21ST INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED17
21-25 AUGUST 2017, THE UNIVERSITY OF BRITISH COLUMBIA, VANCOUVER, CANADA

BRIDGING THE SEMANTIC GAP IN CUSTOMER NEEDS ELICITATION: A MACHINE LEARNING PERSPECTIVE

Wang, Yue (1); Zhang, Jian (2)
1: Hang Seng Management College, Hong Kong S.A.R. (China); 2: Dongguan University of Technology, Hong Kong S.A.R. (China)

Abstract
The elicitation of customer needs (CNs) is a critical step in product development. However, these needs are often expressed in ambiguous, simple language and not in the form of well-defined specifications, causing a semantic gap in the product development process. Traditional methods to bridge the gap rely heavily on human action. Product development teams need to manually link CNs to product specifications in an ad hoc manner. This may be infeasible for large product variant spaces or evolving product families. We propose a machine learning mechanism to automatically bridge the semantic gap. This task is considered as a classification problem, with CNs being the class. The mapping function from product specifications to CNs is learned from training data by using a support vector machine and decision tree classifier. Given a new product variant, the learnt classifier can determine the needs that the product variant can satisfy. Numerical experiments show that the proposed method can achieve very high mapping accuracy. It can also shield product development teams from the tedious labour of linking CNs to product variants, and thus improve the efficiency of needs elicitation.

Keywords: Design informatics, Decision making, Requirements, Semantic gap, Configurator

Contact:
Dr. Yue Wang
Hang Seng Management College
Supply Chain and Information Management
Hong Kong S.A.R. (China)
yuewang@hsmc.edu.hk

Please cite this paper as:
Surnames, Initials: *Title of paper*. In: Proceedings of the 21st International Conference on Engineering Design (ICED17), Vol. 4: Design Methods and Tools, Vancouver, Canada, 21.-25.08.2017.

1 INTRODUCTION

Product design is the process of capturing and transforming customer needs (CNs) into tangible product specifications that designers can follow. It has been acknowledged that the key to product success relies on better understanding of voice of the customer and on better links between customers' preferences and needs (Ulrich and Eppinger, 2011). To successfully design a product, the clarity and completeness of understanding customers' needs and perceived values are essential deciding factors. Extensive research has been conducted to characterise, understand and elicit CNs in product design, such as online product configurator design (Sabin and Weigel, 1998), conjoint analysis (Green and Srinivasan, 1990), the voice of customers (Griffin and Hauser, 1993) and Kansei/affective engineering (Nagamachi, 1989). All of these methodologies depend on customers' abilities and willingness to explicitly express their preferences and needs. For example, most of the current configurators require customers to specify combinations of the product design parameters (DPs) that best meet their needs. However, the difficulty of articulating CNs has been documented. Customers may not possess the necessary expertise of products with which they are unfamiliar (Wang and Tseng, 2014a). They may only be able to express their actual perceptual intents and needs using vague and ambiguous language, such as 'A race car has a powerful feeling with a very strong engine' instead of the technical specification (TS) '3.6L Pentastar DOHC V-6 engine'. It may be difficult, if not impossible, for them to map their subjective needs to the DPs and TSs (Randall et al., 2005). This complexity increases the frustration and confusion that customers encounter and hinders them from accepting product offerings (Blecker and Friedrich, 2006) (Tang et al., 2010). Design defects can also arise as a consequence (Randall et al., 2005).
This difficult situation mainly arises when customers are required to make choices based on the product specifications of products with which they are not familiar, rather than based on their actual perceptual intents and needs. In linguistics, this difference in descriptions is a classic problem known as the 'semantic gap', meaning the discrepancy between two linguistic representations of the same object (Hu et al., 2013). Due to the semantic gap, people using different linguistic scopes (i.e., speaking different languages) cannot communicate with each other. This concept is then further extended to various fields, describing the difficulty of communication at the interface of different parties. Specifically, in the product design field, the semantic gap characterises the differences between product DPs and the intents or requirements of customers. To achieve efficient communication, the mapping relationships between different representation scopes need to be studied. This process is known as 'bridging the semantic gap'.

