Manufacturing Cost Prediction
in the Presence of Categorical and Numeric Design Attributes

by

Eren Sakinc

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
August 6, 2016

Keywords: Cost Prediction, Clustering, Partitioning around Medoids,
Categorical Regression Splines, Kernel Weighting, Tensor Product Splines

Approved by

Alice E. Smith, Joe W. Forehand/Accenture Professor of Industrial and Systems Engineering
Saeed Maghsoodloo, Professor Emeritus of Industrial and Systems Engineering
Nedret Billor, Professor of Mathematics and Statistics
Fadel Megahed, Assistant Professor of Industrial and Systems Engineering

Abstract

Manufacturing processes require not only physical operation capabilities but also non-physical management policies. When designing a new product or manufacturing a customer's new unique design, the focal point is to establish a price which maximizes customer value while still being profitable. Since an irreversible and large amount of capital is tied up in production elements, estimating manufacturing costs accurately is critical. Therefore, final decisions about the product price should be based on analytical approaches, instead of intuitive expectations. Poorly established product prices that are a function of product cost may cause two unfavorable consequences: (1) A potential loss of profit due to the gap between the expected cost and the actual cost, (2) A loss of customers and goodwill due to higher prices than necessary. In this research, we investigate ways of using clustering and spline methods to predict the manufacturing cost of products in the presence of complex numeric and categorical design attributes (cost drivers). The accuracy of the methodology presented in this work is assessed in comparison to a traditional approach, a polynomial regression model. The main concern behind this research is to predict the manufacturing cost of a product quickly and accurately without making assumptions on statistical distributions or estimating model parameters to simplify the complex relationship between categorical and numeric product design attributes and the manufacturing cost.

Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisor, Dr. Alice E. Smith, for her continuous support and patience. I have been extremely fortunate to have her as my advisor giving me the freedom to explore on my own, and at the same time guiding with brilliant ideas. I am really thankful to her for carefully reading and commenting on my research manuscript, especially encouraging me the use of correct grammar. She consistently sets high standards for her students.

I must also recognize my committee members, Dr. Saeed Maghsoodloo, Dr. Nedret Billor and Dr. Fadel Megahed for their time and constructive criticism. In addition, I am deeply thankful to the companies, Radsan, Lux Plastic and Gulecler Socks & Textile, for supplying critical manufacturing data for this research.

This dissertation is dedicated to my family, my mother, Serin, my father, Haydar Ali and my sister, Evrim. I would like to thank them with my most sincere emotions for their unconditional spiritual support throughout writing this dissertation, my doctoral training and my life in general. Without their presence, this work would not have been possible.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ABS	Acrylonitrile Butadiene Styrene
ARE	Absolute Relative Error
CCC	Cubic Clustering Criterion
CCE	Clinical Cost Estimation
crs	Categorical Regression Splines
DS	Dataset
FCM	Fuzzy *c*-means
HA	Hierarchical
MARE	Mean Absolute Relative Error
MARS	Multivariate Adaptive Regression Splines
MCE	Manufacturing Cost Estimation
MSE	Mean Squared Error
MST	Minimum Spanning Tree
N/A	Not Applicable
NHA	Non-hierarchical
NOMAD	Nonsmooth Optimization by Mesh Adaptive Direct Search
PAM	Partitioning Around Medoids
PSF	Calinski and Harabasz’s Pseudo F
PST2	Pseudo T^2

PVC	Poly Vinyl Chloride
RMSE	Root Mean Squared Error
SE	Squared Error
SCE	Software Cost Estimation
TL	Turkish Lira

Chapter 1

Introduction

Regardless of the scale of a manufacturing facility, customers check catalogs and ask price quotes for some specific quantity of products. They even bring their unique product design to manufacturers and ask for an estimate. This even ends up with fierce price negotiation sessions. But, what is the negotiation power of a manager over a product? How much can she/he discount from the regular price tag? It is possible to measure the actual cost of an ongoing product, but is it possible to know the cost of a new and unique design when it has not been actually manufactured?

In economics, we assume that people behave rationally and their aim is to maximize their benefits. If decision makers in a manufacturing factory act parallel to this idea, they identify cost drivers and find the cost of a specific product before they establish its final price. Remember that the basic mathematical expression of the profit is the sale price minus the cost. Since the sale price of a product is determined as a margin of its manufacturing cost, the profit turns out to be a function of the product cost. This is the proof of the dependency and the direct connection of a product price on its actual manufacturing cost.

Manufacturing processes require not only physical operation capabilities but also non-physical management policies. Executives allocate a big investment in machines, raw material and people, who directly and indirectly run the factory. Even though these capital expenditure decisions are made through a series of careful steps, there is still a possibility to observe deviations on the factory floor from what was planned or expected. Therefore, final decisions about the product price should be based on actual cost, instead of ad hoc expectations. Poorly established product prices (remember that it is a function of product cost) may cause two unfavorable consequences: (1) A potential loss of profit due to the gap between the expected cost and the actual cost, (2) A loss of customers and goodwill due to higher prices than competitors in the market.

As time goes on, factories develop new manufacturing skills or improve their current systems. This highly ambitious business environment pushes the boundaries of manufacturers to reduce their manufacturing costs and, eventually, product prices. Manufacturing a product is not only about considering the design details, functionality and style but also considering its monetary attractiveness. This monetary attractiveness can be achieved when the cost of a product is established accurately. Since the profit depends on manufacturing cost, an accurate estimation of manufacturing cost of a product is crucial.

Additionally, when designing a new product or manufacturing a customer's new unique design, the focal point is to establish a price which maximizes the customer satisfaction while being profitable. Since an irreversible and large amount of capital (with respect to the scale of the facility) is tied up in production elements, decision makers in manufacturing systems should be aware of the significance of estimating manufacturing costs accurately before any action is taken. That could be an explanation of why cost estimation efforts have gained substantial attention since the beginning of the modern industrial era. Accurately estimating manufacturing cost of a product is imperative to survive in this highly technological and challenging environment.

1.1 Problem Definition

Statistical tools have always been popular among executive planners when cost estimation effort takes place. Before proceeding forward into statistics, we need to know the cost structure of a product which consists of a collection of cost drivers. A cost driver is defined as any factor which changes the cost of an activity*. From a statistical perspective, cost drivers are explanatory variables that have a contribution to the manufacturing cost of products. That is, the manufacturing cost is the dependent variable which is influenced by cost drivers. Through this dissertation, term analogies for cost drivers are cost variables, design variables, design attributes or, simply, variables and attributes. The variables which have influence on product cost begin from the early product design stage. Shape complexity, main material type, manufacturing tolerances, manufacturing schedule are some of these variables. However, there are other cost drivers, as well as uncertainties, existing in the nature of production systems. Inflation, cost of capital, failure of tooling and machines, local and global regulations and other similar factors cannot be adjusted directly by manufacturing operation planners. Considering these uncertainties, assigning probability density or mass functions to cost drivers while taking correlations into account can be a good start to mimic cost behavior. Along with a point estimate, confidence intervals for the cost of a particular product can be derived as an output of this Monte Carlo simulation. Even though assigning stochastic distributions to cost drivers and then using the Monte Carlo simulation is a practical way to handle uncertainty, is it necessarily realistic? How accurate are these distribution assignments? Or more specifically, is a cost variable coming from the same probability density or mass functions for all products? In addition, many other questions have to be answered confidently before using Monte Carlo simulation.

* According to Chartered Institute of Management Accountants (CIMA)

Similar to the simulation idea just discussed, it is possible to use empirical distributions instead of well-known parametrical distributions for cost variables. No matter how complicated the actual distribution is, we may use Efron's bootstrapping [1] tool to simulate the product cost behavior. One of the questions that arises with bootstrapping is whether we can diagnose the failure of a non-parametric simulation. If a well-known parametric model is fitted to a dataset, consistency of non-parametric bootstrap sampling from the same dataset can be monitored by comparing the consistency of the results with the parametrical benchmark. However, absence of such a parametric model may leave us clueless when the impact of outliers is investigated [2]. Additionally, it may fail to provide appropriate results when the fundamental assumption is not fulfilled. That is, non-parametrical bootstrap samples should be independent and identically distributed from an unknown distribution [1].

The main concern of our research is to predict the manufacturing cost of a product without dealing with probability density or mass function assignments or making strong assumptions on parameters. We will convert physical similarities of products into meaningful mathematical similarities and make product-by-product comparisons. When making product-by-product comparisons, the number of analogies is likely to grow as the number of products grows. Therefore, over a diverse product family, establishing only a single accurate estimation model is challenging and doubtful. This motivates us making comparisons by dividing the whole database of products into neighborhoods until these neighborhoods become sufficiently homogenous. Using statistical terminology, we can call these neighborhoods, groups or clusters. We develop cost estimation models for each cluster. That is, a polynomial regression model is built for each cluster. Since every current and historical product can be represented as points in multidimensional space with respect to their variables, we will investigate in which cluster a new product falls. After assigning the new product to the best cluster, we will use the cluster-specific model to predict its manufacturing cost.

For an illustration, see Figures 1.2, 1.3 and 1.4. Assume that there are only two variables associated with a family of products. The first variable, “Manufacturing Cost”, is the outcome variable, while the second one, “Design Variable”, is the independent. In each figure, black dots represent products on two-dimensional space which are scattered according to the change in the design variable (horizontal axis) and the manufacturing cost (vertical axis). In Figure 1.1, establishing a single linear estimation model (the red colored trend line with the EM label) over a diverse product family may not be as accurate as possible. For product A in the same figure, observing high deviation in the predicted cost (C_A') from the actual cost value (C_A) is very likely when a single estimation model is used.

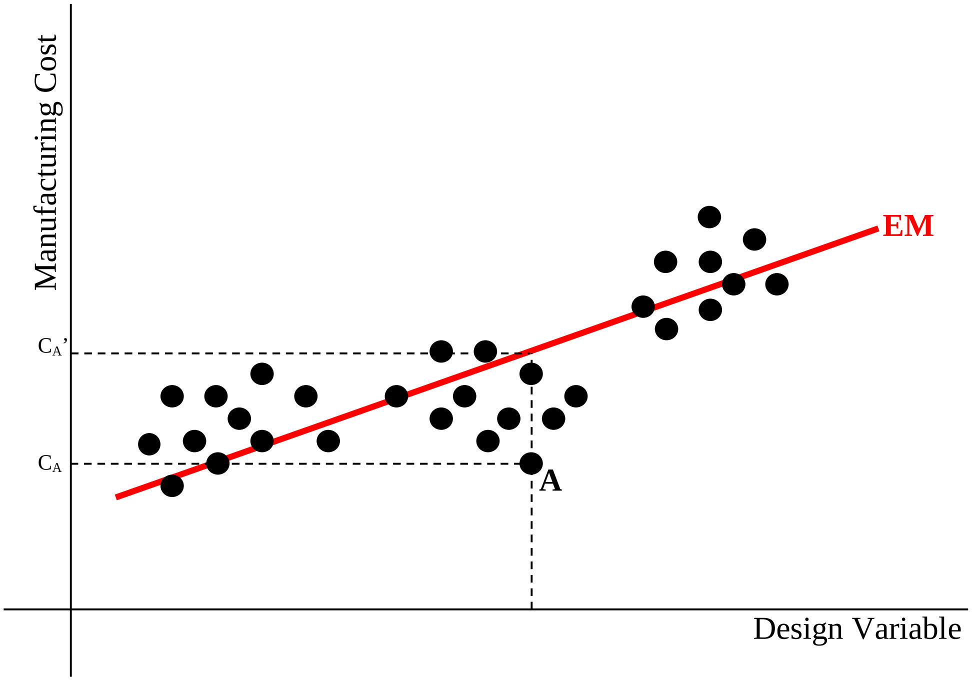


Figure 1.1: Scatter plot of products over Manufacturing Cost vs. Design Variable

However, as shown in Figure 1.2, we may partition points into three meaningful and fairly homogenous clusters according to their design similarities (or dissimilarities). Note that in this figure clusters are formed based on visual observations with respect to the measurement differences between the design variable for each pair of objects. All red dots are the representative objects which are the centers of their corresponding clusters and specifically called “medoids”. As in Figure 1.3, for each cluster, separate estimation models can be developed for a higher predicted cost accuracy than the performance of the original single model EM. EM_1, EM_2 and EM_3 are the models which are specifically established to estimate the cost of products in their underlying clusters. By using the model EM_2, the gap between the predicted cost (C'_A) and the actual cost value (C_A) for product A is relatively smaller than the single model case (as in Figure 1.1). The desired value of the gap between the predicted cost and the actual cost is zero, but in practice, it is very hard to achieve for all products.

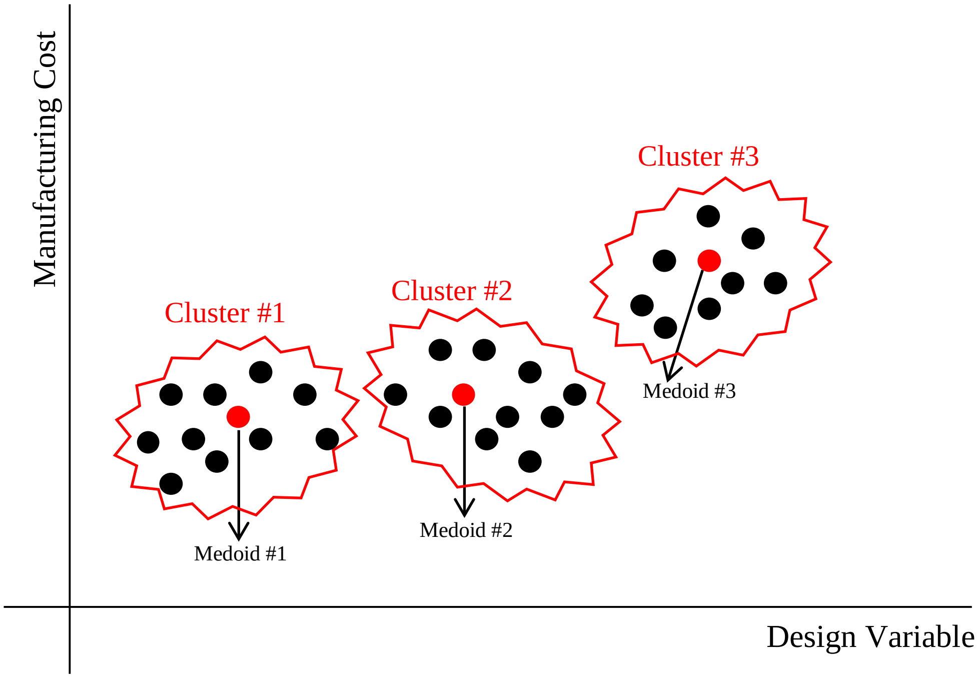


Figure 1.2: Representation of the product family in three clusters

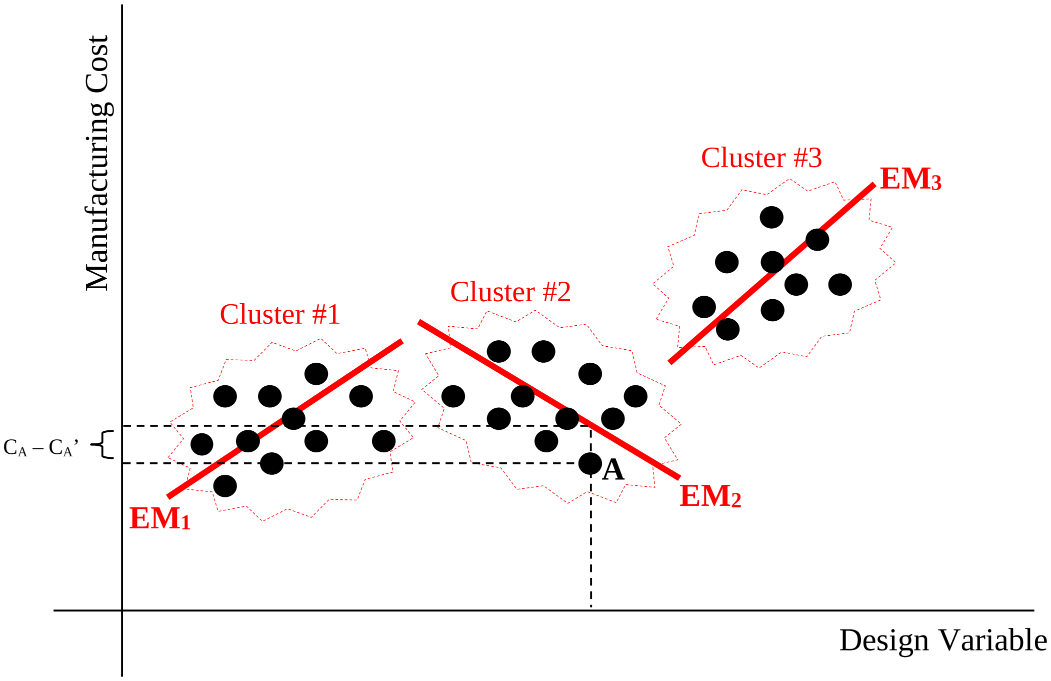


Figure 1.3: Establishing individual cost estimation models for each cluster

When cluster specific models are considered within their defined ranges, at the boundaries they are non-continuous but can form piecewise functions. Since the main concern of this research is to predict the manufacturing cost of a product with non-parametric methods, one alternative is to use splines. We can define a spline as a function that is constructed by piecewise polynomial functions where these polynomial segments connect. Our research also seeks the possibility of building spline models to accommodate cost estimation process with improved accuracy. It enables us the compare the performance of the clustering approach and of the spline approach, and find whether any of these methods are superior to a polynomial regression model built with the entire product stream.

The word “spline” comes from the East Anglian dialect with a meaning of a thin wood or metal piece [3] that was used by shipbuilders and draftsmen to build smooth curves. With the advance of computer technology, using spline models has become a popular approach to produce smooth curves in computer graphics [4]. Spline functions project a high degree of smoothness at the points that segments connect. These connection points are called knots. Figure 1.4 shows a piecewise spline function with two interior knots. That is, the three segments in the function represent three clusters for a single design attribute. Using splines is a potential improvement in the cost estimation process and a powerful candidate to compare the performance of the clustering approach.

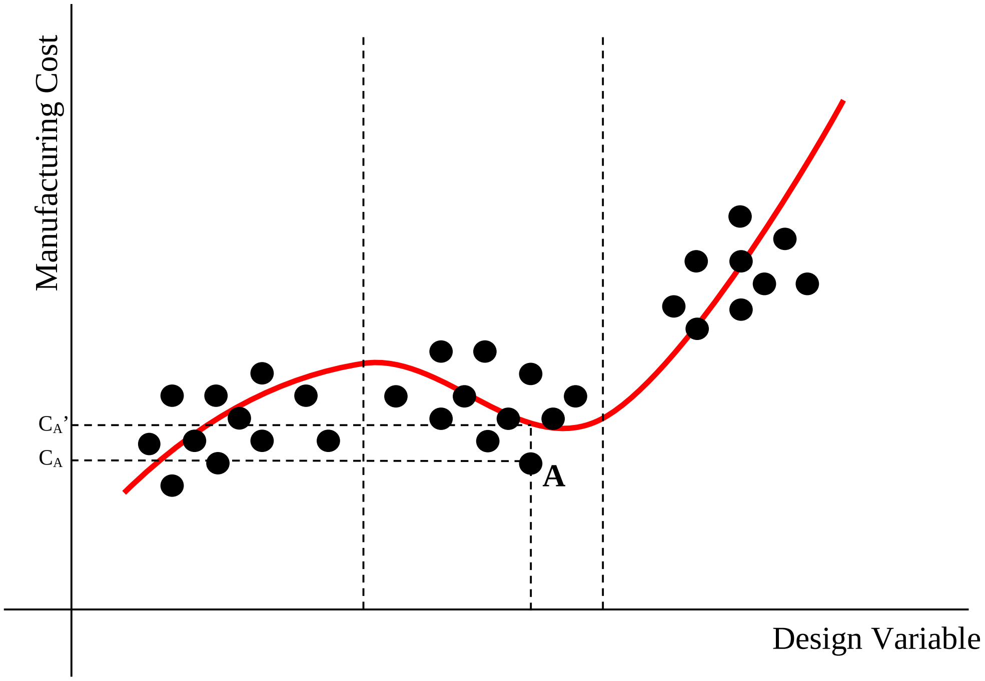


Figure 1.4: Establishing a spline cost estimation model with three segments

We can summarize our research with the following three questions.

Problem 1. *How accurately can the cost of a product be predicted by using a clustering method?*

Our main purpose is to establish the cost of a product as simply and accurately as possible. We decompose the problem into two phases: (1) Grouping similar products together by using the k-medoids algorithm and establishing cost estimation regression models for each group, (2) Assessing the best cluster to assign a new unique product and then predicting the manufacturing cost of the new product with the corresponding cost estimation regression model.

According to the Jain and Dubes [5] definition, clustering analysis is the process of grouping objects into meaningful subsets. Clustering has been very popular among biologists, social scientists, engineers and many other occupations to find significant taxonomies behind data. There are several clustering methods applicable to manufacturing cost estimation problems. Two main categories are hierarchical (HA) and non-hierarchical (NHA) techniques. Selection of the clustering method depends on the specific dataset because of feasibility concerns. Using a HA or NHA technique may not be possible for every instance. That is, some clustering methods are not applicable to some problems because of method limitations.

We investigated ways of using clustering methods to predict the manufacturing cost of a product without actually manufacturing it. We only focused on clustering methods which are applicable to the majority of manufacturing cost estimation problems. Due to the limitations of clustering algorithms on mixed numeric and categorical data, instead of using calculated virtual reference points as cluster centers, using real objects (medoids) is preferred. This is why our starting basis is partitioning according to medoids, specifically the k-medoids algorithm. We need to supply as good as possible clustering content to build estimation models because poorly-constructed clusters would not provide the true and necessary information to calculate a valid and robust estimation model. There are several ways to build estimation models for manufacturing costs estimation purposes. Certainly, the simplest and the most prevalent way is to fit data to a regression model because and this is typically the most sophisticated way used in practice. A more primitive cost estimation approach using experiential knowledge is the dominant method used in practice.

We assess if our prediction is more precise than the prediction in absence of a clustering method. In addition, we also inquire if our prediction is reasonable compared with the actual cost. Constructing regression models is one of the focal points in this research. However, it is necessary to emphasis that other modeling approaches can be used such as neural networks, regression trees, or fuzzy logic techniques. We leave this part open for future research.

Problem 2. *How can the appropriate number of clusters be determined to obtain the most precise estimate in the presence of categorical and numeric design attributes?*

In this study, we considered several approaches to find an appropriate number of clusters which satisfies a certain amount of homogeneity among groups. Even though there is no strictly, statistically binding constraint for the maximum number of clusters, it is preferred to employ the smallest number of clusters while maximizing the inter-cluster variability relative to the within-cluster variability [6]. Our methodology of selecting the appropriate number of clusters is neither deterministic nor arbitrary. We look for consensus among three statistics through plots of the C-index, Gamma and silhouette width, where local peaks of the Gamma and silhouette width combined with local troughs of the C-index is our choice for the number of clusters. Note that this choice may not be unique. If every product forms an individual cluster, it would be pointless to use a clustering method in order to estimate the manufacturing cost. With a similar idea, using a single cluster for the entire product database would be exactly same as not using any clustering method.

The possible number of allocations of products (totally N products) among a fixed number of clusters (k clusters) can be expressed as a combinatorial problem. In mathematical terminology, this is called the pigeonhole principle [7]. It is the placement problem of N pigeons into k pigeonholes, where N is greater than k and at least one pigeonhole must contain more than one pigeon. The most important point in this combinatorial clustering problem is that the products are not identical and transfer of a product to another cluster may violate the assignment principle. Choosing k representative points out of N observations with a combination gives the exact number of assignment possibilities. However, the strategy is not randomly assigning observations to clusters but partitioning them according to the smallest distance from each cluster center (medoid). That is, every possible allocation of N products into k clusters is not meaningful (feasible) because products are assigned only to the closest cluster center. The distance between a product and a cluster center is determined by a similarity (or dissimilarity) measure. A further discussion about similarity measures can be found under section 2.

Figure 1.5 illustrates the feasible allocation strategy with a closer snapshot of Cluster #2 and Cluster #3 in Figures 1.2 and 1.3. The distance between object B and Medoid #2 or in other words, the dissimilarity of object B with Medoid #2, and the distance between object B and Medoid #3 are marked d_1 and d_2, respectively. The smaller the distance between two objects gets, the higher the level of similarity obtained. That is, object B and Medoid #2 are more dissimilar than are object B and Medoid #3. In other words, object B and Medoid #3 are more similar to each other than are object B and Medoid #2. The distance d_1 is larger than the distance d_2 and, therefore, the assignment of object B to Cluster #3 is preferred (or necessary).

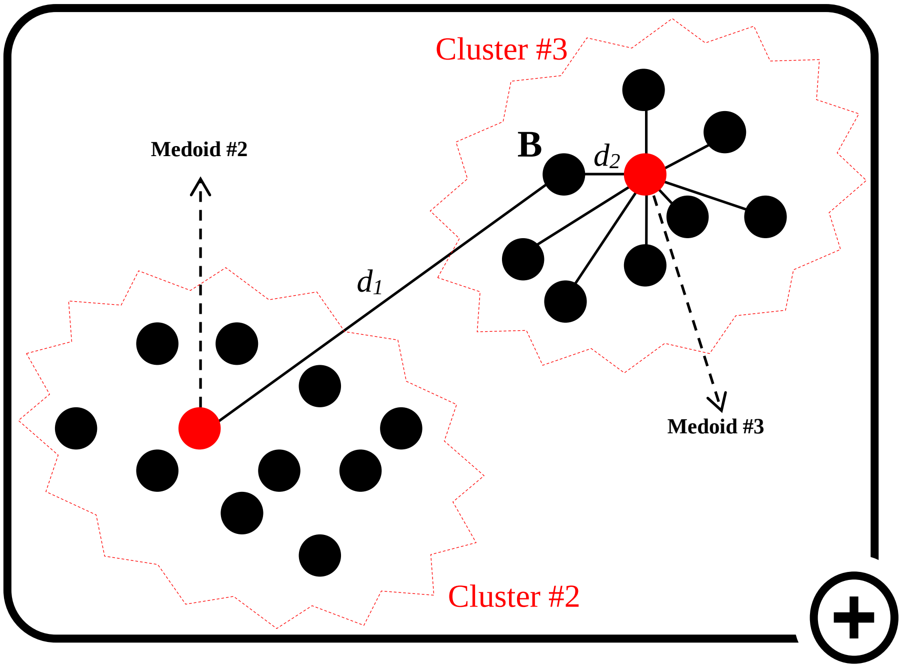


Figure 1.5: Assignment mechanism of objects into clusters

Problem 3. *How accurately can the cost of a product be predicted by using splines?*

We investigated whether implementation of a spline approach provides accurate estimates of manufacturing costs. At the same time, the performance of the underlying splines approach is compared with the clustering approach to discern a possible superiority relationship between them. Predictably, the benchmark cost estimation performance assessment is made with a single regression model built with the entire data that does not consider clustering or piecewise functions.

Splines constitute a reasonable approach for the nonparametric estimation of manufacturing cost functions. Unfortunately, the commonly known splines are restricted to continuous predictors (numeric attributes). This is a disadvantage when it comes using splines on manufacturing cost estimation problems since it is not unexpected to encounter categorical predictors (categorical attributes) in real life cases. However, with some modifications on the traditional continuous-only predictor splines approach, it is possible to accommodate the presence of categorical design attributes without artificial dataset manipulations such as splitting the dataset into continuous and categorical attributes subsets [8]. Tensor product B-splines for numeric design attributes along with kernel-weighting for categorical design attributes are used to handle mixed data to overcome this limitation of standard splines. Further details about establishing non-parametric splines for continuous and categorical predictors can be found in Chapter 3.

1.2 Complexity of Variables

There are two issues rendering this cost estimation problem more complicated: (1) incorporating qualitative and quantitative variables in a dataset simultaneously, (2) the number of variables in a dataset being less than the number of products but still large relative to the number of products. We address the first issue by using applicable clustering and spline techniques and the second issue by removing irrelevant variables. Please refer to Chapters 3 and 4 to read further details about these issues and our approach.

We can classify the variable types into two main categories: continuous (quantitative) variables and discrete (qualitative) variables. Quantitative variables provide information of numerical magnitude. They mainly consist of interval-scaled and ratio-scaled variables. Interval scaled variables are either on positive or negative axes and measured over linear equal intervals. On the other hand, ratio-scaled variables take positive values on a non-linear scale. This might be a transformation of the original continuous variable by an exponential or a logarithmic function. Unlike quantitative variables, qualitative variables provide information of categories. They may take finite discrete numerical values or categorical labels including names. We can categorize the qualitative variables as nominal, ordinal and binary variables. Nominal variables are categorical classifications where ranking among categories does not exist or is not important. However, for ordinal variables, ranking among categories is important and has a meaningful sequence. The transition between categories for ordinal variables is not necessarily distinct or equal but must be logical. Binary variables are a special case of qualitative variables where there are only two possible outcomes (or states). They can either take a value of 0 or 1 where 0 usually represents the absence of a property and at the same time, 1 represents the presence of a property. If the states of a binary variable have equal importance, it is called symmetric binary; otherwise asymmetric binary [9]. Ranking issues among categories, non-distinct borders between transition of categories, and asymmetrical properties of binary variables are challenging aspects of handling categorical variables when mathematical operations are required for further analysis.

In this study, the datasets of the application problems contain at least one kind from each variable type (categorical or numeric). To the best of our knowledge, none of the existing similarity measures (metrics) can handle mixed type of variables in their original form. Using Gower's index [9] is a good alternative for the clustering analysis because it enables us to transform outcomes of different types of variables into a single mathematical value. With the help of Gower's index, an objects-to-objects (products-to-products) distance matrix can be derived. The upper and lower parts of this matrix are symmetrical to each other around the diagonal and the diagonal elements have values of zero. The logical interpretation of a zero-valued diagonal element is the distance from a product to itself is zero. Further discussion about Gower's index is available in the literature review section.

In this study, we have collected four datasets from three manufacturing industries. The representative features have been selected according to the cost drivers for these specific manufacturing processes. The diversity of the manufacturer datasets shows that this study can be extended over different industries by including industry specific design variables.

1.3 Limitations of the Research

We assume that new products are based on some modifications or variations to existing or historical products. That is, introduction of a new product beyond the universe of past products is not advisable because these statistical approaches are strictly interpolation processes. If an estimation model is built based on a categorical universe of small, medium and large, then it is not possible to predict the manufacturing cost of a new design with a categorical design variable value of extra-large. However, for numerical attributes, it is possible to predict costs out of the past range of values because based on an interval scale or ratio scale, there is a logical increment between two different values to determine the relationship between them. However, this is not advisable as these models are designed for interpolation not extrapolation.

The second limitation of this research is that the clustering content is not necessarily optimized. That is, it is not necessary to find the best clustering algorithm to group objects but to find well-constructed clusters. There is a large literature for evaluating the performance of clustering algorithms. Thus, we are not evaluating the performance of multiple clustering algorithms but chose a clustering technique that can handle mixed categorical and numeric design attributes, is not affected by outliers and noisy data, and is already proven to be a robust and powerful method.

This research is strictly limited to non-parametrical approaches to avoid making assumptions concerning statistical distributions. Continuous predictors are not normalized and we assume that all variables come from empirical distributions.

Note that this work assumes commodity production where the size of a batch is not important. All datasets come from mass production facilities. A future extension might consider economies of scale and batch sizing when predicting cost.

1.4 Organization of the Dissertation

We summarize the previous work in the literature in Chapter 2. This chapter consists of four main aspects of the literature, namely the survey of manufacturing cost estimation efforts, clustering methods and similarity measures, splines, and the most specific related work to our research. We propose the framework of our methodologies in Chapter 3. This chapter provides the information about our suggested approaches for clustering, splines, estimation, and result validation efforts. In Chapter 4, we demonstrate a variety of applications of our suggested cost estimation methodologies for four real life datasets, namely socks, electrical grounding elements, lightening protection parts and plastic household products. The last chapter highlights the conclusions of this research and the direction of future research.

