(t-1)}) \tag{8}$$

Then, the recursive relationships can be summarized as follows:

$$\hat{\beta}_{it|(t-1)} = w(x_{i(t-1)}) \tag{9}$$

$$\epsilon_{it} = (\beta_{it} - \hat{\beta}_{it|(t-1)})/r(x_{i(t-1)}) \tag{10}$$

$$x_{it} = f(x_{i(t-1)}) + g(x_{i(t-1)})\epsilon_{it} \tag{11}$$

Therefore, h time-ahead prediction of β_{it} requires only $\epsilon_{i(n+1)}, \ldots, \epsilon_{i(n+h)}$.

In order to obtain prediction distributions, a simulation approach (Hyndman et al. 2008) is adopted, which is simple and can cover all the 30 exponential smoothing models. The simulation approach simulates sample paths or observations β_{it} with initial states x_{it} from the chosen model. The remaining unknown values are future innovations ϵ_{it}, and they can be obtained from a random number generator with an appropriate distribution. An approximate $100(1-\alpha)\,\%$ prediction interval for forecast horizon h is given by the $\frac{\alpha}{2}$ and $1 - \frac{\alpha}{2}$ quantiles of $\beta_{i(t+h)|t}$:

$$\hat{P}^{\frac{\alpha}{2}}_{t+h} = \hat{\beta}^{\frac{\alpha}{2}}_{0(t+h|t)} + \sum_{i \in A} \hat{\beta}^{\frac{\alpha}{2}}_{i(t+h|t)} a_{i(t+h)} \tag{12}$$

$$\hat{P}^{(1-\frac{\alpha}{2})}_{t+h} = \hat{\beta}^{(1-\frac{\alpha}{2})}_{0(t+h|t)} + \sum_{i \in A} \hat{\beta}^{(1-\frac{\alpha}{2})}_{i(t+h|t)} a_{i(t+h)} \tag{13}$$

For example, 90 % of the prediction interval of a market value is given by $\hat{P}^{0.05}_{t+h}$ and $\hat{P}^{0.95}_{t+h}$. The prediction interval should be interpreted as the average prediction success instead of any single case. In other words, 90 % of the time, the real market value will fall within the bounds of intervals.

3.3.3 Performance test for predictive model in PDPFD

In this section, the prediction capabilities of the PDPFD model and the RW model in Sect. 3.3.1 are compared. The hypotheses are (1) the proposed model can provide a similar level of predictive accuracy to the RW model when data have a simple trend (trend of regression coefficients) and (2) the proposed model can predict future values more accurately than the RW model when data have complex patterns (e.g., trend and cycle of regression coefficients).

Data sets with a simple trend and complex patterns were generated randomly with the description of the generation procedures. Each data set contains three factors and one class variable (response or dependent variable) with 100 instances. The goal is to predict one-step-ahead class values using previous data sets. There were a total of 30 data sets from $t = 1$ to $t = 30$, and the prediction results were collected from $t = 11$ to $t = 30$ (i.e., 20 time periods).

As a performance measure, mean absolute error (MAE) was selected as given by Eq. (14):

$$\text{Mean absolute error} = \frac{|b_1 - d_1| + \cdots + |b_m - d_m|}{m} \tag{14}$$

where $b_1, b_2, \ldots, b_m$ are the predicted class values and $d_1, d_2, \ldots, d_m$ are the actual class values.

3.3.3.1 Data with trend For the first hypothesis, the following data generation procedure was applied: (1) the value of each factor was randomly chosen from 1 and 5 for each of the 30 data sets, (2) the base regression coefficients (i.e., $t = 1$) for three factors were randomly chosen between 30 and 40, (3) one of possible trends (increasing 1.5 or decreasing 1.5) was randomly selected and applied to each coefficient from $t = 2$ to $t = 30$, (4) the class values were generated based on the values of the factors and the regression coefficients with some additional randomness, (5) the regression analysis was applied to the generated data sets, and (6) the identified values of regression coefficients were used for predictive modeling. Due to the randomness in step 4, the trend of regression coefficients is not exactly 1.5.

The result of 20 MAEs (each MAE represents the average of absolute errors for 100 instances) showed that the prediction accuracies of the two models were almost identical (Mann–Whitney test, $\alpha = 0.05$, p value $=$ 0.98). Both models predicted one-step-ahead values with less than 1 % error.

3.3.3.2 Data with trend and cycle For the second hypothesis, the same data generation procedure was applied except for (2) the base regression coefficients for three factors were randomly chosen with cyclical patterns (e.g., $t = 1$ between 30 and 40, $t = 2$ between 40 and 50, $t = 3$ between 50 and 60, $t = 4$ between 60 and 70) and (3) one of possible trends (increasing 1.5 or decreasing 1.5) was randomly selected and applied to the regression coefficients of each cycle from $t = 5$ to $t = 30$. As a result of this procedure, similar patterns were repeated for every four-time steps (i.e., cycles).

Table 1 shows the comparison result from both models. Since the RW model depends only on the trend line for the prediction, when data have complex patterns, the proposed model provides a higher prediction accuracy (Mann–Whitney test, $\alpha = 0.05$, p value $= 0$).