2 BACKGROUND AND RELEVANT RESEARCH

Discussion on the importance of reducing semantic gaps in CNs elicitation, although not always explicitly referred to in these terms, has long existed in product design research. Green and Srinivasan stated that the 'semantic gap is a classic problem in conjoint analysis, but most applications ignore the interactive effects of product attributes on consumer preferences' (Green and Srinivasan, 1990). Krippendorff (2006) emphasised that product design should fundamentally make sense to customers. In axiomatic design theory, the semantic gaps between customers and product providers are bridged from the product provider's perspective: it is a systems design methodology using matrix methods to systematically analyse the transformation of CNs into functional requirements (FRs), DPs and process variables (Suh, 2001). With knowledge of their target customers' needs, providers can set the related manufacturing process variables accordingly. Based on this structure, Ramaswamy and Ulrich (1993) proposed another casual network in reverse order, in which product parameters drive function specifications and finally relate to CNs. Thus, designers usually perform mappings between the CN, FR and DP domains. Little research has been conducted on the automation of this mapping.
Bridging the semantic gap has also been studied in marketing science. Randall et al. (2007) introduced the seminal needs-based product configurator design. This needs-based product configurator allows customers to configure products based on their perceived needs instead of detailed product specifications, as with traditional parameters-based product configurators. When using needs-based configurators, customers are presented with a series of descriptions of their desired products. These descriptions are stated in simple words, such as 'I want a light computer so I can carry it easily'. Customers indicate how important or relevant the descriptions are to their needs by sliding a bar on the configurator. After confirming this step, they are directed to the second step showing a reference product based on their initial preference. Customers can then modify the reference product using human

semantic words (e.g., 'lighter', 'cheaper' and 'larger') until they make a final decision. Randall et al. demonstrated that for non-professional users, this needs-based configurator results in better performance than the traditional parameter-based configurators for innovative products. They also verified that when the expertise levels of customers increased, the advantage of using the needs-based product configurator decreased. The whole mechanism provides an intuitive and user-friendly way to elicit CNs.

However, there are critical limitations to the proposed needs-based configurator design. First, product designers predefine the mapping relationships between expressed CNs (a set of plain language in non-technical terms) and design parameters (TSs), in what is basically a pre-tagging process. Manually mapping CNs to design parameters (i.e., the process of manually bridging the semantic gap) is very challenging and can easily be biased, even by professional designers. Designers need to compare every product in the solution space and set rules for manually updating recommendations. As long as there is an update of product series or market change, this process has to be repeated. Therefore, this configurator design cannot be easily implemented for fast-moving products. In addition, Randall et al. used an empirical experiment to prove that the proposed diagram works well mainly for innovative products (e.g., computers) or other services, such as retirement and medical plans, in which semantic gaps are much easier to bridge. In these cases, the main barrier is often that customers have very limited knowledge of the configurable features (e.g., memory space and CPU speed). As long as some understandable explanations are transferred to customers during the configuration process, the configurator's performance can be greatly improved. Therefore, the performance premium drops dramatically once innovative products become widely known of and accepted in markets.

To summarise, a practical approach for automatically bridging the semantic gap between perceptual needs and product specifications in the CN elicitation process is still lacking. We attempt to address this challenge by bridging the semantic gap between product design parameters and the intent of customers. We use machine learning approaches to learn a semantic mapping network and break the barrier between these two domains. A new relative preference-based production configurator and a recommendation framework are built based on the network. Customers do not need to go through the routine configuration process. They are only required to fine-tune a reference product by indicating their relative preferences for each attribute. This could shield customers from the confusion and perplexity encountered with traditional configurators.

3 RESEARCH FRAMEWORK

3.1 Notation and problem definition

To bridge the semantic gaps between customers and product providers, we first generate a description of a product using 'language' from both sides.

