Evaluation of Kano-like Models Defined for Using Data Extracted from Online Sources

Huishi Yin and Dietmar Pfahl(✉)

Institute of Computer Science,
University of Tartu, J. Liivi 2, 50409 Tartu, Estonia
{huishi,dietmar.pfahl}@ut.ee

Abstract. The Kano model is a frequently used method to classify user preferences according to their importance, and by doing so support requirements prioritization. To implement the Kano model, a representative set of users must answer for each feature under evaluation a functional and dysfunctional question. Unfortunately, finding and interviewing users is difficult and time-consuming. Thus, the core idea of our proposed approach is to extract automatically opinions about product features from online open sources (e.g., Q & A sites, App reviews, etc.) and to feed them into the Kano questionnaire to prioritize software requirements following the principles of the Kano model. One problem with our proposed approach is how to pair input extracted from the internet into paired answers to the functional dysfunctional questions. This problem arises because the reviews and comments from online sources that we plan to transform into answers to either the functional or dysfunctional question are usually unpaired. Therefore, the aim of this study is to find a method that produces results resembling those of the traditional Kano model although we only retrieve partial information. We propose two Kano-like models, i.e., the Half- and the Deformed-Kano model, for unpaired answers to functional and dysfunctional questions. In order to analyze the performance of the two proposed models as compared to that of the traditional Kano model, we run several simulations with synthetic data. Then we compare the simulation results to see which Kano-like model produces results that are similar to those of the traditional Kano model. The simulation results show that on average both the Half-Kano and Deformed-Kano models on average generate feature categorizations similar to those of the traditional Kano model. However, only the Deformed-Kano model generates the same range of categorizations as the traditional Kano model. The Deformed-Kano can be used as an approximation of the traditional Kano model when the input is unpaired or partly missing.

Keywords: Kano model · Requirement prioritization · Online source

1 Introduction

In software product management, requirement prioritization is often used for determining which candidate requirements of a feature should be included in a software release. Requirements are also prioritized to minimize risk during development so that the most important or low-risk requirements are implemented first [1, 2]. Several

P. Abrahamsson et al. (Eds.): PROFES 2016, LNCS 10027, pp. 539–549, 2016.
DOI: 10.1007/978-3-319-49094-6_39

methods for classifying and prioritizing software requirements exist. The Kano model is one of the best-known and frequently used methods to do this.

The Kano model was developed by Noriaki Kano in the 1980s [3]. It defines the relationship between user satisfaction and product features. Since the Kano model can be used to prioritize user needs as a function of customer satisfaction, it is one of the most popular methods that address customer needs prioritization. The traditional Kano model defines five categories of user needs that have different effects on user satisfaction. Those are O, A, M, I, R[1] [4–8]. Since it is possible to receive contradictory responses from customers, the category Q (Questionable) is also an option.

In the context of a larger research project, we plan to use advanced machine learning and data mining techniques to help us extract systematically user needs from online open sources for complementing traditional sources for the elicitation and prioritization of software requirements. In order to fuel the information generated in this way into requirements prioritization activities, we intend to use the principles of the Kano model to classify user needs according to their importance. In this paper, we do not report on the automatic elicitation and sentiment analysis of user needs voiced in online open sources. In the following we rather focus on how to use the ideas and principles of the Kano model when the information extracted does not correspond to paired answers of the Kano questionnaire.

2 Related Work

Since the 1980s when the Kano model was first introduced, it has become a popular theory used by researchers and business practitioners across many industries. After an extensive review of the literature on the Kano model, Josip and Darko summarized and evaluated five methods, which classify quality features into the categories defined by the Kano model [9], however, in a different way than Noriaki Kano proposed. The methods analyzed were the original Kano model developed by Noriaki Kano [3], the "Penalty reward contrast analysis" (PRCA) originally proposed by Brandt [10], the "Importance grid" developed by IBM [11], the "Qualitative data methods" including CIT (critical incident technique) developed by Herzberg and ACC (analysis of complaints and compliments) used by Cadotte [12], Oliver [13], Friman, and Edvardsson [14], and the "Direct classification" method proposed by Emery and Tian [15]. Among those five methods, only CIT and ACC have the same assumption, i.e. that "quality features can be categorized by comparing how frequently customers mention it in a positive context or a negative context" [9]. However, the reliability of both CIT and ACC methods, as compared to the Kano model, remains questionable when the frequencies with which customers mention features are low. According to Josip and Darko's research, the Kano model and the direct-classification method are the only methods capable of classifying features.

