Continuous Preference Trend Mining for Optimal Product Design With Multiple Profit Cycles

Jungmok Ma
Enterprise Systems Optimization Laboratory,
Department of Industrial and
Enterprise Systems Engineering,
University of Illinois at Urbana-Champaign,
Urbana, IL 61801
e-mail: jma15@illinois.edu

Harrison M. Kim[1]
Associate Professor
Enterprise Systems Optimization Laboratory,
Department of Industrial and Enterprise
Systems Engineering,
University of Illinois at Urbana-Champaign,
Urbana, IL 61801
e-mail: hmkim@illinois.edu

Product and design analytics is emerging as a promising area for the analysis of large-scale data and usage of the extracted knowledge for the design of optimal system. The continuous preference trend mining (CPTM) algorithm and application proposed in this study address some fundamental challenges in the context of product and design analytics. The first contribution is the development of a new predictive trend mining technique that captures a hidden trend of customer purchase patterns from accumulated transactional data. Unlike traditional, static data mining algorithms, the CPTM does not assume stationarity but dynamically extracts valuable knowledge from customers over time. By generating trend embedded future data, the CPTM algorithm not only shows higher prediction accuracy in comparison with well-known static models but also provides essential properties that could not be achieved with previously proposed models: utilizing historical data selectively, avoiding an over-fitting problem, identifying performance information of a constructed model, and allowing a numeric prediction. The second contribution is the formulation of the initial design problem which can reveal an opportunity for multiple profit cycles. This mathematical formulation enables design engineers to optimize product design over multiple life cycles while reflecting customer preferences and technological obsolescence using the CPTM algorithm. For illustration, the developed framework is applied to an example of tablet PC design in leasing market and the result shows that the determination of optimal design is achieved over multiple life cycles.
[DOI: 10.1115/1.4026937]

Keywords: continuous preference trend mining, product and design analytics, design for multiple profit cycles, model tree

1 Introduction and Background

Product and design analytics relates and utilizes large-scale data for design decisions throughout the life cycle of complex products. The area of analytics is emerging as a promising area for the use of large-scale data generated by the users, original equipment manufacturers (OEMs), markets, and the public, which are often freely available. Recent progress in this area has garnered notable achievements in predictive product design, for example, merging predictive data mining and knowledge discovery methods with product portfolio design [1], predictive trend mining for product portfolio design [2], defining design analytics [3], and linking on-line reviews with product attributes [4,5], to name a few. The area of product and design analytics could be particularly beneficial for the area of product life cycle design and recovery, as product end-of-use state (e.g., life cycle length, product condition, product recovery decision making) is hard to predict. The current paper shows that product and design analytics can serve as a foundation to link product prelife (design and manufacturing), usage, and end-of-life operations with the development of the trend mining algorithm.

1.1 Recovery of End-Of-Life Electronics. Recovery of end-of-life electronic products has become an urgent problem that requires design engineers' attention due to multiple reasons. The first reason is the fast growing e-waste stream. The U.S. Environmental Protection Agency (EPA) estimated that 2.37×10^6 short tons of electronic products were ready for the end-of-life processes in 2009 [6]. Compared to the amount in 1999, almost 50% have increased and only 25% of them were gathered for recycling. The second reason is the fact that electronic products contain toxic and hazardous materials [7]. Lead, mercury, nickel, and palladium are examples that present negative environmental and human health effects. Reckless landfills are not an optimal solution. A third factor of the problem of recovering these products is that electronic products also contain reusable and valuable resources, such as gold, copper, tin, nickel, etc. [7]. Efficient and systematic methods to recover the reusable parts and resources are needed. Fourth, more regulations and responsibilities are emerging. The countries in the European Union have already begun adopting product take-back policy (Extended Producer Responsibility, EPR) since 1991 [8]. The U.S. has also introduced more EPR laws recently compared to 2006 [9]. Fifth, "green consumers" [10] give more pressure to companies regarding their "green" image. Now, their increased awareness of sustainability is a critical factor in determining the demand of target products. Lastly, product recovery and recycling are known to reduce the necessity for fuel consumption and landfill space, and provide substantial benefits environmentally [11].

Some OEMs such as Caterpillar, Xerox, and Sony have shown that a proper recovery system of their end-of-life products not only extends their products' lives and gives some environmental benefits, but also allows for multiple profit cycles [12–14]. These OEMs consider the end-of-life stage as the "relife" stage and return take-back components to "same-as-new" condition to customers. The relife processes or recovery options include reuse, repair, refurbishment, cannibalization, and recycling. By

[1]Corresponding author.

Contributed by the Design Automation Committee of ASME for publication in the Journal of Mechanical Design. Manuscript received May 23, 2013; final manuscript received February 14, 2014; published online April 11, 2014. Assoc. Editor: Bernard Yannou.

introducing remanufactured products back to the market, companies can find new profit opportunities and establish a greener or more environmentally friendly image.

The evidence from these OEMs indicates that the construction of a system for recovery can be a hidden source of profit. However, many factors should be considered in order to determine the profitability of the recovery system. Possible sources of uncertainties are the product's life, the state (product condition) after its life, available quantities for recovery, the reliability of a remanufactured product, customer preferences, and technological obsolescence. Product and design analytics can provide a systematic way to explain these uncertainties for multiple recoveries or multiple life cycles.

1.2 Predictive Trend Mining for Product and Design Analytics. Data mining in the context of product and design analytics was suggested as an alternative for knowledge extraction [15]. Traditionally, there are a few methods for capturing customer requirements and preferences such as quality function deployment, conjoint analysis, and discrete choice analysis. These methods resort to direct or close interactions with target customers and generate stated preference data. The strength of using data mining models is to utilize revealed preference data or accumulated data sets related to customers' actual behavior (e.g., transactional data, sales, and on-line reviews) that usually have characteristics of large volume, unstructured form, and timeliness.

Predictive trend mining is a new and emerging data mining area, which is also known as change mining [16,17] or learning concept drift [18]. Unlike traditional static data mining models with the assumption of stationarity, the predictive trend mining is a dynamic and adaptive model that captures trend or change of customer preferences over time.

A tree based data mining algorithm with predictability was used in the predictive trend mining. Tucker and Kim [2] proposed the *Discrete* Preference Trend Mining (DPTM) algorithm and suggested a classification of attributes as standard, nonstandard and obsolete with respect to a class variable for guiding design engineers. The attributes or features are also known as independent and explanatory variables, and the class variable is a dependent and response variable. It should be noted that due to the fact that the algorithm was developed to deal with discrete class variables and attributes for product portfolio concept generation, the term *Discrete* is added to the original name PTM. For example, five discrete prices {$99, $149, $179, $199, $249} were used as the class variable and no design problem was provided. Ma et al. [19] extended the work and proposed that the predictive trend mining technique called demand trend mining (DTM) can benefit optimal life cycle design problems. Utility was used as the discrete class variable and discrete choice analysis was utilized to calculate expected market shares. However, the nature of optimal design problems often requires continuous variables, e.g., price, cost, demand, etc., and discrete class variables might limit the application of design problems. In order to allow continuous variables while capturing a trend, a new method, Continuous Preference Trend Mining (CPTM), is presented in this paper.

Support vector machine (SVM) [20] is another data mining tool that can be used in the predictive trend mining. The SVM learns by example to classify different groups. Klinkenberg [18] discussed several methods to handle concept drifts based on the SVM. With concept drifts, different weighting schemes for historical data are possible, i.e., each data point over an extended period of time can be removed or utilized based on its age by allocating individual weights. Klinkenberg showed that the performance of his adaptive techniques outperformed that of simple heuristic approaches such as using all data or the most recent data in his simulated experiments. However, the proposed SVM based techniques are not for numeric prediction but for simple binary classification.

The CPTM algorithm will shed light on the initial design problem which has an opportunity for multiple profit cycles. If there are multiple recovery chances for end-of-life electronics in the near future, a trend of customer preferences and technological obsolescence will be traced and captured at the target time for the optimal initial design. The captured information will then be merged with a product design problem.

1.3 Merging Predictive Trend Mining With Design for Multiple Life Cycles. Design for multiple life cycles is a design paradigm that enables design engineers to close the loop of a product life cycle and to manage its multiple life cycles. Leasing or sales of service is a representative example of the management of multiple life cycles as shown in Fig. 1. After designing and manufacturing a product, a lessor (a person possessing a product that is being leased) would lease the product to a lessee (user of the product being leased). At the end of the lease contract or the usage stage, the lessor would take back the product and determine a proper recovery option. If it is profitable, the lessor would lease the product again for a multiple periods of time. Eventually, a product would generate multiple profit cycles, k. Many studies showed that the initial design of a product would determine 70–85% of total life cycle cost and environmental impact [21–23], so the selection of initial design attributes is the focus in this paper, especially from the economic perspective.

In order to combine the design problem with the CPTM, design for multiple life cycles is proposed to be formulated as an optimization problem. The formulation determines the design attributes that maximize the total life cycle profit and generate multiple profit cycles. Only a few studies [24,19] provided mathematical models that realize the total profit from both prelife (design and manufacturing) and end-of-life stages. Design for multiple life cycles will extend these studies.

1.4 Research Approach. In this paper, the CPTM algorithm is developed in order to take large sets of transactional data and extract valuable knowledge of customer purchase patterns. The CPTM is then tested with generated and real data sets. The architecture of the CPTM will help to predict the target class variable that reflects trend of customer preferences and technological obsolescence over time. By merging the continuous, predictive trend mining technique with an optimization model, the proposed framework will produce an optimal product design that maximizes a total unit profit and eventually reveals an opportunity for multiple profit cycles.