Chapter 2

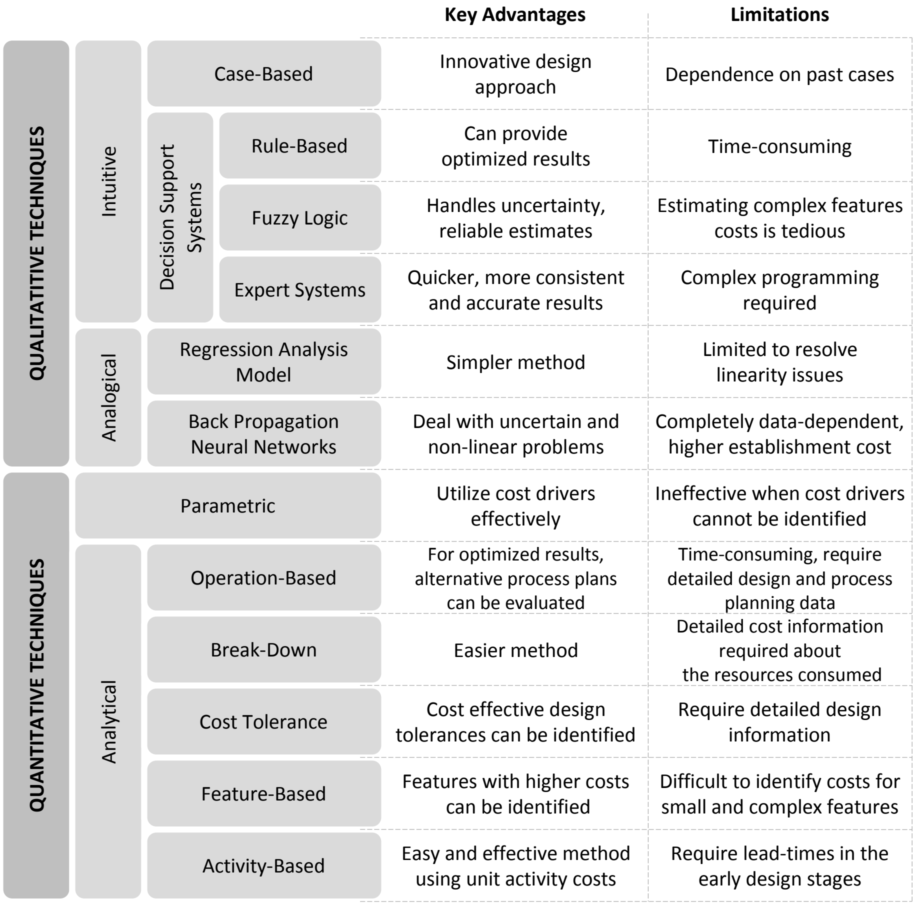
Literature Review

This literature review chapter explores three main topics and the most related aspects of previous work: (1) Manufacturing cost estimation, (2) Clustering methods and similarity measures, (3) Splines and (4) Relevant work. In the first part, manufacturing cost estimation efforts and different classifications of manufacturing cost estimation techniques are discussed. The second part of the literature review gives an extensive review of clustering techniques and their possible advantages. In addition, to complement clustering techniques, the most prevalent similarity measures are summarized. Potential uses of these measures with their benefits and drawbacks are presented under the second part. The third part shows the literature review for splines and the last part consists of the most related publications to our proposed cost estimation methodologies.

2.1 Manufacturing Cost Estimation

Estimating product cost is an inseparable part of manufacturing processes. Even though a manufacturing process is a physical operation, cost estimation has a principal role in all parts of it. Layer et al. [10] point out that manufacturing cost calculations are classified based on the timing of calculations: (1) Pre-calculation, (2) Intermediate calculation and (3) Post-calculation. Pre-calculation estimates the potential costs before actually manufacturing the item. The price of a product is usually declared based on the pre-calculation values when a new unique design has been requested by a customer for a future manufacturing agreement. As a result, higher accuracy in the pre-calculation step is crucial to generate designs where low-cost and high-quality are maintained. On the other hand, actual cost is the interest of the post-calculation phase. Instead of estimated cost drivers, incurred costs are included in the post evaluation step. In the first chapter, we mentioned some of these cost drivers, specifically in Section 1.1. Our research interest is the pre-calculation phase where we seek establishing the cost of a product accurately before actual production takes place. However, we still need historical data of product costs previously recorded based upon the post-calculation. The capability of estimating the cost of a product accurately increases the confidence of a business. Any hesitation during the price establishment phase may result in loss of customers or profit. These are the reasons why cost estimation efforts have been important for all kinds of manufacturers.

There are many publications in the cost estimation area employing empirical and analytical techniques. These techniques provide solutions for chemistry, manufacturing, construction and computer programming applications. Manufacturing cost estimation techniques are classified under two main categories consistently by authorities. Dai et al. [11] and Layer et al. [10] termed these two main categories qualitative and quantitative techniques. However, second-level classifications vary according to subjective opinions. Figure 2.1 has been regenerated from a literature survey of product cost estimation [11] and gives an overview of the key advantages and limitations of the underlying product cost estimation techniques. A total of twelve different product cost estimation techniques were identified by Dai et al. [11]. In their survey, case-based systems, rule-based systems, fuzzy logic systems, expert systems, regression analysis models and backpropagation neural network models are grouped under qualitative techniques, while parametric techniques, operation-based models, break-down models, cost-tolerance models, feature-based models and activity-based models are classified under quantitative techniques. Although, Layer et al. [10] constructs the first level classification of cost estimation the same as Dai et al., they did not include second-level qualitative techniques classification in their work. Therefore, they provided the classification of quantitative cost estimation techniques given in Figure 2.2. Their definition of quantitative techniques includes only three subdivisions: (1) Statistical models, (2) Analogous models and (3) Generative-analytical models.



				Key Advantages	Limitations
QUALITATIVE TECHNIQUES	Intuitive		Case-Based	Innovative design approach	Dependence on past cases
		Decision Support Systems	Rule-Based	Can provide optimized results	Time-consuming
			Fuzzy Logic	Handles uncertainty, reliable estimates	Estimating complex features costs is tedious
			Expert Systems	Quicker, more consistent and accurate results	Complex programming required
	Analogical		Regression Analysis Model	Simpler method	Limited to resolve linearity issues
			Back Propagation Neural Networks	Deal with uncertain and non-linear problems	Completely data-dependent, higher establishment cost
QUANTITATIVE TECHNIQUES	Parametric			Utilize cost drivers effectively	Ineffective when cost drivers cannot be identified
	Analytical		Operation-Based	For optimized results, alternative process plans can be evaluated	Time-consuming, require detailed design and process planning data
			Break-Down	Easier method	Detailed cost information required about the resources consumed
			Cost Tolerance	Cost effective design tolerances can be identified	Require detailed design information
			Feature-Based	Features with higher costs can be identified	Difficult to identify costs for small and complex features
			Activity-Based	Easy and effective method using unit activity costs	Require lead-times in the early design stages

Figure 2.1: Overview of product cost estimation techniques with advantages and limitations [11]

According to Dai et al. [11], case-based systems utilize a database of previously manufactured parts by detecting the most relevant items to the new design with respect to the similarities of their design features. If there are some differences between old designs and the new design, necessary modifications in both design and operational levels should be done. Finally, cost of the new design can be estimated by revising the cost of the benchmark products (old designs) according to the magnitude of change in these modifications. Linear regression analysis models also require some historical data to construct cost trends. This is basically taking the linear correlations into account and building the model with respect to the relationships of independent variables with the dependent variable (product cost). The regression analysis can be extended by identifying more cost drivers and their linear and non-linear relationships. A parametrical cost function can be derived by using these cost drivers with their algebraic relations to mimic the actual cost behavior. This parametric cost estimation technique is claimed to be superior to a simple regression analysis [12]. On the other hand, operation-based cost estimation techniques devise an approach which is the summation of the manufacturing time of individual operations as well as non-value-added activity times. Feature-based cost estimation approaches use a similar idea with operation-based approaches but there is a small difference. That is, instead of identifying solely the value-added and non-value added operations, this method deals with all cost related features including product design geometry, machining requirements and other cost drivers which might have an influence on the total cost. However, identification of these features is subject to the expertise of the person in charge and is limited to his/her information extraction capabilities.

As we discussed above, unlike Dai et al. [11], Layer et al. [10] divided quantitative cost estimation techniques into three categories: statistical, analogous and generative-analytical models. According to their classification, statistical models include regression analysis, neural networks and other optimization techniques. These techniques also require historical data and empirical examination skills. The required historical data usually covers some shape-describing variables as well as semantic product characteristics. Analogous models consider geometrical similarities but also take functional similarities into account. Analogous approaches are similar to statistical models and do not need to be classified differently because both approaches process cost related features in identical ways.

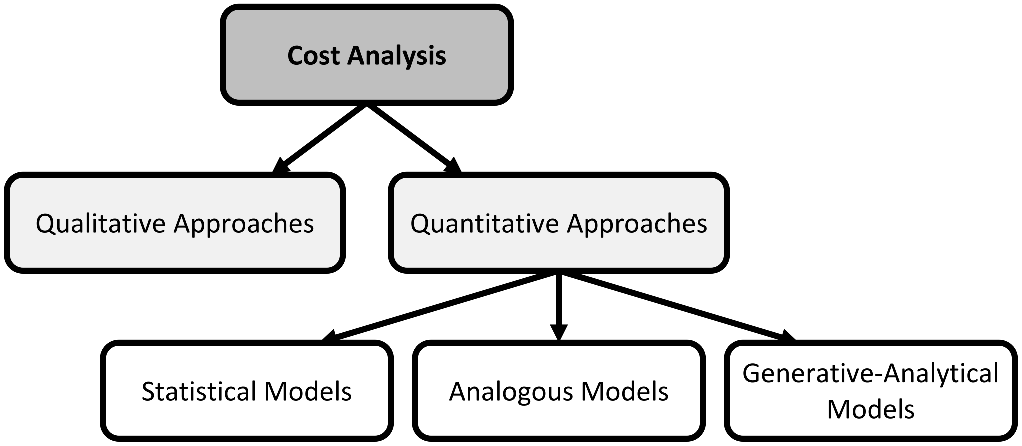


Figure 2.2: Classification of product cost estimation approaches from Layer et al. [10]

Our clustering based cost estimation approach fits none of these classifications strictly but can be considered as a combination of several approaches, namely case-based systems, analogical parametric cost estimation techniques, operation and feature-based models. From the perspective of Layer et al. [10], our approach is also considered an analogous model. In our study, manufacturing cost estimation uses historical data of similarities among previously manufactured products. Therefore, we have identified all cost related features associated with these products and recorded categorical/mathematical values of these features as variables in our data. Lack of such a hybrid comprehensive statistical method in the literature which uses clustering techniques to establish cost models for similar product streams is one of our research motivations.

On the other side, spline functions has never been used as a manufacturing cost estimation tool in the literature. Our curiosity in using such a model motived us developing spline cost estimation models that can accommodate mixed categorical and numeric design attributes. Our spline based cost estimation approach can also be considered a combination of several approaches, namely analogical non-parametric regression analysis along with operation and feature-based models.

2.2 Clustering Methods

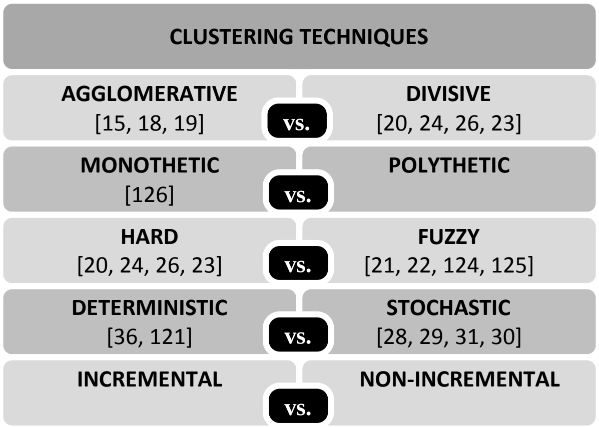
There are many applications of clustering in very different science branches. Jain et al. published a review of clustering methods which explicitly covers the most common techniques available at that time. Even though they grouped clustering methods into two main categories with six sub topics, with currently available information, it is possible to extend this classification with a broader aspect. Please refer to the Figure 2.4 for a comprehensive illustration of clustering classification. The main categories of clustering have been preserved but new subcategories are added to the original representation of Jain et al. [13]. Under this larger clustering umbrella, there are still two main categories: (1) Hierarchical clustering and (2) Non-hierarchical clustering.

Along with the clustering methods mentioned above, Jain et al. [13] discussed alternative techniques for computational clustering implementations. Figure 2.3 gives an overview of these implementations. These approaches are agglomerative vs. divisive, monothetic vs. polythetic, hard vs. fuzzy, deterministic vs. stochastic and incremental vs. non-incremental. In agglomerative approaches, observations of a dataset are merged into groups until all observations form a single cluster. Divisive ones follow the same rule but go backwards by dividing an initial one big group into subgroups. Monothetic vs. polythetic approaches are closely related to serial or simultaneous processing of features, respectively. While serial implementations process features one at a time, simultaneous approaches handle all features at once. According to the hard vs. fuzzy aspect, each object is assigned to a cluster strictly, or with a degree of membership. Therefore, in fuzzy clustering, an object might be a member of several groups with some degrees but the highest value of membership may determine its ultimate assignment. Partitioning clustering techniques can be implemented in two ways: deterministic vs. stochastic, where either a deterministic method or a stochastic search algorithm is used. In Figure 2.3, meta-heuristics are considered as stochastic optimization algorithms. These meta-heuristics have become popular with the development of computational power to handle large datasets. Due to lack of this computational power at the time of early clustering applications, incremental vs. non-incremental approaches were devised. Constraints on processing capabilities and computer memory issues pushed practitioners reduce the dimension of data or handle patterns partially by updating the affected parts instead of re-computing the entire dataset.

2.2.1 Hierarchical Clustering

In terms of computational power, it might be very expensive even for today’s computers to examine all possible clustering combinations. However, many clustering algorithms aim to find satisfactory results without considering all combinations. From its name hierarchical clustering methods build a hierarchy of observations by either a series of successive mergers (agglomerative hierarchical clustering) or a series of successive divisions (divisive hierarchical clustering) [6] in a greedy manner.

One of the biggest advantages of hierarchical methods is the dendrogram. That is, a graphical representation of similarity levels among observations that connects them as branches of a tree. Figure 2.5 is an illustration of a dendrogram for ten observations. In this figure, similarity level is described by distances among these observations. However, these similarities are not limited to distances but correlations, similarity coefficients* or even some other measures can be used to group observations. With the help of a hierarchical clustering method, an entire dataset can be summarized with a single visual output (dendrogram) where possible cluster formations can easily be detected. However, it might be very hard to track connections between observations on a dendrogram due to its chaotic appearance for large datasets. That is a downside of dendrograms. Similarity levels and connections among observations come from a distance matrix and these connection levels may be altered according to which hierarchical method is used. In this research, as can be seen from Figure 2.4, we have discussed five hierarchical methods; however, the most common ones are single linkage, complete linkage, average linkage and Ward's method [14]. Later in the chapter, distance matrices and corresponding similarity measures will be explained in detail.



CLUSTERING TECHNIQUES		
AGGLOMERATIVE [15, 18, 19]	vs.	DIVISIVE [20, 24, 26, 23]
MONOTHETIC [126]	vs.	POLYTHETIC
HARD [20, 24, 26, 23]	vs.	FUZZY [21, 22, 124, 125]
DETERMINISTIC [36, 121]	vs.	STOCHASTIC [28, 29, 31, 30]
INCREMENTAL	vs.	NON-INCREMENTAL

Figure 2.3: Classification of clustering implementations independent of the clustering method

* Please refer to the similarity coefficients on page 42.

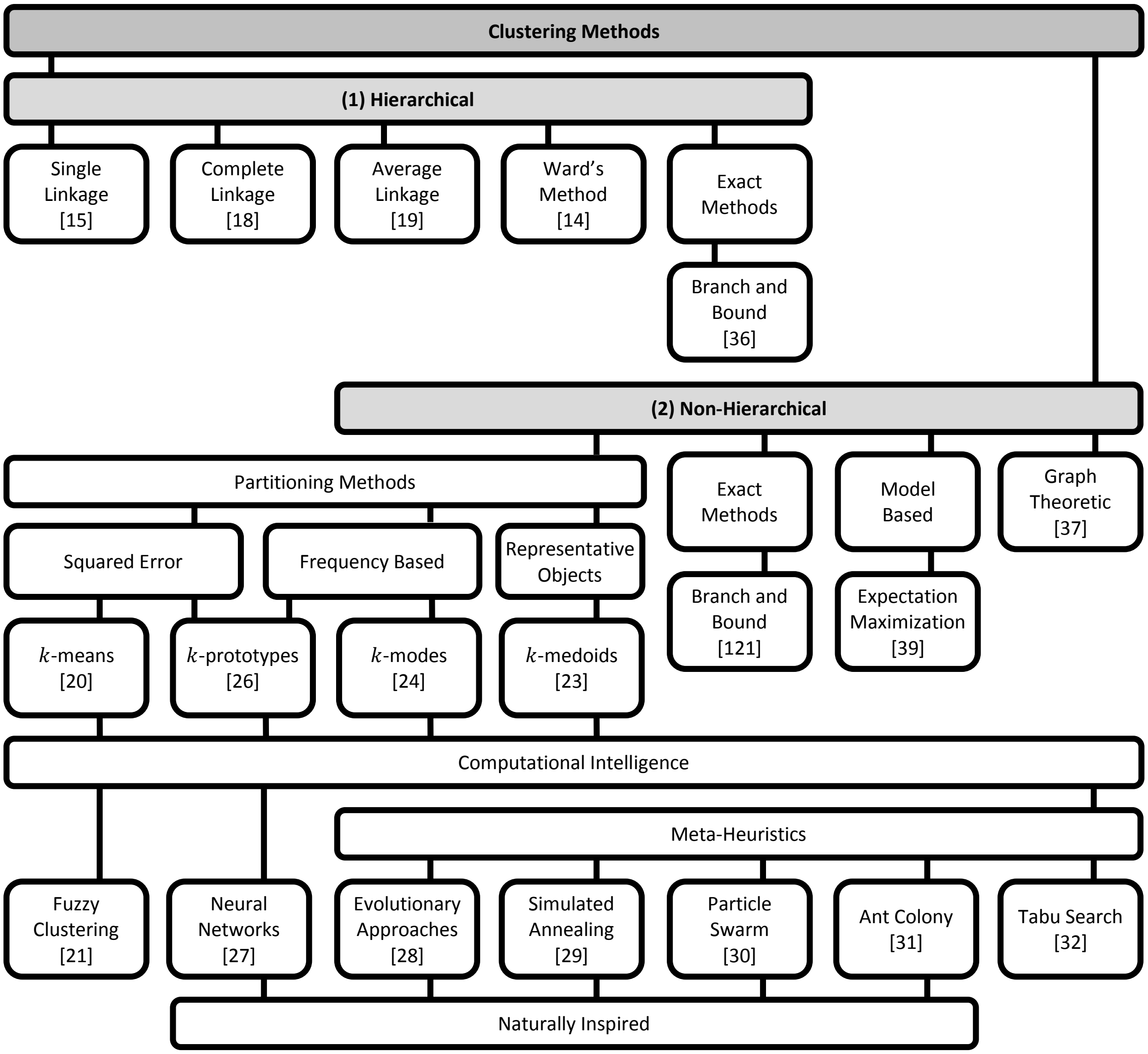


Figure 2.4: Extended classification of clustering methods

The single linkage method was first introduced by Sneath [15] as a part of medical research. In his study, variables are non-numerical values and represented by existence (+) or absence (-) of a particular strain in bacteria data. The aim of the work was to convert non-numerical data into meaningful numbers and suggest taxonomic groups for strains. Basically, the algorithm seeks the nearest entity to merge with an existing cluster or with an individual point. This can be considered as the fusion of entities with the greatest similarities. Unfortunately, since the algorithm seeks minimum distance between entities in a greedy way, it may not be able to detect true connection links among poorly separated clusters [6].

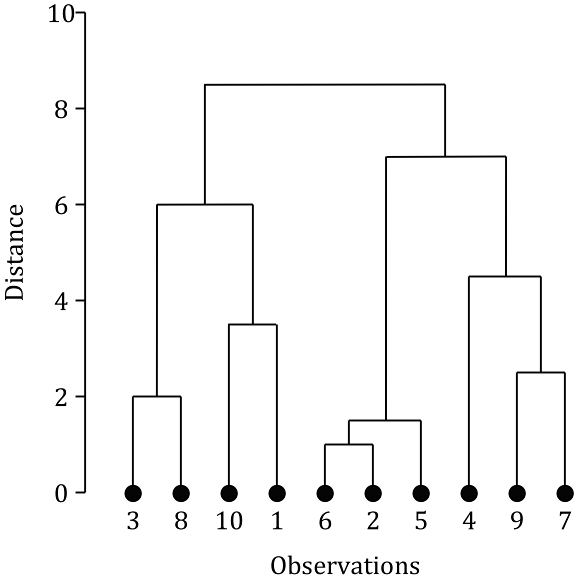


Figure 2.5: A dendrogram for distances among ten objects (observations)

Even though complete linkage method was studied by McQuitty [16] and also by Sokal and Sneath [17] approximately in the same decade, the method was first introduced by Sørenson [18]. In Sørenson's work, the complete linkage method is demonstrated with a non-numerical grouping example. The algorithm works in a similar manner to the single linkage method but with a small nuance. Rather than considering all possible nearest pairs of subjects as in the single linkage criterion, the method considers the furthest neighbors approach. The algorithm guaranties that the most dissimilar items are separated in different clusters. There is a problem of using the complete linkage method when there are more than one equally distant entity to merge with. It is recommended to choose the entity with the highest average similarity coefficient in such a case [17].

Not long after Sneath introduced the single linkage method, Sokal and Michener [19] presented the average linkage method in a university science bulletin. They have categorized bees according to correlation among species. A detailed correlation coefficient selection procedure can be found in their original study. The same logic as in the single and complete linkage methods is applied here with a slight difference. In the average linkage method, the distance between two clusters can be obtained by comparing the average distance between all pairs of items within these two clusters rather than seeking the furthest or nearest neighbors.

Ward [14] has approached the hierarchical clustering problem as a variance minimization problem. Instead of using similarity or correlation coefficient matrices, the method tries to minimize the error sum of squares from cluster means. Unfortunately, Ward’s method can only take quantitative variables into account. The method fails when the dataset includes binary or categorical variables. It also has a binding assumption of the multivariate distribution. Existence of non-normally scattered observations may cause the algorithm to provide poor results. This is an important handicap that we cannot neglect when a dataset contains non-numerical values or numerical but non-normal values.

2.2.2 Non-Hierarchical Clustering

Non-hierarchical clustering methods would be preferable and more efficient to use than hierarchical ones when a large dataset is analyzed. The reason behind that preference is because of the extensive data storage requirement of hierarchical methods. A large amount of data is stored and used for consecutive iterations in hierarchical clustering algorithms. This data may include similarity matrices obtained in every iteration, relationship connection levels and interim dendrograms. Also, as mentioned in the hierarchical methods, dendrogram connections are very hard to follow visually when too many nested sequence of groups exist. While non-hierarchical methods are being used in engineering applications, hierarchical methods are frequently found in biological, social and behavioral science applications [5].This is basically because the ultimate interest of engineering applications is the final content of clusters. On the other hand, the interest for biological, social and behavioral sciences is building taxonomical ranks. We can categorize the non-hierarchical methods into four main topics: (1) Partitioning, (2) Model-based, (3) Graph theoretic, (4) Exact methods, from most to least frequently used.

Without a doubt, the leading algorithm is the k-means (or c-means) clustering method. It was first introduced by MacQueen [20] to allocate observations in a dataset into a pre-determined number of clusters – k. The logic behind the k-means algorithm is to find the content of k partitions by minimizing within cluster variances. That is a reason why the k-means algorithm falls within the squared error category. Recall the name of the algorithm, k-means, notice that k stands for the number of partitions. Also, mean stands for the center point of a cluster where an average is taken for all observation points within a cluster. Overall, k-means refers to k number of center points – centroids. Many modifications to the original work exist in the clustering literature. Even though in the original study, MacQueen suggested to start with k single random points as initial clusters, it is practical to start with an initial partition of the items into k groups. Unfortunately, the final cluster contents obtained by running the k-means algorithm can depend on the initial starting point. Every new k-means run, the algorithm has a possibility to give different results. This issue may significantly impact clustering analysis. To overcome that possible problem, the k-means algorithm should be rerun several times or even compared with results obtained by a different algorithm.

Besides MacQueen’s hard k-means clustering algorithm, a fuzzy version was developed by Dunn [21]. However, Bezdek et al. [22] improved Dunn’s fuzzy clustering approach to use in pattern recognition clustering applications. Bezdek et al.’s fuzzy k-means algorithm is the most commonly used one in fuzzy clustering applications. Unlike the hard k-means, fuzzy k-means clustering (also known as fuzzy c-means or FCM) does not assign observations to clusters during interim steps. Through iterations of fuzzy k-means, the algorithm updates the membership values for each observation that identify the degree of belonging to each cluster. The stopping criterion is same as for the k-means, which is when the percentage of improvement is less than a threshold value. Fuzzy k-means is a computationally intensive algorithm because along with cluster centroids and objects’ distances to each centroid, fuzzy membership values are computed. Fuzzy k-means may overcome the clustering issue when the borders of clusters are not precisely separated. That is, observations may belong to more than one cluster with degrees of membership. Therefore, in the final iteration, observations can be assigned to clusters according to their highest value of membership.

Two decades after the introduction of the k-means algorithm, the partitioning around medoids (PAM) paradigm was developed by Kaufman and Rousseeuw [23]. They called this method, the k-medoids algorithm. The objective of the method is not about minimizing within cluster variability as in k-means. Unlike k-means approach, the method uses real observations as cluster centers and partitions the whole data around these cluster medoids. In other words, instead of devising the error sum of squares approach, the algorithm seeks cluster contents around representative objects based upon average dissimilarity. Allocating the observation points to the nearest medoid is advantageous in many aspects. Since the cluster centers are picked from appropriate elements in the actual dataset, the variables in that dataset do not solely need to be on an interval scale. Kaufman and Rousseeuw also proved that the k-medoids approach gives more robust results than methods based on variance minimization, as with k-means. Additionally, the existence of outliers does not perturb the k-medoids clustering progress.

The k-means algorithm became very handy for clustering large datasets because of its computational efficiency, but unfortunately it is limited to numerical data only. Huang [24] proposed a new clustering method called k-modes to shift the use of k-means method to categorical data. It is a frequency-based algorithm and uses a simple matching similarity coefficient (see similarity coefficients in Section 2.2.3) to deal with categorical variables. The k-modes algorithm replaces the cluster centroids with modes. The working logic of the algorithm is very similar to k-means and stops iterating when the same convergence criterion is met. Similar to k-means, the k-modes algorithm generates locally optimum cluster contents and does not guarantee global optimum solutions. Additionally, k-modes may have some misclassification issues when within-cluster similarities are weak. To address this issue, He et al. [25] suggested adding weights to attribute value matches in the simple matching similarity computations to avoid undesired classifications of objects.

Huang [26] also proposed another algorithm called k-prototypes by integrating the principles of k-means and k-modes to expand the applicability of partitioning algorithms to observations which have mixed numerical and categorical attributes. It specifically devises an objective function which is a combination of k-means and k-modes. Remember that, while the k-means algorithm is based on the Euclidean distances for continuous variables, the latter uses simple matching coefficients for categorical variables. A weighted summation of these two expressions, Euclidean distance and the simple matching coefficient, constitutes the framework of the k-

prototypes algorithm. The k-prototypes algorithm works slightly slower than k-means but it is still an efficient alternative to cluster mixed type of data.

Recall that the information about the impractically of enumerating all possible cluster combinations to find the best clusters. Many adaptive algorithms have been deployed to solve this optimization problem for that specific reason. Naturally inspired clustering applications can be found in the literature such as artificial neural networks [27], evolutionary approaches [28], simulated annealing [29], particle swarm [30] and ant colony optimization [31]. Also, a tabu search algorithm was devised to solve the clustering problem as a discrete optimization problem [32]. Since all of these methods are heuristics, they do not guarantee the optimal allocation of objects into clusters. However, some of these meta-heuristics are robust to the specific initial starting clusters, unlike the k-means algorithm.

Most clustering problems can be solved by these adaptive techniques which can specifically handle solely continuous, solely combinatorial problems, or either. Table 2.1 provides an overview of these adaptive techniques and their applicability among types of clustering problems. In the table, a plus sign (+) represents the possibility of using the underlying adaptive tool for the specific clustering problem. For example, while the simulated annealing algorithm can solve clustering problems in both a continuous and combinatorial sense, the ant colony optimization heuristic is applicable only for combinatorial clustering problems. Because the partitioning around medoids paradigm is a combinatorial case, its applications can be solved by the following adaptive optimizers: simulated annealing, genetic algorithm, tabu search and ant colony. There are several publications which devise these heuristics for PAM clustering problems except for simulated annealing. There is an opportunity to fill the gap with a PAM simulated annealing approach and compare the performance of it with other approaches.

Table 2.1: Overview of meta-heuristics and their applicability for clustering problems

	Compatibility with Problems		
	Continuous	Combinatorial	PAM
Simulated Annealing	+	+	
Genetic Algorithm	+	+	Lucasius et al. [33]
Evolutionary Algorithm	+	N/A	N/A
Tabu Search	N/A	+	Ng and Wong [34]
Ant Colony	N/A	+	Boryczka [35]
Particle Swarm	+	N/A	N/A

Koontz et al. [36] introduced a branch and bound approach for combinatorial hierarchical clustering problems where N objects are grouped into k classes. The other clustering efforts mentioned before are techniques to find sufficient results, usually local optimal clusters. However, the branch and bound method was developed to provide an exact solution to such combinatorial problems with globally optimal clusters. Unfortunately, the branch and bound method is not practical in terms of the computational effort compared with other partitioning methods. However, the time efficiency of the modified branch and bound method [36] for non-hierarchical clustering problems is in comparable time units with other previously mentioned clustering techniques. As a drawback to the branch and bound method, observations in a given dataset are required to be real numbers.

As discussed before, objects in a dataset can be represented as points in multidimensional space. According to the nearest neighborhood approach, these points can be linked to each other if some assumptions hold. This linkage procedure is called the minimal spanning tree (MST) approach in graph theory. Zahn [37] introduced the MST method to detect cluster structures. Even though the MST approach is classified here under non-hierarchical methods, it can be counted as a hierarchical clustering approach since it uses the nearest neighborhood as in the single linkage method. Overall, the MST approach is based on building a tree for N objects in a dataset with $(N-1)$ connections [38]. Even though the algorithm minimizes the sum of these $(N-1)$

connections (similarities), it still requires the same computational time as the other hierarchical linkage methods.

As a preliminary step to the model based clustering, Wolfe [39] introduced multivariate mixture analysis based on two density mixtures: (1) Mixtures of multivariate normal distributions, and (2) Mixtures of multivariate Bernoulli distributions. Regardless of the clustering method, similarity measures used in clustering problems are often subjective or arbitrary. Wolfe devised a maximum likelihood estimation method to avoid arbitrary similarity measure assignment. Since elements in a cluster are different from elements in another one, it would be appropriate to assume that every cluster comes from a probability distribution. In order to represent the whole population, these probability distributions can be combined into one mixture density with underlying mixture weights. There are many parameters required to be estimated in model based clustering algorithms: $k-1$ mixing probabilities (weights, where k is the number of clusters), kd means (where d is the number of variables), and $kd(d+1)/2$ variances and covariances [6]. With accessibility to computational power, the method has gained more use in clustering applications, even though the estimation of a model is very complicated. Everitt [40] extended Wolfe's study by combining categorical and continuous variables into a single joint density function. There are even more extensions to Everitt's work in the model based clustering field. Within the last decade, Moustaki and Papageorglou [41] devised a class mixture model to handle binary, nominal, ordinal and continuous variables in the same dataset by using their appropriate distribution representations: Bernoulli, multinomial, cumulative multinomial and normal, respectively. Our main research concern it to predict the manufacturing cost of a product without assuming statistical distributions or making assumptions about underlying distributions. Therefore, using a model based clustering approach is not appropriate.

2.2.3 Similarity Measures

Most clustering techniques require an assignment of a similarity (or dissimilarity) measure in the very initial step. Selection of a similarity measure should be based on application appropriate logic and in consideration of functional requirements. Poorly chosen measures may lead clustering algorithms to undesired directions. This is also an arbitrary assignment process where many distance measures might be considered; however, only one or a few of them fits the data available and results in the best discrimination between clusters. Higher discrimination power is also what practitioners seek from a similarity measure. A similarity measure should increase as the dissimilarity between two objects increases.