[1] O = One-dimensional Quality, A = Attractive Quality, M = Must-be Quality, I = Indifferent Quality, R = Reverse Quality.

Based on the analysis of the related work, we concluded that the traditional Kano model is the best approach to elicit future customers' perceptions regarding a product's features.

3 Research Goal and Method

To apply the Kano model on data extracted from online open sources, we assume that we have already filtered out the sentiment information expressing a person's feeling from online reviews, comments, and questions, and that we have translated the sentiment information into a data set similar to the format of the Kano model. For example, the statement "I dislike X very much!" represents a very negative answer to the functional question, and "I would be very happy if there is no function X." represents a very positive answer to the dysfunctional answer regarding feature X. We put all answers to the functional questions in one "Yes" (Y) vector, and all answers to the dysfunctional questions in one vector "No" (N) vector.

One of the biggest problems we are facing is how to pair the required input to the Kano model without conducting interviews with real people giving answers to both functional and dysfunctional questions. The reviews and comments from an online source are usually unpaired so that we cannot process the data following the traditional Kano model. Because of this, we need to design Kano-like model algorithms for processing unpaired data.

In this paper, we propose two kinds of Kano-like models, i.e., the Half-Kano model and the Deformed-Kano model. The goal of our study is to compare the performance of the Kano-like models with that of the traditional Kano model and to decide which of the two Kano-like models behaves more similar to the traditional Kano model. Our research method uses simulation experiments with synthetic data (see Sect. 5 for details).

4 Kano-like Models

The two Kano-like models we propose differ in the way how they interpret the unpaired answers derived from online open sources. The assumption of the Half-Kano model is that we only have either answers to the functional question or answers to the dysfunctional question. The assumption of the Deformed-Kano model is that answers to the functional and the dysfunctional questions are from the same group of people, even though we lost the links between answers. One output of the traditional Kano model will only contain one specific category to which a feature is classified to. However, in our study, to be able to compare with the output of the Kano-like models, we use the probability of each category that one feature is categorized into instead of the one category to which the feature is most frequently categorized into. For example, if we get five paired answers about one feature, and each paired answer leads to one category, then we have a list of five categories (e.g., M, M, A, M, O). The traditional Kano model output is that this feature is classified to category M. However, in our study, we say the output is that this feature is 60 % classified to category M, 20 % to A, and 20 % to O.

4.1 Half-Kano Model

To implement the Kano-like models on unpaired data, we assume an extreme case, i.e., that each time when we have an interview with our interviewees, we ask only functional questions or only dysfunctional questions relating to one software feature, hence we only get two groups of responses for functional and dysfunctional questions from different interviewees. In this case, we cannot use two responses from different interviewees to classify one person's satisfaction and based on that derive the satisfaction category the software feature belongs to. However, we can implement an algorithm which calculates probabilities with which a software feature would be classified based on the responses for the functional and the dysfunctional questions. Since the data in the Y and the N vectors are not matched, the Half-Kano model is not a traditional Kano model. Nevertheless, we calculate the probabilities following the traditional Kano model. The difference is that in this method, we use each signal value from Y and N vectors to derive a satisfaction category. Figure 1 shows an example of the process of how the Half-Kano model processes the Y and N vectors when the unpaired input is "1" in the Y vector, and "−2" in the N vector. In this case, we can say that with a probability of 40 % this feature should be classified to the M category, with 30 % probability to the I category, and with 10 % probability each to the R, O, and Q categories.