Product technical specification from designers' point of view: A product is considered to be a bundle of $TS = \{S_1, S_2, ..., S_n\}$ where each variable S_i represents the ith TS of the product and n is the number of TSs. S_i takes values from its choice set $\{s_{i1}, s_{i2}, ..., s_{im_i}\}$ and mi is the cardinality of the ith specification's choices.

Customer needs from customers' point of view: CNs for a product are represented by a set, $CN = \{C_1, C_2, ..., C_k\}$ where each C_i represents a particular need expressed by the customer. We assume that k is a finite number and that a customer may not express needs in all dimensions of k. It should be noted that C_i may also take a finite number of values. For example, for needs of appearance, the value may be 'fancy', 'cool' or 'plain', among others. Each value of a need can be considered as a label of the need. Note that CN as used hereafter, is different from that in the axiomatic design theory. In real practice, it is common for customers to express their needs at a hybrid level, using both rough descriptions and detailed specifications during the needs elicitation process. However, CNs and TSs (referred to as DPs in axiomatic design theory) in axiomatic design theory usually have no overlap.

An example: The TS and CN information of mobile phones are extracted from a Chinese online shopping platform (www.imobile.com.cn) using a Web crawler. The TS set is TS={screen size, screen resolution, processor, RAM, storage space, with or without front-facing camera, with or without duo cameras, capacity of battery, price, mode of network, with or without fingerprint recognition, steel frame or not, colour}. Without loss of generality, we assume that the specifications are ordered as presented. Each specification has a choice set. For example, the value of 'with or without duo cameras' is either 'yes' or 'no' and the specification of 'screen size' includes the numerical values of the screen size of the whole product choice set. For the iPhone 7, one product variant of mobile phone, the corresponding TS set is TS={4.7 inches, 1,334-by-750-pixel resolution, A10 Fusion chip with 64-bit architecture, 128 GB, with front camera, with duo cameras, 14 hours talk time, $749, …, non-steel frame, jet black}. The CN set is CN={appearance, cost-performance, comprehensive functions, powerful functions, resolution, stand-by time, screen size}. The elements in the CN set are identified based on the summary of customer reviews provided by the online shopping platform. Each element in the CN set also has a value. If a customer wants a mobile phone with an appealing appearance, long stand-by time and a big screen size, the corresponding CN set is CN={appealing appearance, NULL, NULL, NULL, NULL, long stand-by time, big screen size}, where NULL indicates that the customer has no particular requirement for the corresponding need. Although 'screen size' appears in both the TS and CN sets, their values are from different vocabularies. For TS, the value of 'screen size' is numerical. However, in the CN set, the values of 'screen size' are 'big', 'medium', 'small' or 'NULL'.

3.2 Mapping procedure

The purpose of this study is to find a way to bridge the semantic gap, such that product searching or customisation tasks can be performed automatically. In addition, one customer's needs may depend on multiple TSs. For example, the need for 'a cool race car' may be related to the TSs of 'colour', 'exterior detail', 'light' and 'wheel', among others. Thus, the mapping functions from each CN to all TS, $g: CN \rightarrow TS$ should be identified. However, the dimension of the CN set is usually much smaller than that of the TS set. Finding the mapping g is usually an ill-posed problem. To handle this issue, we study the dual problem by mapping TS to CN via the function $f_i: TS \rightarrow CN_i$. That is, given a product specification TS set, we should be able to find the corresponding CN set. In this sense, the mapping process is equivalent to a classification problem. Given the mapping function, $f_i: TS \rightarrow CN_i$, and a product variant's TSs, the system should know the CNs that this product variant can satisfy if we consider each value of CN as a label of the class. With this information known, a product can be recommended based on the distance between the label and the customer's expressed needs. Therefore, these two mappings are two sides of the same coin. In this study, only the mapping is conducted based on machine learning algorithms. A general procedure for the training stage is as follows.