The rest of the paper is organized as follows. In Sec. 2, the entire methodology is explored with the CPTM algorithm and an optimal product design model for multiple profit cycles. Section 3 presents performance tests of the CPTM with various data sets. An illustration example of tablet PC design is provided in Sec. 4,

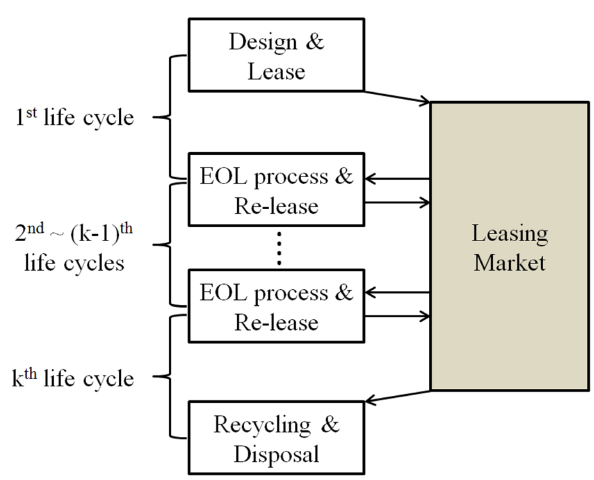

Fig. 1 Product life cycle in leasing market

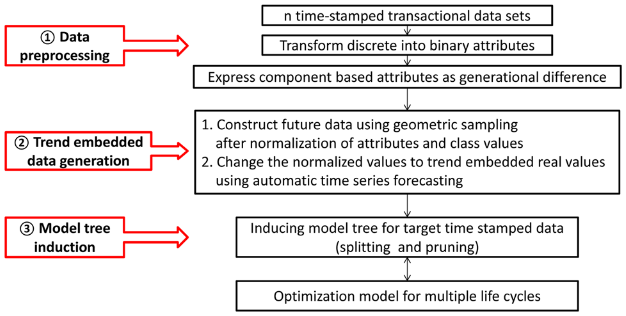

Fig. 2 Overall flow of methodology

and the conclusion and future research directions are presented in Sec. 5.

2 Methodology

The entire framework is divided into two phases. Phase 1 is to implement the CPTM algorithm, which entails data preprocessing, trend embedded future data generation, and model tree induction as shown in Fig. 2. Phase 2 involves an optimal product design for multiple profit cycles by combining the predictive model built from the CPTM.

The schematic of the CPTM algorithm shown in Fig. 3 constructs a predictive model (model tree in Sec. 2.1.3) at time $n+h$ or h periods ahead based on the historical data sets from time 1 to n. The core part of the algorithm is the generation of trend embedded data. Geometric sampling is developed to capture the trend of the relationship between design attributes and class variables by sampling normalized historical data selectively (i.e., ① and ②). Automatic time series forecasting proposed by Hyndman et al. [25] is used to predict future values of design attributes and class variables. By applying the predicted values to the normalized sampled data, unseen future data at time $n+h$ or D^{n+h} can be generated (i.e., ③). Finally, the future model tree or MT^{n+h} can be built based on the trend embedded data (i.e., ④). The two dotted boxes represent the predicted data and model, which are not available initially.

The DPTM, on the other hand, builds a predictive model (decision tree [26]) based on predicted values of Gain Ratio, one of splitting measures. The mathematical form of the Gain Ratio is defined as [27,2]

$$\text{Gain Ratio}(X) = \frac{\text{Entropy}(T) - \text{Entropy}_x(T)}{-\sum_{j=1}^{n} \frac{|T_j|}{|T|} \cdot \log_2 \frac{|T_j|}{|T|}} = \frac{-\sum_{i=1}^{k} p(c_i) \cdot \log_2 p(c_i)[\text{bits}] - \sum_{j=1}^{n} \frac{|T_j|}{|T|} \cdot \text{Entropy}(T_j)}{-\sum_{j=1}^{n} \frac{|T_j|}{|T|} \cdot \log_2 \frac{|T_j|}{|T|}} \tag{1}$$

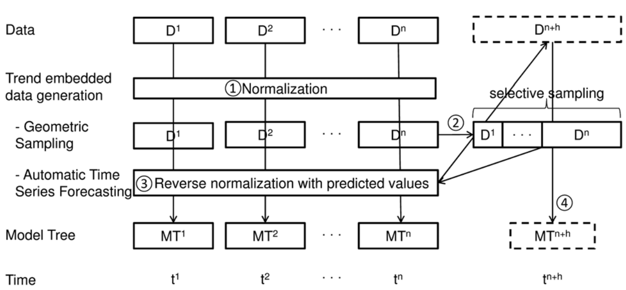

Fig. 3 A schematic of CPTM Algorithm

where X is a set of attributes, T is a data set, and T_j is a subset of the data T after splitting. The denominator represents the information generated by splitting the data set T into n partitions. The numerator represents the amount of uncertainty reduction by splitting on attribute x. Entropy quantifies the expected value of the information in bits. $p(c_i)$ represents the probability mass function of a class variable c_i and k is the number of class values.

The concept of building a tree model based on the predicted Gain Ratio was initially proposed by Böttcher and Spott [16]. The predicted Gain Ratio provides a way to build a future classification tree without real data but there are some disadvantages. First, there is a strong possibility of over-fitting since no pruning process is suggested. Highly branching trees risk over-fitting the training data and performing poorly on new samples. Pruning can help to determine the optimal size of trees. Second, no performance result of built models can be estimated since there is no test data. Third, the Gain Ratio based methods are only applicable to classification models or discrete class variables. It will be shown that by generating the target data, the CPTM algorithm can provide a way to utilize historical data selectively, avoid an over-fitting problem, and identify performance information of constructed model. More importantly, the CPTM adopts a tree induction model that allows use of continuous class variables.

Usually the process of data mining consists of data collection and selection, cleaning and transformation, pattern discovery, and interpretation. In the product design domain, text and web mining [28] provides a way for design engineers to collect and analyze customer preference data (e.g., review data), including identifying product attributes and modeling customer ratings [29–33]. In this paper, our focus is limited to the pattern discovery and interpretation stage.

2.1 Phase 1: Continuous Preference Trend Mining

2.1.1 Data Preprocessing. The first step, data preprocessing is a data preparation technique for trend mining. It starts by gathering and organizing n-time stamped transactional data sets. An example of a data set is shown in Table 6. The data set consists of a set of attributes and one class variable. In the example, there are eight different attributes of a product and a class variable, price, which customers paid in their transactions. Even though any class variables that researchers are interested in can be selected, paid price or market value was used in this study since it is directly related to customer preferences. Sales or demand can be another candidate. There is no restriction on the data except that the class variable should be continuous. Both discrete and continuous attributes can be dealt with by using the proposed approach. In this paper, only one class variable is modeled. In order to allow more than one class variable, a multivariate tree [34] can be used instead of a univariate tree (i.e., model tree).

Next, discrete attributes are transformed to a set of binary attributes. In the case of attributes with significant improvement in their values (which are component based attributes, e.g. Hard drive, CPU, etc.), the values need to be expressed as a *generational difference* [35]. The generational difference is a relative scale that can be acquired by comparing the generational gap between the target part and the latest cutting-edge part which corresponds to the minimum generation or zero. As time passes, a new part is introduced in the market and the generational difference of the existing part is increased. We assume that customers perceive the relative generational gap of components with a given time, and a company has expected values or a roadmap of the components in the near future. The generational difference is utilized to represent the technological obsolescence and its effect over time. In the appendix, an example of generational difference is shown over time and the cutting-edge part has a value of zero.

Before moving to the next step, it can be checked whether structural changes or trends are in the data. In this paper, two possible trends are identified. First, levels under each attribute and class variable can have increasing, decreasing or cyclical patterns.

For example, the display size of cell phones can have an increasing pattern. In order to detect this kind of trend, it is useful to visualize data. Statistically, Spearman's rho test of trend and Mann-Kendalls tau test of trend are available [36]. Second, there are some trends in terms of relationship between design attributes and class variables over time. For example, the memory size of notebooks might be an important factor for the purchase of the products a couple of years ago but some technological advances can change the importance of the memory size in the next year. There is no known method to detect this kind of trend but one possible way is to apply the tests of trend to the coefficients of regression models. If both trends are not detected (i.e., static case), the CPTM will generate the same result with the simple model tree, and other static models (e.g., regression, neural network, SVM, etc.) can be applied to the latest data set or the entire data set depending on the characteristics of data.

2.1.2 Trend Embedded Future Data Generation. The second step is the generation of trend embedded future data. In the previous section, two different trends were introduced in data. The automatic time series forecasting is the technique that captures the first type of trend, and the geometric sampling that is newly proposed in this paper helps to capture an underlying relationship between design attributes and a class variable (i.e., the second type of trend) by selectively utilizing historical data.

When there are a series of time stamped data points, $\{x_1, x_2, \ldots, x_n\}$, where x_t stands for a data point at time t, a couple of different techniques can be applied to forecast a data point at $n+1$ or one-step-ahead forecast. As a heuristic, it is possible to take either the latest data point or the average of all historical data for the forecast. Simple moving average is a method to smooth a time series over last k observations though the selection of k can be a heuristic. Exponential smoothing is one of well-known time series analysis methods, and the simplest form is given by [25]

$$\hat{x}_{n+1} = \lambda x_n + (1-\lambda)\hat{x}_n = \lambda x_n + \lambda(1-\lambda)x_{n-1} + \lambda(1-\lambda)^2 x_{n-2} + \cdots + \lambda(1-\lambda)^3 x_{n-3} + \lambda(1-\lambda)^{n-1}x_1 + (1-\lambda)^n \hat{x}_1 \tag{2}$$

where $\hat{x}_t$ is a forecast at time t and λ is a constant between 0 and 1. The exponential smoothing is a weighted moving average of all time series with exponentially decreasing weights defined by λ. The expanded form shows that recent values have a greater weight than old ones. A total of 30 exponential smoothing models are classified based on the combination of trend, seasonal, and error components. There are two error components (additive and multiplicative), three seasonal components (none, additive, and multiplicative), and five trend components (none, additive, multiplicative, additive-damped, and multiplicative-damped). For example, a model with all additive components can be expressed as (trend + seasonal + error) and a model with all multiplicative components is (trend × seasonal × error). Hyndman et al. [25] provided all the classifications. We adopted the automatic forecasting method [37]. First, apply all the 30 exponential smoothing models and estimate initial states and parameters using maximum likelihood estimation. Second, choose the best model according to one of the following criteria: Akaike's information criterion (AIC), corrected Akaike's information criterion (AICc) or Bayesian information criterion (BIC) [37].

After the geometric sampling process which will be introduced shortly, the sampled normalized data set for the target time is finally transformed to the real value data by applying predicted values of each attribute and class variable. The minimum and maximum values of each attribute and class at the target time are predicted (i.e., the first type of trend) by the automatic time series forecasting algorithm. By adopting the automatic algorithm, users do not need to resort to their own knowledge for models and parameters.