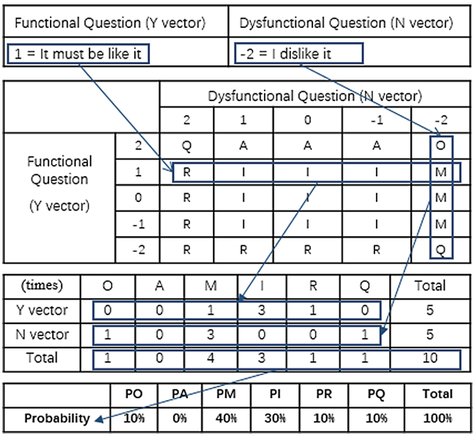

Fig. 1. An example of the process of Half-Kano model (2 = I like it, 1 = It must be like it, 0 = I am neutral, −1 = I can tolerate it, −2 = I dislike it.)

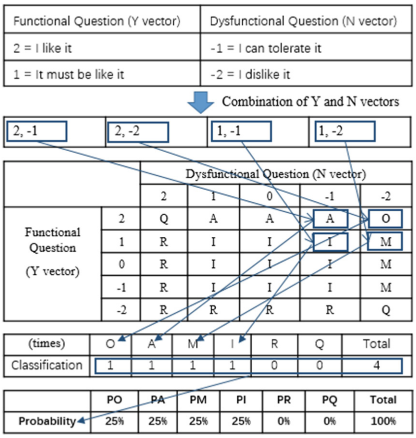

Fig. 2. An example of the process of Deformed-Kano model

The algorithm of probability (P) that vector Y (Functional) and vector N (Dysfunctional) are categorized to the same category (X) can be written as

$$P(cat(Y) = cat(N) = X) = \frac{\sum_{i=1}^{m} F_x cat(Y(i)) + \sum_{j=1}^{n} F_x cat(N(j))}{(m+n) * 5}$$

and

$$F_O cat(Y(i)) = \begin{cases} 1 \text{ if } Y(i) = 2 \\ 0 \text{ if } Y(i) \in \{-2,-1,0,1\} \end{cases} \quad F_A cat(Y(i)) = \begin{cases} 3 \text{ if } Y(i) = 2 \\ 0 \text{ if } Y(i) \in \{-2,-1,0,1\} \end{cases}$$

$$F_M cat(Y(i)) = \begin{cases} 1 \text{ if } Y(i) \in \{-1,0,1\} \\ 0 \text{ if } Y(i) \in \{-2,2\} \end{cases} \quad F_I cat(Y(i)) = \begin{cases} 3 \text{ if } Y(i) \in \{-1,0,1\} \\ 0 \text{ if } Y(i) \in \{-2,2\} \end{cases}$$

$$F_R cat(Y(i)) = \begin{cases} 4 \text{ if } Y(i) = -2 \\ 1 \text{ if } Y(i) \in \{-1,0,1\} \\ 0 \text{ if } Y(i) = 2 \end{cases} \quad F_Q cat(Y(i)) = \begin{cases} 1 \text{ if } Y(i) \in \{-2,2\} \\ 0 \text{ if } Y(i) \in \{-1,0,1\} \end{cases}$$

and

$$F_O cat(N(j)) = \begin{cases} 1 \text{ if } N(j) = -2 \\ 0 \text{ if } N(j) \in \{-1,0,1,2\} \end{cases} \quad F_A cat(N(j)) = \begin{cases} 1 \text{ if } N(j) \in \{-1,0,1\} \\ 0 \text{ if } N(j) \in \{-2,2\} \end{cases}$$

$$F_M cat(N(j)) = \begin{cases} 3 \text{ if } N(j) = -2 \\ 0 \text{ if } N(j) \in \{-1,0,1,2\} \end{cases} \quad F_I cat(N(j)) = \begin{cases} 3 \text{ if } N(j) \in \{-1,0,1\} \\ 0 \text{ if } N(j) \in \{-2,2\} \end{cases}$$

$$F_R cat(N(j)) = \begin{cases} 4 \text{ if } N(j) = 2 \\ 1 \text{ if } N(j) \in \{-1,0,1\} \\ 0 \text{ if } N(j) = -2 \end{cases} \quad F_Q cat(N(j)) = \begin{cases} 1 \text{ if } N(j) \in \{-2,2\} \\ 0 \text{ if } N(j) \in \{-1,0,1\} \end{cases}$$

and F_x is a function that maps the statement X to the set {0,1,3,4}

$$F_x : X \to \{0,1,3,4\}$$

where

$$i \in \{1,2,3\ldots m\}$$

$$j \in \{1,2,3\ldots n\}$$

$$X \in \{O,A,M,I,R,Q\}$$

m is the number of values of Y vector
n is the number of values of N vector

4.2 Deformed-Kano Model

In the Deformed-Kano model, we assume that the responses are from the same group of people, but we lost the links between answers to functional and dysfunctional questions.