- Input: a set of training data $\{(TS_i, CN_i): i = 1,2,...,n\}$ where n is the size of training set. Each set of training data contains a TS and a CN set of the same product variant. The CN set can be considered as the label of the TS set, indicating the class of the TS set.
- Apply a classification algorithm to build the mapping function $f_i: TS \rightarrow CN_i$ for this type of product. A support vector machine (SVM) and a decision tree are used to train the classifier. Details can be found in the next section and in the Appendix.
- Output: The mapping function f_i's.

After training the set of mappings f_i's for this product, we can derive the needs that a new product can satisfy. The procedure is as follows.

- Input: the mapping function learnt in the training stage, $f_i: TS \rightarrow CN_i$, and a new product variant with the specification TS_k.
- Find the corresponding $CN_i = f_i(TS_k)$ for all i's.
- Output: All the needs that the product variant can satisfy.

The proposed mapping is particularly useful for product customisation or an evolving product family, for which manual mapping between CNs and TSs is usually infeasible. For example, BMW claims that the number of variants in its 7 Series line can reach 10^17 (as product variants can be customised) (Zhu et al., 2008). In this case, brute force manual mapping is clearly infeasible. Mapping can improve the efficiency of product searching and providing recommendations. In addition, new product variants may be added to the whole product space. How customers review and evaluate a new product variant may not be known. Leveraging the mapping algorithm makes it easier to determine what kinds of needs a new variant can satisfy.

4 EXPERIMENT PROTOCAL

4.1 Learning algorithm

We use SVM and decision tree methods for classification, as they have been proven to be effective and efficient. The SVM is a supervised learning model for classification tasks (Vapnik, 1999). The basic idea is to transform low dimensional data, which are not linearly separated, into high dimensional data by a kernel function, which would allow them to be separated by a hyper plane. This has been proven to be an effective classification method and has been applied in various applications, such as text classification, computer vision and natural language processing. Traditional SVMs classify two categories. By combining multiple two-class SVM classifiers, it is easy to obtain an ensemble SVM for multiple-category classification. Decision trees are also well-studied methods of predicting the classes to which raw data belong (Quinlan, 1983). They classify objects in the form of tree structures. A set of classification rules is learnt based on training data. Then the learnt rules are generalised to new observations to determine their categories. Some details of SVM classifier are elaborated in Appendix 2. Since, decision tree is a well-known classification algorithm, the introduction is omitted in this paper. We use scikit-learn, a Python language based machine learning toolkit (http://scikit-learn.org/stable/supervised_learning.html#supervised-learning), to train the semantic bridging mapping from functional requirements to product specifications. Each functional requirement is discretised based on the requirements' properties. An SVM classifier with a radial basis function kernel and a decision tree are used to train the classifier (i.e., mapping).

4.2 Data

The training data are the product specifications of and customer-defined functional requirements for a given product. The data are in the format of (CN, TS) as mentioned in Section 3. Heritrix, an open source Web crawler, is used to extract product information from two Chinese ecommerce websites, Jindong (JD.com) and Mobile Home (www.imobile.com.cn). Data on 200 digital camera models and 200 mobile phone types are collected. For each product, the TSs can be obtained from the product's introduction. Customer reviews of each product can also be found on a given product's page. In addition, the two websites summarise customers' reviews using keywords, which are usually in layman's terms. These summaries can be considered to be the expressed CNs. Thus, we can use the Web crawler to obtain both CN and TS data for model training and testing. Screenshots of the summary of customer reviews (i.e., CNs), customer reviews and TSs for the SONY DSC-RX100 camera model are shown in Figure 1 below.