The second type of trend cannot be captured by time series analysis methods since the underlying relationship between design attributes and class variables is hidden. However, similar to the exponential smoothing, required traits include dynamically utilizing all past observations and applying decreasing weights in order to reflect underlying trends of the relationship between design attributes and class variables. Previously proposed trend mining models [16,2] did not consider the dynamics of relative importance of historical data. For example, Böttcher and Spott [16] used a polynomial regression method to predict the future Gain Ratio. This implicitly gave equal weights for all historical data. If historical data is available and older data sets contain more errors (this can be viewed as outliers), the accuracy of the predictive model will be diminished. The CPTM algorithm, on the other hand, provides a dynamic selection of historical data for the reflection of upcoming hidden trends by assigning exponentially decreasing weights to old data sets.

The geometric sampling is a method to sample historical data selectively for the second type of trend. Before sampling, each attribute and class variable should be normalized within a single time step. The tth term of the geometric sampling or a_t which gives the number of instances (data points) that needs to be sampled at time t is given by

$$a_t = a(1-\alpha)^{n-t} \tag{3}$$

where a is an original number of instances, (1-α) is a common ratio in geometric series, α is a smoothing factor $(0 \le \alpha \le 1)$ and n is the latest time. The smoothing factor α can be considered a characteristic of product domain in terms of relationship between design attributes and class variable. Table 1 indicates that when α is close to 1, only the latest data set is useful, and the product domain is technology sensitive and rapidly changing. When α is close to 0, all data sets are valid for future target time and the product domain has a quite insensitive and slowly changing characteristic.

In the geometric sampling, α is defined as a smoothing factor to generate $t=n$ data using $t=1$ to $t=n-1$ data when $t=1$ to $t=n$ data are available. α is obtained by

$$\arg\min_{\alpha} E \tag{4}$$

where E is a performance measure (e.g., error metrics such as mean absolute error, root mean-squared error, and relative squared error, etc.) tested for a model tree constructed from $t=1$ to $t=n-1$ data sets (as train data) with $t=n$ data (as test data). The data for building a model tree can be sampled by Eq. (3). A model tree will be introduced in the next section.

For example, if $t=1$ to $t=10$ normalized data sets are available, using $t=1$ to $t=9$ data sets, a model tree can be constructed with different α values and predicted values of attributes and class variables at $t=10$, and validated with $t=10$ data. Table 2 shows the best α example in terms of the performance measure, mean absolute error which is the average deviation between predicted and observed class variable price, with simulated data sets (each has a thousand instances or $a=1000$). For $\alpha=0.9$, the number of total instances (1111) comes from a thousand instances $(1000(1-0.9)^0)$ from $t=9$ data, a hundred instances $(1000(1-0.9)^1)$ from $t=8$ data, ten instances $(1000(1-0.9)^2)$ from $t=7$ data, and one instance $(1000(1-0.9)^3)$ from $t=6$ data based on Eq. (3). The required numbers are sampled randomly using a random number generator. The sampled normalized data becomes real value data after applying predicted values of

Table 1 α value and product domain

α value	Sampling	Product domain
≈ 1	only the latest data set	technology sensitive, drastically changing
≈ 0	all data sets	insensitive and slowly changing

Table 2 Example of best α selection

α	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Mean absolute error (MAE)	39.6	37.4	35.4	34.8	32.8	33.0	33.4	33.7	34.7	35.5	37.0
Total number of instances	9000	6125	4321	3199	2477	1998	1666	1427	1250	1111	1000

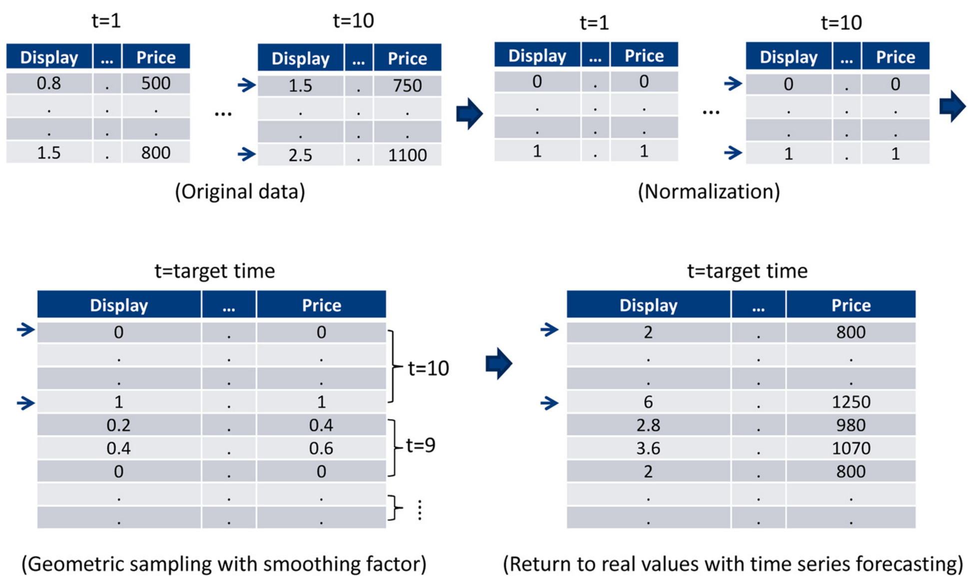

Fig. 4 Graphical example of trend embedded data generation

attributes and class variables. Then, a model tree can be built based on this data and tested with $t = 10$ data. The best α is 0.4 based on the performance measure, mean absolute error.

Based on the selected α, a number of required instances for each data set is determined and sampled as in the example but using $t = 1$ to $t = 10$ data sets this time. By applying predicted values of attributes and class variables at a target time to the sampled normalized data, trend embedded future data is finally generated at the target time. Table 2 shows that the total number of samples can be varied depending on the selected α. Based on the smoothing factor, only the latest data or all data can be used in the extreme case.

A graphical example of the trend embedded future data generation is depicted in Fig. 4. The values of original data sets are normalized within a single time step and then sampled using the geometric sampling method with the selected α. In the example, suppose that the first and the last instances were sampled from the $t = 10$ data set. By applying predicted minimum and maximum values from the time series prediction technique, real values are predicted at the target time. For example, the display size is getting bigger, and the generated target data set reflects the trend (e.g., refer to the small arrows).

By generating future data, two advantages can be achieved. First, performance information of built models can be provided similar to normal data mining processes. The predicted Gain Ratio based models in Sec. 1.2 cannot give test data but the generated data from the CPTM can work as test and validation data. The 10-fold cross-validation technique [28] is a popular way to get the performance (i.e., prediction of class variables) information when a validation data is not available. In the 10-fold cross-validation, the generated data are randomly partitioned into 10 subsamples and validation processes are repeated 10 times. Each time a model is built using 9 subsamples and validated with one remaining subsample. Then, an average performance error can be estimated. Second, pruning can be implemented based on the generated data to reduce the risk of over-fitting. The predicted Gain Ratio based models classify class values without data so that no comparison can be made between a node and subtree for the pruning. In Sec. 3, pruning in the model tree algorithm will be introduced.

2.1.3 Model Tree Induction. The third step is to build a model based on the newly generated data set from the second step. In this step, the knowledge and hidden patterns between the new values of attributes and class variables are mined using a model tree. The result of the model tree is a piecewise linear regression equation depending on a given data set, which can approximate nonlinear functions. Figure 5 shows an example of a model tree. The model tree gives three different linear models to express the nonlinearity with two attributes: A and B. On the other hand, a decision tree that was used in the other trend mining algorithms classifies discrete or categorical class variables.

The M5 model tree was initially proposed by Quinlan [38]. After comprehensive descriptions of model tree induction including a pseudocode by Wang and Witten [39], the model tree has

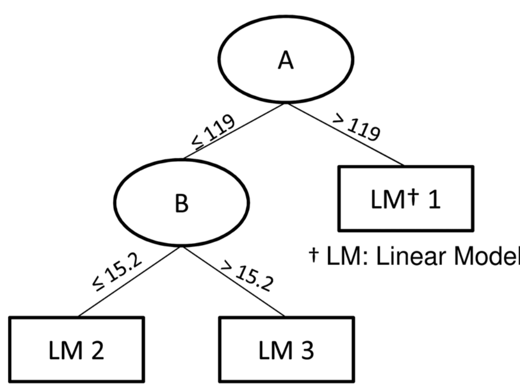

Fig. 5 Example of model tree

received attention from researchers. Wang and Witten's model tree algorithm is known as M5P. The basic operation is splitting, and the splitting is based on standard deviation reduction (SDR) in following equation [39]:

$$\mathrm{SDR} = \mathrm{stdev}(T) - \sum_i \frac{|T_i|}{|T|} \times \mathrm{stdev}(T_i) \tag{5}$$

where stdev() is a standard deviation, $|\cdot|$ stands for the number of instances, T is all instances, $T_1, T_2, \ldots$ are result sets from splitting attributes. An attribute is determined as a node when it has a maximum SDR compared to all other attributes' SDR. If no attribute can reduce a standard deviation of class values, a model tree will be identical to a simple linear regression model. For example, there are two attributes and one class variable in the model tree example in Fig. 5. stdev(T) is the standard deviation of the class values. All possible split points of the two attributes are used to estimate SDRs of the class values after splitting. Then, the one with the maximum SDR becomes the split point and the attribute of the split point becomes the node. The termination criterion of splitting in the M5P is when the number of instances is less than four or when the standard deviation at a node is less than 0.05*stdev(T). Once the splitting operation is finished, instances at the leaf nodes are used to build linear models.

A pruning procedure can reduce size of a tree and the risk of over-fitting. The M5P algorithm uses postpruning or backward pruning, which means the pruning process starts after a tree reaches a leaf node. If the lower estimated error is expected when errors in nonleaf nodes and subtree are compared, the subtrees are pruned to be leaves. The expected error of subtrees is the weighted average of each node's error by the proportion of sample sizes, and the expected error of nonleaf nodes is [39]

$$\frac{n+v}{n-v} \frac{\sum_{\text{instances}} |\text{deviation from predicted class value}|}{n} \tag{6}$$

where n is the number of instances at the nonleaf node and v is the number of parameters in a linear regression model in the node. The second fraction represents the average of absolute difference between the predicted value and the actual class value over each of instances that reach the node. The first fraction is the compensation factor to simplify the regression model in the node. In the appendix, the manual implementation of the model tree in Fig. 5 is provided with sample data. Both splitting and pruning are conducted based on the M5P algorithm.