We sequentially pick a number of the Y vector to combine with each number of the N vector to derive the satisfaction categories, and then we get a list of satisfaction categories. After each value in the Y vector has been combined with all values of the N vector, we calculate the overall proportion of the appearance of each category. Figure 2 shows the example of the process of the Deformed-Kano model when the unpaired input is “2, 1” in Y vector, and “−1, −2” in N vector. The output is that this feature is 25 % classified to category M, 25 % to A, and 25 % to O, and 25 % to I.

The algorithm of probability (P) that vectors Y (Functional) and N (Dysfunctional) are categorized to the same category (X) can be written as

$$\mathrm{P}(\mathrm{cat}(\mathrm{Y}) = \mathrm{cat}(\mathrm{N}) = \mathrm{X}) = \frac{\sum_{\mathrm{i}=1}^{\mathrm{m}} \left(\sum_{\mathrm{j}=1}^{\mathrm{n}} x_{\mathrm{X}}(\mathrm{cat}(\mathrm{Y}(\mathrm{i})) * \mathrm{cat}(\mathrm{N}(\mathrm{j})))\right)}{m * n}$$

and

$$\mathrm{cat}(\mathrm{Y}(\mathrm{i})) * \mathrm{cat}(\mathrm{N}(\mathrm{j})) = \begin{cases} \mathrm{O}\ \mathit{if}\ Y(\mathrm{i}) = 2\ \mathit{and}\ N(\mathrm{j}) = -2 \\ \mathrm{A}\ \mathit{if}\ Y(\mathrm{i}) = 2\ \mathit{and}\ N(\mathrm{j}) \in \{-1, 0, 1\} \\ \mathrm{M}\ \mathit{if}\ Y(\mathrm{i}) \in \{-1, 0, 1\}\ \mathit{and}\ N(\mathrm{j}) = -2 \\ \mathrm{I}\ \mathit{if}\ Y(\mathrm{i}) \in \{-1, 0, 1\}\ \mathit{and}\ N(\mathrm{j}) \in \{-1, 0, 1\} \\ \mathrm{R}\ \mathit{if}\ Y(\mathrm{i}) \in \{-2, -1, 0, 1\}\ \mathit{and}\ N(\mathrm{j}) = 2\ ||\ Y(\mathrm{i}) = -2\ \mathit{and}\ N(\mathrm{j}) \in \{-1, 0, 1\} \\ \mathrm{Q}\ \mathit{if}\ \mathrm{Y}(\mathrm{i}) = 2\ \mathit{and}\ \mathrm{N}(\mathrm{j}) = 2\ ||\ \mathrm{Y}(\mathrm{i}) = -2\ \mathit{and}\ \mathrm{N}(\mathrm{j}) = -2 \end{cases}$$

and x_{X} is a function that map the statement X to the set {0,1}

$$x_{\mathrm{X}} : \mathrm{X} \rightarrow \{0, 1\}$$

where

$$\mathrm{i} \in \{1, 2, 3 \ldots \mathrm{m}\}$$

$$\mathrm{j} \in \{1, 2, 3 \ldots \mathrm{n}\}$$

$$\mathrm{X} \in \{\mathrm{O}, \mathrm{A}, \mathrm{M}, \mathrm{I}, \mathrm{R}, \mathrm{Q}\}$$

m is the number of values of Y vector
n is the number of values of N vector

5 Simulation Study

Simulation Input: There are 31 possible value sets[2] both in the Y and N vectors. For example, value set ID No.1 indicates that the Y and N vectors only contain elements with value ‘−2’. Value set ID No.31 indicates that both vectors contain all possible values, i.e., ‘−2, −1, 0, 1, 2’.