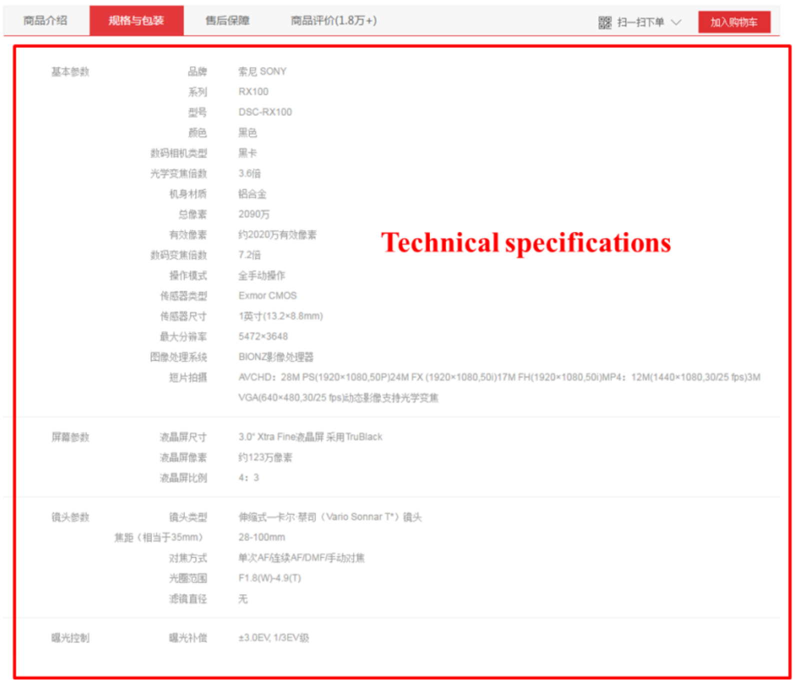

Figure 1. Screenshots of the summary of customer reviews, customer reviews and technical specifications for the SONY DSC-RX100 camera

There are 23 TSs and eight functional requirements to characterise each camera. The detailed lists of the attributes are provided in Appendix 1. 13 TSs and seven functional requirements are used to characterise each mobile phone, as detailed in Example 1 (Section 3.1). It should be noted that the extracted data (i.e., TSs and CNs) are in Chinese and we further transformed them into numerical code. We keep the original screenshot here to illustrate of the source of CN and TS data.

4.3 Experiment procedure and results

For each type of product category, 200 samples are extracted from the website. Five-fold cross validation is used to test the classification results, meaning that the dataset is separated into five parts of equal size. One hundred and sixty samples are used to train the classifiers and the remaining samples are used to test the classification results. Each time, the testing data is changed among the five and the remaining four sets are used as training set, until all of the five sets have been used as testing data. The training time is very short, as the sample size is not very big. For all test data (CN, TS), the TS vector is used as the input to the classification algorithm. The output of the algorithm CN' is then compared to CN to determine whether the TS is correctly mapped. Classification accuracy is used as the measure of performance, which is defined as $accurary = \frac{\#correct\ \ classification\ \ of\ \ CN}{\#testing\ \ data\ \ size}$ .The detailed results for customers' mobile phone and camera needs are shown in Tables 1 and 2.

Table 1. Mapping accuracy for customers' mobile phone needs using the SVM and decision tree classifier

Customer needs	Support vector machine (%)	Decision tree (%)
Appearance	73.3	72.6
Cost-performance	82.5	79.8
Multiple functions or not	86.7	81.0
Powerful functions or not	77.8	73.7
Resolution	92.8	90.4
Stand-by time	94.0	92.6
Screen size	92.7	90.2

Table 2. Mapping accuracy for customers' camera needs using the SVM and decision tree classifier

Customer needs	Support vector machine (%)	Decision tree (%)
Portability	93.0	92.7
Easy to operate	89.6	87.2
General effect of photos	91.8	88.7
Appearance	76.0	72.4
Functions	81.0	78.5
Focus speed	94.8	90.6
Purpose of usage	78.6	76.0
Image quality	82.8	77.3

4.4 Discussion

The SVM classifier outperforms the decision tree classifier in all cases. Two hundred data are available for both mobile phones and cameras. Five-fold cross validation is used to test the performance of classifiers. Thus, there are 160 training samples and 40 testing samples in each round. The overall sample size is not big enough: the SVM can handle smaller data set very well, but the decision tree requires relatively large sets. This may explain why the SVM is better in all of the tasks.