The strength of the PDPFD model comes from the fact that both linear and nonlinear forms of formulations can be used, and the trend of coefficients can be captured in an automatic way. Moreover, the PDPFD model can provide prediction intervals (e.g., forecast value is 60.3 with 80 % prediction interval of 59.8 and 60.8), which can show the uncertainty of market trend (customer preferences) in Sect. 3.3.2. These are characteristics that the RW model cannot achieve.

Now, a general model of predicted market values is formulated. In the next section, the model will be combined with a profit model.

3.4 Individual product design stage

In the individual product stage, there are two levels: enterprise level and engineering level (Wassenaar and Chen 2003; Wassenaar et al. 2005; Chen et al. 2012). As shown in Fig. 3, the market-driven target setting from customer preference data is implemented at the enterprise level, and engineering design with the target is realized at the engineering level.

3.4.1 Enterprise level

At the enterprise level, the objective is to maximize the expected profit while satisfying other constraints:

Maximize

$$\Pi_{n+h}(T_k) = D_{n+h}(P^*_{n+h} - C_{n+h}) \tag{15}$$

Subject to:

$$g(T_k) \leq 0, \quad h(T_k) = 0 \tag{16}$$

where Π_{n+h} is the economic profit at time $n + h$ (h time ahead), T_k is the set of target values (i.e., product architectures with k number), D_{n+h} is the demand or number of orders, P^*_{n+h} is the discounted price or sale price, C_{n+h} is the cost, $g()$ are inequality constraints (e.g., range of k or minimum profit), and $h()$ are equality constraints (e.g., exact number of k).

Table 1 Comparison between RW and PDPFD model over 30 data sets (MAE)

	t11	t12	t13	t14	t15	t16	t17	t18	t19	t20	t21	t22	t23	t24	t25	t26	t27	t28	t29	t30
RW	25.5	142.6	213.5	49.5	25.5	149.7	201.9	49.4	26.2	162.3	183.9	46.1	27.4	156.8	180.9	44.7	28.7	161.4	177	44.2
PDPFD	2.7	3.4	2.8	2.9	3.1	2.6	2.8	3.0	3.0	3.4	2.9	3.0	3.2	3.1	3.0	3.0	2.8	2.6	2.9	2.6

MAE mean absolute error

Equations (17) and (18) show the general models for the discounted price and cost based on the assumptions in Sect. 3.2:

$$P^{*}_{n+h} = f(T_k, d) \tag{17}$$

$$C_{n+h} = f(T_k, k) \tag{18}$$

where *f*() is a scalar function, *d* is the deviation in Eq. (20), which represents the impact of deviations between customers' requirements and product architectures, and *k* is the number of architectures, which represents fixed costs to increase the number of architectures. In order to apply regression analysis, it is assumed that historical data have $k \geq 2$. The data for the cost model at $t = n + h$ are assumed to be available to manufacturers, but the price model at $t = n + h$ should be predicted as discussed in Sect. 3.3. If cost-related data at $t = n + h$ are not available, the same technique used in the price model should be applied.

To solve this problem with large-scale data, a two-step approach is proposed. The proposed process starts from identifying maximum *k*. Then, find each $T_2, \ldots, T_k$ that minimizes deviations from customer requirements. Next, among $T_2, \ldots, T_k$, determine the best one by considering profit prediction along with its prediction intervals at the target time. Note that since this is product family design, more than two product variants (T_2) will be realized.

Step 1 set maximum *k* or number of architectures, and calculate a deviation for all *k* centroids by applying *k*-means clustering

Step 2 calculate profits for all *k* architectures with prediction intervals, and set the target T_k that generates maximum profit

The determination of maximum *k* in this algorithm depends on designers. In general, it is almost impossible for designers to decide the number from large-scale data. However, the maximum number of architectures (*k*) can be estimated not purely by data but jointly by manufacturer's capability and managerial decisions (e.g., the number of production lines allow only a certain number of product variants). If the maximum number *k* cannot be estimated, *k* should be increased enough to the point where no more improvement is possible in the case of a concave profit function. Tucker et al. (2010) used the X-means clustering algorithm (Pelleg and Moore 2000) to automatically select the optimal *k* for product family architecture design, but the maximum *k* should be provided by designers.

The *k*-means clustering algorithm (MacQueen 1967; Witten and Frank 2005) is used since it is simple and effective. The Euclidean distance assumption works well with the *k*-means algorithm. The clustering algorithm partitions a given data set into a fixed number of clusters *k*. It aims at minimizing the objective function, which is within cluster sum of squared errors (SSE) as shown in Eq. (19):

$$f = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - c_i\|^2 \qquad (19)$$

where $x = (x_1, x_2, \ldots, x_n)$ is a set of customer requirements, $C_i = (C_1, C_2, \ldots, C_k)$ is a set of clusters, and c_i is the centroid of cluster C_i (which is the arithmetic mean of points in C_i). The deviation d is defined in Eq. (20) based on Eq. (19):

$$d = \frac{\sum_{i=1}^{k} \sum_{x \in C_i} \|x - c_i\|^2}{n} \qquad (20)$$

The iterative process of the k-means algorithm starts by specifying the number of clusters (k). Then, k points are chosen randomly as cluster centers (c_i) and all instances (x) are assigned to the closest cluster centers in accordance with the Euclidean distance. After the assignment, new cluster centers are recalculated as means. This process is repeated until the same instances are assigned to the same clusters.