[2] $C_5^1 + C_5^2 + C_5^3 + C_5^4 + C_5^5 = 31.$

Simulation Approach: We use the R language to execute the simulation algorithms we proposed in Sects. 4 and 5. We first set the length of Y and N vectors equals to 20, and these 20 numbers are picked from each possible value set to simulate responses of one feature. We combined all 31 possible value sets of Y and N vectors. The total number of possible ways to combine the Y with the N vector value sets is 31 * 31 = 961. In each round simulation, for each combination of value sets of Y and N, we sample data randomly following a chosen distribution, e.g., uniform distribution. Then we run traditional Kano and Kano-like models five times respectively. Next, we calculated the average value of those who have the same value set ID of Y and N vectors and join them together to finally get a table which contains 961 rows and 20 columns (Value set ID of Y and N vectors plus PO, PA, PM, PI, PR, and PQ for the traditional Kano, Half-Kano, and Deformed-Kano model, respectively).

Simulation Hypothesis 1: Deformed-Kano model generates more similar output to the traditional Kano than the Half-Kano model.

We pick the data from one out of 961 rows to show an example of the way to calculate the difference between the traditional Kano model and the Kano-like models. Table 1 shows the way to calculate the difference between traditional Kano and Kano-like models, and the calculation results are shown as well. When we calculate the absolute value of the difference between the two sets of data (Traditional and Half or Traditional and Deformed), the range is 0 to 200 %. Hence, we divide the absolute value by 2 to get the result in the range from 0 to 100 %.

Table 1. An example of the difference between the traditional Kano model and the Kano-like models

	PO	PA	PM	PI	PR	PQ
Traditional (%)	0	20	0	45	25	10
Half (%)	3	15.5	7	40.5	27.5	6.5
Deformed (%)	0	19.5	0	45.5	24.5	10.5
Difference (%)	**Traditional − Half** = (|0 − 3| + |20 − 15.5| + |0 − 7| + |45 − 40.5| + |25 − 27.5| + |10 − 6.5|) /2 = 12.5					
	Traditional − Deformed = (|0 − 0| + |20 − 19.5| + |0 − 0| + 45 − 45.5| + |25 − 24.5| + |10 − 10.5|) /2 = 1					

The lower value of difference represents closer output to traditional Kano model. In the case shown in the Table 1, we can see the Deformed-Kano model's output is closer to the output of the traditional Kano model (difference = 1 %) than the output of the Half-Kano model (difference = 12.5 %).

The ranges and means of the differences between the outputs of the traditional Kano model and the Kano-like models are shown in Table 2. We can see from Table 2 that the range of difference between the traditional and Half-Kano models varies from 10.5 % to 80 %, which is much higher than the range of differences between the traditional and Deformed-Kano models, which is 0 % to 18.74 %. The means show the same trend. 25.99 % between the traditional and Half-Kano models and 4.28 % between the traditional and Deformed-Kano models.

Table 2. The range and means of the difference between the outputs of Traditional Kano model and the Kano-like models

	Traditional-Half (%)	Traditional-Deformed (%)
Ranges	[10.5, 80]	[0, 18.74]
Means	25.99	4.28

To see more clearly the distribution of the differences of outputs between the traditional Kano model and the Kano-like models, we draw 3D and 2D figures. Figures 3 and 4 show that the Deformed-Kano model shows outputs which have lower differences with the outputs of the traditional Kano model.

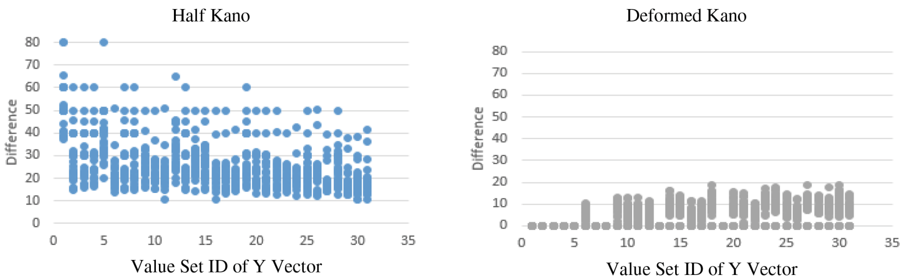

Fig. 3. The projection of the distribution of differences between the traditional Kano and the Kano-like models on Y vector plane