In addition, some CNs have very high classification accuracies, such as the needs for 'high resolution', 'long stand-by time' and 'big screen' for mobile phones and 'fast focus speed' for cameras. Most of these CNs or preferences are actually part of the DP domain. It is common for customers to express their needs at a hybrid level, using both rough descriptions and detailed specifications during the needs elicitation process. The classification accuracies are high for well-articulated needs similar to product specifications, as they can be well linked to the product specifications. For example, the need for 'high resolution' is highly correlated with the DP of 'screen resolution' (see Appendix 1 for the list of CNs and DPs). Vague and imprecise needs, such as 'appealing appearance', may depend on a set of design parameters. Thus, the classification accuracies of such needs are not as high.

5 CONCLUDING REMARK

CN elicitation is as a very important step in the early product development stage (Wang and Tseng, 2014b). With the development of the Internet, online needs elicitation toolkits have become popular in the industry. Examples include product configurator systems, online choice menus and choice navigation systems (Wang and Tseng, 2011). All of these toolkits can automate the needs elicitation process, shield companies from high labour costs and facilitate firm operations. However, the existing methods usually require customers to express their needs in the product specification domain. This may be a challenging task, as customers may not possess the expertise to do so. They may only be able to express their needs in descriptive and plain language, causing semantic gaps in the product specifications domain. Although studies have attempted to bridge such gaps, the methods proposed have various limitations. This study presents a machine learning approach to mapping CNs to product specifications. The mapping problem is modelled as a classification problem. SVM and decision tree classifiers are trained based on the data captured from the two online shopping websites. It is numerically shown that the SVM-based approach can achieve very high accuracy.

This paper does have some limitations. First, the proposed method is evaluated using a small sample size. Further efforts are needed to collect more data and test it on a larger scale. In addition, the proposed method is domain-knowledge free. The advantage of this property is that the model is very general and can be applied to any product. However, to improve the mapping accuracy of particular products and the performance of the classification model, it will be necessary to add more domain knowledge. How to represent and incorporate domain knowledge into the classification model and how to determine which model is more suitable remain challenging tasks to explore.

REFERENCES

Blecker, T. and Friedrich, G. (2006), *Mass Customization: Challenges and Solutions*, Springer. http://dx.doi.org/10.1007/0-387-32224-8

Bottou, L. and Lin, C. (2007), "Support Vector Machine Solvers", In Large Scale Kernel Machines, edited by L. Bottou, Olivier Chapelle, Dennis DeCoste, and Jason Weston, MIT Press, Cambridge, MA.

Green P. and Srinivasan, V. (1990), "Conjoint Analysis in Marketing: New Developments with Implications for Research and Practice", *Journal of Marketing*, vol. 54, pp. 3-19. http://dx.doi.org/10.2307/1251756

Griffin, A. and Hauser, J. R. (1993), "The Voice of the Customer", *Marketing Science*, Vol.12, pp. 1-27. http://dx.doi.org/10.1287/mksc.12.1.1

Hu, T., Zhao J. and Zhao D. (2013), "A Study on the Semantic Gap Between Designer and User in Automobile Design", in International Association of Societies of Design Research, Tokyo, Japan.

Krippendorff, K. (2006), *The Semantic Turn: A New Foundation for Design*, CRC/Taylor & Francis. http://dx.doi.org/10.4324/9780203299951

Nagamachi, M. (1989), *Kansei Engineering*, Tokyo: Kaibundo Publishing Co. Ltd.

Quinlan, R. (1983), "Learning efficient classification procedures", *Machine Learning: an artificial intelligence approach*, Michalski, Carbonell & Mitchell (eds.), Morgan Kaufmann, pp. 463-482. doi:10.1007/978-3-662-12405-5_15

Ramaswamy R. and Ulrich, K. (1993), "Augmenting the House of Quality with Engineering Models", *Research in Engineering Design*, vol. 5, no. 2, pp. 70-79. http://dx.doi.org/10.1007/BF02032576

Randall, T., Terwiesch. and Ulrich, K. (2005), "Principles for User Design of Cutsomized Products", *California Management Review*, Vol. 47, pp. 68-85.