The unique contribution of this Phase 1 is to propose a new data generation scheme for a target time, which reflects two different trends. By applying the model tree algorithm to this predicted data set, this section shows some crucial properties that could not be achieved with the previous models [16,2]: dynamic selection of historical data, avoidance of over-fitting problem, identification of performance information of constructed model, and allowance of a numeric prediction. Section 3 will show empirical test results with higher prediction accuracy.

2.2 Phase 2: Optimal product design for multiple profit cycles. As shown in Fig. 1, products can have multiple life cycles in the leasing market. When design engineers determine the initial product design over the multiple life cycles, they should consider not only the profit from the initial lease, but also the profit from the recoveries and releases, which can be a hidden source of profits. Usually, the latter part is ignored in the initial design stage due to the absence of supporting models. The CPTM results from Sec. 2.1 are expressed as model tree functions and will help to reveal the hidden source of profits.

The optimal product design for multiple profit cycles is formulated as a mathematical model and the overall architecture of the model is depicted in Fig. 6. Model tree functions are used to reflect customer preferences and technological obsolescence over time. In order to address the reliability of target products over time, a reusability function is formulated, which will give probabilities of reusable and nonreusable products. The probabilities affect the cost of end-of-life processes. While the optimizer evaluates the unit profit of a given set of attributes that are decision variables, model tree and reusability functions will take those decision variables and return the unit price and the cost of end-of-life process at a given time t, respectively.

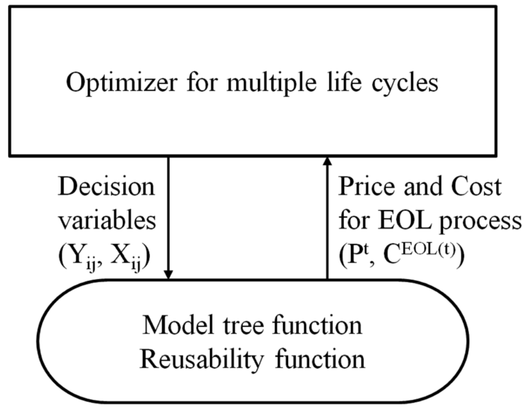

Fig. 6 Architecture of optimal design with CPTM

2.2.1 Problem Statement. The unit profit of a design over multiple life cycles is obtained by a mathematical model. The model is summarized as the following optimization problem:

Objective
- Maximize unit profit of the product for its life cycle

Constraints
- Uniqueness of design attributes

Decision variables
- Target product design attributes

Given inputs
- Historical transactional data as a set of attributes and paid price
- Generational information of parts
- Reliability information
- Cost of manufacturing and new parts
- Cost of reconditioning and logistics

2.2.2 Mathematical Formulation. **Objective Function:** The objective function is expressed as the summation of unit profits, which is the difference between unit price and unit cost at a given time t as

$$\text{Maximize } f = \sum_t \frac{1}{(1+r)^{\delta}} (p^t - c^t) \tag{7}$$

Since the multiple life cycles occur in the future, an annual interest rate r should be applied to discount the value. For the present value, $(1/(1+r)^{\delta})$ is multiplied and δ is the number of the years.

A unit price at time t is derived from the model tree function $MT^t()$ in Eq. (8). Binary decision variables, Y_{ij}, represent the level of noncomponent based attributes such as weight, size, color, etc. and X_{ij} represent the level of component based attributes or replaceable and upgradable attributes such as battery, memory, CPU, etc. A unit cost is divided into three different costs. First, when time t is the starting time (t_1), it is the production of new products, and the unit cost consists of manufacturing costs and forward logistics costs in Eq. (9). The manufacturing cost is affected by X_{ij}. If product attribute i has the level of j, X_{ij} equals 1; otherwise, it equals 0. Second, when time t is the take-back time, it is the remanufacturing of take-back products, and the unit cost

Table 3 Probability of reusable and non-reusable parts at different time *t*

Time	$t=1$	$t=2$	$\cdots$	$t=n$
Prob. of nonreusable parts	$(1-\beta)$	$(1-\beta)^2+\beta(1-\beta)=(1-\beta)$	$\cdots$	$(1-\beta)$
Prob. of reusable parts	β	$\beta^2+(1-\beta)\beta=\beta$	$\cdots$	β

consists of end-of-life process costs, and inverse and forward logistics costs in Eq. (10). Third, when a product eventually reaches the point that is not profitable (t_{end}), it will have a unit price of recycling and a cost of disposal in Eqs. (11) and (12)

$$p^t = MT^t(Y_{ij}, X_{ij}), \quad \text{where } t \neq t_{\text{end}} \tag{8}$$

$$c^t = \sum_j c_j^{\text{manufacturing}} X_{ij} + c^{\text{forwardlogistics}}, \quad \text{where } t = t_1 \tag{9}$$

$$c^t = c^{\text{inverselogistics}} + c^{\text{EOL}(t)} + c^{\text{forwardlogistics}}, \quad \text{where } t = t_{\text{take-back}} \tag{10}$$

$$p^t = p^{\text{recycling}}, \quad \text{where } t = t_{\text{end}} \tag{11}$$

$$c^t = c^{\text{inverselogistics}} + c^{\text{forwardlogistics}} + c^{\text{disposal}}, \quad \text{where } t = t_{\text{end}} \tag{12}$$

Constraints: Equation (13) imposes that each product attribute i has a unique attribute level j. In other words, finding a unique combination of each design attribute is the design problem.

$$\sum_j Y_{ij} = 1, \quad Y_{ij} \in (0,1), \quad \sum_j X_{ij} = 1, \quad X_{ij} \in (0,1) \tag{13}$$

A probability of reusable parts β or a reusability function is defined as the multiplication of each part's reliability at time t in Eq. (14). Equation (15) formulates the cost of end-of-life processes as manufacturing costs with new parts and reconditioning costs with old parts. Reconditioning is conducted with probability β. If a part is not reusable, then a new part should be used. Because a part's manufacturing cost differs by design decisions, remanufacturing with new parts is formulated as a function of X_{ij} with a probability of nonreusable parts $(1-\beta)$. Table 3 shows the probability of reusable and nonreusable parts at different time t with the assumption that the reliability of a product will be back in a state as new after end-of-life processes. Therefore, memorylessness is satisfied.

$$\beta = \prod_i \sum_j \gamma_j(t) X_{ij} \tag{14}$$

$$c^{\text{EOL}(t)} = \sum_j c_j^{\text{manufacturing}} X_{ij}(1-\beta) + c^{\text{reconditioning}} \beta \tag{15}$$

The contribution of this Phase 2 is to formulate the optimal product design model with multiple life cycles. In order to address some issues on the multiple life cycles such as customer preferences, technological obsolescence, and reliability over time, model trees from the CPTM and reusability functions are combined in the optimization model.

3 Performance Test of CPTM

In this section, a set of different data are tested with the CPTM algorithm. In order to understand the mechanism of the CPTM, Secs. 3.1 and 3.2. provide simple data sets. A real data set is also tested in Sec. 3.3 to verify the performance of the CPTM algorithm in a real situation. Section 3.4 deals with the most complex data that will be used for the statistical analysis and the application study in Sec. 4.

Four different static models were compared with the dynamic model, CPTM: linear regression, model tree (M5P), support vector machine (SMOreg), and neural network (Multilayer Perceptron). Weka [40] was used to implement these models, and the names in the parenthesis represent the equivalent algorithms. For the automatic time series forecasting, R [41] was used with the package, *forecast* [37]. All static models construct a predictive model based on the latest data set (*latest* in Table 5) or all historical data sets (*all* in Table 5) as heuristics. On the other hand, the CPTM utilizes all historical data selectively and builds a predictive model based on the generated data set. It is important to realize that the CPTM algorithm also uses the model tree but the difference is in the use of trend embedded target data.

As a performance measure, mean absolute error (MAE) and root mean-squared error (RMSE) were used [28]. Equation (16) and (17) show the MAE and the RMSE with the predicted class values, $b_1, b_2, \ldots, b_m$ and the actual class values, $d_1, d_2, \ldots, d_m$.

$$\text{Mean Absolute Error} = \frac{|b_1 - d_1| + \cdots + |b_m - d_m|}{m} \tag{16}$$

$$\text{Root Mean Squared Error} = \sqrt{\frac{|b_1 - d_1|^2 + \cdots + |b_m - d_m|^2}{m}} \tag{17}$$

3.1 Test With Data Generated From Stationary Linear Mapping Function. The data shown in Fig. 7 were generated by stationary linear mapping functions over time. There are two nominal attributes, A and B, and one class variable, Class. The values of column A increase by two and those of column B by one over time, which represents the first type of trend. In order to generate the class values, the first five instances used a mapping

A	B	Class
1	5	4.4
2	10	9.3
3	15	14
4	20	18.6
5	25	23.1
6	30	8.6
7	35	10
8	40	11.3
9	45	12.6
10	50	14.1

t=1

A	B	Class
3	6	5.7
4	11	10.4
5	16	14.9
6	21	19.3
7	26	24.2
8	31	9.2
9	36	10.9
10	41	12.2
11	46	13.4
12	51	15.2

t=2

A	B	Class
5	7	6.6
6	12	11.6
7	17	15.9
8	22	20.6
9	27	25.4
10	32	10.6
11	37	12
12	42	13.1
13	47	14.4
14	52	15.8

t=3

...