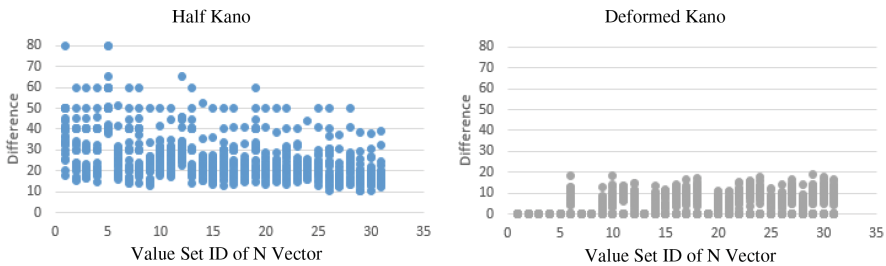

Fig. 4. The projection of the distribution of differences between the traditional Kano and the Kano-like models on N vector plane

Simulation Hypothesis 2: The Deformed-Kano model provides similar outputs as the traditional Kano model.

According to the simulation results, we found that when the input value of the Y vector or the N vector belongs to {−2}, {−1}, {0}, {1}, {2}, {−2, 0}, {−2,1}, {0, 1}, {−2, 0, 1}, the difference always equals zero, which means 477 out of 961 (49.6 %) output combinations of Deformed-Kano and traditional Kano model show no difference.

When the input value of Y vector and N vector does not belong to $\{-2\}$, $\{-1\}$, $\{0\}$, $\{1\}$, $\{2\}$, $\{-2, 0\}$, $\{-2,1\}$, $\{0, 1\}$, $\{-2, 0, 1\}$, the difference will always be more than zero. The simulation results show that 484 out of 961 (50.4%) combinations show differences between the outputs of the traditional Kano and the Deformed-Kano model with a range in 1% to 18%. In addition, the average values are less than 11 %.

6 Threats to Validity

There are several limitations and threats to validity linked to our simulation study. First of all, both Half-Kano and Deformed-Kano models are not fulfilling the requirements of the traditional (interview-based) Kano model, because the input taken from online sources will most of the time not generate paired answers to functional and dysfunctional questions, and the number of answers extracted from online sources will not be balanced. The latter point is particularly limiting our study as comparisons with the traditional Kano model could only be done with sets of balanced data. Thus, we had to restrict our simulation experiments to cases of balanced data. Another limitation is the choice of distributions used in our simulation experiments. Since we have not yet started with extracting real data from online sources, we do not know what empirical distributions of values in the Y and N vectors are realistic. Therefore, we chose a neutral approach and sampled from uniform distributions in our simulation experiments. Finally, we noticed that the relative amount of questionable categorizations might deviate from the typical numbers when using the traditional Kano model. However, since we exclude questionable data from the comparisons between the Kano-like and traditional Kano model, this limitation does not influence our comparison results.

It should be mentioned that the comparison of the Kano-like models used in our simulation study with the traditional Kano model is not based on a single category resulting from the majority of categorizations per interview (in the traditional Kano model) and per vector matching (in the Kano-like models). Instead, we compare the distributions (expressed as probabilities) categorizations per feature. Although this is a deviation from the procedures used by the traditional Kano model, our approach could be considered as giving richer output. Assume, for example, an extreme case where you get 1000 paired answers, with 501 answers leading to category ‘M’ and 499 answers leading to category ‘I’. The output of the traditional Kano model is that this feature should be categorized as “Must be” (M). If only this final categorization is conveyed, one will not know that 49.9 % of the interviewees considered this feature to be “Indifferent” (I).

7 Discussion

It is clear from our simulation study that the Deformed-Kano model produces outputs that are more similar to those of a traditional Kano model than what the Half-Kano model produces.

Although the Deformed-Kano model only partly works like the traditional Kano model, based on our simulation experiments, we found that 49.6 % of its outputs have no difference to the outputs of the traditional Kano model, and for those 50.4 % outputs showing a difference, the differences are very small, with an average of less than 11 %.