Besides the discrimination power of a similarity measure, there is another important point we should discuss. In practical applications, categorical and numerical variables may exist in the same dataset. If there are more than one type of variable associated with observation points, it would be impractical to separate variables and assess them with individual cluster analyses. The hard part is to reconcile cluster content obtained by these separate analyses if they are different from each other [9]. However, Strehl and Ghosh [42] introduced the cluster ensemble approach, which utilizes this idea. That is, synthesizing the results of several clustering algorithms to achieve final partitions. Within a short time frame after Strehl and Ghosh's study, He et al. [43] adopted the cluster ensemble idea to treat heterogeneous data. Their research consists of two consecutive phases: (1) In the first phase, they divide the mixed data into two classes of solely categorical variable and solely numerical variable subsets. For each class, they run a cluster analysis which is exclusively specified to treat that kind of underlying variable. At the end of the first phase, they label objects with a new attribute that represents the clusters in which they fall. (2) In the second stage, the algorithm uses a categorical clustering approach based on only the new attributes to find the ultimate cluster contents. Every variable may not contribute equally to form the final partitions. Therefore, the cluster ensemble approach may need modification if the individual variable contributions are not equally weighted. Figure 2.6 is regenerated from He et al. [43] and summarizes the framework of the cluster ensemble approach for mixed numeric and categorical data.

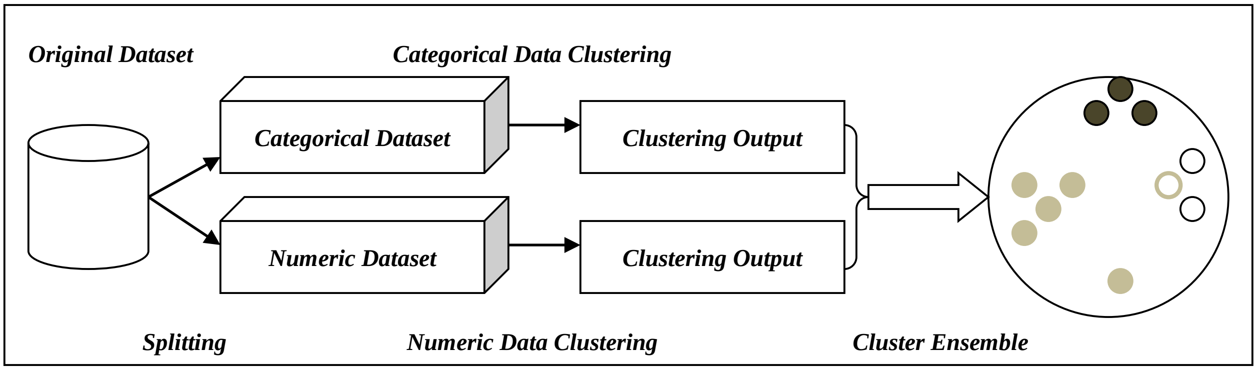


Figure 2.6: Overview of the cluster ensemble approach for mixed variables [43]

Rather than using a cluster ensemble approach, it would be more straight-forward to cluster once with the full dataset. That is, instead of synthesizing the results of separate cluster analysis for different variable types, performing a single cluster analysis [9]. This is possible in two ways: (1) Reduce all variables into binary variables, (2) Reduce all variables into interval-scaled values. The first one is possible if the underlying numerical variable planes are cut into two parts with some specific threshold values. Unfortunately, this sacrifices information [9]. The second approach is practical if there are at most two states for nominal variables. When more than two states exist for a nominal variable, the second approach would fail due to improper discrimination among states of it. That is, the numerical coding for each nominal state may not represent the actual distinction among the states. Additionally, asymmetrical binary variables are treated symmetrically according to this approach which would cause a scaling issue [9]. However, combining different type of variables into a single proximity matrix is found to be more convenient than these two approaches [9, 44, 45, 46, 47].

Let us return to our original subject, similarity measures. We provide the ten most commonly used similarity measures in clustering applications below. We summarize examples and applicable fields of these similarity measures in Table 2.2.

Here are these similarity measures:

i. Euclidean Distance

ii. Scaled Euclidean Distance

iii. Mahalanobis Distance

iv. Minkowski Metric

v. Canberra Metric

vi. Czekanowski Coefficient

vii. Chebyshev Distance

viii. Pearson Correlation

ix. Cosine Similarity

x. Similarity Coefficients for binary variables

Table 2.2: Some application areas of the similarity measures

Similarity Measure	**Area of Application**
Euclidean Distance	mostly with *k-means* clustering method [20]
Scaled Euclidean Distance	mostly when the variation of variables are very different
Minkowski Metric	mostly with fuzzy *k-means* clustering with different weights [22]
Mahalanobis Distance	to handle correlated data (in elliptical shape)
Canberra Metric	to detect computer intrusions [48]
Czekanowski Coefficient	mostly for biological taxonomy [49]
Chebychev Distance	mostly with fuzzy *k-means* clustering with sup norm [50]
Pearson Correlation	widely used for analyzing gene expression data [51]
Cosine Similarity	to compare documents in text mining / document clustering [52]
Similarity Coefficients	to handle binary type of variables / scientific taxonomy [6]

Euclidean Distance

This is a straight line distance between two points. Even though it is the most common distance measure and very easy to apply, when data points in multidimensional space are scattered like an elliptical cloud (that is, the variables have high correlations with each other), then usage of that measure is not appropriate. Euclidean distance becomes meaningful when the data cloud has a circular shape without any significant correlations among variables (See Figure 2.7 where v_1 and v_2 represent variables one and two, respectively). Additionally, the Euclidean distance works well for two or three dimensional data when clusters are isolated [13]. The formulation of the Euclidean distance is given in Equation 2.1. The lower case letter d is the distance between objects x and y. The superscript letter T is the transpose operation for the underlying vector. Throughout the following distance measure descriptions, the notation d and $d(x, y)$ serve the same purpose.

$$d(x, y) = \sqrt{(x - y)^T (x - y)} \tag{2.1}$$

Scaled Euclidean Distance

Individual variables in a dataset might have very different variances and ranges with incomparable units of measures. If so, all variables need to be scaled by their individual standard deviations because Euclidean distance gives equal emphasis to each variable. This normalization amplifies the contribution of relatively small averaged variables to the main response variable. This overall adjusted metric is called scaled Euclidean distance. The formulation of scaled Euclidean distance is given in Equation 2.2. S is a diagonal matrix where its elements consist of corresponding variable variances. S^{-1} is the algebraic notation of the inverse of matrix S and the superscript letter T is the transpose operation for the underlying vector.

$$d(x, y) = \sqrt{(x - y)^T S^{-1} (x - y)} \tag{2.2}$$

Mahalanobis Distance

The Euclidean distance and scaled versions of it neglect covariance terms between variables. When variation in each axis is vastly different and linear relationships exist among these variables, using an elliptical distance would be more appropriate than using other distance measures. The most prevalent elliptical distance is the Mahalanobis distance which takes covariance terms (correlations) into account. The formulation of Mahalanobis distance is given in Equation 2.3 below. $\mathit{\Sigma}$ represents a covariance matrix and $\mathit{\Sigma}^{-1}$ is the inverse of it. The superscript letter T is the transpose operation for the underlying vector. When $\mathit{\Sigma}$ is an identity matrix, Mahalanobis distance is equivalent to Euclidean distance.

$$d(x, y) = \sqrt{(x - y)^T \, \Sigma^{-1} \, (x - y)} \tag{2.3}$$

Minkowski Metric

This is a generalized distance metric. When its weighting parameter m is 1, the Minkowski metric becomes a linear expression and it is called taxicab geometry (also known as city block distance, rectilinear distance, Manhattan distance, or L_1 norm). When m is *2*, the metric becomes the Euclidean distance. Changing the parameter m in the Minkowski metric changes the weight of differences between data points. The formulation of the generalized Minkowski Metric is given in Equation 2.4 where the lower case letter p is the total number of variables.

$$d(x, y) = \left[\sum_{i=1}^{p} |x_i - y_i|^m\right]^{1/m} \tag{2.4}$$

Canberra Metric

This can only handle non-negative variables. For multivariate data analysis applications, it does not suffer from normality assumptions [48]. The metric normalizes the rectilinear distance between two points with respect to the summation of corresponding measurements. The formulation of the Canberra Metric is given in Equation 2.5 where p is the total number of variables.

$$d(x, y) = \sum_{i=1}^{p} \frac{|x_i - y_i|}{(x_i + y_i)} \tag{2.5}$$

Czekanowski Coefficient

This also requires non-negativity assumption of data variables. The Czekanowski coefficient also normalizes the distance between two objects with respect to the summation of corresponding measurements. When individual variables in a dataset have very different variances or are in incomparable units of measure, using the Czekanowski Coefficient may not be appropriate. The formulation of the Czekanowski coefficient is given in Equation 2.6 below where the lower case letter p is the total number of variables.

$$d(x,y) = 1 - \frac{2\sum_{i=1}^{p} \min(x_i, y_i)}{\sum_{i=1}^{p}(x_i + y_i)} \tag{2.6}$$

Chebyshev (Tchebychev) Distance

This is not a unitless measure but a metric. Chebyshev distance is a special case of the Minkowski metric when the weighting exponent m goes to infinity. It is also known as the sup distance, the supremum (sup) norm, the uniform norm or L_∞ norm. It would not be appropriate to use for elliptical shaped data because it does not scale for variable variances or covariances. As a practical sense, when the largest variance is greater than four times the smallest variance, we may consider that the individual variances are divergent. In that case, Chebyshev distance becomes meaningless. Formulation of Chebyshev distance is given in Equation 2.7 where p is the total number of variables.

$$d(x,y) = \max_{1 \le i \le p} |x_i - y_i| \tag{2.7}$$

Pearson Correlation Distance

This is derived from the most common measure of correlation which is the degree of linear correlation among two variables. Its full name is “Pearson product moment correlation coefficient” [53]. Its range is within $[-1, 1]$ interval where -1 represents a perfect negative correlation and +1 represents a perfect positive correlation between two variables. The formulation of the Pearson correlation distance is given in Equation 2.8 where n is the total number of observations. In this case x and y are not observation pairs in a dataset but they are variable pairs.

$$d(x,y) = \frac{1 - r_{xy}}{2} \tag{2.8}$$

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Cosine Similarity

This is derived from the inner (dot) product of two vectors. Cosine similarity is closely related with the angle between two vectors. If the angle between two vectors is relatively small, they roughly point in the same direction. This can be considered as similarity of these vectors. Even though the original representation is not for distance of pairs – $d(x,y)$, for consistency, we used the same distance notation as in other measures. The formulation of cosine similarity is given below in Equation 2.9 where θ is the angle between vectors X and Y. When θ is 180°, $cos(\theta)$ equals to -1 and when θ is 0°, $cos(\theta)$ equals to 1. The resulting distance ranges from -1 to 1 which correspond perfect dissimilarity and perfect similarity, respectively.

$$d(x,y) = \cos(\theta_{xy}) = \frac{X \cdot Y}{L_X \, L_Y} \tag{2.9}$$

$$X \cdot Y = \sum_{i=1}^{p} x_i y_i \qquad and \qquad L_X = (X \cdot X)^{\frac{1}{2}}$$

Similarity Coefficients

When observations in a dataset cannot be represented by meaningful quantitative continuous variables, then pairs of points need to be compared by a different method. Similarity characteristics between pairs can be mathematically represented by binary values. The following table and coefficient expressions are taken from Johnson and Wichern [6]. The original tables have been slightly modified to maintain consistency of notation. Lower case letters a, b, c and d represent the frequency of the binary match, (1-1), (1-0), (0-1) and (0-0), respectively, in Table 2.3. The most commonly used eight similarity coefficients are given in Table 2.4. These quantify the resemblance between objects x and y considering the weighting emphasis on the match type. For the following two tables below, p is the total number of binary variables used to compare objects (items) in a dataset.

Table 2.3: Frequency of matches for item *x* and item *y*

		Item y		
		1	0	Total
Item x	1	a	b	a + b
	0	c	d	c + d
Totals		a + c	b + d	a + b + c + d = p

Combining one or more of these similarity measures with some of the similarity coefficients with weighting factors to treat mixed numeric and categorical data might be an effective approach. Thus, the objective function of the k-prototypes algorithm is a weighted summation of the Euclidean distance (for continuous variables) and the simple matching coefficient (for categorical variables). That is, the objective function of k-prototypes attempts to integrate a quadratic expression with a linear expression. Instead, consolidating two linear or two quadratic similarity expressions might be more mathematically appropriate.

Table 2.4: The most commonly used similarity coefficients

#	Similarity Coefficient	Explanation
1	$\frac{a+d}{p}$	Equal weights for 1-1 matches and 0-0 matches Simple matching coefficient [54]
2	$\frac{2(a+d)}{2(a+d)+b+c}$	Double weight for 1-1 matches and 0-0 matches [6]
3	$\frac{a+d}{a+d+2(b+c)}$	Double weight for unmatched pairs [55]
4	$\frac{a}{p}$	No 0-0 matches in numerator [6]
5	$\frac{a}{a+b+c}$	0-0 matches are treated as irrelevant The Jaccard coefficient [56]
6	$\frac{2a}{2a+b+c}$	Double weight for 1-1 matches and no weight for 0-0 matches [57]
7	$\frac{a}{a+2(b+c)}$	Double weight for unmatched pairs and no weight for 0-0 matches [17]
8	$\frac{a}{b+c}$	Ratio of matches to mismatches excluding 0-0 matches [6]

Table 2.5 gives a comprehensive summary of these ten similarity measures discussed above. The attributes included in the table are specifically chosen considering the scope of our application problems. These are the aspects of correlation consideration, handling only numeric data, handling only categorical data, handling mixed numeric and categorical data, non-negativity requirement, scaling for ranges of variable and elliptical shaped data, modifiable weights, sensitivity to outliers, unitless measure and metric properties, and, lastly, but most importantly, compatibility of these measures with our application problems. As you can see from this table, none of the existing similarity measures are completely compatible with our requirements in their

original forms. Notice that, a plus sign (+) points out the presence of the feature for a particular similarity measure.

Table 2.5: Summary of the most common similarity measures

	Consider Correlations	Handle Numeric Data	Handle Categorical Data	Handle Mixed Numeric and Categorical Data	Non-negativity Requirement	Scale for Elliptical Data	Scale for Range	Modifiable Weight for Differences	Sensitive to Outliers	Unitless Measure	Distance Metric	Compatibility to Our Work
Euclidean Distance		+							+		+	
Scaled Euclidean Distance		+				+			+	+	+	
Minkowski Metric		+						+	+		+	
Mahalanobis Distance	+	+				+				+	+	
Canberra Metric		+			+		+			+		
Czekanowski Coefficient		+			+		+			+		
Chebychev Distance		+									+	
Pearson Correlation	+	+				+				+		
Cosine Similarity		+				+	+					
Similarity Coefficients			+				+			+		

As we discussed earlier in this chapter, combining different types of variables into a single proximity matrix is found to be more convenient than using a cluster ensemble technique [9, 44, 45, 46, 47]. Unfortunately, the existing similarity measures cannot handle mixed numeric and categorical variables. Using Gower's index [9] to construct a proximity matrix is a good alternative for the clustering analysis because it enables us to transform outcomes of different types of variables into a single mathematical value including categorical and numeric variables. The original form of Gower's index handles interval, nominal and binary data as a similarity coefficient between 0 and 1. Kaufmann and Rousseeuw [9] described a slight generalization of this coefficient which covers ordinal and ratio variables in addition to the ones mentioned for the original index. With a simple transformation, Gower's original similarity coefficient [47] can be converted into a dissimilarity value between 0 and 1. Kaufmann and Rousseeuw [9] transformed the similarity coefficients into dissimilarities by using the simple expression given in Equation 2.10. Note that the similarity to dissimilarity transformation is possible with this expression because the similarity coefficient is within $[0,1]$ where 0 represents no similarity and 1 represents perfect similarity. In Equation 2.10, $s(i,j)$ is the similarity between objects i and j, and $d(i,j)$ is the dissimilarity between same objects. After the transformation of the equation, $d(i,j)$ lies within the same interval but this time 0 and 1 switch their roles to perfect similarity and no similarity between objects, respectively. These improvements to the original index enable us to build a dissimilarity matrix which can be used later as an input for cluster analysis. The only downside for Gower's index is that the index is linear. The discrimination capacity of the index might not be as powerful as a quadratic or a higher degree polynomial expression.

$$d(i,j) = 1 - s(i,j) \tag{2.10}$$

A generalization of Gower's index* [9] represented by $d(i,j)$ is provided below.

$$d(i,j) = \frac{\sum_{f=1}^{p} \delta_{ij}^{(f)}\, d_{ij}^{(f)}}{\sum_{f=1}^{p} \delta_{ij}^{(f)}}$$

$$\delta_{ij}^{(f)} = \begin{cases} 1 & if\ both\ X_{if} and\ X_{jf}\ are\ not\ missing \\ 0 & if\ X_{if}\ and/or\ X_{jf}\ missing \\ 0 & if\ f\ is\ an\ asymmetric\ binary\ and\ X_{if} \&\ X_{jf}\ has\ a\ 0-0\ match^{\dagger} \end{cases}$$

If f is a binary or nominal variable:

$$d_{ij}^{f} = \begin{cases} 1 & if\ X_{if} \neq X_{jf} \\ 0 & if\ X_{if} = X_{jf} \end{cases}$$

If f is an interval-scaled or ordinal variable:

$$d_{ij}^{f} = \frac{|X_{if} - X_{jf}|}{R_f}$$

$$R^{(f)} = X_{max}^{(f)} - X_{min}^{(f)}$$

2.3 Splines

The word "spline" comes from the East Anglian dialect with a meaning of a thin wood or metal piece [3] that was used by shipbuilders and draftsmen to build smooth curves. With developments in computer technology, using spline models has become a popular approach to

* $d(i,j)$, p, f, X_{if}, and $R^{(f)}$ are the dissimilarity between objects i and j (Gower's dissimilarity index), the number of variables, the index of a variable, the value of object i for variable f, the range of variable f (if f is an ordinal or an interval-scaled variable), respectively.

† For further details about 0-0 match, please refer to the similarity coefficients on page 42.

produce smooth curves in computer graphics [4]. It is widely accepted that the first mathematical reference to the word “spline” in polynomial piecewise approximation is Schoenberg’s article of 1946 [58]. However, the ideas have their roots in the aircraft and ship-building industries [4].

Let m be the spline order where the polynomial degree D is $m-1$. A spline of degree D is a continuous function formed by connecting polynomial segments of degree D so that: (1) the function has $D-1$ continuous derivatives, and (2) the D^{th} derivative is constant between knots. The points where the segments connect are called knots of the spline and spline functions project high degree of smoothness at these points. Let $\xi_0 < \xi_1 < \cdots < \xi_N < \xi_{N+1}$ be the sequence of knots where N is the number of interior knots, and ξ_0 and ξ_{N+1} are end (boundry) knots. A spline degree of D can be represented as a power series in Equation 2.11 with the truncated power function notation given in Equation 2.12. In Equation 2.11, α_j and γ_i are spline coefficients.

$$S(x) = \sum_{j=0}^{D} \alpha_j x^j + \sum_{i=0}^{N+1} \gamma_i (x-\xi_i)_+^D \tag{2.11}$$

$$(x-\xi_i)_+^D = \begin{cases} (x-\xi_i)^D & x > \xi_i \\ 0 & otherwise \end{cases} \tag{2.12}$$

Splines constitute a reasonable approach for the nonparametric estimation of manufacturing cost functions. Unfortunately, the commonly known splines are restricted to continuous predictors (attributes). This is a disadvantage when it comes to the generalization of using splines for manufacturing cost estimation problems since we may encounter mixed categorical and numeric predictors. In this research, as can be seen from Figure 2.7, we discussed three main spline approaches: univariate, multivariable models. Univariate approaches consist of four spline functions, namely interpolating splines, smoothing splines, basis splines (B-splines) and penalized B-splines (P-splines). These can be extended to multivariable cases with tensor product and additive models.

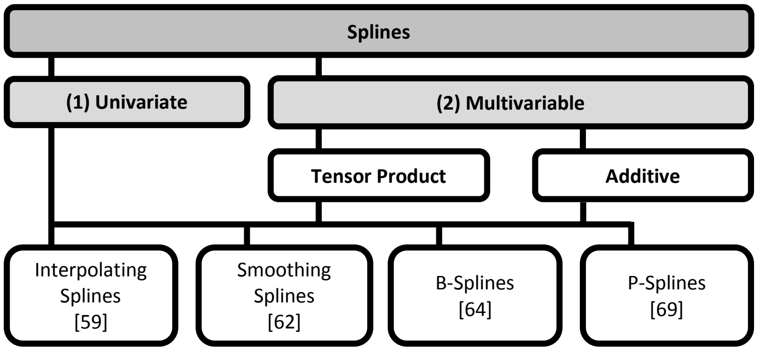


Figure 2.7: Classification of spline models in terms of variable complexity

Interpolating splines [59] is a sophisticated form of interpolation where the interpolant is a piecewise polynomial spline. It is a popular technique for designing planar curves [60] and is preferred over regular polynomial interpolation because the interpolation error is relatively small even for low order of spline functions because it creates less possibility of wild oscillations between data points [61]. The most common spline interpolation is cubic interpolating splines because they produce an interpolated function that is continuous through to the second derivative. That is, cubic interpolating splines produce a curve that appears to be smooth and seamless. The interpolation interval is first divided into small subintervals. Each of these subintervals is interpolated by using a D^{th} degree polynomial. The polynomial coefficients are chosen carefully to satisfy certain conditions depending on the interpolation method but generic requirements are function continuity at the knots and passing through all data points.

The smoothing spline is a technique to fit a smooth curve to a noisy set of data by placing knots at all observation points. It is introduced [62] and generalized [63] by Reinsch. Smoothing splines circumvent the knot selection problem since they just use all input points as knots. Smoothing splines considers two main key factors: (1) the goodness-of-fit to the data, and (2) the roughness penalization (smoothing parameter). The most important concern for smoothing splines is choosing this smoothing parameter and it can be done through cross-validation. If a smoothing spline is needed to fit non-uniformly shaped noisy data, the results can be unsatisfactory due to visiting every data point.

The truncated power function representation is not well suited for computational efficiency because successive terms tend to be highly correlated. A numerically more stable representation of splines can be written as linear combinations of a set of basis functions called B-splines. B-splines was a major development in spline theory and is now the most used in spline applications and software. The term "B-spline" was introduced by Curry and Schoenberg [64]. B-spline is a generalization of the Bézier curves using the de Boor recursion formula [65]. The B-spline function $B(x)$ of degree D is a parametric curve composed of a linear combination of basis B-splines $B_{i,D}(x)$ and is given in Equation 2.13 where the β_i terms are called "control points" or "de Boor" points. Notice that in the equation we use the same notation as in Equation 2.11. The de Boor recurrence relation formula is given in Equation 2.14 where the $B_{i,j}$ functions are called the "i^{th} B-spline basis functions of order j". Notice that, the index of the knots $i = 0, \dots, N + 2m - 1$ is from the augmented knots after resetting the first element of the augmented knot set where $\xi_{-(m-1)} = \cdots = \xi_0 < \xi_1 < \cdots < \xi_N < \xi_{N+1} = \cdots = \xi_{N+m}$. For the recursive calculations, any division to zero is defined as zero.

$$B(x) = \sum_{i=0}^{N+D} \beta_i B_{i,D}(x), \quad x \in [\xi_0, \xi_{N+1}] \tag{2.13}$$

$$B_{i,0}(x) = \begin{cases} 1 & if\ \xi_i \le x < \xi_{i+1} \\ 0 & otherwise \end{cases}$$

$$B_{i,j+1}(x) = \frac{x - t_i}{t_{i+j+1} - t_i} B_{i,j}(x) + \frac{t_{i+j+2} - x}{t_{i+j+2} - t_{i+1}} B_{i+1,j}(x) \tag{2.14}$$

Penalized B-splines (P-splines) is an intermediate solution between regression and smoothing splines proposed by Eilers and Marx [66]. B-splines are attractive for non-parametric modelling but choosing the appropriate number of knots with their locations is a significant issue. Eilers and Marx proposed a roughness penalization procedure by starting with a relatively large number of knots but still less than one per observation. This method combines the reduced knots of regression splines with the roughness penalty of smoothing splines where the coefficients are determined partly by the data to be fitted and partly by an additional penalty function that aims to avoid over fitting.

Tensor product splines is an extension to the one-dimensional spaces of polynomial splines over a space of multi-dimensional splines by taking tensor products. Because of the outer product nature of the multi-dimensional space, many properties of polynomial splines in one dimension are retained, such as working with single dimension B-spline functions [67]. Tensor product models consider interaction terms between univariate spline functions. We will use an approach which takes the tensor products of spline functions into account to handle multiple predictors. For further details about tensor products, please see the next chapter.

Hastie and Tibshirani [68] proposed using additive models with separate smoothers to accommodate non-linear covariate effects. Additive models can also be used with spline functions that can extend univariate spline models over multidimensional space. It is a blend of generalized linear models with additive models and provides the potential for better fit to data than parametric models but with some loss of interpretability. Unfortunately, unlike tensor product spline models, additive models lack interaction terms but are very easy to implement with multiple linear regression or regression splines. We discuss the performance of additive models compared to tensor product splines in sections 4.3 and 4.4.

Multivariate adaptive regression splines (MARS) is a non-parametric regression technique generalization of spline methods for function fitting introduced by Freidman [69]. Freidman extended the univariate properties of splines to multiple variables considering complex relationships between predictors. However, according to the non-parametric regression procedure, no assumption is required for the underlying functional relationships between the dependent variable and predictors. That is, MARS employs the tensor product representation with a very large number of eligible knot locations on each variable. The MARS model is a weighted sum of truncated power basis functions (as in B-splines). It can handle both continuous and categorical data [70], but requires a discretization of continuous data into ordinal categories. This is called recursive partitioning where continuous data is partitioned into disjoint regions. MARS has a tendency to perform better for only numeric data. This is one of the downsides that makes us seek a better alternative to accommodate mixed categorical and numeric design attributes. Furthermore, MARS restricts the maximum degree of interaction terms to one.

2.4 Related Work

The first two parts of the literature review chapter focused on the possible cost estimation and clustering techniques and an overview of their basic applications; however, this section is a concentration of the most relevant work with our research. In this section, we present applications of clustering techniques alone, cost estimation techniques alone and their combination for numerical data, categorical data or both.

In this dissertation, our aim is to distinguish appropriate clustering methods or similarity measures that can handle heterogeneous types of data. That is, a clustering technique (or a complementary similarity measure) that can process a dataset with both categorical and numerical variables. Any specific clustering technique or similarity measure which fails to satisfy our variable treatment requirements would be inadmissible in this study. Due to this constraint, we are able to eliminate most of the methods discussed earlier in the chapter. However, it is necessary to consider more information about the selection of clustering methods and/or similarity measures. We provide a summary of the most common clustering techniques with their computational complexities in Table 2.6 which highlights the clustering techniques which can process mixed categorical and numerical variables. Even though enumerations of the whole search space are not quite clustering techniques, they are included in the table as a benchmark to illustrate the complexity of search space. The information in Table 2.6 is a snapshot of our clustering technique selection reasoning and it is discussed in the next chapter in further detail.

One of the most relevant studies that have been conducted so far is the work of Angelis and Stamelos [71] concerning software cost estimation. That is, the estimation of the required effort to develop specific software based on analogies with previously undertaken projects. Even though this is not a manufacturing cost estimation attempt, it is relevant to our research. Angelis and Stamelos developed a non-parametric bootstrap simulation tool to investigate the accuracy of the underlying estimation methodology which is constructed on Euclidean, Manhattan and Chebyshev distances between an active project and historical projects. The idea of software cost estimation based on analogies existed in the literature previously but the study of Angelis and

Stamelos is the most comprehensive and illustrative work. Although this work specifically uses similarities between historical projects and an active project in the development phase with an emphasis on Gower's index, it does not employ any clustering technique or an estimation model such as regression models or neural networks.

Table 2.6: Overview of the most common clustering methods

Clustering Technique	Computational Complexity[*]		Type of Data[†]			Sensitivity to Outliers	Best Data Set Size[‡]	Initial Seed Dependence	Comments
	Time	Space	C	N	M				
Enumeration[§]		$C(N,K)$	+	+	+	No	S	No	Impractical / prohibitive
Enumeration[**]		$K^N/K!$	-	+	-	No	S	No	Impractical / prohibitive
Single Linkage	$O(N^2)$	$O(N^2)$	+	+	-	Yes	S	No	Good for taxonomy
Complete Linkage	$O(N^2)$	$O(N^2)$	+	+	-	No	S	No	Not sensitive to outliers
Average Linkage	$O(N^2)$	$O(N^2)$	+	+	-	No	S	No	Good for taxonomy
Ward's Method	$O(N^2)$	$O(N^2)$	-	+	-	Yes	S	No	Sensitive to normality
k-means	$O(NKd)$	$O(N+K)$	-	+	-	Yes	L	Yes	Easy to implement
k-medoids	$O(Kd(N-K)^2)$	$O(N+K)$	+	+	+	No	S	No	Relatively complex
k-modes	$O(NKd)$	$O(N+K)$	+	-	-	No	S – L	Yes	Best for binary data
k-prototypes	$O(NKd)$	$O(N+K)$	+	+	+	Yes	S – L	Yes	Efficient as k-means
Branch & Bound	N/A	Varies	-	+	-	No	S	No	Gives exact solution
Model Based	$O(N\log N)$	N/A	+	+	+	No	S – L	No	Non-arbitrary similarity
Graph Theoretic	$O(N^2)$	$O(N^2)$	-	+	-	No	S	No	For irregularly-shaped clusters
Meta-Heuristics	Varies	Varies	+	+	+	No	L	Possibly	Gives solutions fast
Cluster Ensemble	Varies	Varies	+	+	+	No	S	Varies	Consolidation issues

Lee et al. [72] proposed a two-phase software cost estimation method which is based on clustering analysis and neural networks for mixed numerical and categorical data. For quantitative attributes, they used average Euclidean distance. On the other hand, for nominal attributes, the Jaccard coefficient[††] is calculated. Finally, the value of the Jaccard coefficient and the average Euclidean distance which represent the similarity between two objects (software projects) for their

[*] N: Number of objects, K: Number of clusters, d: Number of variables (dimension)

[†] C: Categorical, N: Numerical, M: Mixed Categorical and Numerical

[‡] S: Small, L: Large

[§] Enumeration expression is written for combinatorial problems where K objects are chosen out of N observations as cluster centers

[**] Enumeration expression is written for combinatorial problems where N observations are allocated into K clusters with the nearest mean

[††] Please refer to the similarity coefficient #5 on page 43.

corresponding variable type are consolidated into a single resemblance coefficient. After repeating this combining process for all object pairs, a matrix of resemblance coefficients is obtained which later becomes the main feed to hierarchical clustering analysis. The output of the clustering analysis is the input to the neural network training phase. A neural network which is trained using the output of clustering analysis promises higher accuracy than a non-cluster-integrated neural network. This is another motivation for us to devise clustering-based manufacturing cost estimation models. As a downside, their work was limited to single linkage hierarchical clustering without the existence of ordinal and binary variables.

Khoshgoftaar and Xu [73] extended software cost estimation efforts with a fuzzy *c-means* (k-means) clustering approach. Because software experts define the level of complexity according to their subjective opinions, using cost associated variables which take certain numerical values does not reflect the true nature of software cost estimation efforts. Hence, this research accounts for the imprecision and vagueness of expert knowledge with linguistic variables and fuzzy rules. In the data pre-processing phase, all linguistic and numerical variables are transformed into software cost attributes which determine multiplying factors. These multiplying factors take only numerical values and show the complexity of development effort for their corresponding attribute. The fuzzy c-means approach devised by Khoshgoftar and Xu is an intra-cluster variance minimization algorithm and has similar disadvantages as does the k-means algorithm. Although the whole method appears to handle mixed numerical and categorical variables, in fact, the clustering module itself is unfortunately limited to numerical data. Additionally, cost attributes which are specifically defined for software cost estimation efforts are not equivalent to manufacturing cost variables. Therefore, software cost estimation applications do not substitute for a study in the manufacturing cost estimation field.

The performance of multivariate adaptive regression splines (MARS) for software cost estimation efforts was investigated by Pahariya et al. [74]. It has a similar aim to our methodology to establish cost estimation models based on historical data because with MARS, continuous and categorical variables can be modeled [70] but with the downsides of MARS that were discussed in the previous chapter. In their analysis, the majority of software project attributes have been eliminated. That is, slightly more than two thirds of the attributes were removed from the dataset including the categorical variables. The real challenge in our methodology is dealing with mixed numeric and categorical variables, and Pahariya et al.'s work is not very helpful considering unreasonable simplifications in the data preparation phase.