The k-means clustering algorithm has some disadvantages as follows. First, it is necessary to specify the number of cluster k by designers. It was discussed above how to constrain the k for product family architecture design. Second, its performance can be significantly diminished with high-dimensional data. New k-means clustering algorithm with high-dimensional data was proposed by Sun et al. (2012), and various dimensionality reduction techniques were discussed in the literature such as principle components analysis (Witten and Frank 2005), kernel trick (Witten and Frank 2005), data compression (Chan et al. 2012), feature selection (Witten and Frank 2005; Tucker et al. 2010). If data are really high-dimensional (e.g., DNA, tweets), special clustering techniques should be applied (Kriegel et al. 2009). Third, the algorithm converges to local minima. Initial starting points can affect the result, and repeating the algorithm with different starting points is required. Note that these disadvantages are common in any clustering algorithm.

3.4.2 *Engineering level*

The engineering-level problems can be stated as follows: find a design solution that minimizes the deviations between design targets from Sect. 3.4.1 and actual responses while satisfying design constraints:
Minimize

$$f_k + \|T_k - R\|_2^2 \qquad (21)$$

Subject to:

$$g(T_k) \le 0, \quad h(T_k) = 0 \qquad (22)$$

where f_k is the local product design objective function(s) (e.g., minimize weights), T_k is the target vector cascaded down from the enterprise level, and the R is the response vector obtained from the analysis model $r(x)$ (e.g., engineering-level analytical models to calculate the response of the targets).

3.5 Product family design stage

As discussed in Sect. 2.1, multiple-platform scale-based product family design is studied with known common parameters. The goal in this stage is to find clusters of values under each common parameter for exploring the possibility of multiple platforms while maintaining the performances of products. The clustering is based on similarity without the prior knowledge of cluster numbers. There are a few clustering techniques to allow this task: expectation maximization (EM) (Dempster et al. 1977; Witten and Frank 2005; Do and Batzoglou 2008) and X-means clustering (Pelleg and Moore 2000). Both of them are extended versions of the k-means clustering method, which is used in the individual product design stage. Based on empirical test results for the product family design stage, the EM algorithm is used in this stage.

The EM clustering algorithm is a generalization of maximum likelihood estimation when the given data set is incomplete or there are unobserved latent variables. The goal is to estimate parameter $\hat{\theta}$ that maximizes the log-likelihood $\log P(x, z; \theta)$, where x is the observed variable and z is the latent variable. The EM iteration alternates between the expectation (E) step, which calculates a probability distribution over possible completions of missing data with the initial guess of parameters, and the maximization (M) step, which re-estimates the parameters using these completions. Do and Batzoglou (2008) provided a simple coin-flipping example of the EM algorithm.

In the clustering task, the unobserved latent variables are the assignments of observed values to clusters, and the parameters are the means and covariance matrices of the selected distributions representing each cluster. Therefore, the E-step calculates the cluster probabilities with the guessed parameters. The M-step calculates the parameters (i.e., cluster means and covariances) by maximizing the likelihood of the distributions.

Based on the result of the EM clustering, multiple values can be allowed for each common parameter. Whether one constant (i.e., single platform) or multiple constant values (i.e., multiple platforms) are used for common parameters depends on designers. Finally, the engineering-level optimization problem should be re-solved with respect to reduced design variables (i.e., scaling variables) with fixed common parameters.

4 Case study: universal motor family design

4.1 Background and data generation

The design of a universal motor family (Simpson et al. 2001) is used to demonstrate the effectiveness of the proposed model and provide a comparison of the proposed model and a clustering-based model (Tucker et al. 2010). Universal electric motors are the most common components in power tools (e.g., electric saws, drills, and drivers) and in household appliances (e.g., blenders, vacuum cleaners, and washing machines). Figure 6 shows the schematic of a universal motor. There are eight design variable as inputs in Table 2. A mathematical model provided by Simpson et al. (2001) returns four performance outputs: power (P), torque (T), mass (M), and efficiency (η) of motors as a function these eight design variables. The objective of this case study is designing a family of universal electric motors that maximizes the expected profit for the next market trend (customer preferences) based on accumulated data.

Three data sets (*data set 1*, *data set 2* and *data set 3*) were generated using the generation procedure in Sect. 3.3.3 (data with trend) with manually generated new orders. Figure 7 shows the new orders in *data set 1* and *data set 2*, which needs to be clustered. Each data set contains twelve historical (6-month interval) transactional data, one new order data, and one cost-related data. Each datum has one million instances (i.e., a total of 14 million instances for each data set). The embedded artificial trends in *data set 1* are shown in Table 3. For example, the coefficients of efficiency have an increasing trend over time in comparison with other factors, which indicates customers pay more attention to the factor as time passes. For the remaining sections, only *data set 1* is used for discussion except for the comparative study in Sect. 4.3.3.