Although the Deformed-Kano model does not work exactly like the traditional Kano model, it has several advantages. Firstly, it can handle unpaired and unbalance input (Y and N vectors). Secondly, the ability to process unpaired and unbalanced data but nevertheless producing similar results than the traditional Kano model makes the analysis of user preferences cheaper as all steps can be automated and now costly interviews are needed. Thirdly, it is difficult to guarantee the representativeness of the opinions voiced by a small set of selected interviewees. Using data from online sources has the potential of generating a more complete and thus more realistic input to the analysis of user preferences.

8 Conclusions

According to our simulation experiments, we find that the results of using the Deformed-Kano model are always close to the results of the traditional Kano model. Because of that, we consider the Deformed-Kano model to be a good approximation of the traditional Kano model. Moreover, the Deformed-Kano model can be used even when the input is unbalanced or partly missing. Therefore, we believe that the low cost of using the Deformed-Kano model combined with the possibility to use unbalanced data compensates for the potential lack of paired data when comparing with the traditional Kano model.

Acknowledgement. This research was supported by the Estonian Research Council.

References

1. Lehtola, L., Kauppinen, M., Kujala, S.: Requirements prioritization challenges in practice. In: Bomarius, F., Iida, H. (eds.) PROFES 2004. LNCS, vol. 3009, pp. 497–508. Springer, Heidelberg (2004). doi:10.1007/978-3-540-24659-6_36
2. Berander, P., Andrews, A.: Requirements prioritization. In: Aurum, A., Wohlin, C. (eds.) Engineering and Managing Software Requirements, pp. 69–94. Springer, Heidelberg (2005)
3. Kano, N., Nobuhiku, S., Fumio, T., Shinichi, T.: Attractive quality and must-be quality. J. Jpn. Soc. Qual. Control **14**(2), 39–48 (1984)
4. Berger, C., Blauth, R., Boger, D., Bolster, C., Burchill, G., DuMouchel, W., Timko, M.: Kano's methods for understanding customer-defined quality. Center Qual. Manage. J. **2**(4), 3–35 (1993)
5. Jiao, J.R., Chen, C.H.: Customer requirement management in product development: a review of research issues. Concurrent Eng. **14**(3), 173–185 (2006)
6. Kai, Y.: Voice of the Customer: Capture and Analysis. McGraw-Hill, New York (2007)
7. Sharif Ullah, A.M.M., Tamaki, J.I.: Analysis of Kano-model-based customer needs for product development. Syst. Eng. **14**(2), 154–172 (2011)

8. http://foldingburritos.com/kano-model/. Accessed 15 June 2016
9. Mikulic, J., Prebezac, D.: A critical review of techniques for classifying quality attributes in the Kano model. Managing Serv. Qual. An Intl. J. **21**(1), 46–66 (2011)
10. Brandt, R.D.: A procedure for identifying value-enhancing service components using customer satisfaction survey data. In: Surprenant, C. (ed.) Add Value to Your Service, pp. 61–65. American Marketing Association, Chicago (1987)
11. Vavra, T.G.: Improving Your Measurement of Customer Satisfaction: A Guide to Creating, Conducting, Analyzing, and Reporting Customer Satisfaction Measurement Programs. ASQ Quality Press, Milwaukee (1997)
12. Cadotte, E.R., Turgeon, N.: Dissatisfiers and satisfiers: suggestions from consumer complaints and compliments. J. Consum. Satisfaction Dissatisfaction Complaining Behav. **1**, 74–79 (1988)
13. Oliver, R.L.: Satisfaction: A Behavioral Perspective on the Consumer. McGraw-Hill Series in Marketing Show all Parts in this Series. McGraw-Hill, New York (1997)
14. Friman, M., Edvardsson, B.: A content analysis of complaints and compliments. Managing Serv. Qual. Intl. J. **13**(1), 20–26 (2003)
15. Emery, C.R., Tian, R.G.: Schoolwork as products, professors as customers: a practical teaching approach in business education. J. Educ. Bus. **78**(2), 97–102 (2002)