Wolfe et al. [75] conducted research on estimating total direct medical costs of people with rheumatoid arthritis. These medical costs include physician and healthcare worker visits, medications, diagnostic tests and procedures, and hospitalization. It is an extensive 3-year study on more than 7,500 patients (data observations) using demographic variables such as age, ethnic origin, education level, medical history. According to their statistical findings, the effect of age on total costs indicated a V-shaped scatter. To model this relatively complex age vs. cost relationship, they used linear splines with a single interior knot. Even though, Wolfe et al. implemented an approach to estimate the cost based on categorical and numeric demographic predictors, they only used an integer scale numeric variable, age, to develop spline models. Their approach is primitive relative to our manufacturing cost estimation methodology and it does not account for mixed categorical and numeric data.

Another cost estimation related research was done by Almond et al. [76] about the hospitalization costs of low birth weight on heavier and lighter infants from twin pairs born in the United States. In this study, health of a newborn is modeled using categorical and numeric variables, namely birth weight, race, education, age, and maternal smoking during pregnancy. To quantify the health status of a newborn, among these five variables, only birth weight factor is used to build a piecewise linear spline model. Unfortunately, none of the categorical factors have been considered in the spline model. Moreover, the outcome of the parametric model they devised is the health of a newborn, not the hospital cost because actual cost values associated with these hospitalization cases did not exist in the dataset. That is, there is no corresponding actual cost data to assess the performance of the estimation model. Almond et al. calculated hospital cost by adding generic expenses for each treatment performed on a newborn. The research lacks two aspects compared with our cost estimation methodology: (1) Not utilizing categorical variables in the spline model, and (2) Not using actual cost values to evaluate the performance of the underlying parametric model.

Deploying a spline approach has been prevalent for estimating medical costs incurred by treating diseases, especially in real clinical trials. Carides et al. [77] presented a procedure for estimating the mean cumulative cost of long-term treatment on two clinical studies: (1) Heart failure clinical trial of left ventricular dysfunction, and (2) Ulcer treatment. A two stage estimator of survival cost with parametric regression, and a non-parametric regression with cubic smoothing splines are devised to exploit the underlying relationship between total treatment cost and survival time. The results of the parametric and non-parametric approaches are dominant in performance compared with commonly used two cost estimation methods in medical industry. However, only continuous covariates are used in the two-stage model and the effect of both categorical and numeric attributes associated with each of these clinical studies were not considered.

Valverde and Humphrey [78] developed translog, Fourier, and cubic spline models to predict the cost effects of 20 individual bank mergers. The motivation behind this research was to accurately estimate the decrease in unit costs due to the merger. The underlying performance metric was the actual cost changes affecting all merging banks. Unfortunately, the proposed models were able to identify the sign of the merger-associated cost change only in one-third of the cases. Only two numeric variables were under consideration in the cubic spline models: (1) Value of loans, and (2) Value of securities (and other assets). Accuracy of predicting the sign of the cost change suffered from the limited number of data observations in this research. Also, categorical merger bank attributes were not implemented in the cost estimation efforts.

Table 2.7 highlights the most relevant cost estimation literature using clustering techniques/splines and type of data. A "+" sign indicates that the underlying research is in which specific area of application, what kind of approach is devised, and what type of data is used. For instance, Carides et al. [77] implemented spline models to estimate clinical costs by using numeric data. As you notice, clustering techniques or spline models have not been used in manufacturing cost estimation efforts because of the complex relationships between categorical and numeric design attributes. It is our motivation in this research to evaluate performance of the mentioned techniques and models in manufacturing.

Table 2.7 Overview of the most relevant research

	Area of Application*			Estimation Approach		Type of Data†			Comments
Article	SCE	CCE	MCE	Clustering	Splines	C	N	M	
Angelis and Stamelos [71]	+							+	Analogical relationships used
Lee at al. [72]	+			+				+	No ordinal or binary variables
Khoshgoftaar and Xu [73]	+			+			+		Subjective attribute assignments
Pahariya et al. [74]	+				+		+		Omitted majority of variables
Wolfe et al. [75]		+			+		+		Considered one variable in splines
Almond et al. [76]		+			+		+		Used estimated medical costs
Carides et al. [77]		+			+		+		Promising estimation results
Valverde and Humphrey [78]					+		+		Limited data with poor accuracy

* SCE: Software Cost Estimation, CCE: Clinical Cost Estimation, MCE: Manufacturing Cost Estimation

† C: Categorical, N: Numeric, M: Mixed Categorical and Numerical

Chapter 3

Methodology

In this chapter, we propose the cost estimation approach in detail. The purpose of this chapter is to discuss the main components of the proposed cost estimation methodologies. The chapter consists of three parts: (1) Clustering based cost estimation approach, (2) Spline based cost estimation approach, (3) Validation of manufacturing cost estimation models. The first part discusses the clustering technique and dissimilarity measure we employ to handle mixed categorical and numeric data. Also, it focuses on how to build cost estimation models for each cluster along with how to determine the appropriate number of clusters. The second part discusses which spline model we use to address complex relationships between mixed numeric and categorical design attributes. Furthermore, this section specifically shows the kind of parameters we used in the spline model building phase. The last part shows the validity of the corresponding estimation models for each approach, cluster specific models and spline models.

3.1 Clustering Cost Estimation Approach

3.1.1 Grouping Products

Our clustering cost estimation approach is a two-phase process. In the first phase, we use all historical products to evaluate possible clustering formations and to build a cost estimation model for each cluster. There are several clustering techniques that we can use with combinations of dissimilarity measures. However, the main question is if they can handle mixed numeric and categorical data. The second phase is the cost prediction phase in which a new design is assessed for the best cluster fit and then the corresponding cost estimation model is used. According to design similarities between a new design and the existing clusters established in the first phase, we select the best cluster to which the new design should be assigned. Once the best cluster is found, the remaining part is to use the cluster specific cost estimation model to predict the manufacturing cost of the new design. Figures 3.1 and 3.2 are illustrations of proposed methodology for the first and second phases, respectively.

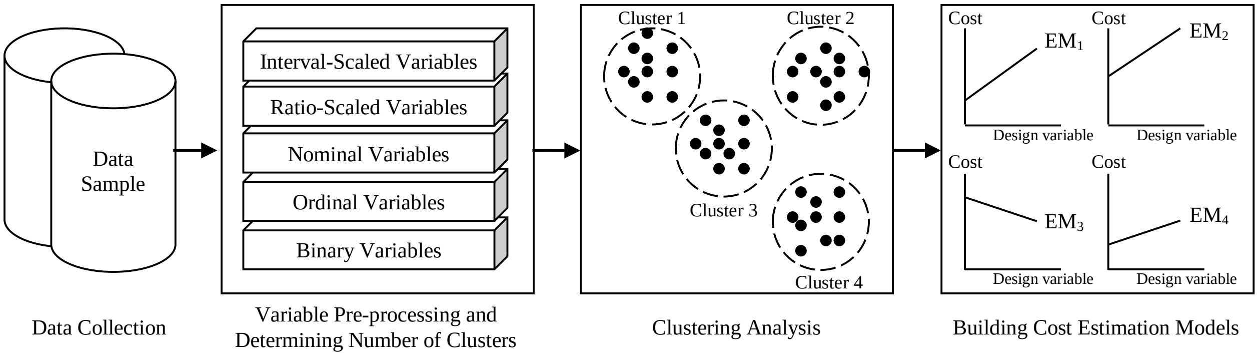


Figure 3.1: The first stage of the methodology: cluster analysis and calculating estimation model for each cluster

3.1.2 Determining the Number of Clusters

Determining the number of clusters is a minor mathematical problem. Nevertheless, it is significant and has the capability of causing the prediction accuracy to be either very satisfactory or very poor. Unfortunately, there is no definitive methodology for determining the number of clusters [79]. In a practical sense, graphically assessing the data scatter is a good start but when there are more than two or three dimensions (i.e., variables), this is not as practical as it first appears. Also, when the data is mixed with categorical and numeric values, it is very hard to identify clusters visually.

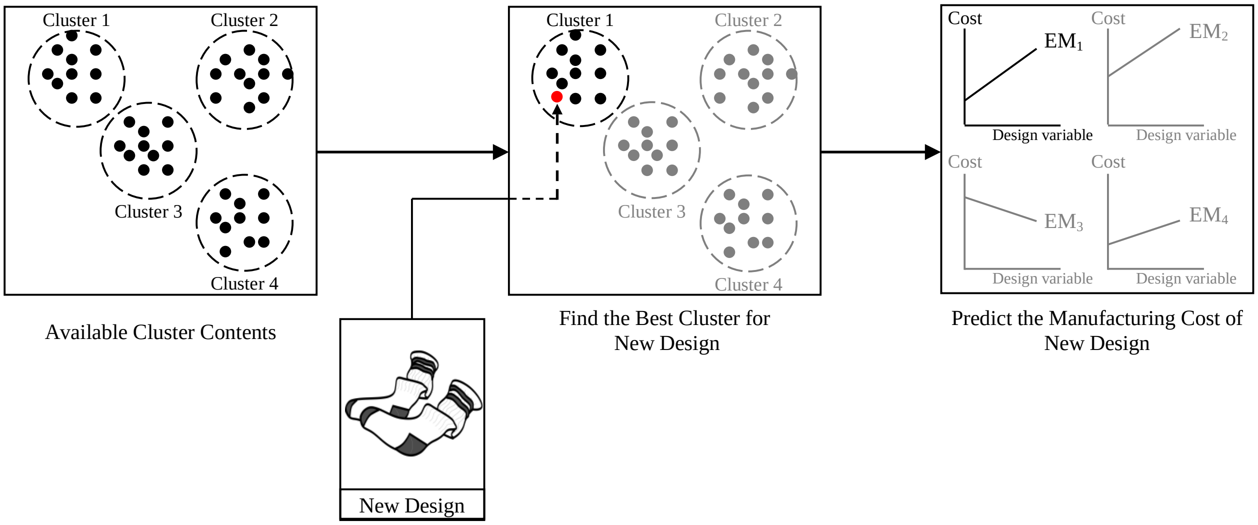


Figure 3.2: The second stage of the methodology: finding the best cluster and predicting the manufacturing cost of a new design

Even though it is possible to have an idea of how many product groups exist in a database based on experts' opinions in a company, the groups are usually not distinct or the given opinions do not represent the similarities among products perfectly. Opinions of experts may be misleading during this cost estimation process because there are many possible logical classifications of products such as according to physical appearance, material similarities, or even similar manufacturing stages. The distinction power of a similarity measure becomes very crucial in this phase because it forms the basis of these comparisons among products or products with clusters. Since opinions of experts are subjective, they may not lead to proper product groupings. Additionally, there are no labels in the product database which indicate the cluster category for each product. Absence of labels for product groups makes the whole clustering process unsupervised. That is, no target clusters are known for products and the proper number of groups is not known a priori. During the cluster analysis stage, we need to choose the appropriate number of clusters. This is directly linked with how many cost estimation models are required to be built at the end of the first phase.

Many techniques have been developed and a number of statistics devised by statisticians to determine the appropriate number of clusters. According to Milligan and Cooper's extensive investigation [80] on 30 statistics for four hierarchical clustering methods, the top five performers are Calinski and Harabasz's pseudo F (PSF) [81], Duda and Hart's $J_e(2)/J_e(1)$ ratio [82] which can be transformed into pseudo T^2 (PST2) statistics, Dalrymple-Alford's C-index [83], Baker and Hubert's Gamma [84] which was adopted from Goodman and Kruskal's gamma (γ) [85], and Beale's F-ratio [86]. Sarle's cubic clustering criterion (CCC) [87] also has competitive performance and identifies the usage of too many clusters at the highest rate among these 30 statistics. Unfortunately, these statistics are specifically built for hierarchical clustering methods, but they may still provide some useful information about the appropriate number of clusters for non-hierarchical clustering. Since Milligan and Cooper's study is based on a series of simulated datasets with two to five distinct non-overlapping clusters, it is not appropriate to draw a generalized framework from their results. This is because the performance of a statistic (or a method) can depend on the clustering algorithm used and the structure of the clusters [88].

The pseudo F statistic is the ratio of inter-cluster variance to within cluster variance [81]. The intention of the statistic is to capture the tightness of clusters [89]. Large values of PSF are an indication of well separated clusters and a correct number of partitions [79]. However, when the dataset that is being analyzed consists of discrete variables, PSF can be expected to increase when the number of partitions increase, so sudden jumps or local peaks are more meaningful than the global maximum value [79]. Equation 3.1 is the formulation of PSF where k is the number of clusters and n is the total number of observations. Also, B_k is the between-cluster sum of squares matrix and W_k is the within-cluster sum of squares matrix for k partitions. The trace of a square matrix is the sum of its main diagonal elements.

$$PSF(k) = \frac{trace(B_k)/(k-1)}{trace(W_k)/(n-k)} \tag{3.1}$$

Duda and Hart's $J_e(2)/J_e(1)$ ratio [82] proposes a criterion to assess whether an existing cluster (m^{th} cluster) should be divided into two subclusters (clusters k and l), or not. According to the ratio, $J_e(2)$ is the sum of squared errors within the cluster when the data are divided into two clusters and $J_e(1)$ is the sum of squared errors when there is only one cluster present. It requires a hypothesis testing procedure where a suboptimal partition is rejected if the ratio is smaller than an approximate critical value [82]. The correct number of clusters can be obtained when the hypothesis is first rejected. The critical value is a function of a standard Normal score (z), the number of variables (p), and the number of observations (n). The formulation of the ratio and the critical value are given in Equations 3.2 and 3.3, respectively. W_k, W_l and W_m are the sums of the squared errors within clusters k, l and m, respectively. The $J_e(2)/J_e(1)$ ratio is closely related to PST2 [90] and this relationship is clearly shown in Equation 3.4 where n_k and n_l are the number of observations in clusters k and l. The difference between two clusters that are being merged at a given step is quantified by PST2. Hence, a smaller PST2 value indicates more distinct clusters. Since PST2 is computed for several different numbers of clusters, it would be better to plot PST2 with respect to the number of clusters and read the plot from right to left. The correct number of clusters is the previous number (if we read from right to left) to where a distinct jump is observed in the PST2 value [79].

$$\frac{J_e(2)}{J_e(1)} = \frac{W_k + W_l}{W_m} \tag{3.2}$$

$$Critical\ Value = 1 - \frac{2}{\pi p} - z\sqrt{\frac{2(1 - 8/\pi^2 p)}{np}} \tag{3.3}$$

$$\frac{J_e(2)}{J_e(1)} = \frac{1}{1 + (PST2)/(n_k + n_l - 2)} \tag{3.4}$$

Sarle developed the cubic clustering criterion (CCC) to test the hypothesis that the data has been sampled from a uniform distribution on a hyperbox* [87]. When this hyperbox is divided into clusters, these clusters are shaped roughly like hypercubes. The CCC value is very accurate in determining the existence of too many clusters [80]. While positive values of CCC (values exceeding 2 or 3) can be interpreted as the presence of good clusters, large negative values indicate the existence of outliers that should be removed before further analysis [87]. Also, Sarle pointed out the failure of CCC when variables are highly correlated or cluster structures are irregularly shaped. This is because of the violation of the main assumption of hypercubical clusters. The formulation of CCC is given in Equation 3.5 where R^2 is the observed value for the proportion of the sum of squares explained by the clusters and $E(R^2)$ is the expected R^2 that is obtained by clustering uniformly distributed points into hypercubes. The letter K represents a variance stabilizing transformation and detailed information about it can be found in Sarle's technical report [87].

* A p-dimensional right parallelepiped

$$CCC = ln\left[\frac{1 - E(R^2)}{1 - R^2}\right]K \tag{3.5}$$

The user manual of the SAS statistical software [79] suggests monitoring consensus among PSF, PST2 and CCC, where each statistic designates a particular number of clusters according to its individual decision criteria. Even though these statistics assume that all variables are ratio or interval scaled continuous values, it is still possible to find the appropriate number of clusters for purely categorical or mixed categorical and continuous data by monitoring some other statistics such as Dalrymple-Alford's C-index [83], Baker and Hubert's Gamma [84] or Rousseeuw's silhouette width [91]. These three statistics operate on a dissimilarity matrix and a vector of integers indicating the cluster number to which each observation is assigned. Since a part of our research is clustering mixed categorical and numeric data, we can simply derive a dissimilarity matrix and then use it to calculate the C-index, Gamma and silhouette width statistics. Monitoring these three statistics are more relevant than the others for mixed categorical and numeric datasets.

Dalrymple-Alford's C-index [83] was assessed by Dimitriadou et al. [92] on artificially created high-dimensional binary datasets that were clustered by two different algorithms. Since the index operates on a distance matrix, all pairwise distances in a dataset must be computed and stored. Computation of all pairwise distances was found to be prohibitive for large datasets [92] before the early 2000's but this is not the case anymore. The C-index is expressed in Equation 3.6 where d_w is the sum of all within cluster distances. The number of clusters which minimizes the C-index should be chosen. The index varies within the $[0,1]$ interval.

$$C\text{-}index = \frac{d_w - \min(d_w)}{\max(d_W) - \min(d_w)} \tag{3.6}$$

Baker and Hubert devised an index called Gamma [84] which was adopted from Goodman and Kruskal's gamma (γ) [85] to use in clustering applications. The index basically compares within cluster distances with between cluster distances [80] where a pair of distances is considered consistent (inconsistent) if the within cluster distance is less (greater) than the between cluster distance [88]. The main question concerning Gamma is whether the assignment quality of a clustering algorithm can be reasonably achieved by grouping objects arbitrarily [5]. Gamma was found to be one of the best performing statistics among the 30 considered by Milligan and Cooper [80]. The expression of Gamma is given in Equation 3.7 where $s(+)$ and $s(-)$ represent the number of consistent and inconsistent comparisons, respectively. Since the numerator of the Gamma index is normalized with respect to the total number of consistencies and inconsistencies, it takes a value within $[-1, 1]$. The number of clusters should be chosen as the one which maximizes the index in the positive region.

$$\gamma = \frac{s(+) - s(-)}{s(+) + s(-)} \tag{3.7}$$

Another index which is applicable for mixed numeric and categorical data is Rousseeuw's silhouette width [91]. It was devised to assess how well each object lies within its assigned cluster. Even though the silhouette width was first developed for partitioning around medoids, it is possible to use it in any context for which a distance matrix can be derived. Unfortunately, neither Milligan and Cooper's study [80] nor Dimitriadou et al.'s study [92] evaluated and compared the performance of the silhouette width with other indexes or statistics. The fundamental procedure behind this approach is plotting the average silhouette widths for the entire dataset that are obtained from different choices for the number of clusters, and selecting the number of clusters which maximizes the index. The silhouette width has a range of -1 to 1 where the positive region indicates a better classification. A value in the negative region indicates a poor classification. A value above 0.5 is interpreted as the existence of a strong cluster structure depending on how close it is to 1 [9]. That is, the value of the index approaches 1 when the quality of classification increases. The expression of the silhouette width $s(i)$ is given in Equation 3.8. In the expression, $a(i)$ is the average dissimilarity of the i^{th} observation with all other observations within its assigned cluster, and $b(i)$ is the minimum average dissimilarity of the same observation with all other observations in each cluster which the observation i is not a member of. According to the intrinsic logic of the silhouette width, the index is not defined when the number of clusters is 1.

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{3.8}$$

There are several other indexes and statistics that are either less meaningful or not meaningful for mixed categorical and numeric data such as including Davies and Bouldin index [93], Tibshirani et al.’s gap index [94], Krzanowski and Lai index [95], and the Hartigan index [96]. Thus, we will not benefit from such indexes. Our methodology of selecting the appropriate number of clusters is neither deterministic nor arbitrary, but it is consistent with and also as simple as the one defined in the user manual of SAS for numeric data [79]. We look for consensus among three statistics, namely C-index, Gamma and silhouette width, and these statistics can be applied regardless of the type of data. We seek local peaks of the Gamma and silhouette width combined with local troughs of the C-index to choose the number of clusters. Note that this choice may not be unique.

3.1.3 Choice of Clustering Algorithms

One of the most important aspects of this research is choosing a clustering algorithm which satisfies the requirements. That is, to devise a clustering technique which can handle mixed numeric and categorical data and can provide the most satisfactory cluster assignment. As we mentioned earlier in the first chapter, a complementary similarity measure with high discriminating power is crucial; however, the choice of clustering algorithm is more important than the choice of similarity measure [5]. Most of the hierarchical and non-hierarchical clustering techniques are eliminated due to the limitation of those algorithms on the type of data. To conduct a cluster analysis for mixed data in terms of variable types, the applicable clustering algorithms are narrowed down to two partitioning algorithms, namely k-medoids and k-prototypes.

The final cluster contents obtained by running the k-prototypes algorithm can depend on the initial starting point. In each k-prototypes run, the algorithm may give similar but different results. To overcome this issue, the k-prototypes algorithm should be run several times or even compared with results obtained by a different algorithm such as k-medoids. Similar to the k-means algorithm, it is also sensitive to outliers because it works based on the similar error minimization principle. Recall that a weighted summation of Euclidean distance and the simple matching coefficient constitutes the framework of the k-prototypes algorithm. That is, the objective function of k-prototypes attempts to integrate a quadratic expression with a linear expression. Instead, consolidating two linear or two quadratic similarity expressions might be more mathematically appropriate. That is the reason why the weighting coefficient in the objective function determines the contribution of each variable type. The main purpose of the weight is to avoid favoring any type of variable, categorical or numeric. However, there is no exact or definitive heuristic to calculate the appropriate weighting coefficient. Thus, k-prototypes has some drawbacks when grouping products with mixed categorical and numeric design attributes.

The k-medoids algorithm was found to be more robust than any clustering technique that uses the error sum of squares [23]. To clarify, the k-medoids algorithm is not based on minimizing the error sum of squares. Instead, it finds a set of representative observations (medoids) for each cluster and then allocates all other remaining observations to these clusters according to the closest distance to each medoid. Recall that a medoid is an observation from the dataset and represents the center of a cluster. This is advantageous in three aspects: (1) Possibility of clustering mixed data when a dissimilarity matrix can be derived, (2) Possibility of handling outliers, and (3) Elimination of making assumptions about underlying distributions such as multivariate normality.

We employ the k-medoids clustering algorithm as described in Kaufmann and Rousseeuw [9]. They implemented the k-medoids algorithm in a program called “PAM”. According to Kaufmann and Rousseeuw’s “PAM”, the algorithm consists of two phases. These phases are called BUILD and SWAP. The first phase, BUILD, constructs an initial solution of k representative objects and the second phase, SWAP, attempts to improve the set of representative objects. The objective function of the algorithm is to minimize the sum of distances (dissimilarities) of each object to their closest representative object. Figure 3.3 and Figure 3.4 show the steps of the BUILD and SWAP phases, respectively. These steps are taken from Kaufmann and Rousseeuw [9] without any modifications. Note that, the process of building an initial solution continues until k objects are found. The first two steps of the SWAP phase (steps 1 and 2) are carried out to calculate the effect of a swap between objects i and h on the value of clustering. Additionally, the last two steps of the same phase (steps 3 and 4) decide whether a swap is accepted. The algorithm considers all potential swaps. Notice that once the cluster centers (medoids) are determined during an iteration, all other remaining observations are allocated to the clusters according to the closest distance to each medoid.

1 Consider an object i which has not yet been selected.

2 Consider a non-selected object j and calculate the difference between its dissimilarity D_j with the most similar previously selected object, and its dissimilarity $d(j, i)$ with object i.

3 If this difference is positive, object j will contribute to the decision to select object i. Therefore we calculate:

$$C_{ji} = \max(D_j - d(j, i), 0)$$

4 Calculate the total gain obtained by selecting object i:

$$\sum_j C_{ji}$$

5 Choose the not yet selected object i which

$$\underset{i}{\text{maximizes}} \sum_j C_{ji}$$

Figure 3.3: The steps of the BUILD phase in “PAM” [9]

Reynolds et al. proposed a medoid based clustering algorithm [97] and compared its performance with k-medoids. However, they named the actual k-medoids algorithm PAM and their algorithm k-medoids. To clarify the ambiguity, we do not call Reynolds et al.’s algorithm k-medoids. Reynolds et al.’s algorithm and k-medoids differ according to their move operators and the construction of initial solutions. Since Reynolds et al.’s algorithm is an adaptation of k-means, it has similar drawbacks to the k-means algorithm. Recall that the performance of k-means can be dependent on the choice of the initial solution. Reynolds et al.’s algorithm chooses k objects at random to be the initial cluster medoids. However, the k-medoids algorithm carefully selects the initial representative objects with the BUILD phase of “PAM”. Therefore, the final cluster structure does not depend on a randomized initial solution, but it may require more computational power. The k-medoids algorithm moves from one solution to another one with the SWAP phase of “PAM”. According to the move operator of Reynolds et al.’s algorithm, in every iteration medoids for each cluster are calculated by finding object i within the cluster that minimizes Equation 3.9. In the equation, C_p is the p^{th} cluster ($p = 1,2, \dots, k$) and $d(i, j)$ is the dissimilarity of objects i and j. We will use the actual k-medoids algorithm in this research, since it outperforms Reynolds et al.’s algorithm by every performance measure except the execution time [97].

$$\sum_{j \in C_p} d(i, j) \tag{3.9}$$

3.1.4 Regression Models

Regression analysis is used to predict a dependent variable from one or more independent variables (predictors). One of the main purposes of regression analysis is to predict an unknown variable based on a regression function established with historical data on the predictors. Regression models are very easy to implement for developing predictive models in different application areas. Especially for manufacturing cost prediction, it is one of the methods that comes first to mind because it is the most sophisticated and frequently used way according to our observations on multiple manufacturing facilities. Manufacturing cost of a product is influenced by some independent factors (product design attributes or, simply, cost drivers). If we look at the linear relationship between the manufacturing cost and only one cost driver at a time, these figures may not represent a realistic analysis. Some of these cost drivers may have complementary, competitive or even more complex relationships with each other. For instance, printing a pattern on a modal Lycra material can be more laborious than printing on a cotton material for socks manufacturing. That is why the manufacturing cost is likely to increase relatively more when a pattern is printed on modal Lycra fabric rather than on cotton fabric. Thus, we should consider all cost relations at the same time. In a regression model for the manufacturing cost estimation problem, the outcome (or dependent) variable is the manufacturing cost, and independent (explanatory) variables are the cost drivers (design attributes in this case). While the manufacturing cost is a numeric continuous value, independent variables can be either categorical labels or numeric values. We assume a 5% confidence level for determining the significance of independent variables and their interactions. Checking interactions between variables is crucial because some variables create antagonistic or synergetic effects which may significantly impact the cost of a product. Some of the variables and interaction terms are eliminated if these are irrelevant or have statistically non-significant contribution on the cost value.

To reduce the computational load and also to avoid over parameterization issues we developed linear models. However, the performance of quadratic regression models is also assessed along with linear models. We develop cluster specific estimation models with the output of the cluster analysis. In other words, we constructed k regression models where k represents the number of clusters. That is, one regression model is developed for each cluster. To compare the true prediction performance of our clustering based cost estimation approach, we develop a regression model for the entire data that is built without any cluster analysis.

1 Consider a non-selected object j and calculate its contribution C_{jih} to the swap:

a If j is more distant from both i and h than from one of the other representative objects, C_{jih} is zero.

b If j is not further from i than from any other selected representative object ($d(j,i) = D_j$), two situations must be considered:

b1 j is closer to h than to the second closest representative object

$$d(j,h) < E_j$$

where E_j is the dissimilarity between j and the second most similar representative object. In this case the contribution of object j to the swap between objects i and h is

$$C_{jih} = d(j,h) - d(j,i)$$

b2 j is at least as distant from h than the second closest representative object

$$d(j,h) \geq E_j$$

In this case the contribution of object j to the swap is

$$C_{jih} = d(j,h) - D_j$$

It should be observed that in situation b1 the contribution C_{jih} can be either positive or negative depending on the relative position of objects j, h and i. Only if object j is closer to i than to h is the contribution is positive, which indicates that the swap is not favorable from the point of view of object j. On the other hand, in situation b2 the contribution is always positive because it cannot be advantageous to replace i by an object h further away from j than from the second closest representative object.

c j is more distant from object i than from at least one of the other representative objects but closer to h than any representative object. In this case the contribution of j to the swap is

$$C_{jih} = d(j,h) - D_j$$

2 Calculate the total result of a swap by adding contributions C_{jih}:

$$T_{ih} = \sum_j C_{jih}$$

3 Select the pair (i,h) which

$$\underset{i,h}{\text{minimizes}}\, T_{ih}$$

4 If the minimum T_{ih} is negative, the swap is carried out and the algorithm returns to step 1. If the minimum T_{ih} is positive or 0, the value of the objective cannot be decreased by carrying out a swap and the algorithm stops.

Figure 3.4: The steps of the SWAP phase in “PAM” [9]

3.2 Spline Cost Estimation Approach

Our spline cost estimation approach is also a two-phase process. In the first phase, we use all historical products to build a spline cost estimation model. There are a number of different spline functions available for practitioners to use for estimation purposes. However, the main concern is handling mixed numeric and categorical data. The second phase is the cost prediction phase in which the manufacturing cost of a new design is assessed. Figure 3.5 illustrates the proposed spline methodology for the first and second phases. The first phase shows the preliminary steps, including data collection and variable pre-processing, then the spline model building step. Once the desired spline model is established, the cost of a new product can be predicted using the underlying model.

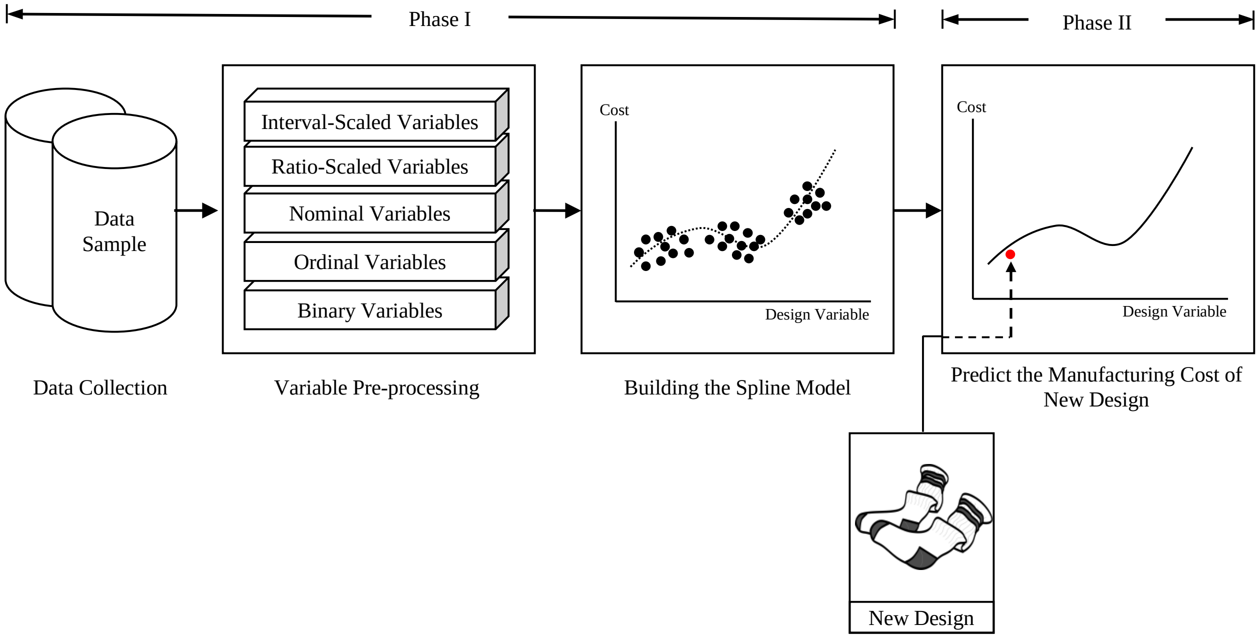


Figure 3.5: Building spline models and predicting the manufacturing cost of a new design

3.2.1 Choice of Spline Model

Racine et al. [98] considered the problem of estimating relationships using regression splines when categorical and continuous predictors are present. As we mentioned earlier in the review of clustering techniques, one alternative to accommodate mixed design attributes is splitting the categorical and continuous variables into two separate subsets. The same approach can be pursued for splines but there may be a consolidation as with the cluster ensemble techniques, as discussed in the literature review section. Instead of sample splitting, Racine et al. [98] devised a spline approach by combining the regression splines with kernel functions to handle existence of categorical variables.