A	B	Class
15	12	12.4
16	17	17
17	22	21.5
18	27	26
19	32	30.5
20	37	15.2
21	42	16.9
22	47	18.1
23	52	19.7
24	57	21.2

t=8

A	B	Class
15	12	12.4
16	17	16.8
17	22	21.5
18	27	26.2
19	32	30.7
20	37	15.6
21	42	17.1
22	47	18.2
23	52	19.8
24	57	21.1
18	27	26.3
21	42	16.8
⋮	⋮	⋮

Total 70 instances

Trend embedded data set for t=8

Fig. 7 Data from stationary linear mapping function and generated future data

Table 4 Forecast results

			$t=7$ (Latest)	Forecast	$t=8$ (Target)
Stationary Linear Data	A	Min	13	15	15
		Max	22	24	24
	B	Min	11	12	12
		Max	56	57	57
	Class	Min	11.3	12.42	12.4
		Max	29.7	30.73	30.5
Stationary nonlinear Data	A	Min	13	15	15
		Max	22	24	24
	B	Min	11	12	12
		Max	56	57	57
	Class	Min	11.75	12.37	11.79
		Max	56.9	65.55	64.7
Real data	Class	Min	1	0.9	0
		Max	379	379	395

Table 5 Performance results

			MAE	RMSE
	Dynamic	CPTM	1.30	1.57
Stationary linear Data	Static (latest/all)	Linear Regression	3.93/3.83	5.12/5.06
		Model Tree	3.87/2.59	5.44/4.93
		SVM	3.80/3.71	5.12/5.31
		Neural Network	3.61/5.46	4.99/6.89
Stationary nonlinear data	Dynamic	CPTM	6.77	8.88
	Static (latest/all)	Linear Regression	11.80/14.64	15.75/17.35
		Model Tree	11.86/8.65	18.34/15.96
		SVM	12.73/15.68	18.19/20.98
		Neural Network	13.24/15.50	19.59/20.58
Real data	Dynamic	CPTM	13.70	18.40
	Static (latest/all)	Linear Regression	21.9/25.13	31.1/33.54
		Model Tree	18.20/14.56	25.8/9.02
		SVM	18.79/20.65	34.86/33.10
		Neural Network	17.32/20.96	21.81/24.97

function, $\text{Class} = 0.1 * A + 0.9 * B + \text{Random}(-0.2 \sim 0.2)$ and the remaining five instances used a mapping function, $\text{Class} = 0.4 * A + 0.2 * B + \text{Random}(-0.2 \sim 0.2)$ with some randomness in the functions, which represents the second type of trend. Since this is a stationary case, all data sets from $t=1$ to $t=8$ have the same mapping functions.

The goal is to construct a predictive model for $t=8$ data with $t=1$ to $t=7$ data sets. First, the values of each attribute and class variable were normalized within a single time step. Second, based on Eq. (4), the smoothing factor, $\alpha=0$, was selected using the built model tree from $t=1$ to $t=6$ data with different αs on Eq. (3) and tested with $t=7$ data in terms of the MAE. Then, the selected α gave the number of samples from each normalized data set based on Eq. (3). Since $\alpha=0$, all 70 normalized data were sampled. Third, the automatic time series forecasting was conducted for the original values of attributes A,B, and class variable Class in Table 4. By applying the predicted minimum and maximum values to the sampled normalized data, Fig. 7 shows the resulted trend embedded data set for the target time $t=8$. Finally, the model tree algorithm was applied to the generated data set and the built model tree is the predictive model from the CPTM algorithm. The model tree was pruned so that 10 linear models were reduced to only two linear models. The pruned model showed almost the same performance accuracy compared to the unpruned tree. Table 5 shows the result of the performance test.

3.2 Test With Data Generated From Stationary Nonlinear Mapping Function. The data shown in Fig. 8 were generated by stationary nonlinear mapping functions over time. Two nominal attributes, A and B are the same as in Sec. 3.1, but nonlinear mapping functions were used: $\text{Class} = 0.01 * A^2 + 0.9 * B + \text{Random}$ $(-0.2 \sim 0.2)$ for the first five instances and $\text{Class} = 0.2 * \sqrt{A} + 0.3 * B + \text{Random}(-0.2 \sim 0.2)$ for the last five instances.

The goal is to construct a predictive model for $t=8$ data with $t=1$ to $t=7$ data sets, and Fig. 8 shows the trend embedded data set for the target time. The smoothing factor, $\alpha=0.3$, was selected using the built model from $t=1$ to $t=6$ data and tested with $t=7$ data. Also the automatic time series forecasting was conducted in Table 4. Based on the smoothing factor, 27 normalized instances were sampled and the predicted values were applied to them. The model tree was pruned so that eight linear models were reduced to only two linear models. The pruned model showed a little bit higher performance accuracy compared to the unpruned tree. Table 5 shows the result of the performance test.

3.3 Test With Real Data. Second-hand values or buy-back prices of cell phones [42] were tested with the CPTM. Since the data set was obtained with the list of target cell phones, all attribute values were the same but buy-back prices were varied over time. Due to market penetration, the market value of the same products has a tendency to go down over time. After

Table 6 Example of data set (decision variables and snapshot of data)

Display size (inch)	Weight (lbs)	Hard drive (GB)	CPU (technology)	Graphics card (technology)	Memory (GB)	Battery (hours)	Touchscreen (technology)	Price ($)
9 (Y_{11})	0.8 (Y_{21})	40 (X_{11})	Core 2 duo (X_{21})	HD G (X_{31})	4 (X_{41})	6 (X_{51})	Touch D (X_{61})	p1
10 (Y_{12})	1 (Y_{22})	80 (X_{12})	Core 2 e (X_{22})	HD G 2000 (X_{32})	6 (X_{42})	12 (X_{52})	Touch C (X_{62})	p2
11 (Y_{13})	1.5 (Y_{23})	120 (X_{13})	Core i3 (X_{23})	HD G 2500 (X_{33})	8 (X_{43})	18 (X_{53})	Touch B (X_{63})	p3
12 (Y_{14})	2 (Y_{24})	250 (X_{14})	Core 2 i5 (X_{24})	HD G 3000 (X_{34})	16 (X_{44})	24 (X_{54})	Touch A (X_{64})	⋮
		320 (X_{15})	Core 2 i7 (X_{25})	HD G 4000 (X_{35})	32 (X_{45})			
		500 (X_{16})	Core 2 i7 e (X_{26})					
10	1.3	40	Core 2 i7	HD G 2500	4	6	Touch D	950
10.5	0.8	80	Core 2 duo	HD G 2000	8	24	Touch B	910
12	0.9	320	Core 2 i5	HD G 4000	32	18	Touch C	1,200
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

A	B	Class
1	5	4.4
2	10	9.5
3	15	14.6
4	20	19.8
5	25	25.1
6	30	9.7
7	35	11.2
8	40	12.7
9	45	14.1
10	50	15.7

t=1

A	B	Class
3	6	6.3
4	11	11.6
5	16	16.9
6	21	22.3
7	26	28.4
8	31	9.7
9	36	11.5
10	41	12.9
11	46	14.3
12	51	16.2

t=2

A	B	Class
5	7	8.6
6	12	14.6
7	17	20.1
8	22	26.2
9	27	32.6
10	32	10.4
11	37	12.0
12	42	13.2
13	47	14.6
14	52	16.2

t=3

…

A	B	Class
15	12	33.4
16	17	41
17	22	48.7
18	27	56.6
19	32	64.7
20	37	11.8
21	42	13.6
22	47	14.9
23	52	16.7
24	57	18.3

t=8

A	B	Class
15	12	30.2
16	17	38.5
17	22	47.3
18	27	56.4
19	32	65.6
20	37	12.4
21	42	14.3
22	47	15.7
23	52	17.8
24	57	19.4
15	12	26.2
19	32	65.6
⋮	⋮	⋮

Total 27 instances

Trend embedded data set for t=8

Fig. 8 Data from stationary nonlinear mapping function and generated future data

preprocessing the original data, monthly data sets of 155 cell phones from June 2009 to March 2010 were tested with 10 different attributes: camera pixel, talk time, touch screen, weight, memory slot, wiFi, MP3, GPS, bluetooth, and 3G.

The goal is to construct a predictive model for the $t=10$ data with $t=1$ to $t=9$ data sets. The smoothing factor, $\alpha=0$, was selected using the built model from $t=1$ to $t=8$ data and tested with $t=9$ data. Also the automatic time series forecasting was conducted in Table 4. Table 5 shows the result of performance test with this real data.

3.4 Test With Data Generated From Nonstationary Linear Mapping Function. Twenty four data sets were generated randomly with assumed ranges of attributes and some trends reflecting real-world tablet PC leasing markets. Each data set has 200 instances and 8 different attributes shown in Table 6. The first part of Table 6 explains levels of each attribute which are decision variables explored in Sec. 2.2.2. The second part indicates an example of the generated data. The data set shows transactional history and the class variable is the price that customers paid. Since this is data from nonstationary linear mapping functions, mapping functions with some randomness vary over time.

The goal is to construct a series of predictive models for the $t=n+1$ data using $t=1$ to $t=n$ data sets. n were increased by 1 from 11 to 23. For example, for an unseen $t=12$ data set, static models were constructed using the latest data set $t=11$ while the CPTM mined a trend from $t=1$ to $t=11$ data sets and constructed a tree model from a predicted data set at $t'=12$ with the calculated smoothing factor 0.5. Then, both models were evaluated

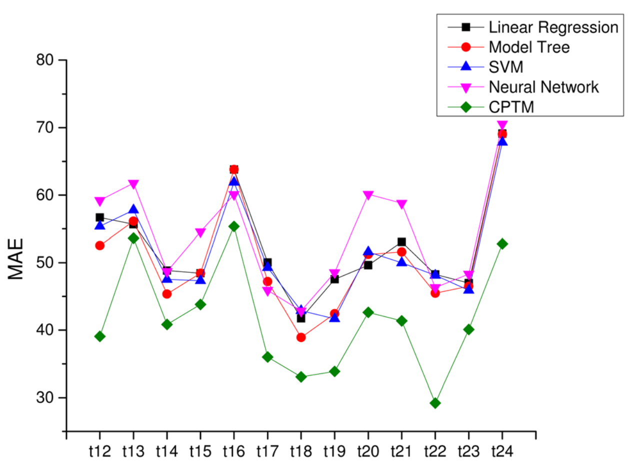

Fig. 9 Comparison of one time-ahead prediction accuracy between static and dynamic model

with real $t=12$ data set in terms of the MAE. This procedure continued up to $t=24$ (total 13 times) for one time-ahead prediction and the results are shown in Fig. 9.

In order to obtain a statistically valid conclusion between static models and the CPTM, both parametric (F and T-test) and nonparametric (Mann-Whitney) tests were employed. With a significance level of alpha $=0.05$, the accuracy of the CPTM model was significantly higher than that of static models with the generation of trend embedded data.

3.5 Discussion. The CPTM algorithm showed good predictive performances in comparison to the four well-known static models in Table 5. From the cases of data sets generated from simple stationary linear and nonlinear mapping functions, it is relatively clear to look at the effect of the geometric sampling and the time series prediction of attributes and class variables as shown in Figs. 7, 8, and Table 4. The geometric sampling helped to reflect the trend of relation between attributes and class variables over time. The automatic time series forecasting also gave good approximations of future attribute and class values. In both cases, smoothing factors were close to zero, which makes sense in that stationary mapping functions were applied over time.