4.2 Profit modeling

Two key factors (torque and efficiency) were assumed to be identified for the estimation of discounted price and cost functions. The discounted price and cost functions at one step ahead (i.e., 6 months later) were formulated in Eqs. (23) and (24):

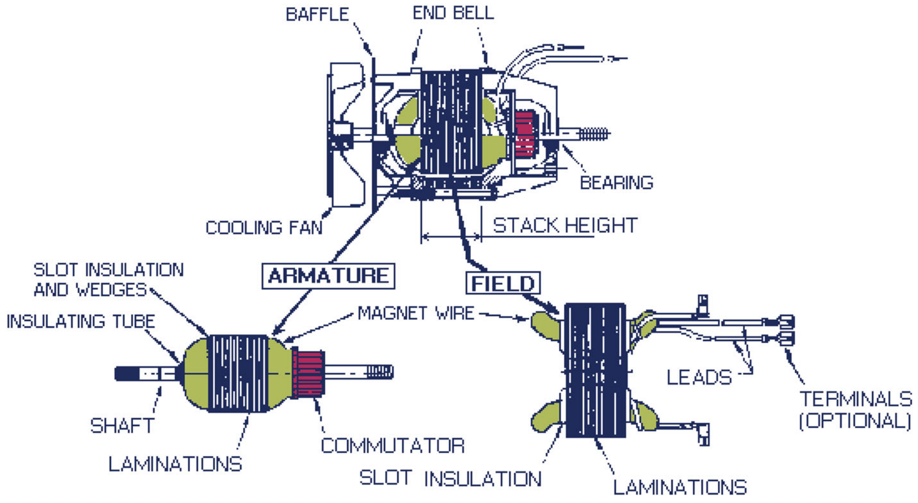


Fig. 6 Universal motor schematic (*Source*: Simpson et al. 2001)

Table 2 Design variables and ranges of universal motors

Variable	Definition	Range
N_c	Number of wire turns on the motor armature	$100 \leq N_c \leq 1500$ turns
N_s	Number of wire turns on each field pole	$1 \leq N_s \leq 500$ turns
A_{wa}	Cross-sectional area of the armature wire	$0.01 \leq A_{wa} \leq 1\,\text{mm}^2$
A_{wf}	Cross-sectional area of the field wire	$0.01 \leq A_{wf} \leq 1\,\text{mm}^2$
r	Radius of the motor	$0.01 \leq r \leq 0.1\,\text{m}$
t	Thickness of the motor	$0.0005 \leq t \leq 0.1\,\text{m}$
I	Current drawn by the motor	$0.1 \leq I \leq 6.0$ A
L	Stack length	$0.0566 \leq L \leq 10\,\text{cm}$

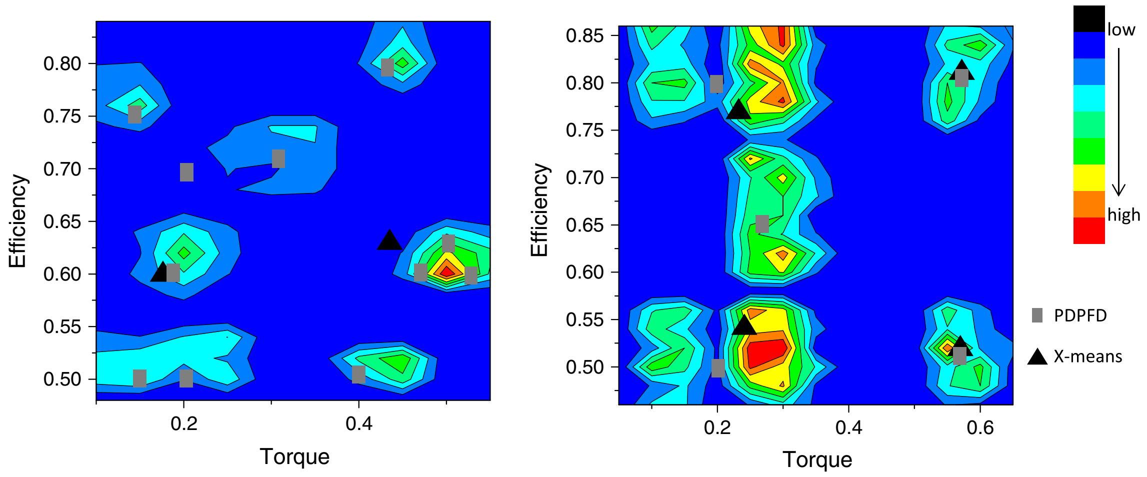


Fig. 7 New orders in data set 1 (*left*) and data set 2 (*right*)

$$\hat{P}^*_{n+h} = \beta_{0(n+1)} + \beta_{1(n+1)} \sum_{i=1}^{k} a_{1i} + \beta_{2(n+1)} \sum_{i=1}^{k} a_{2i} + \beta_{3(n+1)} d \tag{23}$$

$$\hat{C}_{n+1} = \gamma_{0(n+1)} + \gamma_{1(n+1)} \sum_{i=1}^{k} a_{1i} + \gamma_{2(n+1)} \sum_{i=1}^{k} a_{2i} + \gamma_{3(n+1)} k \tag{24}$$

where a_1 is the torque, a_2 is the efficiency of a universal motor, d is the deviation in Eq. (20), and k is the number of product architectures. Since the demand (D_{n+1}) is given as the customers' new orders, the profit model at time $n+1$ is formulated by Eq. (15). In order to maximize the profit, both the deviation and the number of architectures should be minimized. However, these two components are conflicting each other. When the number of architectures is increased, the deviation is decreased accordingly or vice versa. Both Eqs. (23) and (24) use the constant impact of the deviation and the number of product architectures, which can cause errors for estimation.