In this research, we need to model complex relationships of categorical and numeric variables. A range of kernel regression methods have been proposed to model such relationships [99]. We used the same approach as described in Racine et al. [98] to accommodate the existence of categorical and numeric design attributes since the method demonstrates robust performance on both simulated and real world data without breaking the data into subsets of continuous only and categorical only variables.

Racine et al. [98] proposed tensor-product polynomial splines weighted by kernel functions method to estimate the unknown conditional mean in the location-scale model given in Equation 3.10. In the model, $g(\cdot)$ is an unknown function, $\sigma(\mathbf{X},\mathbf{Z})$ is the standard deviation function, ε represents noise, $\mathbf{X} = \left(X_1, \dots, X_q\right)^T$ is a q-dimensional vector of continuous predictors and $\mathbf{Z} = (Z_1, \dots, Z_r)^T$ is an r-dimensional vector of categorical predictors. Tensor products constitute the framework of interaction terms in regression spline models.

$$Y = g(\mathbf{X}, \mathbf{Z}) + \sigma(\mathbf{X}, \mathbf{Z})\varepsilon \tag{3.10}$$

The notation used in their study [98] is listed in the five items below without any modification:

1. $\mathbf{z} = (z_s)_{s=1}^{r}$ where z_s takes c_s (a finite positive constant) different values in $D_s \equiv \{0, 1, \dots, c_s - 1\}$, $s = 1, \dots, r$
2. $(Y_i, \mathbf{X}_i^T, \mathbf{Z}_i^T)_{i=1}^{n}$ is an i.i.d. copy of $(Y, \mathbf{X}^T, \mathbf{Z}^T)$ where *n* is the number of observations.
3. $\mathcal{B}(\mathbf{x})$ is the tensor-product polynomial spline basis. $\mathcal{B}(\mathbf{x}) = B_1(x_1) \otimes \dots \otimes B_q(x_q)$ where B_j is the B-spline basis matrix for the predictor *j* and $\otimes$ is the Kronecker product.
4. $L(\mathbf{Z}, \mathbf{z}, \lambda)$ is a product categorical kernel function where $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_r)^T$ is the vector of bandwidths for each of the categorical predictors.
5. $\beta(\mathbf{z})$ is a $\mathbf{K}_n \times 1$ vector, where $\mathbf{K}_n = \prod_{l=1}^{q} K_l$ and $K_l = N_l + m_l$, N_l is the number of interior knots, and m_l is the spline order assuming $1 \le l \le q$

The non-parametric function $g(\mathbf{x}, \mathbf{z})$ can be approximated by $\mathcal{B}(\mathbf{x})^T \beta(\mathbf{z})$, where $\beta(\mathbf{z})$ can be estimated by minimizing the weighted least squares criterion given in Equation 3.11. In the equation, $L(\mathbf{Z}_i, \mathbf{z}, \lambda)$ is a variant of Aitchison and Aitken's [100] univariate categorical kernel function and shown in Equation 3.12. For ordinal categorical variables, $l(Z_s, z_s, \lambda_s) = \lambda_s^{|Z_s - z_s|}$ is used to offset with the estimation bias [101] in Equation 3.11.

$$\hat{\beta}(\mathbf{z}) = \arg \min_{\beta \epsilon \mathbb{R}^{\mathbf{K}_n}} \sum_{i=1}^{n} \{Y_i - \mathcal{B}(\mathbf{X}_i)^T \beta(\mathbf{z})\}^2 \, L(\mathbf{Z}_i, \mathbf{z}, \lambda) \tag{3.11}$$

$$l(Z_s, z_s, \lambda_s) = \begin{cases} 1, & when\ Z_s = z_s \\ \lambda_s, & otherwise \end{cases}$$

$$L(\mathbf{Z}, \mathbf{z}, \lambda) = \prod_{s=1}^{r} l(Z_s, z_s, \lambda_s) = \prod_{s=1}^{r} \lambda_s^{1(Z_s \neq z_s)} \tag{3.12}$$

Let $\mathbf{B} = [\{\mathcal{B}(\mathbf{X_1}), \dots, \mathcal{B}(\mathbf{X_n})\}^T]_{n \times \mathbf{K}_n}$. Let $\mathcal{L}_z = diag\{L(\mathbf{Z_1}, \mathbf{z}, \lambda), \dots, L(\mathbf{Z_n}, \mathbf{z}, \lambda)\}$ be a diagonal matrix where $i \epsilon [1, n]$. If $\mathbf{Y} = (Y_1, \dots, Y_n)^T$, then $\hat{\beta}(\mathbf{z})$ defined in Equation 3.11 can be written as Equation 3.13.

$$\hat{\beta}(\mathbf{z}) = (n^{-1}\mathbf{B}^T \mathcal{L}_z \mathbf{B})^{-1}(n^{-1}\mathbf{B}^T \mathcal{L}_z \, \mathbf{Y}) \tag{3.13}$$

Racine et al. [98] implemented their work in R with a package called “crs”. The package is appealing for applied researchers because it uses a framework for nonparametric regression splines to address the existence of categorical and numeric variables as it was proposed in their study. We used the same package in R and applied it on our cost estimation problems. The package provides the convenience of automatic parameter tuning as well as the flexibility of manual parameter adjustment for each input.

3.2.2 Knot Placement and Smoothing Parameters

Choosing the number of knots with their locations has also been in the center of attention for researchers since it affects the smoothness and eventually the performance of spline models. There are two common approaches to determine the location of knots [102]: (1) Knots can be placed based on equally spaced quantiles where the number of observations in each segment is equal, (2) Knots can be places at equally spaced intervals. The “crs” package has the flexibility to use the desired option but most significantly, it chooses the knot placement strategy automatically based on whichever method provides better output. That is, the package does not require any extra effort from the user for the decision of knot placement.

There are many approaches available in the literature to find the appropriate number of knots for splines such as Bayesian curve-fitting with using a reversible-jump Markov chain Monte Carlo approach [103], devising various criteria to choose a penalty parameter [66], using a roughness penalty to adjust the locations of knots in penalized splines [104]. However, the common sense for parameter optimization in splines is adopting a cross-validation approach [66, 69, 104, 105, 106, 107, 108]. These parameters include selection of bandwidths (smoothing parameters), number of interior knots, and spline orders. Hall and Racine [108] utilized cross-validation to select the values of both polynomial degree and bandwidths for polynomial kernel regression. Racine et al. [98] adopted the same tradition and extended Hall and Racine’s work one step further by minimizing the cross-validation function, $CV(N, \lambda)$, given in Equation 3.14 for choosing the number of knots and smoothing parameters. The notation in the equation is consistent with their notation presented earlier. In addition to that, N is the vector of the number of interior knots, λ is the vector of smoothing parameters (bandwidths), and $\hat{\beta}_{-i}(Z_i)$ denotes the leave-one-out estimate of β. Parameters N and λ are not used in the cross-validation function directly but are used to calculate the current tensor product spline model coefficients during cross-validation iterations.

$$CV(N, \lambda) = n^{-1} \sum_{i=1}^{n} \left(Y_i - B_m(X_i)^T \hat{\beta}_{-i}(Z_i) \right)^2 \tag{3.14}$$

The package “crs” offers two search options to optimize the number of interior knots along with the value of bandwidths: (1) Exhaustive search, (2) Nonsmooth optimization by mesh adaptive direct search, NOMAD [109]. The number of interior knots for each continuous predictor is an integer value and the bandwidths for each categorical predictor is a value between $[0,1]$. That is, the optimization problem is a mixed-integer model. Clearly, using an enumeration based method such as exhaustive search might be computationally expensive for large datasets considering the number of categorical and numeric variables. In the “crs” package, the NOMAD approach was adopted to leverage recent advances in mixed-integer problems and also to avoid the computational burden of using a brute-force method like exhaustive search. As a summary, the “crs” package optimizes the combination of spline degrees in given intervals and knot placement strategy along with number of interior knots and the bandwidths with using a cross-validation function. There are three available cross-validation functions in the package, namely generalized cross-validation [110], expected Kullback-Liebler cross-validation [111], and least-squares cross-validation. We used the default approach, least-squares cross-validation, since it is computationally less time consuming with sufficient results. Note that in some cases, the optimal spline degree is found to be zero, and also the bandwidth is one. It means the corresponding variables are automatically removed from the model. To achieve better performance from the cost estimation spline model and to develop a parsimonious model, we do not include these removed (redundant or non-value adding) variables in the final model, because these are irrelevant. For details about other fine tuning parameters and further reading please refer to the R “crs” package [8].

3.3 Validation of the Methodologies

Without a validation tool, the prediction of manufacturing costs would be incomplete or meaningless because the ultimate purpose of this research is to predict the cost of a new and unique product (a future product). There are three commonly used non-parametric techniques for estimation of bias. In our research, the bias is the difference between actual cost and the estimated cost. These techniques are Quenouille's jackknife [112], Efron's bootstrapping [1] and cross-validation. Since the jackknife is closely related as a special case of cross-validation and bootstrapping, we present only the latter two methods with further detail.

Non-parametrical Bootstrapping

Chernick defines bootstrapping as a large group of resampling procedures from the original dataset [113]. Efron, who is known as the inventor of the simple non-parametric bootstrap, defines bootstrap as a computer-based method for estimating the standard error of a parameter that is estimated from an unknown distribution. Efron suggests putting equal probabilities on all observed values in an empirical distribution and drawing random bootstrap samples with replacement from that population [1]. It is possible to use Efron's non-parametric bootstrap procedure by drawing random samples to validate the performance of our manufacturing cost estimation methodologies.

Efron's bootstrapping is easy and simple to apply compared with other resampling methods [115]. It is also straightforward to derive standard errors and even confidence intervals for complex parameters of interest or complex probability distributions. Furthermore, an analytic expression of the estimator is not required [113]. There are three major factors may lead bootstrapping to fail to provide consistent results [2]: (1) Incomplete data: Missing observations may determine the accuracy of the parameter that we are estimating. (2) Dependent data: Since there is a need to adjust estimation of the variance of distribution for correlated data, actual variance and estimated variance with bootstrapping may differ because there is no apparent way to find a general joint distribution for dependent data. (3) Noisy data: If there is a parametric density function, it can be used to compare as a benchmark with results of non-parametric bootstrapping. However, if data is coming from an empirical distribution, there is no benchmark performance measure. In such case, noise cannot be easily discerned.

In resampling methods, there are two sources of errors, and often a combination of these two occurs [2]: (1) statistical error and (2) simulation error. Statistical error depends on the magnitude of the difference between the actual distribution and the fitted distribution. It can be reduced or even removed entirely by choosing a better estimator. However, simulation error depends totally on sampling from the fitted distribution due to using empirical estimates rather than exact properties [2]. Simulating the system for a nearly infinite time may eliminate the simulation error but this is practically impossible. There is an easier way to deal with simulation error by choosing an appropriate number of Monte Carlo replications. To find the necessary number of replications for a desired level of accuracy, a three-step method is suggested by Andrews and Buchinsky [116]. However, we do not need to consider these steps because using leave-one-out cross-validation would overcome these issues without extra effort. Although we consider the non-parametric bootstrap as a validation tool, cross-validation is more meaningful for our application because in a real world example, we predict the manufacturing costs of products one at a time. Since cross-validation replicates the whole process for the total number of observations, the cost of each product is predicted once rather than predicted several times due to random bootstrap samples with replacement. Furthermore, the datasets that are being analyzed may not be large enough to apply a non-parametrical resampling technique for validation purposes.

Thus, leave-one-out cross-validation is a better choice for such cases. Unfortunately we do not know which validation method is superior but we infer cross-validation adequately validates the predictive power of the models within a reasonable error level considering the reasons mentioned above.

Cross-Validation

Cross validation can be a useful statistical tool to evaluate model validation when fitting an estimation model to a set of data [2]. That is, cross-validation is a model testing technique, not a model construction method. In the cross-validation procedure, there are two easy steps to follow: (1) Divide the dataset into two equal parts randomly and use the first half for model fitting (training subset), and (2) Predict the second half according to the fitted model based on the first half (validation subset). After developments in computer technology, the cross-validation procedure has been improved by leaving out only one data point at a time then fitting the model for the rest of the data points and finally computing the bias for the point being left out.

Quenouille [112] introduced a non-parametric approach based on sequentially deleting observation points and recomputing the estimation of the parameter of interest [1]. This non-parametric estimate of bias method was later called jackknife and it is also known as leave-one-out cross-validation. After Quenoullie intoduced this powerful tool for statistical analysis, Tukey [117] suggested a formula for non-parametric estimate of variance that was derived from the recomputed statistics. For further details and mathematical expressions about the jackknife estimate of bias, please refer to Efron’s study [1]. It is worthwhile to mention that the jackknife estimate of variance (leave-one-out cross-validation) is biased and usually greater than the true variance [115].

Because of the reasons discussed in non-parametrical bootstrapping, we use the leave-one-out cross-validation tool in our study to validate the performance of the estimation models that are being constructed with or without cluster analysis. An observation is left out to test a cost estimation model that is built or trained with the remaining observations in the dataset. The observation being left out for every replication can be considered as an external test data point since it is not used in the cluster analysis nor model building phases. To clarify, the left out observation does not participate in any part of the cluster analysis nor for constructing estimation models but is used for testing the accuracy of the methodologies. First, we conduct a cluster analysis and then build cluster specific cost estimation models based on the entire data except the left out observation. Second, we find the cluster in which the left out observation falls. Finally, we test the corresponding cluster specific estimation model with the left out data point. With the same logic, first we build a spline model leaving one product out of the data sample. Second, we evaluate the spline model validity with the left out observation point.

3.4 Summary of the Suggested Methodologies

In summary, we consider three different ways to predict the manufacturing cost of products, namely clustering analysis, splines, and the conventional way. The conventional way is the benchmark comparison method that is based on a simple regression model built with the entire product stream. This comparison helps us to evaluate the degree of improvement when a clustering algorithm or a spline model is devised. As an extension to Figures 3.1, 3.2 and 3.5, we summarize and illustrate our methodologies with the chart given in Figure 3.6. All alternatives are named with labels on the right top corners and we call them manufacturing cost estimation 1 (MCE 1), manufacturing cost estimation 2 (MCE 2) and manufacturing cost estimation 3 (MCE 3), respectively.

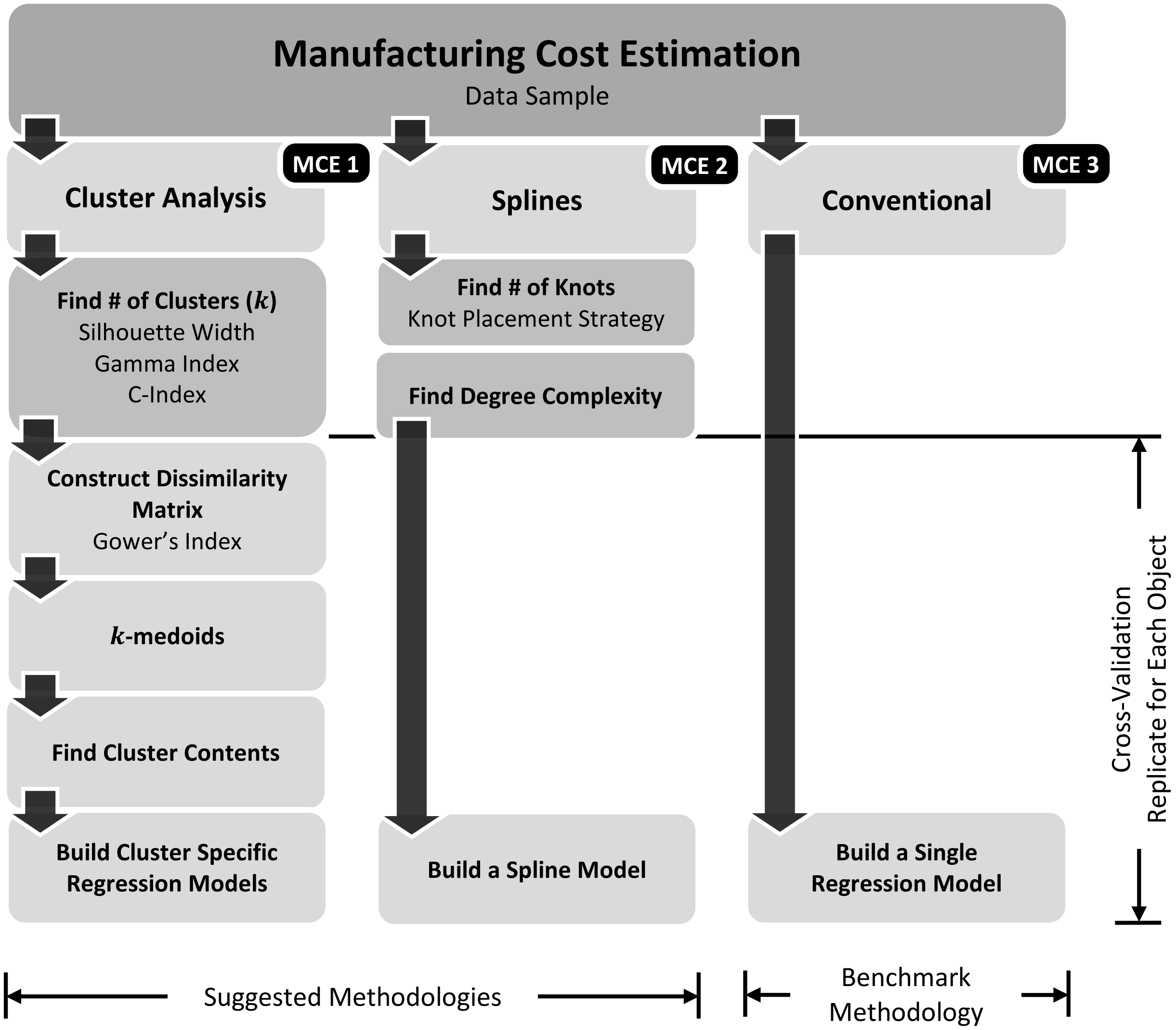


Figure 3.6: Summary of the proposed manufacturing cost estimation methodologies

Chapter 4

Real World Cost Estimation Applications

In this chapter, we apply our manufacturing cost estimation methodology on four datasets from three different industries. We discuss these real world problems from least to most complexity according to their sizes in terms of number of numeric and categorical variables and observations. The data was collected from socks, electromagnetic parts and plastic tools manufacturing factories in Ankara and Konya, Turkey. Mixed numeric and categorical design attributes, cost drivers or other variables comprise in these datasets. Due to the confidentiality agreements that were signed with these companies, we cannot state any brand names or product codes. Further information about these industries and datasets are given under each application problem.

4.1 Company and Dataset Descriptions

4.1.1 Socks Manufacturing Data

The first application problem dataset was collected from a socks manufacturer which produces copyrighted and licensed socks for some major brands in Europe and USA. They also produce the merchandise for the local market. Germany, Australia, Belgium, Denmark, France, Spain, Sweden, Italy, Lithuania and Poland are some of the primary countries in the European region to which their products are being imported. The company launched their production in

1996. Since that time the company has been operating with advanced technological machinery and a professional staff in a 100,000 square feet indoor factory and a 325,000 square feet outdoor area. Their range of products consists of sports, casual and formal/dress socks for women, men, children and infants. The pictures of sample socks from the most recent collection for women, men and children are shown in Figure 4.1 from left to right, respectively. The products are kept in separate warehouses for raw materials, spare parts, semi-products and finished socks. The manufacturing processes include pattern design, knitting, toe seam, washing-softening, pattern printing, final quality control and packaging. Steam, silicon and antibacterial washing are the types of washing-softening operations. In the printing department, the company is capable of applying lithographs, holograms, and also heat transfer, embroidery, rubber, acrylonitrile butadiene styrene (ABS), and caviar bead prints. Pairing and quality control, labeling, assortment and packaging are usually the final operations in the facility.

Figure 4.1: Some sample socks from the most recent collection of the company

The dataset that we collected from the company’s database contains information for 76 products of women and men’s socks. There are nine variables associated with these products, and eight of these variables are qualitative (categorical), namely raw material, pattern, elasticity, woven tag, heel style, leg style, fabric type and gender. The only quantitative variable measured on a continuous scale in this dataset is the actual cost which is recorded in Turkish Lira (TL) money units. Table 4.1 is the summary of the dataset and associated attributes. The columns of the table are variable name, data type, variable type, and categories (for categorical data) or range (for numeric data) from left to right, respectively. For nominal variables, the order of categories is not important since there is no logical transition between categories. However, for ordinal variables, categories represent the order of the labels from the lowest to the highest category in its ordinal scale. For instance, elasticity is an ordinal variable that can take a value from “None” to “Double”. In this case, “None” represents the lowest elasticity level and “Double” represents the highest possible elasticity level of the sock material.

The R script we developed creates the dissimilarity matrix, cluster contents, regression models and splines models based on an input vector of variable types. Interval-scaled continuous, ratio-scaled continuous, nominal, ordinal, symmetric binary and asymmetric binary predictors are coded as 1, 2, 3, 4, 5 and 6, respectively. The dependent variable, actual cost, is coded as 0. For instance, the input vector for the socks production data is [3 5 4 5 5 4 5 5 0]. This vector represents the type of each predictor, raw material (nominal), pattern (symmetric binary), elasticity (ordinal), woven tag (symmetric binary), heel (symmetric binary), leg style (ordinal), fabric type (symmetric binary), gender (symmetric binary), and actual cost (dependent variable), respectively.

Table 4.1: Summary of the socks manufacturing dataset

Variable Name	Data Type	Variable Type	Categories/Range
Raw Material	Categorical	Nominal	Bamboo Lycra Cotton Lycra Cotton Coolmax Lycra Organic Cotton Lycra Modal Lycra
Pattern	Categorical	Symmetric Binary	Yes No
Elasticity	Categorical	Ordinal	None Plain Derby Curly Double
Woven Tag	Categorical	Symmetric Binary	None Label
Heel	Categorical	Symmetric Binary	None Plain
Leg Style	Categorical	Ordinal	None Short Medium Long
Fabric Type	Categorical	Symmetric Binary	Plain Towel
Gender	Categorical	Symmetric Binary	Women Men
Actual Cost	Numeric	Interval Scale	$[0, \infty)$

4.1.2 Electrical Grounding Parts Data – Tubular Cable Lugs

The second application problem dataset was collected from an electromagnetic parts manufacturer which produces lightening protection elements, grounding materials, metal masts for various purposes and cabins for specific purposes. The company manufactures these products for both local and foreign markets. They also provide project services including installation of lightening conductor and grounding systems, and also maintenance of this equipment. Furthermore, the company evaluates existing systems on site for protection performance and international standards fulfillment. Since its establishment in 1953, the company has been following world-wide standards with the most recent techniques and high tech equipment. Steel, copper, stainless steel, aluminum, brass, bronze, cast iron, plastic and concrete are the primary raw materials used to manufacture these static grounding systems. In the facility, they are able to coat these materials with electro galvanization, hot deep galvanization, electro copper coating, electro tin coating, electro Chromium-Nickel (Cr-Ni) coating, black insulation and green-yellow insulation.

The dataset that we collected from the company's database contains information for various tubular cable lugs of 68 observations. The pictures of some of these cable lugs are shown in Figure 4.2. There are 12 variables associated with these 68 observations, namely lug type, cross-section, hole diameter, number of holes, gap between holes, material weight, process time, inner diameter, outer diameter, coating type, coating time and the actual cost. Ten of these variables are quantitative attributes and nine of them are recorded on continuous scales. These nine continuous valued variables are cross-section, hole diameter, gap between holes, material weight, process time, inner diameter, outer diameter, coating time and the actual cost, and their units are recorded in mm^2, mm, mm, kg, mm, mm, minutes and TL, respectively. The remaining one quantitative variable takes integer values. The label of the strictly integer valued quantitative variable is the number of holes, and it does not have any measurement units. There are at most two holes on a lug and the minimum number of holes is zero. DIN, forend, long, standard and forend standard are the categories of the variable lug type. In Figure 4.2, the groups of four pieces, from left to right, represent various sizes of DIN, forend standard and long type of tubular lugs. Coating type has two categories, tin and none, where none indicates that there is no coating on the piece. Since only one cable lug has been labeled with none in the dataset, there are not sufficient observations to reveal the true contribution of the coating type variable on the actual cost values. Table 4.2 is the summary of the dataset and its associated attributes. The input vector for the tubular cable lugs production data is [3 1 1 1 1 1 1 1 1 3 1 0].

Figure 4.2: Sample tubular cable lugs for electrical grounding

Table 4.2: Summary of the tubular cable lugs manufacturing dataset

Variable Name	**Data Type**	**Variable Type**	**Categories/Range**
Lug Type	Categorical	Nominal	DIN Forend Forend Standard Long Standard
Cross-section	Numeric	Interval Scale	$[0, \infty)$
Hole Diameter	Numeric	Interval Scale	$[0, \infty)$
Number of Holes	Numeric	Interval Scale	0, 1, 2, …
Gap b/w Holes	Numeric	Interval Scale	$[0, \infty)$
Material Weight	Numeric	Interval Scale	$[0, \infty)$
Process Time	Numeric	Interval Scale	$[0, \infty)$
Inner Diameter	Numeric	Interval Scale	$[0, \infty)$
Outer Diameter	Numeric	Interval Scale	$[0, \infty)$
Coating	Categorical	Nominal	None Tin
Coating Time	Numeric	Interval Scale	$[0, \infty)$
Actual Cost	Numeric	Interval Scale	$[0, \infty)$

4.1.3 Lightening Protection Parts Data – Air Rods

The third application problem dataset was collected from the same electromagnetic parts manufacturer as in the second problem and includes information about 197 air rods for lightening protection purposes. The pictures of some of these protective air rods are shown in Figure 4.3. In this figure from left to right, respectively, the types of air rods are long, tube, multiple points and Eratec* types. In the dataset, there are 10 variables associated with these 197 observations. Five of these variables take continuous numeric values and the remaining five are categorical labels. The numeric variables are rod diameter, rod length, screw size, material weight and the actual cost. The values of these variables are measured with these units, respectively: mm, mm, mm, kg and TLs. The screw size takes a value of zero when there is no screw used, and the actual minimum screw size is 8.5 mm. The categorical variables are screw type, main material, coating, raw material and screw nut coating. In Table 4.3 the summary of the dataset and its associated attributes are shown. The input vector for the air rods production data is [1 1 1 3 1 3 3 3 3 0].

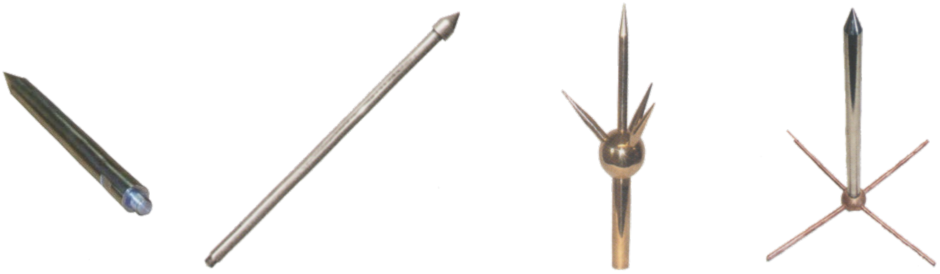


Figure 4.3: Sample air rods for lightening protection

* "Eratec" is a type of air rod made from four pieces of thermo-welded conductive material.

Table 4.3: Summary of the air rods manufacturing dataset

Variable Name	**Data Type**	**Variable Type**	**Categories/Range**
Rod Diameter	Numeric	Interval Scale	$[16, \infty)$
Rod Length	Numeric	Interval Scale	$[150, 6000]$
Screw Size	Numeric	Interval Scale	$[8.5, 16]$
Screw Type	Categorical	Nominal	None Interior Screw Exterior Screw
Material Weight	Numeric	Interval Scale	$[0, \infty)$
Main Material	Categorical	Nominal	Aluminum Copper Iron-Steel Bronze Gray Cast Iron Stainless Steel Brass Plastic
Coating	Categorical	Nominal	No Coating Electro-Galvanizing Hot Dip Galvanizing Electrodeposited Copper Electrodeposited Tin Electrodeposited Cr-Ni Black Insulation Yellow Green Insulation
Raw Material	Categorical	Nominal	Aluminum Rod Ø16 Aluminum Rod Ø20 Brass Rod Ø16 Brass Rod Ø20 Copper Rod 16 x 3000 Copper Rod 16 x 3500 Copper Rod 20 x 3000 Copper Rod 20 x 6000 Stainless Rod Ø16 Stainless Rod Ø20 Transmission Ø16 Transmission Ø20
Screw Nut Coating	Categorical	Nominal	No Screw Nut Non-Coated Galvanized Stainless Brass
Actual Cost	Numeric	Interval Scale	$[0, \infty)$

4.1.4 Plastic Products Data

The last dataset was taken from a plastic parts manufacturer which produces kitchenware, food and non-food storage containers, and salad, pastry, bathroom, and hanger accessorizes for foreign markets across four continents such as Argentina, Turkmenistan, Russia, South Africa, Spain and Kuwait. The company is also in close collaboration with markets, malls, chain purchasers and wholesalers of the local Turkish market. They have been in the plastic industry since 1989 with an indoor facility of 55,000 square feet. They hold several international standard certificates for production quality, safety and environmental sustainability.

In this dataset, there are many products with completely different physical shapes. However, we may group them according to their raw material types, manufacturing processes/operations or some other factors. Some of these products are spoons, vegetable peelers, containers and paper towel racks, as can be seen in Figure 4.4. The dataset covers 51 variables for 130 plastic products. Actually, there are 10 main categories of variables, raw material, press, vacuum, paint, sticker, wall plug, labor complexity, and actual cost. There are 13 variables under the raw material category where 12 of them are binary and one is numeric. These 12 variables represent the type of raw material such as anti-shock, acrylonitrile butadiene styrene (ABS), poly carbon and carbon fiber. If a material is used in the main material mixture for a particular product, the value of the underlying material variable takes one, otherwise zero. The only variable measured on a continuous scale is mixture weight under the raw material subject. It is recorded in grams. The second variable category is press which actually stands for the pressing process. There are three machine groups in the company that can perform press operations. Tederic, TSP and Haitian are the names of these machine groups. There are eleven, eight and four different machines under the Tederic, TSP and Haitian groups, respectively. Every machine corresponds to a variable in the dataset. There can be multiple alternative machines to perform the same operation; however, if a machine is used for any step of production for a particular product, its variable takes a numeric value representing the machining time. If the underlying machine is not used for that product, the value of that machine's variable takes a value of zero. The next variable category is for the vacuuming process. There are two variables under the vacuum topic: (1) Poly vinyl chloride (PVC) type for the vacuuming process, and (2) the number of vacuums required. The PVC type is a categorical variable and the number of vacuums takes discrete numeric values. Under the boxing category, there are seven variables. Six of these variables are numeric variables and one of them is a categorical variable. These variables are number of items in a box, net weight, gross weight, length, width, depth of the box and the type of the boxing material. Each remaining category corresponds to a single variable. Package, paint material weight, sticker, wall plug, labor complexity and actual cost are, respectively, binary, numeric, binary, binary, ordinal and numeric variables. The unit of the paint material weight is grams. Also, the actual cost is recorded in TLs. Furthermore, the labor complexity is tracked according to the complexity of the manufacturing and assembly operations and ranked from 1 (easiest) to 3 (most complex) sequentially. In Table 4.4, the summary of the dataset and its associated attributes are shown.

We used acronyms to represent each of these four datasets. We named the application problems dataset 1 (DS 1), dataset 2 (DS 2), dataset 3 (DS 3) and dataset 4 (DS 4) for the socks manufacturing, the tubular cable lugs, the air rods, and the plastic products problem sets, respectively.