The real data in Sec. 3.3 had an interesting data structure. The values of attributes were fixed but the class variable continued to change, which is why there are only predictions for class variables in Table 4. It is important to realize that even though the forecast result was similar to the latest data, the geometric sampling improved the predictive performance alone. Empirical tests showed that without a precise prediction of attribute values, the CPTM algorithm worked well with the geometric sampling. Also, without any knowledge of customer preferences and their trend over time in the used product market, the selected smoothing factor can indicate that underlying relations between attributes and class variables were quite stationary in the interval of one month. This work with real data has great potential and can provide some directions from data collection to real application in different design domains.

The last case of data from nonstationary mapping functions represents a very complex data structure, and the question was whether the CPTM worked well in this case. Due to the nonstationary nature of data, the prediction error of the CPTM was close to other static models, e.g., at $t=13$ and $t=14$, etc., in Fig. 9. However, statistical results showed that an overall performance of the CPTM is better than other models with this data.

Among those four static models, the model tree was selected for the purpose of direct comparison since the CPTM algorithm also adopts the model tree for the prediction of class variables. From all the tested data cases, the predictive performance of the CPTM outperformed that of the model tree, and this indicates that the generation of a trend embedded data set improved the accuracy.

While conducting these experiments, a total of five possible sources of variation on the result were observed: smoothing factor, model tree, time series prediction, selection of samples or random number generator, and size of samples. With the data sets from simple stationary linear and nonlinear mapping functions, it is difficult to construct a good model due to the small number of instances. Random sampling can also have a great impact with the small sample size. However, the impact from the last two factors can be minimized with large-scale data. The model tree algorithm in the CPTM is known to be fast or capable of dealing with large number of instances and attributes [43]. Empirical tests with a data set which is similar to the real buy-back price data (10 attributes) in Sec. 3.3 showed that the model tree took 1.34 s with 10^4 instances and 10.44 s with 10^5 instances running on an Intel Core i5 2.5 GHz Processor.

Moreover, the important observations are the facts that the model trees were pruned to avoid an overfitting and could generate performance information of the models by applying the 10-fold cross-validation technique. Also, continuous attributes and class variables were allowed in the models. These are aforementioned benefits of the CPTM over the DPTM from generating the trend embedded future data.

4 Application

The overall methodology in Sec. 2 was applied to tablet PC design in the leasing market. The same data sets described in Sec.3.4. were used. Weka and R in Sec. 3 also provided necessary tools for the model tree induction and the automatic time series forecasting.

4.1 Problem Setting.

4.1 Problem Setting. Tablet PCs are wireless, portable touch screen-operated computers. It is assumed that feasible candidate design attributes are defined in Table 6. It is expected that the start of leasing time is $t=12$ and the company has accumulated data sets from $t=1$ to $t=11$. A manufacturer (and lessor) should manage multiple life cycles of its tablet PC by taking back leased products and releasing after processing for the next usage-life. The goal of this problem is to find the optimal tablet designs for multiple profit cycles while considering customer preferences, technological obsolescence, and reliability. Given inputs and assumptions are as follows:

Given inputs

- Historical transactional data as a set of attributes and price
- Generational difference information in the appendix
- Reliability information in the appendix
- Manufacturing and new parts cost in the appendix
- Reconditioning cost = \$120
- Logistics cost: forward logistics = \$5, reverse logistics = \$5

Assumptions

- 2-year time frame (No disposal stage)
- Leasing period is fixed at six months
- After end-of-life processes, the reliability of a product will be back in a state as new
- Upgrade is not considered so that there are no compatibility issues during EOL processes
- Time for logistics and remanufacturing is negligible compared to that of the leasing period length

4.2 Applying CPTM. Since it was assumed that the tablet PC will have 6 months leasing time over the 2-year time frame, the number of life cycles is 4 and predictive models from $t=12$ to $t=15$ are needed by design. Static models were constructed using heuristics e.g. only the $t=11$ data set or all historical data. For the CPTM, it predicted 1 time, 2 time, 3 time, and 4 time-ahead data sets using $t=1$ to $t=11$ data sets selectively and built model trees from the predicted data sets. Table 7 presents the results. At $t=12$ all split points of the 8 attributes were used to estimate SDRs of the class vales after splitting. Since the split point 2.5 of the attribute CPU maximized the standard deviation reduction of the class values, the CPU became the first node with branches of less than or equal to 2.5 and greater than 2.5. After all of the splitting, instances at the leaf nodes were used to build linear models. The resulted model tree was pruned so that it had only three linear models, while the original tree had 169 linear models. By pruning the tree, the built model's performance was decreased based on the generated data or training data (e.g., 14.6% more errors than the unpruned tree) but the prediction accuracy of the model was improved with real $t=12$ data (e.g., 1.3% less errors than the unpruned tree) due to the generalization. The 10-fold cross-validation in Weka was used to get the performance information of the built model from the training data (i.e., the predicted data set at $t'=12$) and the prediction accuracy was calculated from the real data (i.e., the real data set at $t=12$). This shows that unpruned trees have a strong chance to be over-fitted. The comparison result between static models with the latest data set and the CPTM is shown in Fig. 10. Finally, these linear models will be used in the optimization model.

Table 7 CPTM results of illustration example

At $t=12, MT^{12}$ is defined as follows: CPU $\leq$ 2.5: LM1 CPU > 2.5: | Hard drive $\leq$ 2.5: LM2 | Hard drive > 2.5: LM3 LM num: 1 Class(Price) = − 1.0472 * Weight + 0.6497 * Hard Drive − 8.8437 * CPU − 28.1698 * Graphics Card −17.0256 * Memory − 10.779 * Battery life − 8.6369 * Touchscreen + 1014.3514 LM num: 2 Class(Price) = 7.8489 * Display size − 3.4677 * Weight − 12.11 * Hard drive − 16.2778 * CPU − 30.4275 * Graphics card − 20.6762 * Memory − 3.8257 * Battery life − 1.9893 * Touchscreen + 940.6956 LM num: 3 Class(Price) = 8.6802 * Display size − 3.4677 * Weight − 4.4711 * Hard Drive − 18.8712 * CPU − 21.8863 * Graphics card − 2.8183 * Memory − 32.5704 * Battery life − 1.9893 * Touchscreen + 990.9373
At $t=13, MT^{13}$ is defined as follows: Hard drive $\leq$ 2.5: LM1 Hard drive > 2.5: LM2 LM num: 1 Class(Price) = 8.3753 * Display size − 33.9229 * Hard Drive − 20.2652 * CPU − 37.4899 * Graphics card − 11.9472 * Memory − 7.3387 * Battery life − 8.3119 * Touchscreen + 993.0399 LM num: 2 Class(Price) = 0.7584 * Display size − 21.0345 * Weight − 6.1539 * Hard Drive − 18.1878 * CPU − 8.9427 * Graphics card − 7.4079 * Memory − 31.1933 * Battery life − 0.505 * Touchscreen + 1103.8625
At $t=14, MT^{14}$ is defined as follows: LM1 LM num: 1 Class(Price) = 16.3777 * Display size + 10.698 * Hard drive − 17.8321 * CPU − 20.8245 * Graphics card − 14.2723 * Memory − 19.7954 * Battery life + 853.3004
At $t=15, MT^{15}$ is defined as follows: LM1 LM num: 1 Class(Price) = 7.481 * Hard drive − 27.5058 * CPU − 14.6723 * Graphics card − 18.1991 * Memory − 28.7345 * Battery life − 6.7278 * Touchscreen + 1107.0882

4.3 Design for Multiple Profit Cycles. Table 8 shows the mathematical formulation of the application derived from Sec. 2.2.2. The objective function consists of unit profits from four lease contracts and the interest rate, 3%, was assumed. Prices or market values that reflect the trend of customer preferences and

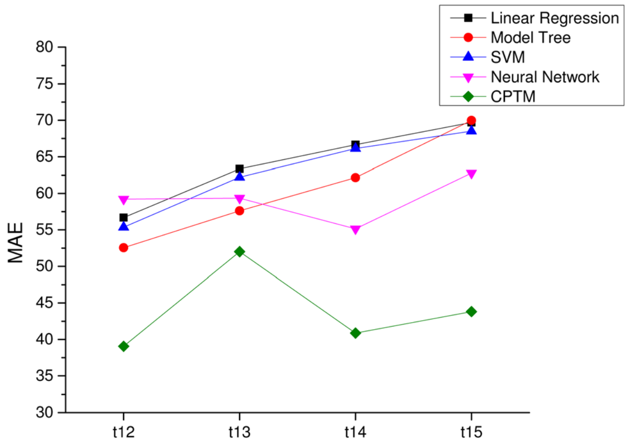

Fig. 10 Comparison of 1, 2, 3, and 4 time-ahead prediction accuracy between static and dynamic model

technological obsolescence were formulated with the model tree functions depicted in Table 7.

4.4 Discussion. Similar to the previous section for the CPTM performance, Fig. 10 indicates that the accuracy of the CPTM outperformed that of those static models even with the multiple time-ahead predictions. The accuracy was measured by the mean absolute error which is the average deviation between predicted and observed class variable price. Based on Fig. 10, the CPTM was adopted as the predictive model for this application.

The CPTM result in the Table 7 shows model trees constructed from the CPTM algorithm. At $t=12$ and $t=13$, multiple linear regression models were built and at $t=14$ and $t=15$, simple regression models were formulated to explain the class variable price. At $t=12$, the model tree consists of three linear models: LM1, LM2, and LM3. When the attribute CPU has a generational difference less than 2.5, the first linear model, LM1, is selected. If the attribute CPU has a generational difference greater than 2.5 then the attribute hard drive will work as a splitting criterion. Again, if the attribute hard drive has a generational difference less than 2.5, then the second linear model, LM2, will be selected. Otherwise, the third linear model, LM3, will be used. In each linear model, eight different design attributes in Table 6 with a constant term explain the class variable.