Randall, T., Terwiesch. and Ulrich, K. (2007), "User Design of Customized Products", *Marketing Science*, Vol. 26, pp. 268-280.

Sabin, D. and Weigel, R. (1998), "Product Configuration Frameworks – A Survey", *IEEE Intelligent Systems*, Vol. 13 No. 4, pp. 42–49. http://dx.doi.org/10.1109/5254.708432

Suh, N. (2001), *Axiomatic Design: Advances and Applications*, Oxford University Press.

Tang, Z., Chen X. and Xiao J. (2010), "Using the Classic Grounded Theory Approach to Understand Consumer Purchase Decsion in Relation to the First Customized Products", *Journals of Product and Brand Management*, Vol. 19, pp. 181-197.

Ulrich, K. and Eppinger, S. (2011), *Product Design and Development*, 5th edition, McGraw-Hill Education.

Vapnik, V. (1999). *The Nature of Statistical Learning Theory.* Berlin: Springer. http://dx.doi.org/10.1007/978-1-4757-3264-1

Wang, Y. and Tseng, M. (2011), "Adaptive Attribute Selection for Configurator Design via Shapley Value", *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Vol. 25 No 1, pp. 189–199.

Wang, Y. and Tseng, M. (2014a), "Incorporating Tolerances of Customers' Requirements for Customized Products", *CIRP Annals-Manufacturing Technology*, Vol. 63, pp. 129-132.

Wang, Y. and Tseng, M. (2014b), "Identifying Emerging Customer Requirements in Early Design Stage by Applying Bayes Factor Based Sequential Analysis", *IEEE Transactions on Engineering Management*, Vol. 61 No. 1, pp. 129-137.

Zhu, X., Hu, S. and Yoram, K. (2008), "Modeling of Manufacturing Complexity in Mixed-Model Assembly Lines", *Journal of Manufacturing Science and Engineering*, Vol. 130:051013. http://dx.doi.org/10.1115/1.2953076

ACKNOWLEDGMENTS

This paper is partially supported by UGC/FDS14/E02/15. We would like to thank the reviewers for their comments. The authors contribute equally to this work. The data set used in this paper is available upon request.

APPENDIX

1. TSs and CNs of digital camera used in the experiment:
   Technical specifications: price, pixel, weight, size of LCD screen, LCD panel ratio, touch screen or not, rotated LCD or not, colour, type, Optical zoom, body material, sensor type, sensor size, image processing system, type of lens, focal length, focus mode, white balance mode, shutter speed, image stabilization function, selfie function, burst function, extra function.
   Customer needs: portability, easy to operate, general effect of photos, appearance, powerful functions, focus speed, purpose of usage, image quality
2. Introduction of SVM
   Given a set of training set $\{(x_i, y_i) : i = 1,2,...,n\}$, the objective of SVM is to fit the data into a function f, which satisfies the following mathematical programming.

$$\min_{w,b,\xi,\xi^*} \quad \frac{1}{2} w^T w + C\sum_{i=1}^{n} \xi_i + C\sum_{i=1}^{n} \xi_i^* \tag{1}$$

Subject to

$$w^T f(x_i) + b - y_i \le \varepsilon + \xi_i \tag{2}$$

$$y_i - w^T f(x_i) - b \le \varepsilon + \xi \tag{3}$$

$$\varepsilon, \xi_i, \xi_i^* \ge 0, \quad i = 1,...,n \tag{4}$$

where w is the vector variable, b, ξ_i and ξ_i^* are positive slack variables. C is a positive parameter to determine the trade-off between the flatness of f and the tolerance of deviations larger than ε .
A large body of literature has been devoted to the optimization method of solving the SVM. Details can be found in (Bottou and Lin, 2007).