Table 4.4: Summary of the plastic products manufacturing dataset

	Variable Name	Data Type	Variable Type	Categories/Range
Raw Material	Cristal	Categorical	Symmetric Binary	Yes, No
	Anti-Shock	Categorical	Symmetric Binary	Yes, No
	PP	Categorical	Symmetric Binary	Yes, No
	ABS	Categorical	Symmetric Binary	Yes, No
	Poly Carbon	Categorical	Symmetric Binary	Yes, No
	NAT ABS	Categorical	Symmetric Binary	Yes, No
	Randum	Categorical	Symmetric Binary	Yes, No
	ESM	Categorical	Symmetric Binary	Yes, No
	i20	Categorical	Symmetric Binary	Yes, No
	Carbon Fiber	Categorical	Symmetric Binary	Yes, No
	Stainless Steel	Categorical	Symmetric Binary	Yes, No
	PVC	Categorical	Symmetric Binary	Yes, No
	Weight	Numeric	Interval Scale	$[0, \infty)$
Pressing Process	Tedeceric 100_1	Numeric	Interval Scale	$[0, \infty)$
	Tedeceric 100_2	Numeric	Interval Scale	$[0, \infty)$
	Tedeceric 110	Numeric	Interval Scale	$[0, \infty)$
	Tedeceric 120	Numeric	Interval Scale	$[0, \infty)$
	Tedeceric 140	Numeric	Interval Scale	$[0, \infty)$
	Tedeceric 188_1	Numeric	Interval Scale	$[0, \infty)$
	Tedeceric 188_2	Numeric	Interval Scale	$[0, \infty)$
	Tedeceric 188_3	Numeric	Interval Scale	$[0, \infty)$
	Tedeceric 230_1	Numeric	Interval Scale	$[0, \infty)$
	Tedeceric 230_2	Numeric	Interval Scale	$[0, \infty)$
	Tedeceric 280	Numeric	Interval Scale	$[0, \infty)$
	TSP 120_1	Numeric	Interval Scale	$[0, \infty)$
	TSP 120_2	Numeric	Interval Scale	$[0, \infty)$
	TSP 150_1	Numeric	Interval Scale	$[0, \infty)$
	TSP 150_2	Numeric	Interval Scale	$[0, \infty)$
	TSP 220	Numeric	Interval Scale	$[0, \infty)$
	TSP 250	Numeric	Interval Scale	$[0, \infty)$
	TSP 360_1	Numeric	Interval Scale	$[0, \infty)$
	TSP 360_2	Numeric	Interval Scale	$[0, \infty)$
	Haitian 110	Numeric	Interval Scale	$[0, \infty)$
	Haitian 150_1	Numeric	Interval Scale	$[0, \infty)$
	Haitian 150_2	Numeric	Interval Scale	$[0, \infty)$
	Haitian 250	Numeric	Interval Scale	$[0, \infty)$
Vacuuming	PVC Type	Categorical	Ordinal	0 15 20
	# of Vacuums	Numeric	Interval Scale	$[0, \infty)$
Boxing	# in box	Numeric	Interval Scale	1, 2, 3, …
	Net Weight	Numeric	Interval Scale	$[0, \infty)$
	Gross Weight	Numeric	Interval Scale	$[0, \infty)$
	Length	Numeric	Interval Scale	$[0, \infty)$
	Width	Numeric	Interval Scale	$[0, \infty)$
	Depth	Numeric	Interval Scale	$[0, \infty)$
	Type	Categorical	Nominal	Blister Polybag Display Box Bound Card PVC Shrink Sticker Box
	Package	Categorical	Symmetric Binary	Yes, No
	Paint Weight	Numeric	Interval Scale	$[0, \infty)$
	Sticker	Categorical	Symmetric Binary	Yes, No
	Wall Plug	Categorical	Symmetric Binary	Yes, No
	Labor Complexity	Categorical	Ordinal	1 2 3
	Actual Cost	Numeric	Interval Scale	$[0, \infty)$

Figure 4.4: Sample plastic products including kitchen tools, food containers and towel holders

4.2 Cluster Analysis and the Number of Clusters

As discussed earlier in Chapter 3, we used Kaufmann and Rousseeuw's k-medoids algorithm as it was implemented in "PAM" [9]. We coded the manufacturing cost estimation models in R, and for the cluster analysis, the package "cluster" with its contingent packages were utilized. In the cluster analysis and cost estimation phases, the actual cost of a product is the dependent variable and all other variables are predictive ones. The first target is to determine the appropriate number of clusters. The C-index, the Gamma and the average silhouette width graphs are the primary tools to choose the appropriate number of clusters. We plotted the values of the underlying indices from 2 to 20 clusters. As expected the value of Gamma and the average silhouette width increase as the number of clusters increase. The value of the C-index decreases as the number of clusters increases which is consistent with the pattern of the other two indices. The graphs of these three indices with respect to the number of clusters are given in Figure 4.5, 4.6, 4.7 and 4.8 for the application problems DS 1, DS 2, DS 3 and DS 4, respectively.

Remember that our policy is to seek a consensus among these three graphs. For DS 1, a settlement point of the indices is seven clusters as shown in Figure 4.5 with the black points where a local trough is observed right before a dramatic jump in the C-index. Furthermore, at the point of seven clusters, local peaks can be observed one step before the sudden drops in Gamma and silhouette width trends. For DS 2, the silhouette width does not have any value higher than 0.5. However, a local peak is observed at 11 clusters. When we compare the performance of the other two indices with the silhouette width, 11 is a reasonable value as the appropriate number of clusters. Furthermore, after 11 clusters, the cluster contents become unbalanced where too many observations accumulated in some groups. For DS 3, we picked the point where the silhouette width goes above 0.5 for the first time because a value above 0.5 indicates robust clustering structure. After 14 clusters, the value of silhouette width stagnates right below the 0.5 line. If we check the consistency of silhouette width with the other two statistics, we can see that 14 clusters is a good choice. For DS 4, the silhouette width never moves higher than 0.5, but there is a sudden drop in the C-index value at 10 clusters. When the Gamma index is considered, the value increases slowly to the point at 10 clusters and after that it becomes stable. Combining the information derived from these statistics, we can conclude that 10 is a proper value. There are several other possible points that these indices suggest, but 7, 11, 14 and 10 are the most conspicuous points for DS 1, DS 2, DS 3 and DS 4, respectively, when we monitor these graphs from left to right simultaneously. Table 4.5 shows the number of observations allocated to each cluster using k-medoids based on the best number of clusters for each application dataset. When we analyze the individual observations in each cluster, it is easy to see that categorical variables play an important role in forming the cluster contents. Also, we plotted the minimum (min), maximum (max) and average (mean) actual cost values of products allocated in each cluster in Figures 4.9, 4.10, 4.11, and 4.12 for DS 1, DS 2, DS 3 and DS 4, respectively. These graphs are provided to illustrate how actual cost values strongly overlap among clusters for the most cases. It is interesting to observe that the similarity of products does not necessarily follow the same similarity pattern of the actual cost values. Since multiple cost drivers contribute to product cost, there is no single factor determining the cluster contents. The interaction of a collection of cost drivers are more influential than a single variable for each product.

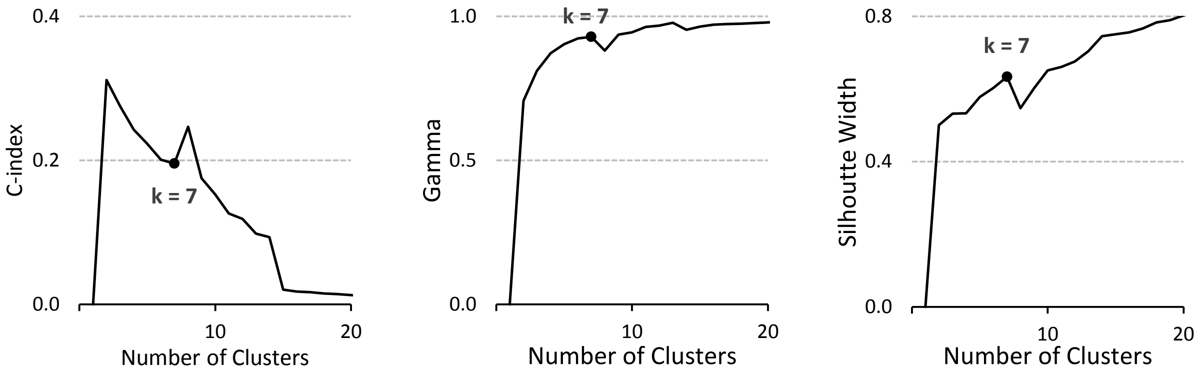


Figure 4.5: C-index, Gamma and silhouette width plots for DS 1 of 76 products

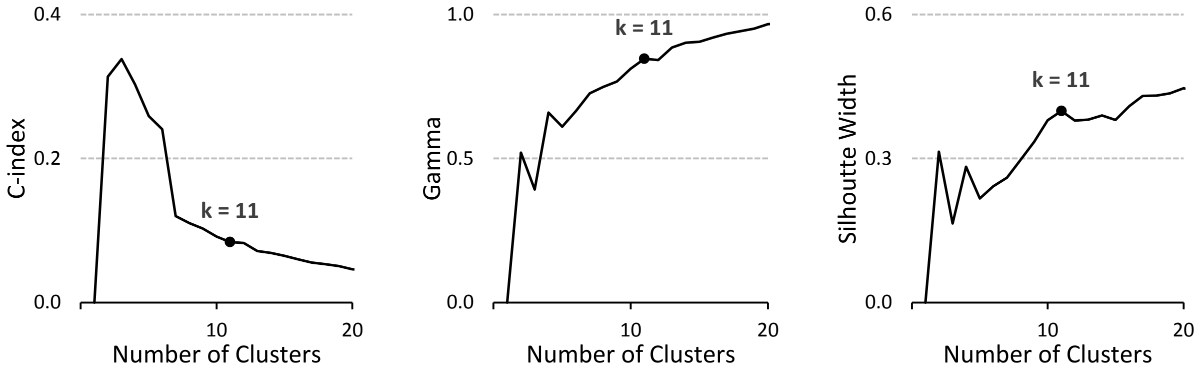


Figure 4.6: C-index, Gamma and silhouette width plots for DS 2 of 68 products

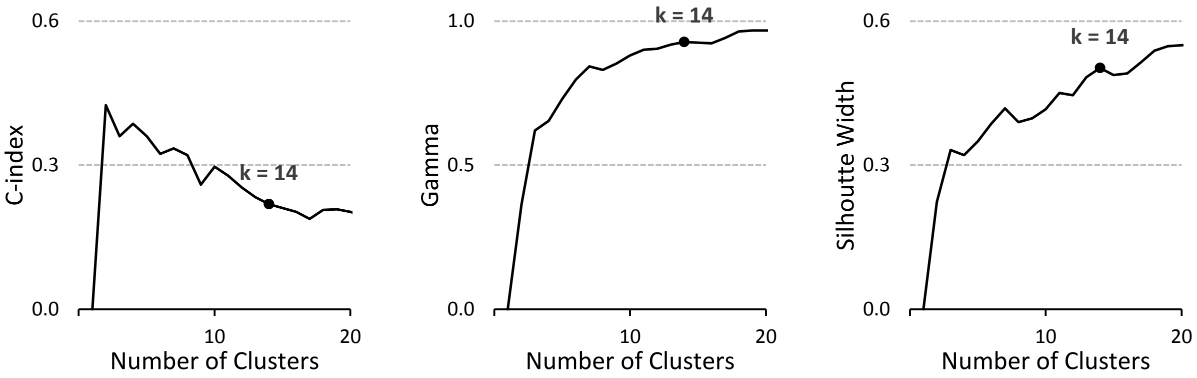


Figure 4.7: C-index, Gamma and silhouette width plots for DS 3 of 197 products

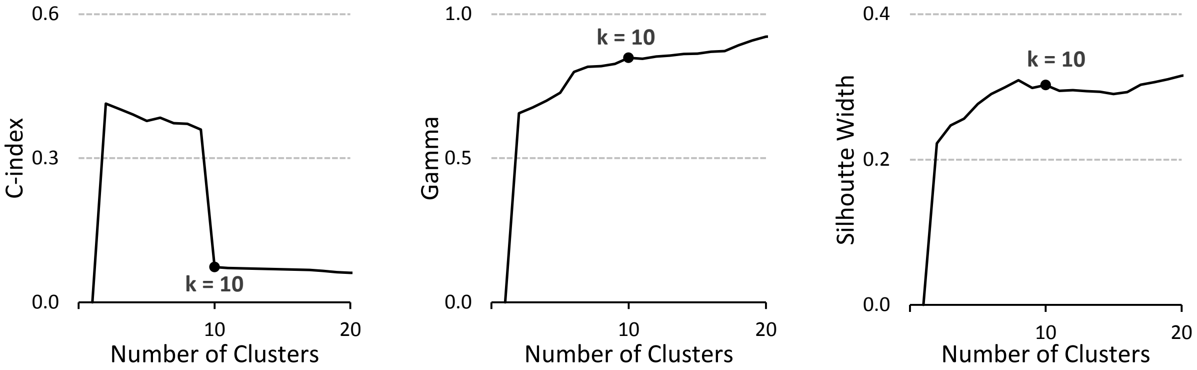


Figure 4.8: C-index, Gamma and silhouette width plots for DS 4 of 130 products

Table 4.5: The number of observations in each cluster for the application problems

Cluster No	DS 1	DS 2	DS 3	DS 4
1	37	10	26	24
2	11	9	23	20
3	11	8	23	17
4	6	8	17	16
5	5	7	16	15
6	3	5	16	10
7	3	5	14	8
8		5	13	8
9		4	9	7
10		4	9	5
11		3	8	
12			8	
13			8	
14			7	

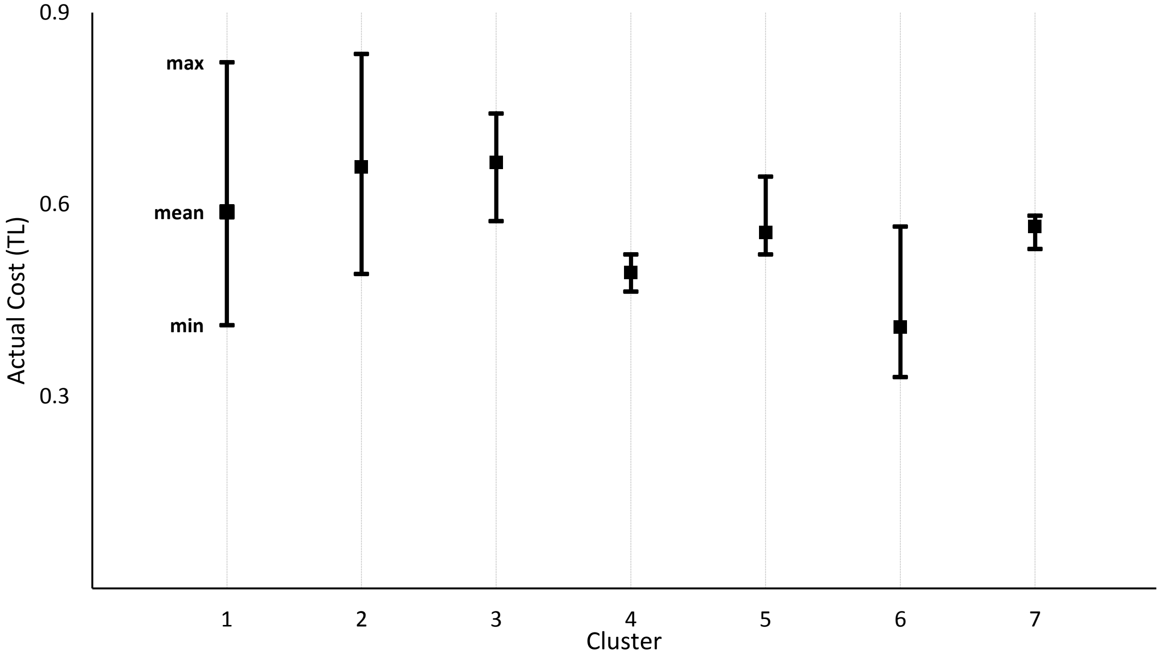


Figure 4.9: The minimum (min), maximum (max) and average (mean) actual cost values of objects allocated in each cluster for DS 1

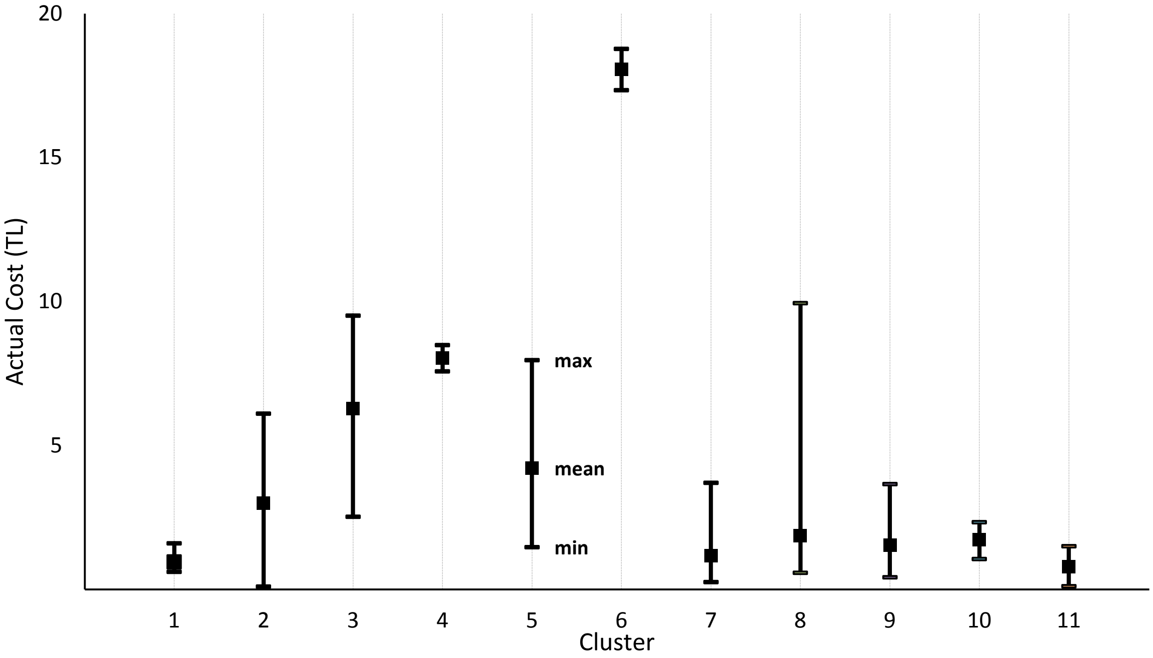


Figure 4.10: The minimum (min), maximum (max) and average (mean) actual cost values of objects allocated in each cluster for DS 2

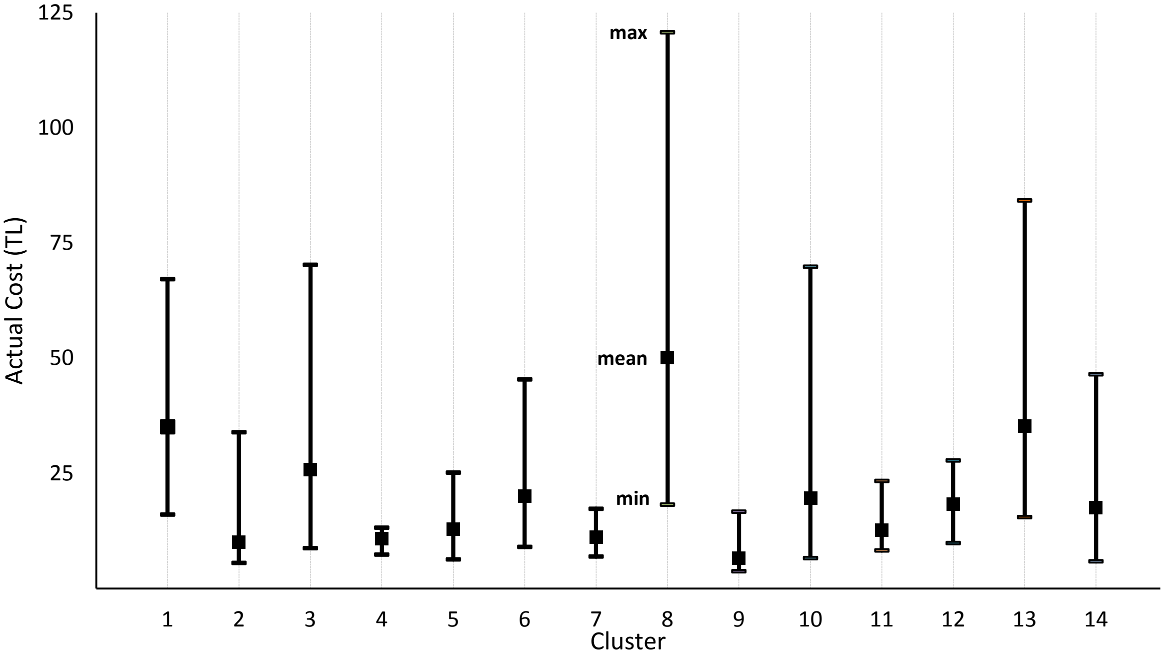


Figure 4.11: The minimum (min), maximum (max) and average (mean) actual cost values of objects allocated in each cluster for DS 3

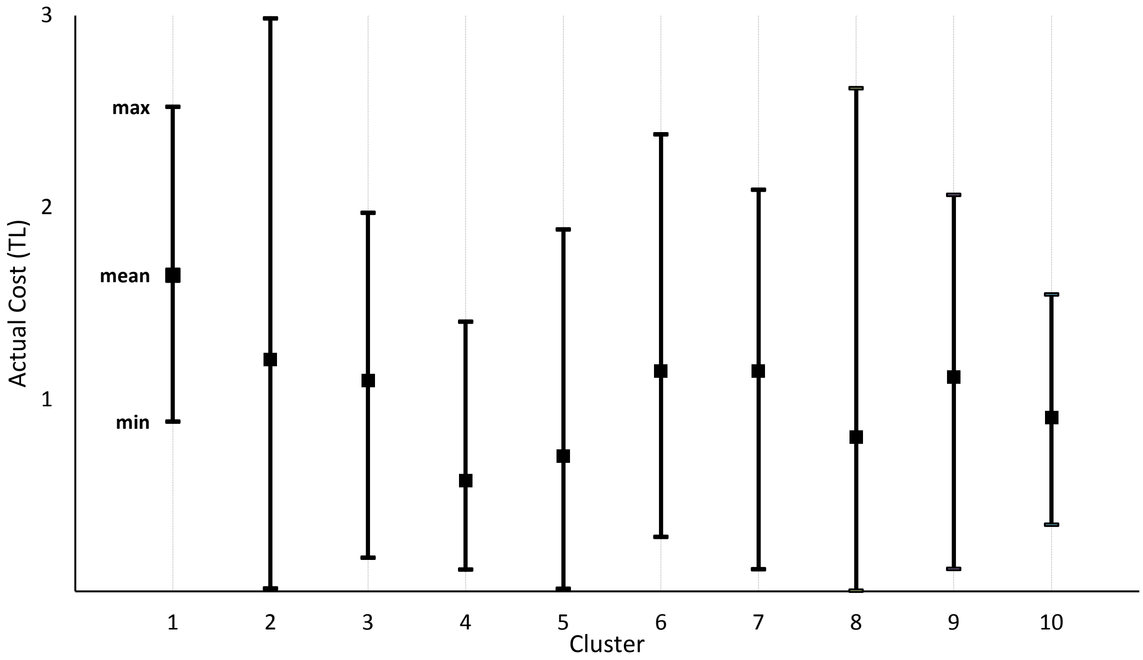


Figure 4.12: The minimum (min), maximum (max) and average (mean) actual cost values of objects allocated in each cluster for DS 4

4.3 Spline Model Parameters

As discussed in Chapter 3, we used the R package called “crs” to build spline models in the presence of categorical and numeric design attributes. The “crs” function input parameters used in our computer code are provided in Table 4.6.

“degree.max” and “degree.min” indicate the maximum and the minimum polynomial degree of each continuous predictor, respectively. We used the default value, 10, as the maximum polynomial degree because increasing the degree of the piece-wise functions will obviously increase the prediction accuracy for the current data but will likely result in overfit models. According to our observations, none of the continuous predictors came out to be a higher degree than cubic splines considering the cross-validated set of parameters. When the polynomial degree of a predictor is zero, the variable is automatically removed from the spline model due to its irrelevance.

“segments.max” and “segments.min” represent the maximum and minimum number of segments for each of the continuous predictors. The number of segments corresponds to the number of pieces in the spline model and is closely related to the number of knots where the number of knots minus one is the number of segments. We used the default value, 10, because the performance of spline models significantly declined beyond 5 segments when we ran the script for all of the datasets on hand.

“cv” shows what kind of search is used to optimize the mathematical spline parameters such as number of knots or categorical predictor bandwidths. We used “NOMAD” because it is the suggested optimization module for the “crs” package due to the reasons mentioned in the previous chapter. “cv.func” indicates which cross-validation method should be used to select predictor bandwidths. We used the least squares approach, “cv.ls”, by default because according to the preliminary runs, the function provided consistent result. Even though computational efficiency is not an issue for the datasets we worked on, for generalization of the spline cost estimation method, using the least squares approach is a promising choice since it requires modest computational effort.

“complexity” indicates whether the model complexity is determined by the polynomial degree of the spline, “degree”, or by the number of knots, “knots”, or both, “degree-knots”. In each optimization iteration, a different combination of spline degree and number of knots for each variable is assessed along with the categorical predictor bandwidths to minimize the underlying cross-validation function. If “degree” is chosen for the model complexity, the “crs” function requires a numerical value to be entered for the number of knots for each continuous variable because the cross-validation does not consider the number of knots in the optimization process. Similarly, if “knots” is chosen for the model complexity, a vector of integer values should be entered. This vector specifies the spline degree for each continuous predictor. We used “degree-knots” as the model complexity since ultimate goal is establishing the most suitable spline models to achieve an accurate cost estimation model. Also, we assumed that there is no further information available for the datasets. In manufacturing cost estimation applications in general, the degree complexity and the number of knots are not known a priori.

“basis” indicates whether interaction terms in the spline model should be included. That is, to include interaction terms “tensor” should be entered as the basis and “additive” otherwise. We ran the spline model script with both “additive” and “tensor” inputs initially. The results show that using tensor products (that is, including interaction terms) provided slightly more accurate results. Using tensor products is supposed to be computationally costly but even for the two big application problems (DS 3 and DS 4) in terms of number of variables and number of observations, the tensor product spline model generated results in a reasonable time. Thus, computational efficiency is not a concern for the size of data in the application problems, or indeed for most cost estimation tasks.

For the final input parameter, “knots”, we let the cross-validation decide the best knot placement strategy. Knot placement is an art by itself and there is considerable research done in this area as we discussed in the previous chapter. “quantiles” specifies knots placed equally spaced quantiles where an equal number of observations lies in each segments. On the other hand, “uniform” specifies knots placed at equally spaced intervals in the continuous range of a variable. We used “auto” as the “knots” input parameter value. The “crs” model output consistently demonstrated that the automatic knot placement module locates the knots for each continuous predictor based on a “quantiles” placement strategy because it provided better results than the “uniform” strategy according to the minimization of the cross-validation function (Equation 3.14). However, to keep the manufacturing cost estimation methodology as general as possible, using the “auto” function seems to be the best alternative rather than limiting it to either “quantiles” or “uniform”.

Table 4.6: The “crs” function input parameters used to build the spline models

Parameter	**Value**
degree.max	10
degree.min	0
segments.max	10
segments.min	1
cv	NOMAD
cv.func	cv.ls
complexity	degree-knots
basis	tensor
knots	auto

4.4 Results and Discussion

As we discussed in Chapter 3, we devised a leave-one-out cross-validation tool to leverage the data for both validation and model building. That is, an observation is left out to predict its manufacturing cost based on MCE 1, MCE 2 and MCE 3 that are built from the remaining observations in a dataset. This process is replicated for the number of observations in a dataset to complete one full turn for the entire data. Without a validation tool, our methodology would not have credibility to be used in a real life business environment. This validation module is fully integrated in the same R script.

For each product, the performance criteria we considered are absolute relative error (ARE) and squared error (SE). Due to the size of the datasets, it is not practical to provide these two error values for each data point. Instead, providing the error graphs for each dataset and generating a summary of these key factors is more efficient. In this research, we refer to the error as the difference between actual cost and the predicted cost. ARE and SE are calculated according to Equations 4.1 and 4.3, respectively. ARE and SE are computed for each observation where i represents the observation number and n is the total number of observations. Mean absolute relative error (MARE) is the average absolute percentage deviation from the actual cost over all observations and given in Equation 4.2. Mean squared error (MSE) is the average squared deviation from the actual cost values for each observation point and given in Equation 4.4. Root mean squared error (RMSE) is the square root of MSE and its mathematical expression is provided in Equation 4.5.

$$ARE_i = \frac{(Actual\ Cost)_i - (Estimated\ Cost)_i}{(Actual\ Cost)_i} \tag{4.1}$$

$$MARE = \frac{1}{n}\sum_{i=1}^{n} ARE_i \tag{4.2}$$

$$SE_i = [(Actual\ Cost)_i - (Estimated\ Cost)_i]^2 \tag{4.3}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n} SE_i \tag{4.4}$$

$$RMSE = \sqrt{MSE} \tag{4.5}$$

In Table 4.7, we present the performance metrics of each cost estimation approach, termed MCE 1, MCE 2, and MCE 3 for the application problems, DS 1, DS 2, DS 3, and DS 4. These metrics include MARE, RMSE, the minimum ARE (Min ARE), and the maximum ARE (Max ARE) over the validated predictions for each product. Notice that MCE 2 does not have error defined for DS 1 in the table since this dataset does not contain any continuous predictors to form a spline basis. Thus, MCE 2 is not applicable (N/A) for DS 1. The minimum values of MARE, RMSE and Max ARE are depicted in bold in Table 4.7 for each dataset. According to MARE values, the most accurate cost estimation approach is MCE 1 based on overall performance. However, MCE 2 generates slightly more accurate predictions for DS 3 compared to MCE 2. Clearly, MCE 3 was outperformed by MCE 1 and MCE 2. It is certainly consistent with our suggestion that over a diverse product family, building a single regression model is doubtful. To capture complex relationships between cost drivers and the manufacturing cost requires a more complex methodology than a simple regression. It is really hard to decide which cost estimation method is superior between MCE 1 and MCE 2 because first, MCE 2 is not applicable to the first dataset (DS 1) since there are no continuous predictors to build a spline basis. The second and the most important reason is MCE 1 was bettered by MCE 2 for DS 2 in terms of MARE and RMSE values. Also, when we consider Max ARE values, MCE 2 did better than MCE 1. The Max ARE gap for DS 4 is more than 100% on MCE 1 and MCE 2. It is worthwhile to mention that MCE 1 and MCE 2 were able to predict the manufacturing cost of products with very good accuracy. Figure 4.13 shows the performance of the cost estimation methods over the four data application problems in terms of the MARE values given in Table 4.7.

Table 4.7: Performance metrics of each cost estimation model for the application problems

MARE

	MCE 1	**MCE 2**	**MCE 3**
DS 1	**6.25%**	N/A	8.54%
DS 2	**4.98%**	38.70%	49.82%
DS 3	5.81%	**4.08%**	15.42%
DS 4	**12.39%**	17.55%	33.83%

RMSE

	MCE 1	**MCE 2**	**MCE 3**
DS 1	**5.75%**	N/A	6.51%
DS 2	**8.86%**	104.29%	140.26%
DS 3	355.72%	**138.07%**	615.92%
DS 4	**17.71%**	23.95%	34.20%

Min ARE

	MCE 1	**MCE 2**	**MCE 3**
DS 1	0.00%	N/A	0.00%
DS 2	0.00%	0.00%	0.00%
DS 3	0.00%	0.00%	0.00%
DS 4	0.00%	0.00%	0.00%

Max ARE

	MCE 1	**MCE 2**	**MCE 3**
DS 1	**49.12%**	N/A	49.82%
DS 2	**46.67%**	162.01%	429.52%
DS 3	56.04%	**26.23%**	64.36%
DS 4	203.54%	**94.73%**	233.79%

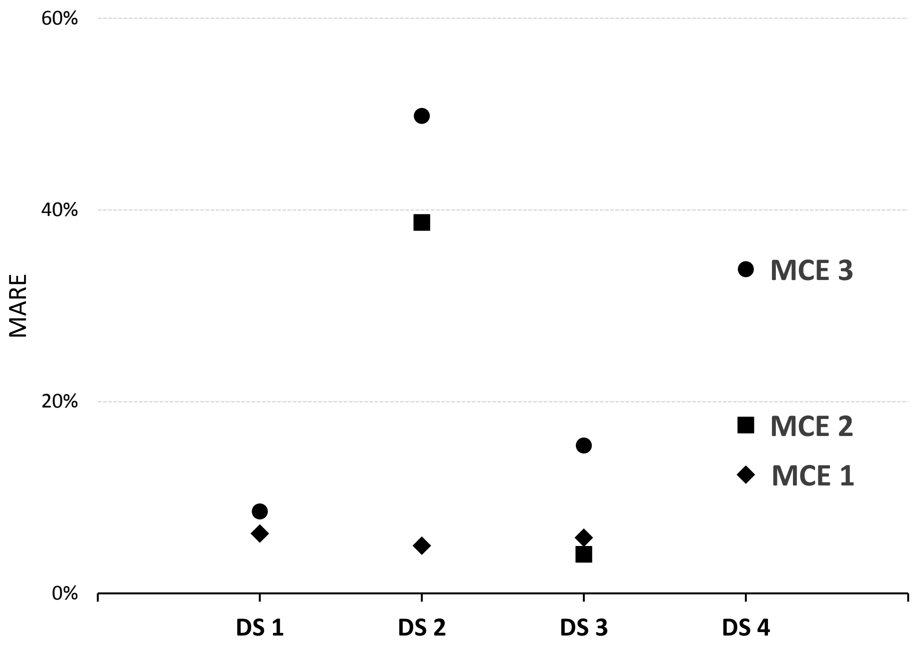


Figure 4.13: Performance of the cost estimation approaches in terms of MARE

We also evaluated the performance of spline models by setting the maximum polynomial degree to 1 to make a fair comparison between MCE 2 and MCE 1, and MCE 2 and MCE 3 because MCE 1 and MCE 3 are basically linear models. Furthermore, we removed the interaction terms in the spline models by setting the "basis" input as "additive" because MCE 1 and MCE 3 do not consider interaction terms. The performance difference between the default tensor product MCE 2 model and the linear additive MCE 2 model was minimal and these changes did not affect its overall accuracy. The linear additive MCE 2 model still outperformed MCE 3 by far. We can conclude that even considering suboptimal spline model parameters, MCE 2 is a better alternative than MCE 3.