Excel solver with an evolutionary algorithm was used to solve the illustration design problem. The selected designs are shown in Table 9, and the total life cycle unit profits are revealed in Table 10. The selected best design attributes are 12-in. in display size, 0.8-lbs in weight, 40-GB in hard drive, Core 2 i7 e in CPU, HD G 4000 in graphics card, 8-GB in memory, 24-hour in battery life, and touch C in Touch screen with the total life cycle unit profit of

Table 8 Mathematical formulation for illustration example

Objective function

$$\text{Maximize}\, f=(p^{12}-c^{12})+\frac{1}{(1.03)^{0.5}}(p^{13}-c^{13})+\frac{1}{(1.03)}(p^{14}-c^{14})+\frac{1}{(1.03)^{1.5}}(p^{15}-c^{15})$$

$$p^{12}=MT^{12}(Y_{ij},X_{ij}),\, p^{13}=MT^{13}(Y_{ij},X_{ij}),\, p^{14}=MT^{14}(Y_{ij},X_{ij}),\, p^{15}=MT^{15}(Y_{ij},X_{ij})$$

$$c^{12}=\sum_j c_j^{\text{manufacturing}}X_{ij}+c^{\text{forwardlogistics}}$$

$$c^{13}=c^{\text{inverselogistics}}+c^{\text{EOL}(13)}+c^{\text{forwardlogistics}}$$

$$c^{14}=c^{\text{inverselogistics}}+c^{\text{EOL}(14)}+c^{\text{forwardlogistics}}$$

$$c^{15}=c^{\text{inverselogistics}}+c^{\text{EOL}(15)}+c^{\text{forwardlogistics}}$$

Constraints

$$h1:\sum_j Y_{ij}=1$$

$$h2:\sum_j X_{ij}=1$$

$$h3:c^{\text{EOL}(t)}=\sum_j c_j^{\text{manufacturing}}X_{ij}(1-\beta)+c^{\text{reconditioning}}\beta$$

$$h4:\beta=\prod_i\left(\sum_j \gamma_j X_{ij}\right)$$

$$h5:Y_{ij},X_{ij}\in(0,1)$$

Table 9 Result of optimal tablet pc design

	Display size (inch)	Weight (lbs)	Hard drive (GB)	CPU (technology)	Graphics card (technology)	Memory (GB)	Battery (hours)	Touch screen (technology)
CPTM	12	0.8	40	Core 2 i7 e	HD G 4000	8	24	Touch C
Linear Regression (latest/all)	12 / 12	0.8 / 0.8	40 / 40	Core 2 duo/Core 2 i7 e	HD G 4000/HD G 3000	32/32	24/24	Touch D/Touch A

Table 10 Result of total life cycle unit profit

		$t=12$	$t=13$	$t=14$	$t=15$	Total life cycle
CPTM	Profit($)	419	430	386	327	1562
	Price($)	949	994	951	875	3769
	Cost($)	530	564	565	548	2207
Linear Regression(latest/all)	Profit($)	477/392	390/418	346/376	232/343	1445/1529
	Price($)	962/972	919/1000	875/958	745/909	3501/3839
	Cost($)	485/580	529/582	529/582	513/566	2056/2310

$1,562. There are other design results from linear regression models (i.e., static model) with the two heuristics in Table 9. First, the "only latest data set case" selected different CPU, memory, and touch screen attributes. It is interesting that the model generated much more profits at $t = 12$ but the design selected by the CPTM brought more profits over the life cycle as shown in Table 10. Second, the "all data set case" selected different graphic card, memory, and touch screen attributes. This model generated more profits than the other heuristic but fewer profits than the CPTM. The illustration concludes that the proposed framework can identify the optimal design that maximizes the total life cycle profit based on historical transactional data sets.

From the result obtained from the artificially generated data, it can be argued that customers are very sensitive about the technological obsolescence for CPU, graphics card and battery attributes (refer to Table 6). Manufacturers should use the latest cutting edge technology for these parts. At the same time, due to the popularity of cloud storage and external storage devices, the capacity of a hard drive seems to have become less important to customers. This suggests that manufacturers can place less priority on hard drive capacity. The proposed framework enabled this type of insight, which is not readily available under the previous trend mining approaches.

The illustration does not consider the option to upgrade for the initial design selection problem. However, an additional decision making process can determine the proper end-of-life options including upgrades [35]. Given the target time, manufacturers can decide whether the decrease of generational difference (i.e., upgrade) is better than reconditioning for each component. The life cycle management plan can then be set up including upgrades.

5 Conclusion and Future Work

In this paper, a new predictive trend mining algorithm, CPTM, is developed in the context of product and design analytics. Unlike traditional, static data mining algorithms, the CPTM does not assume stationarity, and dynamically extracts valuable knowledge of customers over time. By generating trend embedded future data, the CPTM algorithm not only shows higher prediction accuracy in comparison with static models but also provides essential properties that could not be achieved with the previous trend mining algorithms: dynamic selection of historical data, avoidance of over-fitting problem, identification of performance information of constructed model, and allowance of a numeric prediction. Also, the optimization model for multiple life cycles is formulated as a binary integer programming model and combined with the CPTM result. Using the proposed framework, design engineers can select the optimal design for the target product that can generate multiple profit cycles. The illustration example of tablet PC design showed that the optimization model with the CPTM can reveal hidden profit cycles successfully.

In the future, it will be interesting to observe the impact of the prediction interval in the CPTM algorithm even though there are multiple sources of variation as discussed in Sec. 3.5. Different optimal design solutions can be obtained based on the interval. The optimization model in the illustration example was simplified in order to show the application of the CPTM. Additionally, compatibilities among different parts, different product life cycles, and product families can be considered for more interesting and realistic problems. Finally, instead of having a set of attributes as a priori, capturing of emerging attributes and management of dynamic attribute sets would be possible tasks in the future.

Acknowledgment

The work presented in this paper was supported by the National Science Foundation under Award No. CMMI-0953021. Any opinions, findings and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Nomenclature

a = number of instances for each data set
a_t = number of instances that will be sampled at time
CPTM = continuous preference trend mining
c^{disposal} = unit cost of disposal of a take-back product
$c^{\text{EOL}(t)}$ = unit cost of end-of-life processes at time t
$c^{\text{forwardlogistics}}$ = unit cost of forward logistics from factory to customers
$c^{\text{manufacturing}}$ = unit cost of manufacturing a product
$c^{\text{reverselogistics}}$ = unit cost of reverse logistics from customers to factory
c^t = unit cost of a product at time t
EOL = end-of-life
i = index for attributes
j = index for levels or options under one attribute
MAE = mean absolute error
MT^t = model tree at time t
$p^{\text{recycling}}$ = unit price of recycling a take-back product
p^t = unit price of a product at time t
r = interest rate
RMSE = root mean squared error
SDR = standard deviation reduction
X_{ij} = component based attribute as a binary decision variable
Y_{ij} = noncomponent based attribute as a binary decision variable
α = smoothing factor or characteristic of product domain
β = probability of reusable parts
γ_j = reliability of reusable part j

Appendix

A.1 Manual Implementation of Model Tree. Based on the sample data in Table 11, the model tree in Fig. 5 is manually built. The sample data has two attributes, A and B, and one class variable C.

Table 11 Sample data for model tree

A	B	C
200	14.5	10
140	20	26
90	14.4	29
98	13.5	32
86	16	34
50	24	44

Table 12 Determining a root node of model tree

C	A	midpoint	stdev(T)	stdev(T1)	stdev(T2)	SDR
44	50	68	11.2	N/A	9.5	N/A
34	86	88	11.2	7.1	9.8	2.3
29	90	94	11.2	7.6	11.4	1.7
32	98	119	11.2	6.5	11.3	3.1
26	140	170	11.2	6.9	N/A	N/A
10	200	N/A				
C	B	midpoint	stdev(T)	stdev(T1)	stdev(T2)	SDR
32	13.5	14	11.2	N/A	12.4	N/A
29	14.4	14.5	11.2	2.1	14.4	0.9
10	14.5	15.3	11.2	11.9	9.0	0.7
34	16	18	11.2	11	12.7	−0.4
26	20	22	11.2	9.5	N/A	N/A
44	24	N/A				

Table 13 Determining the second node of model tree

C	A	midpoint	stdev(T)	stdev(T1)	stdev(T2)	SDR
44	50	68	6.5	N/A	2.5	N/A
34	86	88	6.5	7.1	2.1	1.9
29	90	94	6.5	7.6	N/A	N/A
32	98	N/A				
C	B	midpoint	Stdev (T)	Stdev (T1)	stdev(T2)	SDR
32	13.5	14	6.5	N/A	7.6	N/A
29	14.4	15.2	6.5	2.1	7.1	1.9
34	16	20	6.5	2.5	N/A	N/A
44	24	N/A				

Table 12 shows all standard deviation reduction (SDR) calculations for determining the root node and splitting point of the model tree. First, the class variable and each attribute are grouped, and values of the attribute are sorted from smallest to largest. For each mid-point, calculate the standard deviation of class values in the divided groups. For example, with the mid-point 88 (the second row), stdev($T1$) and stdev($T2$) represent the standard deviation of {44, 34} and {29, 32, 26, 10}. stdev(T) represents the standard deviation of all the class values. Then, the final column SDR is calculated based on Eq. (5), and the midpoint that produces the maximum SDR is the splitting point (i.e., 119 of attribute A).

Table 14 Assumed information of generational difference

$t=11$	Hard drive	CPU	Graphics card	Memory	Battery	Touch screen
	5	5	4	4	3	3
	4	4	3	3	2	2
	3	3	2	2	1	1
	2	2	1	1	0	0
	1	1	0	0		
	0	0				
$t=12$	Hard drive	CPU	Graphics card	Memory	Battery	Touch screen
	6	5	4	5	3	3
	5	4	3	4	2	2
	4	3	2	3	1	1
	3	2	1	2	0	0
	2	1	0	1		
	1	0				
$t=13$	Hard drive	CPU	Graphics card	Memory	Battery	Touch screen
	6	6	5	5	5	3
	5	5	4	4	4	2
	4	4	3	3	3	1
	3	3	2	2	2	0
	2	2	1	1		
	1	1				
$t=14$	Hard drive	CPU	Graphics card	Memory	Battery	Touch screen
	8	7	6	6	5	5
	7	5	5	5	4	4
	6	4	4	4	3	3
	5	3	3	3	2	2
	4	2	2	2		
	3	1				
$t=15$	Hard drive	CPU	Graphics card	Memory	Battery	Touch screen
	9	8	6	7	5	5
	8	7	5	6	2	4
	7	6	4	5	1	3
	6	4	3	4	0	2
	5	3	2	3		
	4	2				

Table 15 Assumed information of reliability

$t=11$	Hard drive	CPU	Graphics card	Memory	Battery	Touch screen
	1	1	1	1	1	1
	1	1	1	1	1	1
	1	1	1	1	1	1
	1	1	1	1	1	1
	1	1	1	1		
	1	1				
$t=t+1$	Hard drive	CPU	Graphics card	Memory	Battery	Touch screen
	0.95	0.95	0.96	0.98	0.99	0.99
	0.98	0.95	0.97	0.98	0.99	0.97
	0.99	0.98	0.98	0.99	0.99	0.97
	0.99	0.99	0.98	0.99	0.98	0.91
	0.99	0.99	0.93	0.96		
	0.95	0.93				

When the value of attribute A is greater than 119, only two instances reach the node. The termination criterion of the M5P algorithm (i.e., less than four instances) stops further splitting for this branch. For the other branch, there are four instances and the standard deviation at the node (6.5) is greater than the other criterion (i.e., 0.05*stdev(T) = 0.56). After removing the instances that are greater than 119, the same procedure can be applied as shown in Table 13. In this case, two splitting points (i.e., 88 of attribute A and 15.2 of attribute B) produce the same SDR so that either of them can be selected and the model performance will be the same. Figure 5 shows the case that 15.2 of attribute B is selected. All the nodes have less than four instances so that the splitting operation of the model tree is completed. Finally, the instances at the leaf nodes are used for regression models. Due to the small number of instances, all leaf nodes take a simple model i.e. LM1: C = 18, LM2: C = 30.5, and LM3: C = 39.