Table 3 shows the historical regression coefficients of the discounted price fitted for historical data. The exponential smoothing was applied to model each time series (e.g., Torque from $t = 1$ to $t = 12$) using the *forecast* package (Hyndman and Khandakar 2008) in R (R Development Core Team 2008). The mean column of Table 4 contains the point estimation of one-step-ahead prediction (i.e., $t = 13$). The automatic forecasting method in Sect. 3.3.1 provided required parameters and initial states. Table 4 also shows lower (i.e., lo80 and lo95) and higher (i.e., hi80 and hi95) bounds of 80 and 95 % prediction intervals based on the simulation method in Sect. 3.3.2. Instead of having the assumption of normally distributed errors, re-sampled errors or bootstrapping techniques were used to simulate future values. The cost model at $t = 13$ is provided in the right side of Table 4.

4.3 Individual product design stage

4.3.1 Enterprise level

It was assumed that the maximum number of architectures was determined as 15 based on the manufacturer's capability and production environment. Positions of product architectures that minimize deviation errors for the one million new orders were identified using the k-means algorithm in Weka (Hall et al. 2009). Since the k-means algorithm is the local optimizer, multiple seed values (10 different values) were used to get the k best clusters. Figure 7 shows the result with $k = 11$ (left) and $k = 5$ (right).

The profit model in Eq. (15) at $t = n + 1$ (i.e., $t = 13$) is now available. By utilizing Eqs. (6), (12), and (13), profits for mean, 80, and 95 % prediction intervals can be calculated as shown in Table 5. The top 4 ks were selected according to their expected profits. Though the selection of k is dependent on designers, the important fact is that the prediction intervals give the uncertainties of the predicted profit model. For example, T_{11} can have the profit range from 0.47 to 7.59 million dollars, while T_{15} can have the range from -1.16 to 8.48 million dollars with a 80 % prediction interval. It was assumed that the designer chose 11 architectures (T_{11}) with the expected profit of 4.03 million dollars. Then, the target T_{11} in Fig. 7 (left) was cascaded down to the engineering level.

4.3.2 Engineering level

The local objective function f_k [from Eq. (21)] in this case study is the mass function of a universal motor. A mathematical universal motor model (Simpson et al. 2001) is used as the analysis model $r(x)$ in Eq. (21). Therefore, the objective function is to minimize the mass of motors and

Table 3 History of regression coefficients for discounted price

	$t = 1$	$t = 2$	$t = 3$	$t = 4$	$t = 5$	$t = 6$	$t = 7$	$t = 8$	$t = 9$	$t = 10$	$t = 11$	$t = 12$
Torque	34.99	34.50	34.20	34.00	33.50	33.09	32.79	32.70	32.49	32.19	31.79	31.19
Efficiency	22.01	22.49	22.8	23.00	23.50	23.60	23.60	23.60	23.80	24.79	25.49	26.29
Deviation	−18.00	−18.10	−18.20	−18.3.	−18.50	−18.70	−19.10	−19.10	−19.29	−19.49	−19.69	−19.89
Intercept	−0.0077	0	0	0	−0.0002	−0.0002	−0.0005	−0.0003	0	0.0001	0.0014	0.0001

deviations between the target T_{11} and the response R while satisfying design constraints in Table 6.

The Generalized Reduced Gradient (GRG) algorithm in Excel was used to solve this problem. Table 7 shows the engineering-level optimization result with T_{11} from the enterprise level (i.e., T and η column).

4.3.3 Comparative study

As shown in the previous sections, the proposed algorithm combines the clustering-based and market-driven approaches together for the target setting of the individual product design stage. In this section, PDPFD and a previous clustering-based approach (Tucker et al. 2010) are compared to validate the performance of the proposed algorithm.

The clustering-based method (Tucker et al. 2010) used the X-means clustering algorithm (Pelleg and Moore 2000) to design aerodynamic particle separators. Out of 1000 data points, the X-means clustering found five cluster centroids (i.e., architectures) based on the Bayesian information criterion (BIC) (Pelleg and Moore 2000). From these five architectures (with the maximum BIC score), five product variants could be realized. However, the clustering-based method calculated the production cost after determining the five product architectures. In contrast, the proposed algorithm considers the expected profit while simultaneously determining the product architectures. By design, PDPFD can generate profits that are equal to or greater than profits from the clustering-based method, while BIC scores can be reduced.