To test the significance of the prediction accuracy among three statistical approaches, MCE 1, MCE 2, and MCE 3, we follow the same approach deployed by Smith and Mason [118]. We used a paired t-test to evaluate the significance of the mean of the differences in AREs. This is done by first subtracting the ARE of a MCE approach from another one and then using the t-distribution to test the null hypothesis where the mean of the differences is equal to zero. In Table 4.8, p-values for the paired t-tests on the mean of the differences are given. For instance, let us consider DS 2 for the pair of MCE 1 and MCE 2. First, the ARE of MCE 2 is subtracted from MCE 1 and then a paired t-test is performed to check the significance of the mean of the differences between these two cases. The associated p-value for this test is dramatically less than 0.001. Therefore, at a confidence of 95%, this value is significant where the p-value proves that these two approaches produce significantly different results. When we combine it with the known information about the MARE values, we can conclude that MCE 1 finds better results than MCE 2 for the second application problem.

Considering all pair-wise manufacturing cost estimation hypothesis test results in Table 4.8, based on the p-values, we reject all of the null hypotheses where the null hypothesis suggests the mean of the differences is zero. That is, all cost estimation approaches, MCE 1, MCE 2 and MCE 3 produce significantly different ARE results than each other at a 95% confidence level. Therefore, we can conclude that there is a clear dominance in the performance of MCE 1 compared to MCE 3 and MCE 2 compared to MCE 3. However, for the MCE 1 and MCE 2 pair, we could not conclude if one of them is superior method over the other one because for only DS 2, MCE 1 demonstrates a clear dominance when MARE values are considered. For DS 3, MCE 2 turns out to be the best approach but very close in performance to MCE 1. For the last application dataset, DS 4, MCE 1 finds slightly more accurate estimated values than MCE 2.

Table 4.8: p-values for the paired t-tests of the pairs of MCE approaches

DS 1	**MCE 3**	**MCE 2**
MCE 1	9.12×10^{-8}	N/A
MCE 2	N/A	

DS 2	**MCE 3**	**MCE 2**
MCE 1	6.65×10^{-9}	2.68×10^{-9}
MCE 2	0.0003	

DS 3	**MCE 3**	**MCE 2**
MCE 1	2.52×10^{-17}	3.44×10^{-15}
MCE 2	3.62×10^{-18}	

DS 4	**MCE 3**	**MCE 2**
MCE 1	1.50×10^{-22}	1.33×10^{-18}
MCE 2	2.58×10^{-25}	

We also considered the sensitivity of MARE with respect to the number of clusters for MCE 1. As expected, MARE decreases as the number of clusters increases and finally it converges to a limit value. The limit MARE values are around 5%, 3%, 4% and 11% for the application problems DS 1, DS 2, DS 3 and DS 4, respectively. That is, increasing the number of clusters does not much affect the accuracy of the estimated cost values beyond the values that we established for datasets 1 to 4 of 7, 11, 14 and 10 clusters, respectively. Figure 4.14 shows this change in MARE values when the number of clusters increases for each application dataset. Recall that the choice of the number of clusters may not be unique. Even though increasing the number of clusters results in more accurate estimates, it might be over parameterized.

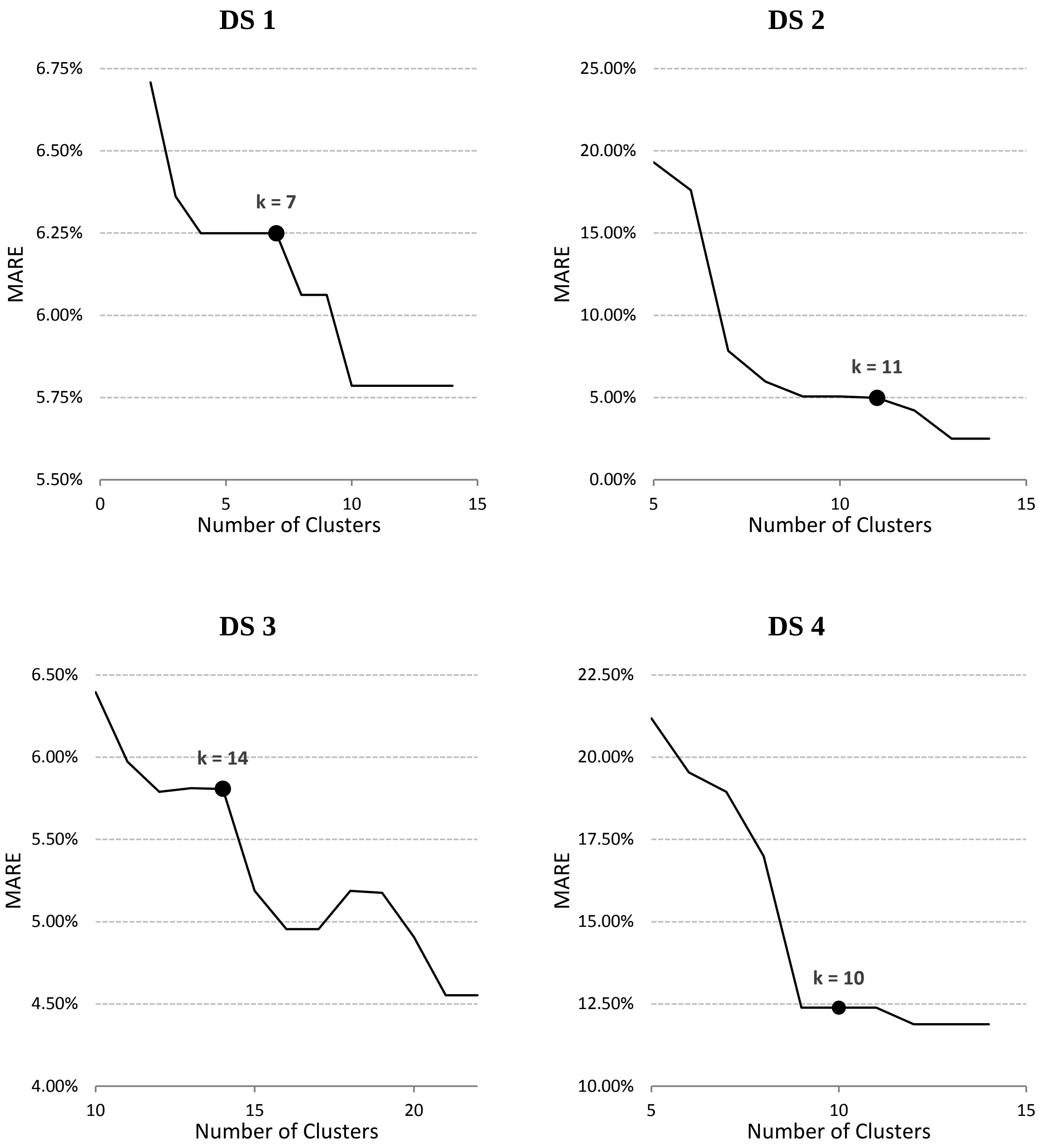


Figure 4.14: MARE vs. number of clusters of each application problem for MCE 1

Monitoring the actual cost vs predicted cost graphs along with the error graphs helps evaluate the fit of predictive models. Figures 4.15, 4.16, 4.17 and 4.18 illustrate the performance of the MCE methods compared to the actual cost values for problems DS 1, DS 2, DS 3, and DS

4, respectively. In these figures, the actual cost values are sorted in ascending order to enhance the visibility of the error values. Existence of categorical predictors certainly influences the prediction precision due to the ranking issues among categories and non-distinct borders between transitions of categories. It is a challenge especially for problems DS 1 and DS 4. The Max ARE, MARE and RMSE values indicate that MCE 1 performed the best on all datasets compared to MCE 2 and MCE 3. On the other hand, MCE 2 demonstrated very good performance on DS 3 but only acceptable performance on DS 2 and DS 4. Notice that there is no graph provided for the predicted values from MCE 2 for DS 1 because, MCE 2 is not applicable due to lack of a continuous predictor to form the basis of the spline function. The appropriate statistic to measure the proportion of total variance explained by an estimation model is the coefficient of determination and it is termed R^2. That is, the coefficient of variation provides a measure of how well the actual cost of products are predicted by the cost estimation models. We provide the R^2 values for each MCE model in Table 4.9. The maximum R^2 value for each data set is in bold to show the best model fit among MCE 1, MCE 2 and MCE 3. The R^2 values of MCE 1 and MCE 3 from the table show that finding a well suited model for DS 1 is challenging due to lack of relevant continuous predictors in the dataset. Adding more variables to the MCE models for DS 1 might increase the true explanatory power of the models but unfortunately the dataset was strictly limited to only eight categorical predictors. However, this dataset is atypical as most manufactured products include both numeric and categorical cost drivers. For the other datasets, each MCE approach is able to explain the total variability with a high R^2. Thus, none of the MCE models suffer from low prediction accuracy except in the instance for DS 1. For a better illustration of R^2 (R-sq) values, we plotted the fitted values (predicted cost) by observed values (actual cost) in Figures 4.19, 4.20, 4.21, and 4.22 for DS1, DS 2, DS 3 and DS 4, respectively.

Table 4.9: Coefficient of determination (R^2) values for the MCE approaches

$\boldsymbol{R^2}$

	MCE 1	**MCE 2**	**MCE 3**
DS 1	**63.49%**	N/A	53.19%
DS 2	**99.94%**	91.50%	84.63%
DS 3	96.83%	**99.52%**	90.49%
DS 4	**93.69%**	88.46%	76.47%

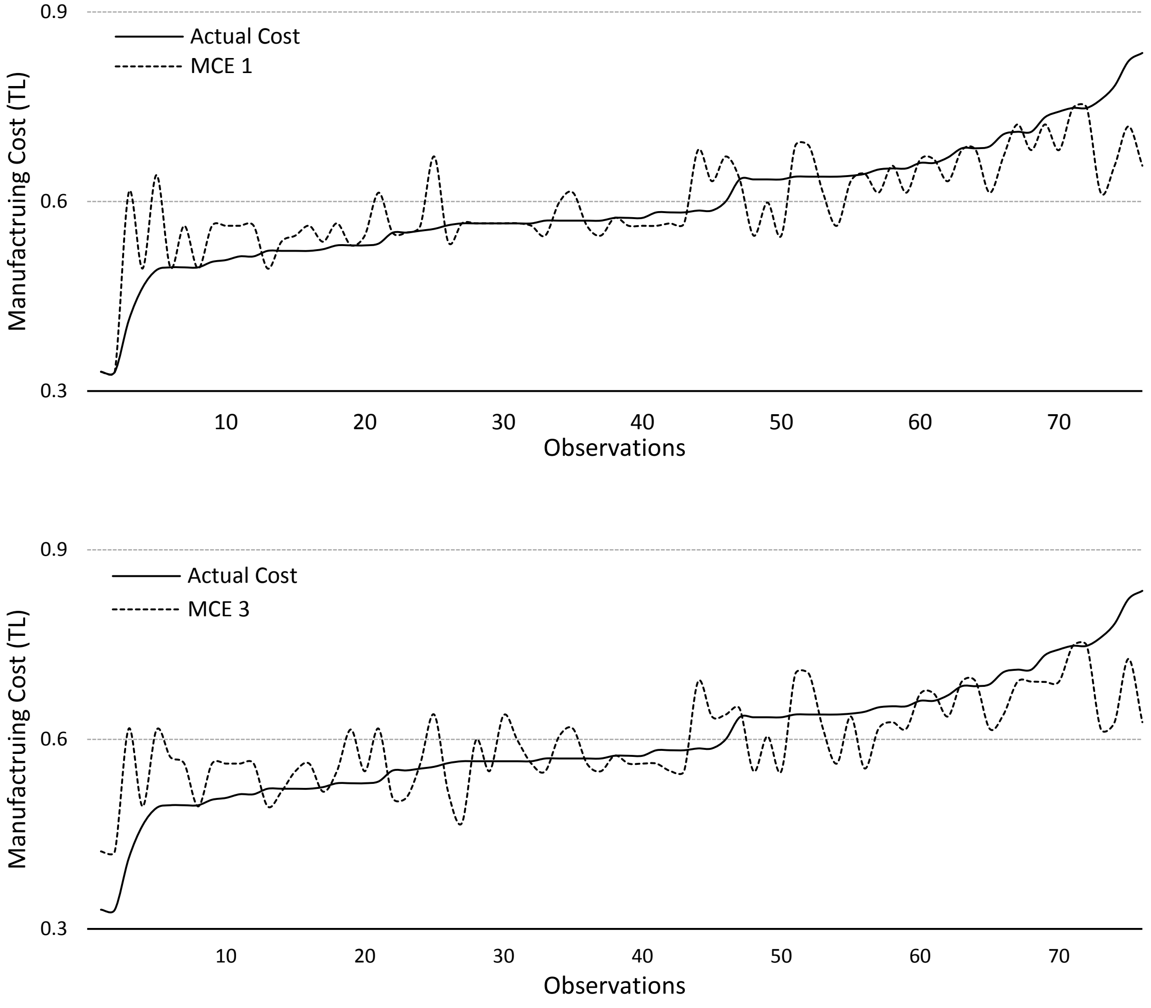


Figure 4.15: Performance of the MCE models: Actual vs. Predicted Cost for DS 1

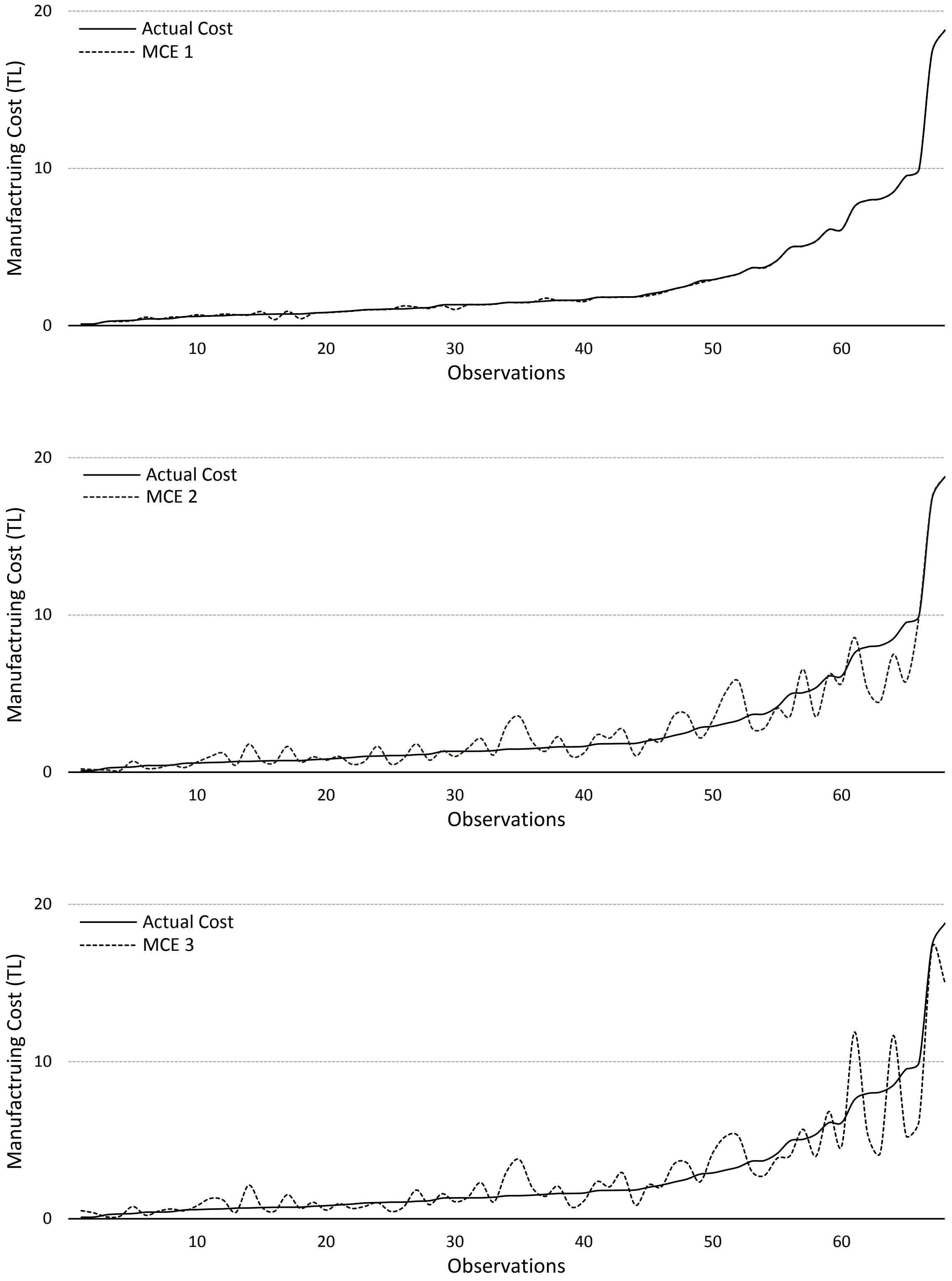


Figure 4.16: Performance of the MCE models: Actual vs. Predicted Cost for DS 2

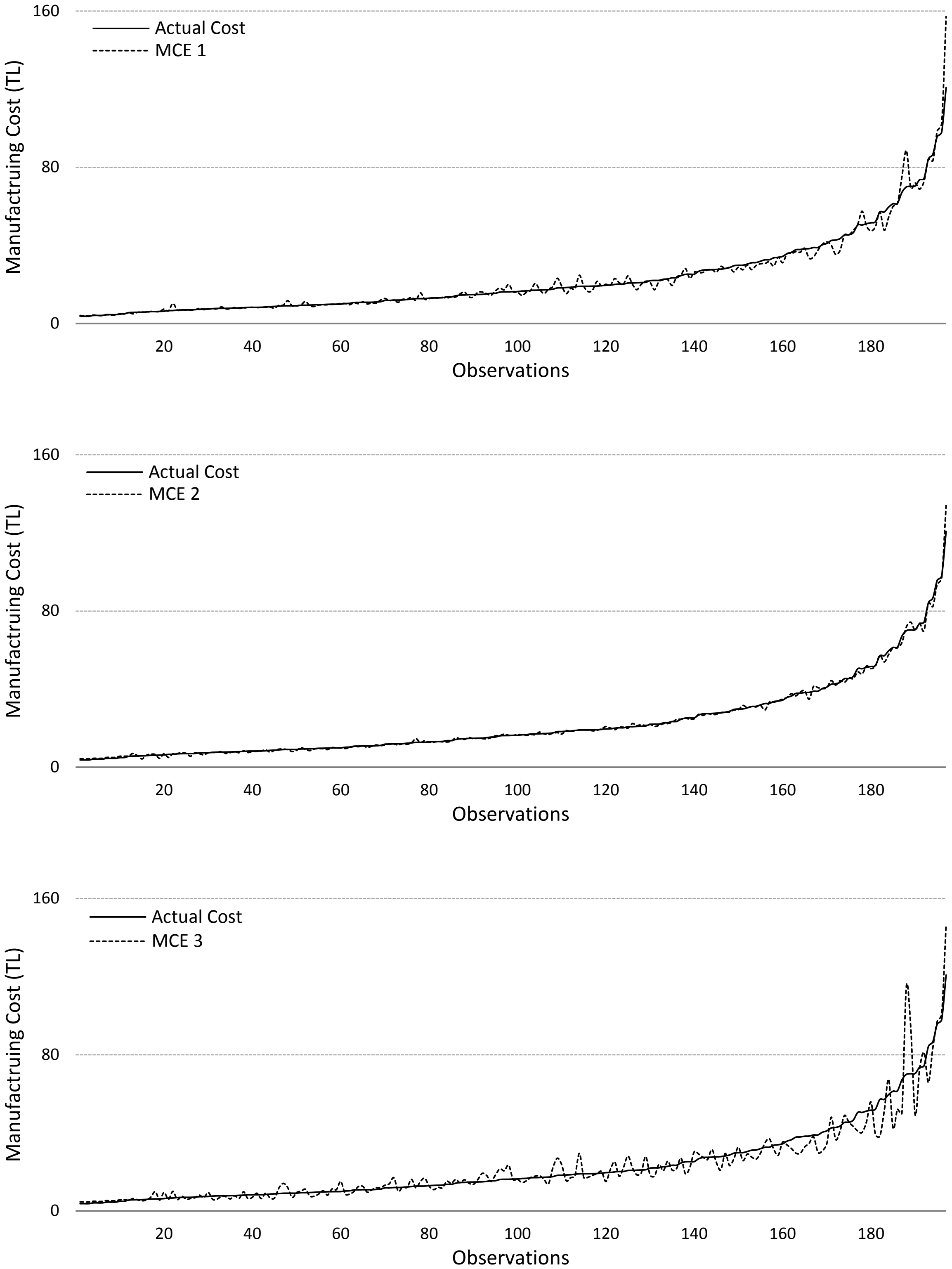


Figure 4.17: Performance of the MCE models: Actual vs. Predicted Cost for DS 3

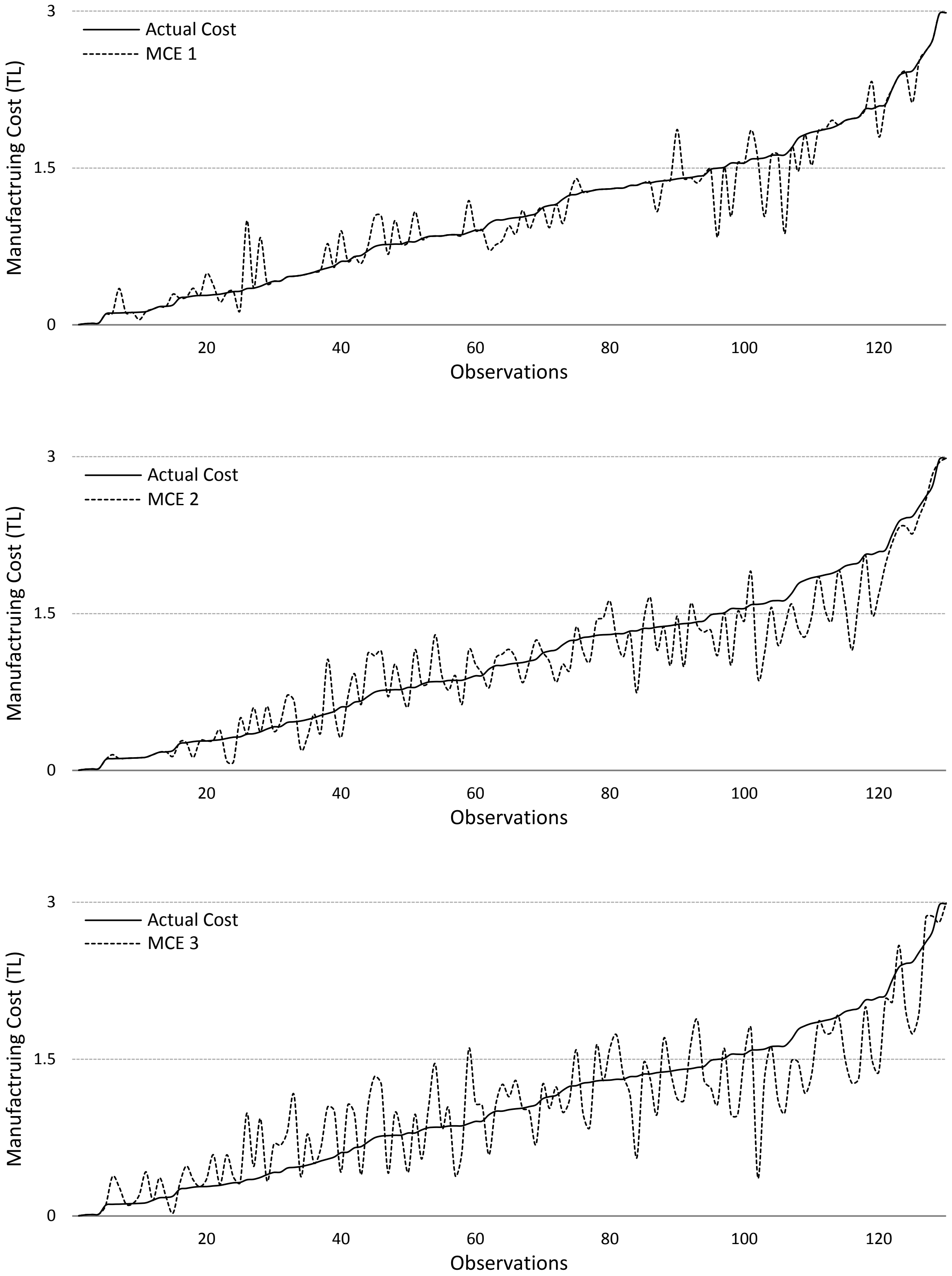


Figure 4.18: Performance of the MCE models: Actual vs. Predicted Cost for DS 4

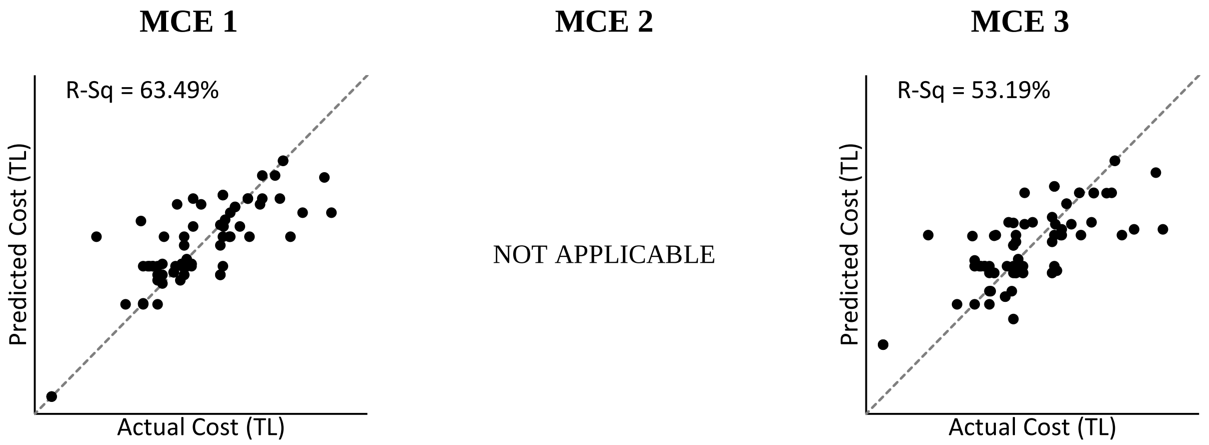


Figure 4.19: Fitted values (predicted cost) vs. observed values (actual cost) along with the R^2 values (R-Sq) for DS 1

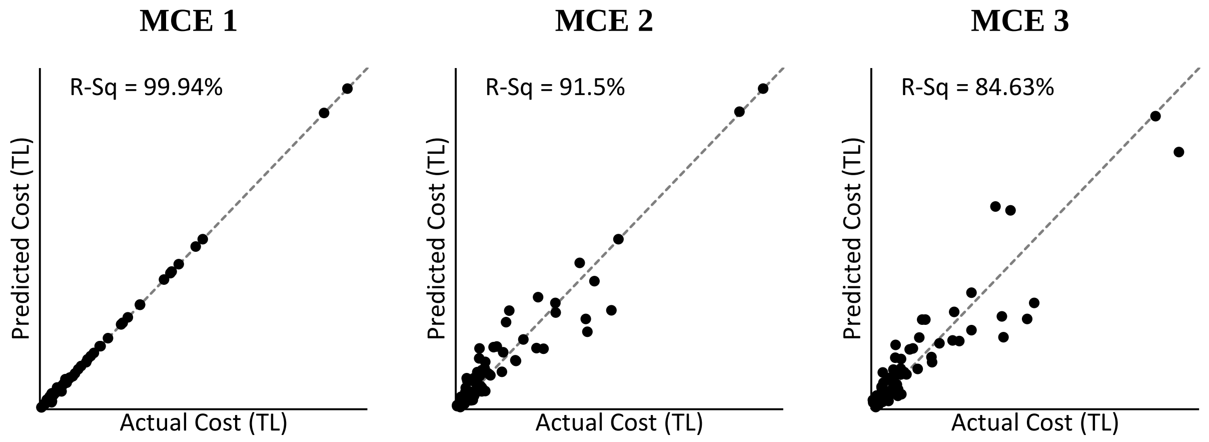


Figure 4.20: Fitted values (predicted cost) vs. observed values (actual cost) along with the R^2 values (R-Sq) for DS 2

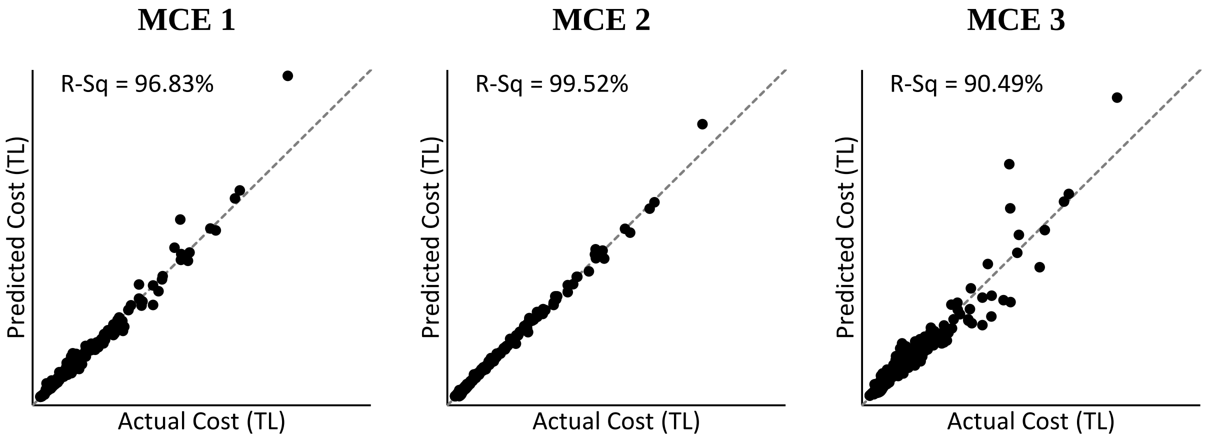


Figure 4.21: Fitted values (predicted cost) vs. observed values (actual cost) along with the R^2 values (R-Sq) for DS 3

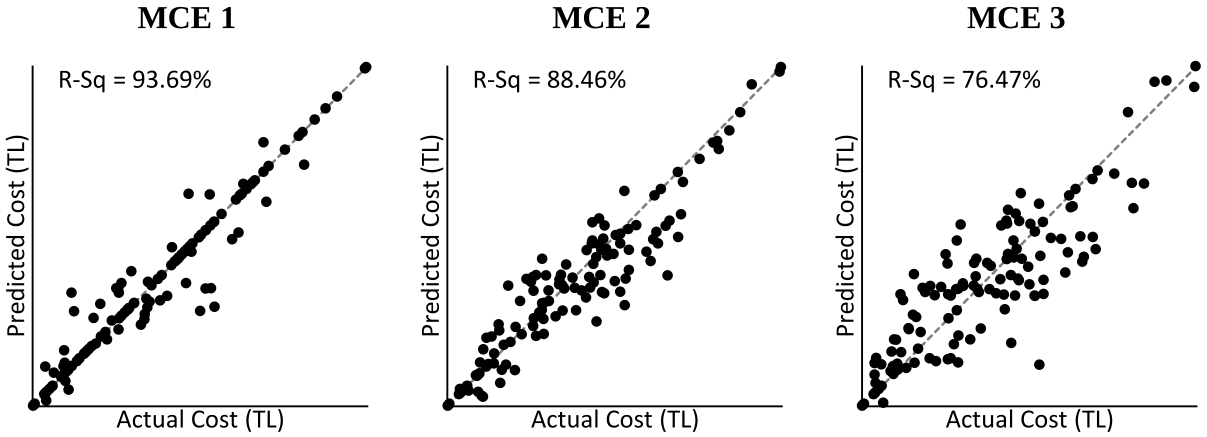


Figure 4.22: Fitted values (predicted cost) vs. observed values (actual cost) along with the R^2 values (R-Sq) for DS 4

As a final point, it is important to discuss the spread of error intervals. For DS 1, MCE 1 and MCE 3 resulted in an error distribution within the $[-0.21, 0.19]$ TL interval. However, ARE values are most important when judging the prediction performance of a model since the magnitude of error is a relative percentage of the actual cost of a product. The error intervals for DS 2 vary among different cost estimation methods. That is, the MCE 1 error spread is between – 0.21 and + 0.35 TL from the actual cost values. However, for the same dataset MCE 2 produces errors in the $[-2.48, 3.74]$ TL interval and MCE 3 predictions deviate from the actual cost with ± 4.29 TL. It clearly shows that MCE 1 dominates the other two cost estimation methods based on smaller prediction error intervals in general. The error values fluctuate within [-36.26, 9.34], [-13.26, 4.36] and [-44.98, 21.21] for DS 3 according to MCE 1, MCE 2 and MCE 3, respectively. MCE 2 certainly demonstrates the best performance among these three approaches in terms of overall error precision. For the last application problem, DS 4, [-0.65, 0.76], [-0.52, 0.83] and [-0.71, 1.23] error spreads have been observed for MCE 1, MCE 2 and MCE 3, respectively. Please refer to Figures 4.23, 4.24 and 4.25 for a better understanding in the error distributions. Notice that observations are sorted in ascending order for the actual cost values and these observations are in the same order as they are in Figures 4.15, 4.16, 4.17 and 4.18. As you can clearly see from these graphs, manufacturing cost estimation models 1, 2 and 3 either overestimate the cost, perfectly locate the actual cost, and sometimes underestimate it. The distribution of errors are not skewed to the negative or positive regions but the magnitude of error increases as the actual cost increases. Therefore, there is not enough evidence to conclude that the estimation models are biased. The minimum, the maximum and percentile distributions of the error values generated by the cost estimation approaches are provided in Figure 4.26 for DS 1 to DS 4. For MCE 1 in the DS 2 and DS 4 graphs, the boxes are not visible because the upper and lower 25th percentile and the median values are all zeros. Notice that in Figures from 4.23 to 4.26 the error values are given in the original units, Turkish Lira (TL). The spread between the maximum and the minimum error values does not explain much because the most important measures for the prediction accuracy are the relative percentage errors. That is, instead of the magnitude of the error itself, the relative error values like the maximum ARE, the minimum ARE and MARE provide more valuable information.