A pruning procedure compares the expected error of leaf nodes and a nonleaf node. The nonleaf node B has two leaf nodes and their expected error can be calculated as follows: the absolute difference between the predicted and the actual class value is averaged at the each node and weighted by the proportion of sample sizes $((2/4)(|(29-30.5|/2)+(|32-30.5|/2))+(2/4)((|34-39|/2)+(|44-39|/2))=3.25)$. The internal regression model at the node B (C = 1.31*B + 12.59) is then used to calculate the expected error (2.05) based on Eq. (6). Also by dropping the parameter of the internal regression model (C = 34.75), another expected error (4.63) can be calculated but this model can be ignored due to the higher expected error. Since the expected error of the node B is lower than that of the leaf nodes, the tree should be pruned and the internal regression model becomes a leaf node. Similarly, by comparing the node A and leaf nodes, the pruning operation can be determined and it turns out that the tree should be pruned. After the pruning procedure, one regression model (C = −0.21*A + 52.21) replaces the three regression models.

A.2 Assumed Information in Application. Tables 14–16 show the assumed information of generational difference, reliability, and cost for manufacturing and new parts, respectively.

Table 16 Assumed information of cost for manufacturing and new parts ($)

Hard drive	CPU	Graphics card	Memory	Battery	Touch screen
40	60	90	20	40	50
55	70	110	30	50	60
75	90	130	40	60	85
90	100	140	80	70	100
100	110	160	135		
120	125				

References

[1] Tucker, C. S., and Kim, H. M., 2008, "Optimal Product Portfolio Formulation by Merging Predictive Data Mining With Multilevel Optimization," ASME J. Mech. Des., **130**(4), pp. 991–1000.

[2] Tucker, C. S., and Kim, H. M., 2011, "Trend Mining for Predictive Product Design," ASME J. Mech. Des., **133**(11), p. 111008.

[3] Van Horn, D., Olewnik, A., and Lewis, K., 2012, "Design Analytics: Capturing, Understanding and Meeting Customer Needs Using Big Data," ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2011), Paper No. DETC2012-71038.

[4] Tucker, C. S., 2011, "Data Trend Mining Design for Predictive Systems Design," Ph.D. thesis, University of Illinois, Chicago, IL.

[5] Rai, R., 2012, "Identifying Key Product Attributes and Their Importance Levels From Online Customer Reviews," ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2011), Paper No. DETC2012-70493.

[6] Environmental Protection Agency, 2011, "Electronics Waste Management in the United States Through 2009," U.S. EPA, May, Report EPA No. 530-R-11-002.

[7] Sodhi, M. S., and Reimer, B., 2001, "Models for Recycling Electronics End-Of-Life Products," OR Spektrum, **23**(1), pp. 97–115.

[8] Fishbein, B. K., 1998, "EPR: What Does it Mean? Where is it headed?," P2: Pollution Prevention Rev., **8**(4), pp. 43–55.

[9] Product Stewardship Institute, 2012, "Extended Producer Responsibility State Laws." Available at: http://productstewardship.us (accessed in May 2013).

[10] Wagner, S., 2003, *Understanding Green Consumer Behaviour: A Qualitative Cognitive Approach*, Consumer Research and Policy Series, Taylor & Francis Group.

[11] Environmental Protection Agency, 2011, "Benefits of the Remanufacturing Exclusion: Background Document in Support of the Definition of Solid Waste Rule," June, Washington, DC.

[12] Hucal, M., 2008, "Product Recycling Creates Multiple Lives for Caterpillar Machines," Peoria Magazines, September.

[13] King, A., Miemczyk, J., and Bufton, D., 2006, "Photocopier Remanufacturing at Xerox uk a Description of the Process and Consideration of Future Policy Issues," *Innovation in Life Cycle Engineering and Sustainable Development*, D. Brissaud, S. Tichkiewitch, and P. Zwolinski, eds., Springer Netherlands, pp. 173–186.

[14] Parker, D., and Butler, P., 2007, "An Introduction to Remanufacturing." Available at: http://www.remanufacturing.org.uk (accessed in May 2013).

[15] Kusiak, A., and Smith, M., 2007, "Data Mining in Design of Products and Production Systems," Annu. Rev. Control, **31**(1), pp. 147–156.

[16] Böttcher, M., Spott, M., and Kruse, R., 2008, "Predicting Future Decision Trees From Evolving Data," Proceedings of ICDM'08, pp. 33–42.

[17] Böttcher, M., 2011, "Contrast and Change Mining," Wiley Interdiscip. Rev.: Data Mining Knowledge Discovery, **1**(3), pp. 215–230.

[18] Klinkenberg, R., 2004. "Learning Drifting Concepts: Example Selection vs. Example Weighting," Intell. Data Anal., **8**(3), pp. 281–300. Available at: http://www.iospress.nl/

[19] Ma, J., Kwak, M., and Kim, H. M., 2014. "Demand Trend Mining for Predictive Life Cycle Design," J. Clean. Prod.

[20] Vapnik, V. N., 1998, *Statistical Learning Theory*, Wiley-Interscience, Hoboken, NJ.

[21] Fixson, S. K., 2004, "Assessing Product Architecture Costing: Product life cycles, Allocation Rules, and Cost Models," ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2004), Paper No. DETC2004-57458.

[22] Duverlie, P., and Castelain, J. M., 1999, "Cost Estimation During Design Step: Parametric Method Versus Case Based Reasoning Method," Int. J. Adv. Manuf. Technol., **15**(12), pp. 895–906.

[23] Seo, K., Park, J., Jang, D., and Wallace, D., 2002, "Approximate Estimation of the Product Life Cycle Cost Using Artificial Neural Networks in Conceptual Design," Int. J. Adv. Manuf. Technol., **19**(6), pp. 461–471.

[24] Zhao, Y., Pandey, V., Kim, H. M., and Thurston, D., 2010, "Varying Lifecycle Lengths Within a Product Take-Back Portfolio," ASME J. Mech. Des., **132**(9), p. 091012.

[25] Hyndman, R., Koehler, A., Ord, J. K., and Snyder, R., 2008, *Forecasting with Exponential Smoothing: The State Space Approach*, Springer-Verlag, Berlin, Heidelberg.

[26] Quinlan, J. R., 1993, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Series in Machine Learning, Morgan Kaufmann Publishers.

[27] Quinlan, J. R., 1986, "Induction of Decision Trees," Mach. Learn., **1**(1), pp. 81–106.

[28] Witten, I., and Frank, E., 2005, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed., The Morgan Kaufmann Series in Data Management Systems, Elsevier Science.

[29] Cheung, K.-W., Kwok, J. T., Law, M. H., and Tsui, K.-C., 2003, "Mining Customer Product Ratings for Personalized Marketing," Decision Support Syst., **35**(2), pp. 231–243.

[30] Archak, N., Ghose, A., and Ipeirotis, P. G., 2011, "Deriving the Pricing Power of Product Features by Mining Consumer Reviews," Manage. Sci., **57**(8), pp. 1485–1509.

[31] Ferreira, L., Jakob, N., and Gurevych, I., 2008, "A Comparative Study of Feature Extraction Algorithms in Customer Reviews," 2008 IEEE International Conference on Semantic Computing, pp. 144–151.

[32] Abulaish, M., Jahiruddin, Doja, M. N., and Ahmad, T., 2009, "Feature and Opinion Mining for Customer Review Summarization," Proceedings of the 3rd International Conference on Pattern Recognition and Machine Intelligence, PReMI'09, Springer-Verlag, pp. 219–224.

[33] Decker, R., and Trusov, M., 2010, "Estimating Aggregate Consumer Preferences From Online Product Reviews," Int. J. Res. Market., **27**(4), pp. 293–307.

[34] De'ath, G., 2002, "Multivariate Regression Trees: A New Technique for Modeling Species-Environment Relationships," Ecology, **83**(4), pp. 1105–1117.

[35] Kwak, M., and Kim, H. M., 2011, "Market-Driven Positioning of Remanufactured Product for Design for Remanufacturing With Part Upgrade," ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2011), Paper No. DETC2011-48432.

[36] Yue, S., Pilon, P., and Cavadias, G., 2002, "Power of the Mannkendall and Spearman's Rho Tests for Detecting Monotonic Trends in Hydrological Series," J. Hydrol., **259**(14), pp. 254–271.

[37] Hyndman, R., and Khandakar, Y., 2008, "Automatic Time Series Forecasting: The Forecast Package for R," J. Stat. Softw., **27**(3), pp. 1–22.

[38] Quinlan, J. R., 1992, *Learning With Continuous Classes*, World Scientific, Singapore, pp. 343–348.

[39] Wang, Y., and Witten, I. H., 1997, "Inducing Model Trees for Continuous Classes," Proceedings of the 9th European Conference on Machine Learning Poster Papers, pp. 128–137.

[40] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H., 2009, "The Weka Data Mining Software: An Update," SIGKDD Explor. Newsl., **11**(1), pp. 10–18.

[41] R Development Core Team, 2008, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.

[42] Kwak, M., Kim, H. M., and Thurston, D., 2012, "Formulating Second-Hand Market Value as a Function of Product Specifications, Age, and Conditions," ASME J. Mech. Des., **134**(3), p. 032001.

[43] Shrestha, D. L., and Solomatine, D. P., 2006, "Machine Learning Approaches for Estimation of Prediction Interval for the Model Output," Neural Netw., **19**(2), pp. 225–235.