Data sets 1, 2, 3 were utilized for this comparative study. The X-means clustering algorithm in Weka (Hall et al. 2009) was used with the minimum (2) and maximum (15) number of architectures. Table 8 shows both results from the proposed and clustering-based methods. When PDPFD generated more architectures, the averages of within cluster sum of squared errors (SSE) were lower than that of the clustering-based method. The clustering-based method generated lower expected profit at the end because it maximized the BIC score first, and then the profit was calculated sequentially with the determined number of product architectures. The PDPFD algorithm explored all the k values (e.g., $k = 2$–15) and determined the best one by comparing expected profits. This shows the benefit of the combination of clustering-based and market-driven combined approaches in product family architecture design as introduced in Figs. 1 and 2.

4.4 Product family design stage

In this section, multiple-platform scale-based product family design is conducted with known common parameters from Simpson et al. (2001) [radius of the motor (r) and

Table 4 Regression coefficients for discounted price and cost at $t = 13$

For discounted price	Mean	lo80	hi80	lo95	hi95	For cost	Mean
Torque	30.86	30.68	31.03	30.59	31.13	Torque	26.0
Efficiency	27.07	26.63	27.50	26.40	27.73	Efficiency	24.8
Deviation	−20.10	−20.14	−20.06	−20.16	−20.04	*k*	2.5
Intercept	0.00027	−0.00262	0.00316	−0.00414	0.00469	Intercept	0

Table 5 Architecture rankings based on prediction intervals of profit

		Mean	lo80	hi80	lo95	hi95
Rank	The best	11/4.03	11/0.47	15/8.48	5/−1.01	15/11.04
[*k*/profit ($ MM)]	Second	15/3.66	5/−0.15	11/7.59	11/−1.41	11/9.47
	Third	13/2.68	7/−0.27	14/6.80	4/−1.48	14/9.14
	Fourth	14/2.40	6/−0.72	13/6.79	7/−1.48	13/8.89

k is the number of architectures

Table 6 Design constraints for universal motors

Name	Constraint
Magnetizing intensity (H)	$H \leq$ 5000 A · turns/m
Feasible geometry	$t < r$
Power (P)	$P = 300$ W
Mass (M)	$M \leq$ 2.0 kg

thickness of the stator (t)]. Based on the result (i.e., 11 motors) from the individual product design stage, the EM clustering in Weka (Hall et al. 2009) was applied to find clusters within the two common parameters. Two clusters were identified for the radius of the motor (r). Then, the engineering-level optimization problem was resolved with respect to the six free design variables with the two fixed common parameters (i.e., r and t). Table 9 shows the result of the optimization problem, which indicates two different platforms based on r and t (i.e., 4.74/2.21 and 2.21/2.21) shared by motors. The average weight of the motor family was increased by 30.2 % (from 0.86 to 1.12 kg), but all weight constrains were satisfied (i.e., less than 2 kg).

5 Closing remarks and future work

This article addresses how to determine optimal product family architectures with customer preference data. The proposed model expands clustering-based approaches to

Table 7 Universal motor specifications and performance responses

Motor no.	Product specifications (design variables)								Responses			
	N_c	N_s	A_{wf} (mm^2)	A_{wa} (mm^2)	I (A)	r (cm)	t (mm)	L (cm)	T (Nm)	η (%)	P (W)	M (kg)
1	998	105	0.476	0.347	3.72	3.05	2.73	2.34	0.30	70	300	0.984
2	998	105	0.430	0.416	5.25	4.91	2.44	1.69	0.40	49.8	300	0.809
3	998	105	0.431	0.467	3.81	4.57	2.41	1.58	0.20	68.5	300	0.637
4	997	36	0.149	0.149	5.22	1.47	1.47	1.88	0.10	50.0	300	0.218
5	997	75	0.346	0.346	4.24	2.51	2.51	4.49	0.49	61.6	300	1.294
6	997	101	0.560	0.560	3.29	2.62	2.62	4.50	0.41	79.4	300	1.821
7	995	61	0.255	0.255	3.47	1.67	1.67	2.26	0.10	75.3	300	0.406
8	995	72	0.335	0.334	4.41	2.49	2.49	4.46	0.50	59.3	300	1.252
9	995	53	0.213	0.213	4.35	1.82	1.82	2.63	0.17	60.0	300	0.443
10	995	45	0.199	0.199	5.15	1.82	1.82	2.62	0.2	50.7	300	0.422
11	995	70	0.319	0.319	4.41	2.42	2.42	4.23	0.45	59.3	300	1.126

Table 8 Result of comparative study

	k	Average SSE	BIC	Cost ($ MM)/k	Revenue ($ MM)/k	Expected profit ($ MM)
Data set 1						
PDPFD	11	0.003	−933	25.76	26.12	4.03
Clustering-based method	2	0.104	−907	25.17	24.83	−1.57
Data set 2						
PDPFD	5	0.021	−934	28.56	28.65	0.42
Clustering-based method	4	0.032	−795	27.00	26.71	−1.10
Data set 3						
PDPFD	2	0.096	−781	27.50	124.50	0.194
Clustering-based method	4	0.025	−716	30.25	78.44	0.192

k is the number of architectures

Table 9 Universal motor family design with fixed r and t

Motor no.	Product specifications (design variables)								Responses			
	N_c	N_s	A_{wf} (mm^2)	A_{wa} (mm^2)	I (A)	r (cm)	t (mm)	L (cm)	T (Nm)	η (%)	P (W)	M (kg)
1	1229	54	0.195	0.195	5.21	2.21	2.21	0.75	0.30	70	300	1.003
2	1227	116	0.475	0.475	5.25	4.74	2.21	1.31	0.40	49.8	300	1.975
3	1196	159	0.562	0.562	3.81	4.74	2.21	0.93	0.20	68.5	300	1.999
4	1437	54	0.220	0.220	5.21	2.21	2.21	0.64	0.10	50.0	300	0.346
5	1437	67	0.412	0.411	4.23	2.21	2.21	3.86	0.49	61.6	300	1.329
6	1050	86	0.597	0.597	3.29	2.21	2.21	5.69	0.41	79.4	300	1.894
7	1050	81	0.279	0.278	3.47	2.21	2.21	1.32	0.10	75.3	300	0.462
8	1050	64	0.354	0.353	4.41	2.21	2.21	5.18	0.50	59.3	300	1.262
9	1050	65	0.223	0.222	4.35	2.21	2.21	1.78	0.17	60.0	300	0.467
10	1050	55	0.208	0.207	5.15	2.21	2.21	1.77	0.2	50.7	300	0.443
11	1050	64	0.333	0.333	4.41	2.21	2.21	4.66	0.45	59.3	300	1.128

incorporate a market-driven approach. The market-driven approach provides a profit model in the near future to determine the optimal position and number of product architectures among product architecture candidates generated by the *k*-means clustering algorithm. An extended market value prediction method is proposed to capture the trend of customer preferences and uncertainties in predictive modeling.

The predictive, data-driven product family design (PDPFD) framework consists of the individual product design stage and the product family design stage. The individual design stage is a bi-level optimization model. At the enterprise level, an updated market value prediction method is suggested using the exponential smoothing (Hyndman et al. 2008). In comparison with the original model (Rutherford and Wilhelm 1999), the proposed predictive model not only showed the better prediction accuracy for data with complex patterns but also provided prediction intervals which represent the uncertainties of customer preferences. The *k*-means clustering algorithm is suggested to capture the effect of deviations between product architectures and customer requirements. Then, the optimal position and number of product architectures can be determined to maximize the expected profit without predefined market segment information. With this market-driven target, the engineering-level optimization problem is formulated and solved to find designs which minimize deviations from the target. The next stage is the product family design stage where the EM clustering algorithm is applied to find clusters within known common parameters so that the possibility of multiple platforms can be explored. Finally, the engineering-level optimization is resolved with reduced design variables and common parameters.

A universal electric motors design example is used to demonstrate the implementation of the proposed framework in a hypothetical market. The comparative study shows that the PDPFD algorithm maximizes the expected profit, while clustering-based models do not consider market so that less profit can be achieved.

The proposed algorithm starts with a maximum number of architectures which is mainly dependent on the manufacturer's capability and production condition. If k is too big, then processing time for the algorithm will be very long accordingly. More efficient ways should be explored to find the lower and upper bound of the number of k in the future. Furthermore, throughout this study, a scenario with no competition was used. It will be interesting to consider competing products for market-driven target setting as possible future work. With competitors' products, the assumption of the closest product selection may be not hold.

References

Agard B, Kusiak A (2004) Data-mining-based methodology for the design of product families. Int J Prod Res 42(15):2955–2969

Bezdek JC (1981) Pattern recognition with fuzzy objective function algorithms. Kluwer, Norwell

Chan KY, Kwong C, Hu B (2012) Market segmentation and ideal point identification for new product design using fuzzy data compression and fuzzy clustering methods. Appl Soft Comput 12(4):1371–1378

Chen C, Wang L (2008) Product platform design through clustering analysis and information theoretical approach. Int J Prod Res 46(15):4259–4284. doi:10.1080/00207540701199693

Chen W, Hoyle C, Wassenaar H (2012) Decision-based design: integrating consumer preferences into engineering design. Springer, Bücher

Dai Z, Scott M (2007) Product platform design through sensitivity analysis and cluster analysis. J Intell Manuf 18(1):97–113. doi:10.1007/s10845-007-0011-2

Damodaran P, Wilhelm WE (2005) Branch-and-price approach for prescribing profitable feature upgrades. Int J Prod Res 43(21):4539–4558. doi:10.1080/00207540500168139

Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. J R Stat Soc Ser B (Methodological) 39(1):1–38

DesAutels P, Berthon P (2011) The PC (polluting computer): forever a tragedy of the commons? J Strateg Inf Syst 20(1):113–122. doi:10.1016/j.jsis.2010.09.003

de Weck OL, Suh ES, Chang DD (2003) Product family and platform portfolio optimization. In: 2003 ASME design engineering technical conference, American Society of Mechanical Engineers, Chicago, Illinois, DETC03/DAC-48721

Do CB, Batzoglou S (2008) What is the expectation maximization algorithm? Nat Biotechnol 26(8):897–899

Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The weka data mining software: an update. SIGKDD Explor Newsl 11(1):10–18

Hyndman R, Khandakar Y (2008) Automatic time series forecasting: the forecast package for R. J Stat Softw 27(3):1–22

Hyndman R, Koehler A, Ord J, Snyder R (2008) Forecasting with exponential smoothing: the state space approach. Springer, Berlin

Kira K, Rendell LA (1992) A practical approach to feature selection. In: Proceedings of the ninth international workshop on machine learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, ML92, pp 249–256

Kriegel HP, Kröger P, Zimek A (2009) Clustering high-dimensional data: a survey on subspace clustering, pattern-based clustering, and correlation clustering. ACM Trans Knowl Discov Data 3(1):1–58. doi:10.1145/1497577.1497578

Kumar D, Chen W, Simpson TW (2009) A market-driven approach to product family design. Int J Prod Res 47(1):71–104

Kwak M, Kim H (2013) Market positioning of remanufactured products with optimal planning for part upgrades. J Mech Des 135(1):011,007. doi:10.1115/1.4023000

Lei N, Moon SK (2015) A decision support system for market-driven product positioning and design. Decis Support Syst 69:82–91. doi:10.1016/j.dss.2014.11.010

Ma J, Kim H (2014) Continuous preference trend mining for optimal product design with multiple profit cycles. J Mech Des 136(6):1–14. doi:10.1115/1.4026937

Ma J, Kwak M, Kim HM (2014) Demand trend mining for predictive life cycle design. J Clean Prod 68:189–199. doi:10.1016/j.jclepro.2014.01.026

MacQueen J (1967) Some methods for classification and analysis of multivariate observations. In: Le Cam LM, Neyman J (eds) Proceedings of the 5th Berkeley symposium on mathematical statistics and probability, vol 1, University of California Press, Berkeley, CA, USA, pp 281–297

Messac A, Martinez MP, Simpson TW (2002) Introduction of a product family penalty function using physical programming. J Mech Des 124(2):164–172. doi:10.1115/1.1467602

Moon S, Kumara SRT, Simpson TW (2006) Data mining and fuzzy clustering to support product family design. In: 2006 ASME design engineering technical conference, American Society of Mechanical Engineers, Philadelphia, Pennsylvania, DETC2006-99287

Nayak RU, Chen W, Simpson TW (2002) A variation-based method for product family design. Eng Optim 34(1):65–81. doi:10.1080/03052150210910

Pelleg D, Moore A (2000) X-means: extending k-means with efficient estimation of the number of clusters. In: Proceedings of the 17th international conference on machine learning. Morgan Kaufmann, pp 727–734

Pirmoradi Z, Wang GG, Simpson T (2014) A review of recent literature in product family design and platform-based product development. In: Jiao J, Siddique Z, Hölttä-Otto K (eds) Advances in product family and product platform design. Springer, New York, pp 1–46

R Development Core Team (2008) R: a language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0

Rutherford DP, Wilhelm WE (1999) Forecasting notebook computer price as a function of constituent features. Comput Ind Eng 37(4):823–845

Simpson T (2004) Product platform design and customization: status and promise. AI EDAM 18:3–20

Simpson T, Maier J, Mistree F (2001) Product platform design: method and application. Res Eng Des 13(1):2–22

Simpson T, Bobuk A, Slingerland L, Brennan S, Logan D, Reichard K (2012) From user requirements to commonality specifications: an integrated approach to product family design. Res Eng Des 23(2):141–153

Simpson T, Jiao J, Siddique Z, Hölttä-Otto K (2014) Advances in product family and product platform design: methods & applications. Springer, New York

Sun W, Wang J, Fang Y (2012) Regularized k-means clustering of high-dimensional data and its asymptotic consistency. Electron J Stat 6:148–167. doi:10.1214/12-EJS668

Train K (2003) Discrete choice methods with simulation. Cambridge University Press, Cambridge

Tseng MM (1998) Design for mass customization by developing product family architecture. In: 1998 ASME design for manufacture conference, American Society of Mechanical Engineers, Atlanta, GA, DETC98/DFM-5717

Tucker CS, Kim HM, Barker DE, Zhang Y (2010) A relieff attribute weighting and X-means clustering methodology for top-down product family optimization. Eng Optim 42(7):593–616

Ulrich K, Eppinger S (2012) Product design and development. McGraw-Hill, New York

Wassenaar HJ, Chen W (2003) An approach to decision-based design with discrete choice analysis for demand modeling. J Mech Des 125(3):490–497. doi: 10.1115/1.1587156. http://link.aip.org/link/?JMD/125/490/1

Wassenaar HJ, Chen W, Cheng J, Sudjianto A (2005) Enhancing discrete choice demand modeling for decision-based design. J Mech Des 127(4):514–523. doi: 10.1115/1.1897408. http://link.aip.org/link/?JMD/127/514/1

Wilhelm WE, Damodaran P, Li J (2003) Prescribing the content and timing of product upgrades. IIE Trans, pp 647–664. doi:10.1080/07408170390214509

Witten I, Frank E (2005) Data mining: practical machine learning tools and techniques. The Morgan Kaufmann Series in Data Management Systems, 2nd edn. Elsevier Science, Amsterdam