Our cost estimation methodologies, MCE 1 and MCE 2, can be replicated as long as an identical R code is used. The performance of the clustering technique we employed, k-medoids, is not contingent on the random number seed because of the reasons discussed in Chapters 2 and 3. Also, NOMAD carefully optimizes the underlying spline parameters within the given interval. If one of these parameters is altered, the resulting spline model would be different than the original one. However, the "crs" package can identify a good parameter combination [98] including the number of segments, the knot placement strategy, and the categorical predictor bandwidth values with respect to the minimization of the underlying cross-validation function (Equation 3.14).

We extended the analysis by increasing the order of the linear regression models for MCE 1 and MCE 3 to observe the effect of polynomial degree on the major performance factor, MARE. Second degree (quadratic) predictors of the continuous design variables are included in the regression models. It is not logical to take the second degree of categorical predictors except ordinal ones. Even for the ordinal variables, there is no distinctive transition between categories. Let us consider the variable "Leg Style" in the socks manufacturing dataset. Leg Style has four categories, none, short, medium and long. There are no defined values for the upper and lower bounds of short, medium and long length for the leg style. Thus, adding a quadratic term to the regression model is not analytical. That is the reason why the categorical predictors are kept linear and only quadratic terms are added for the continuous predictors. In Table 4.10, we present the performance metrics for the second degree regression models of MCE 1 and MCE 3 for the application problems, DS 2, DS 3, and DS 4. We call the quadratic models MCE 1Q and MCE 3Q, respectively, in the table. Notice that DS 1 is not included in the second degree regression analysis because there are no continuous predictors in the dataset.

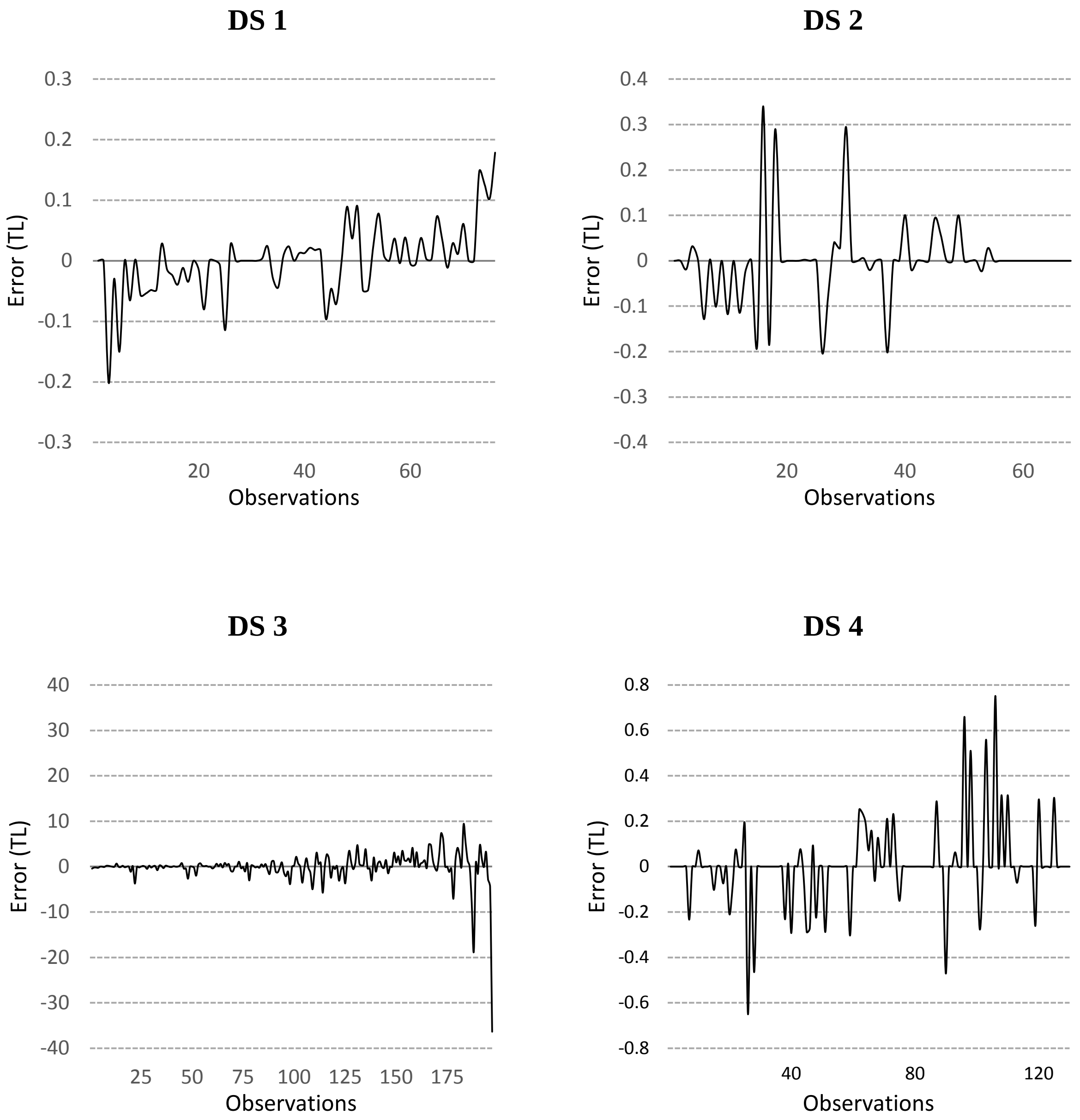
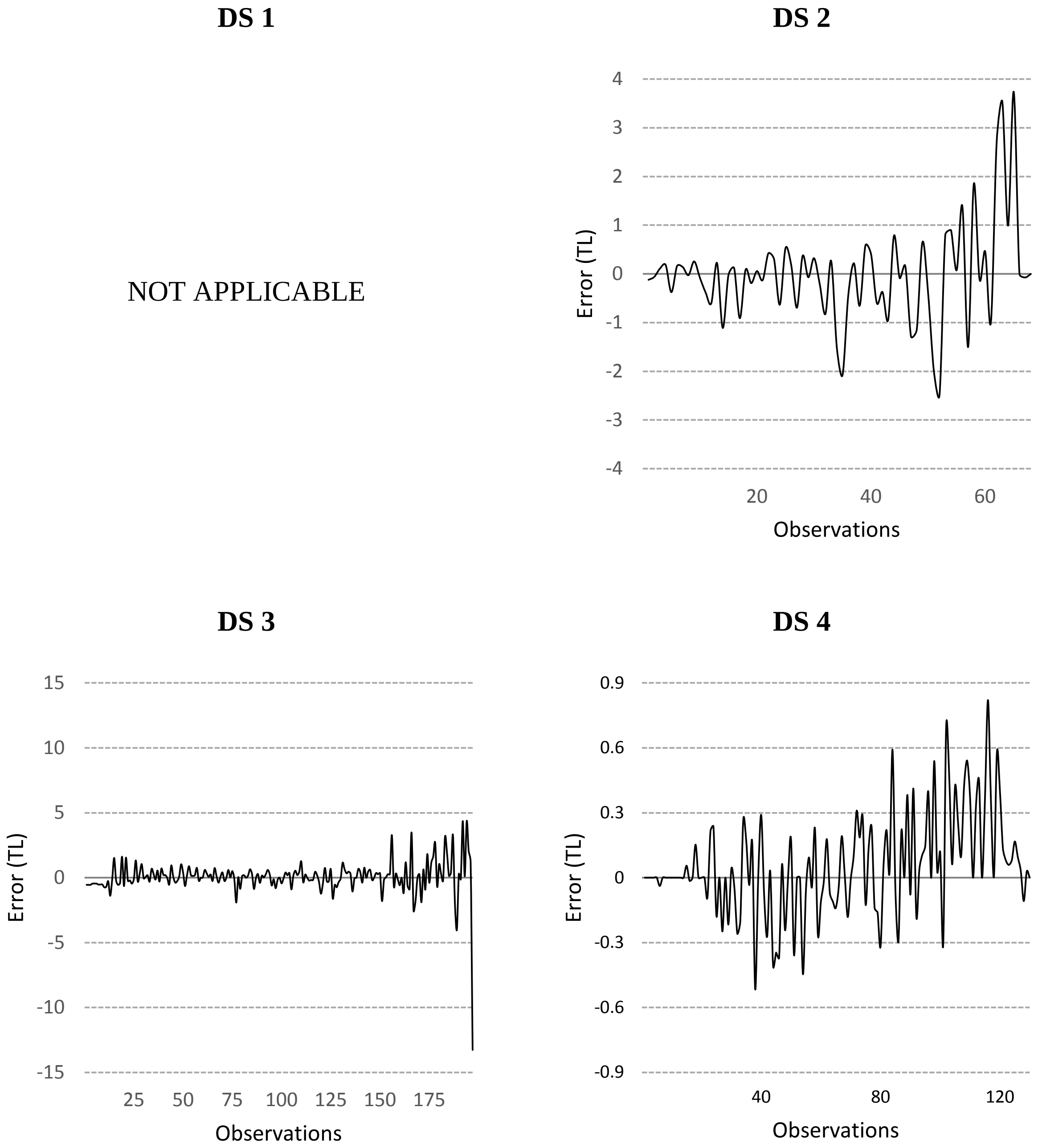


Figure 4.23: Observed prediction errors based on the performance of MCE 1



DS 1

DS 2

DS 3

DS 4

Figure 4.24: Observed prediction errors based on the performance of MCE 2

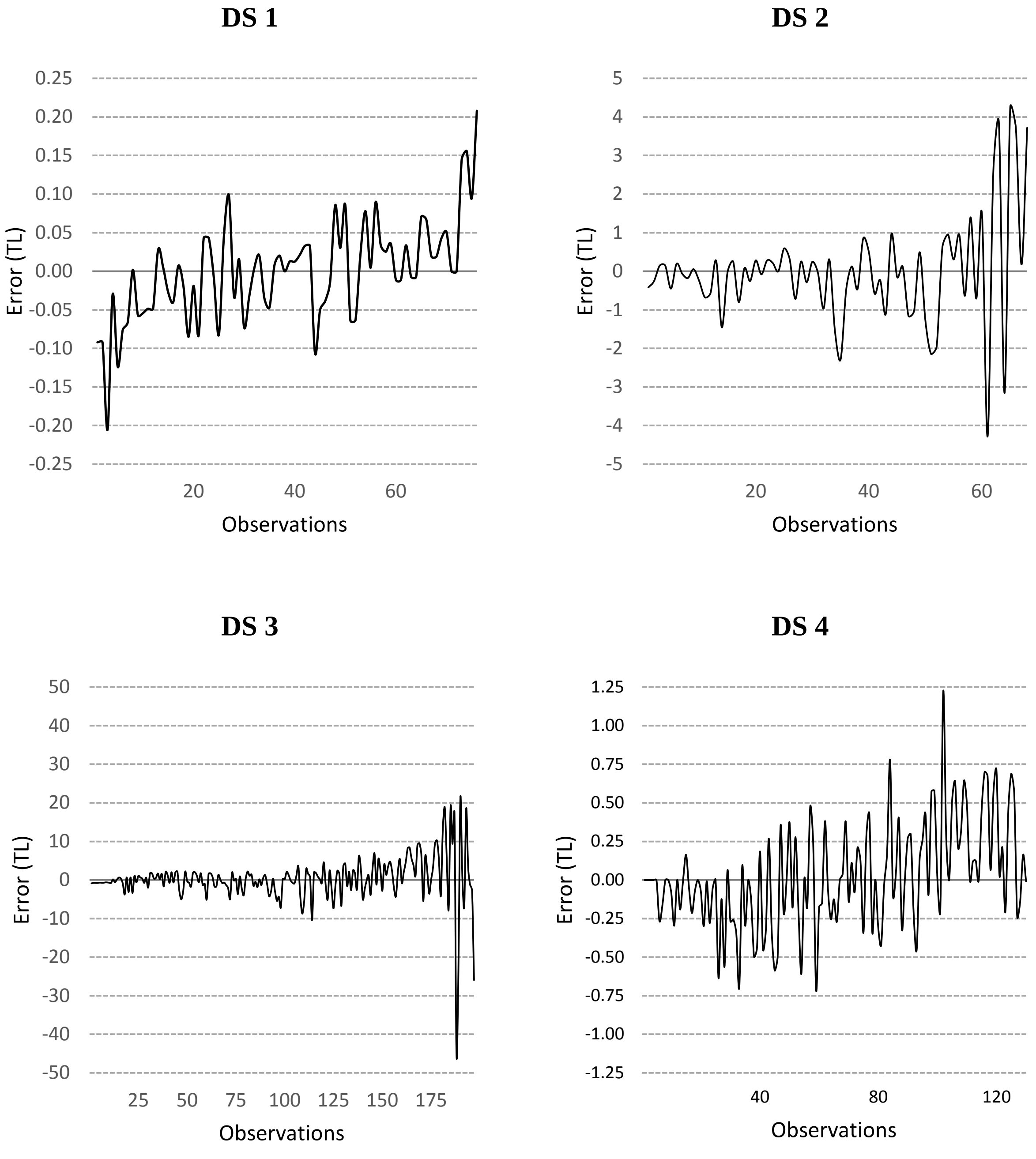


Figure 4.25: Observed prediction errors based on the performance of MCE 3

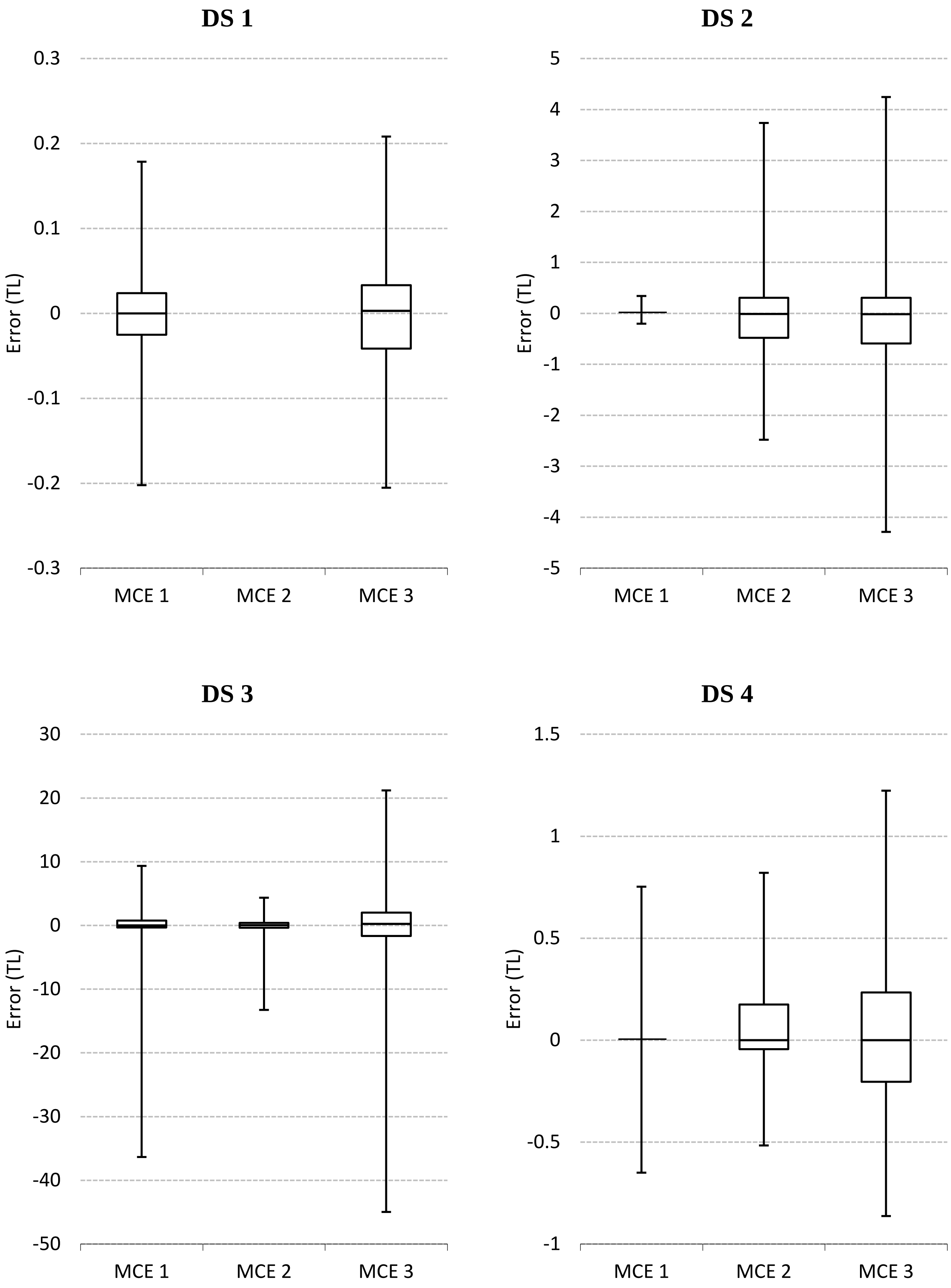


Figure 4.26: Box and whisker plots of error values generated by the MCE approaches

Table 4.10: Performance metrics of the quadratic estimation models for the application problems

DS 2

	MCE 1Q	MCE 3Q
MARE	4.37%	43.22%
Min ARE	0.00%	0.00%
Max ARE	81.12%	440.98%
MSE	0.42%	93.49%
RMSE	6.50%	96.69%

DS 3

	MCE 1Q	MCE 3Q
MARE	2.31%	12.22%
Min ARE	0.00%	0.00%
Max ARE	23.26%	44.41%
MSE	188.45%	1537.48%
RMSE	137.28%	392.11%

DS 4

	MCE 1Q	MCE 3Q
MARE	9.69%	36.68%
Min ARE	0.00%	0.00%
Max ARE	250.10%	399.26%
MSE	1.34%	6.80%
RMSE	11.60%	26.08%

Compared with the linear models, as expected, there are slight improvements in the MCE 1 model fit and MARE values for DS 2, DS 3 and DS 4. There is considerable improvement for DS 2 and DS 3 MARE performances when the quadratic terms are included. For DS 4, MSE and RMSE values are improved but the MARE value deteriorated from the linear MCE 3 model. Adding quadratic terms is expected to increase the overall prediction accuracy, but it may result in overfit models for the cross-validated (that is, new) observations. Even though increasing the polynomial degree of the regression models seems like a good alternative to improve prediction accuracy, the models may become over parameterized. Considering the DS 4 MARE values, for even only one case, we proved that the improvement is not consistent as the polynomial degree increases. As a good illustration of the MARE (given in the y-axis) performances of MCE 1 (linear) vs. MCE 1Q (quadratic) and MCE 3 (linear) vs. MCE 3Q (quadratic) regression models for DS 2, DS 3 and DS 4, refer to Figures 4.27 and 4.28, respectively.

Regardless of what parametric or non-parametric approach is chosen, manager and customer confidence in these methods is a known challenge. When designing a new product or manufacturing a customer's new unique design, the focal point is to establish a price which maximizes customer value while still being profitable. Since an irreversible and large amount of capital is tied up in production elements, predicting manufacturing costs accurately is significant. Poorly established product prices may cause a loss of profit due to the gap between the expected cost and the actual cost or a loss of customers due to higher prices than competitors in the market. Our sophisticated cost estimation methodologies consistently demonstrated significantly better accuracy than a traditional regression approach. The cross-validated prediction results fortify the credibility of our suggested methodologies. There is no clear dominance between the categorical spline regression based approach and the cluster based cost estimation approach, but we can observe that the latter method is relatively more reliable when its performance on MARE is considered over four application problems as a whole.

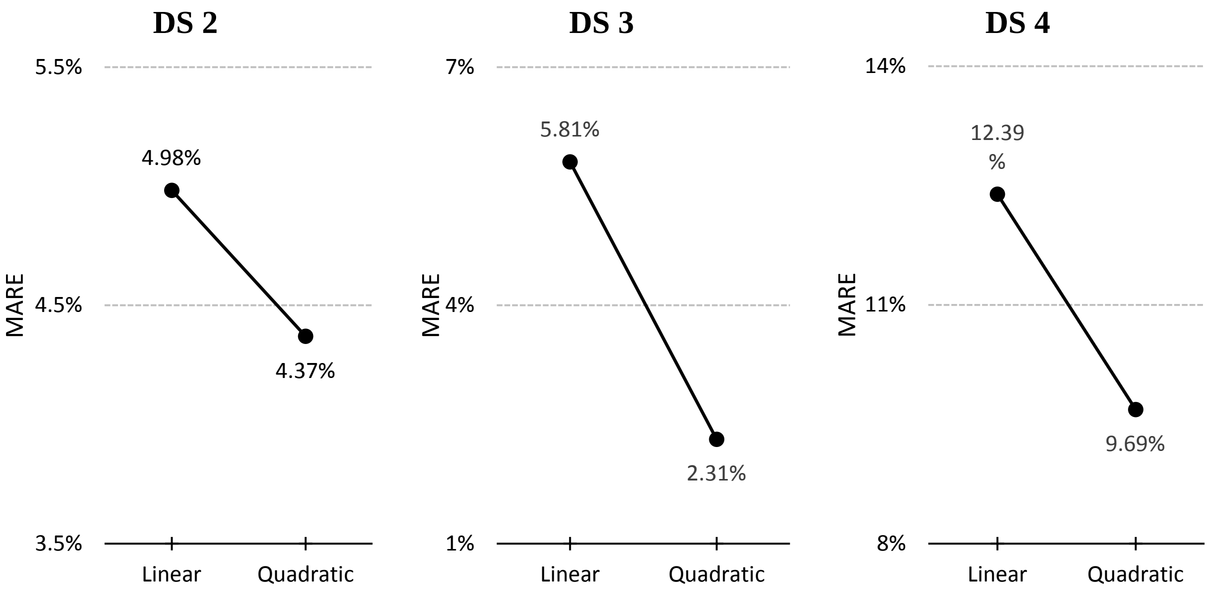


Figure 4.27: MARE values for linear (MCE 1) vs. quadratic (MCE 1Q) regression models

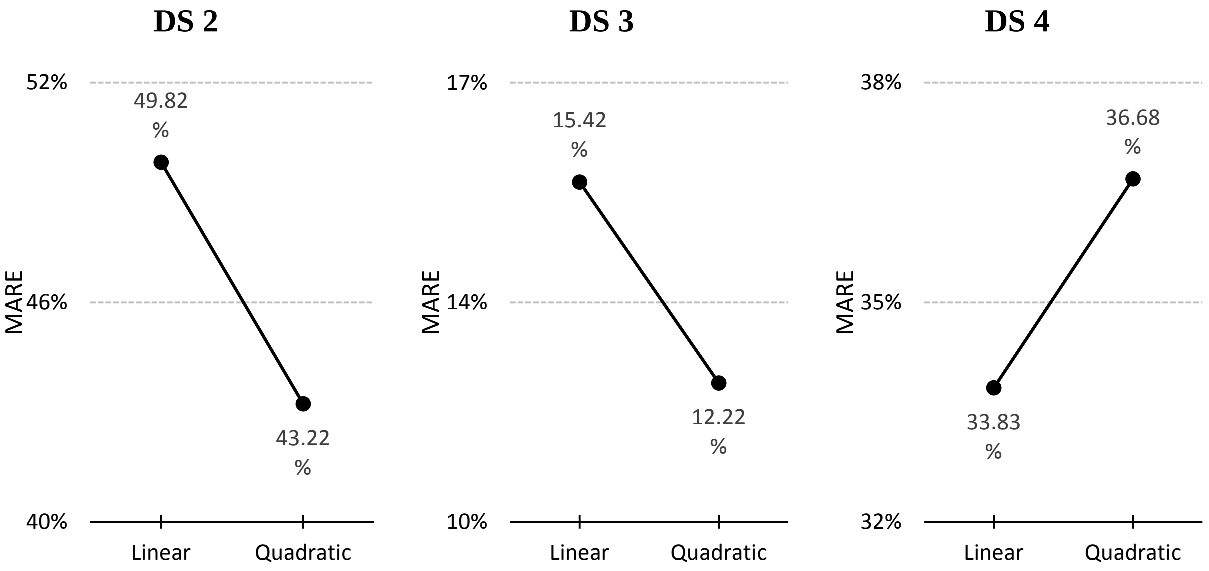


Figure 4.28: MARE values for linear (MCE 3) vs. quadratic (MCE 3Q) regression models

4.5 Manufacturing Cost Estimation User Interface

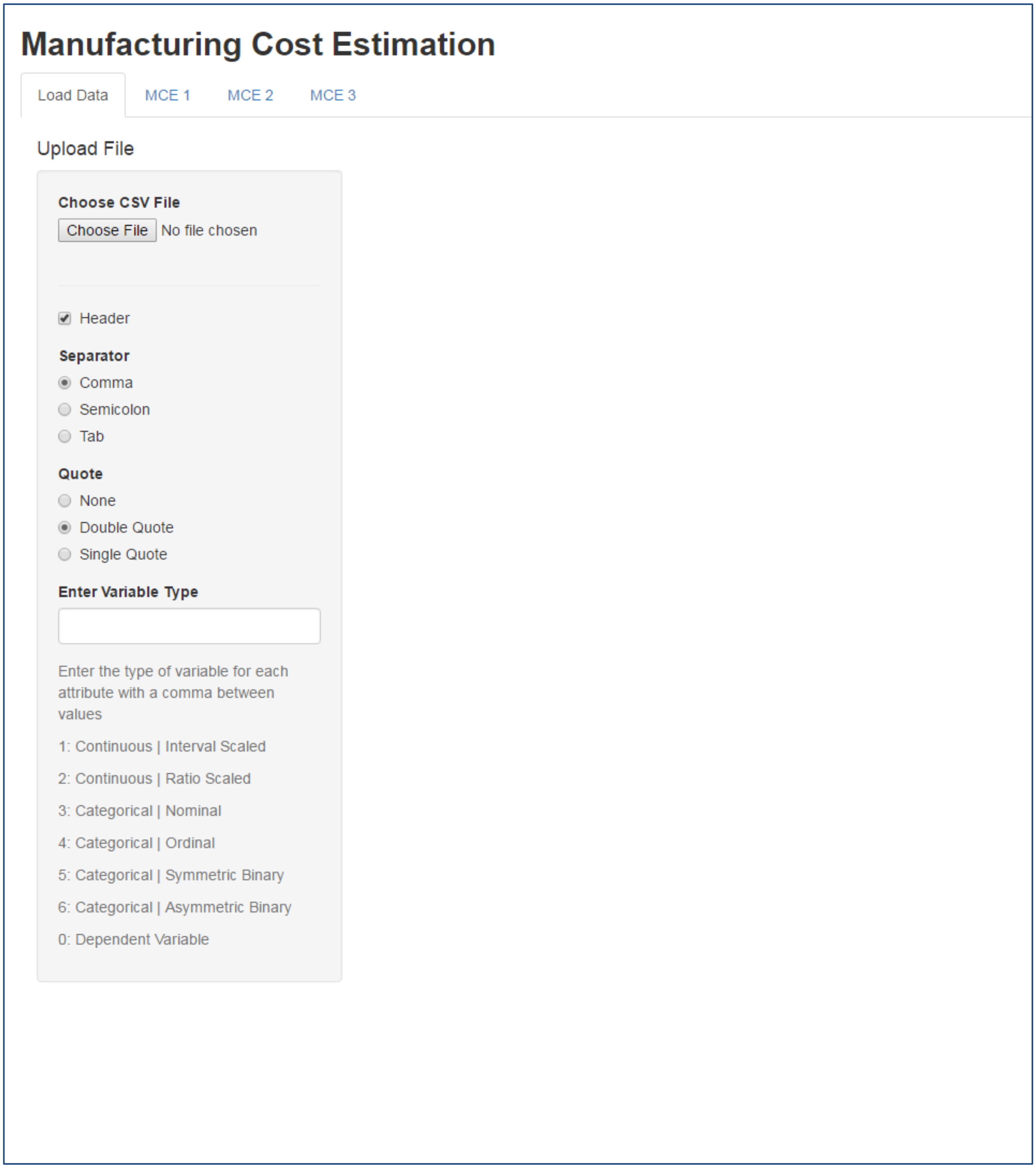
One of the most important post analysis tasks is to generalize the R code to solve any problem with different sizes of datasets. It is a milestone to enable the technology transfer of the cost estimation approaches, clustering and spline based, with an end user graphical interface. We built an interface using the R package called “shiny” [119]. It is a web application framework to turn R scripts into interactive web applications. It has two main components: (1) Server side component that is responsible for the computational tasks and rendering plots and tables, (2) User interface component that is the actual interactive web interface with input entering elements such as check boxes, radio and action buttons, and other numeric and text input boxes. The interface is a web based application and we plan to publish it online in the future for cost estimation practitioners. It consists of four main tabs: (1) Load Data, (2) MCE 1, (3) MCE 2, and (4) MCE 3.

The “Load Data” tab is for uploading a dataset to the system in a comma separated values format. In this tab, the user enters a vector representing the variable types as discussed in section 4.1. A screenshot of the data loading tab is given in Figure 4.29. When the data upload is complete, user is notified with a success message and a table form of the dataset as shown in Figure 4.30.

The “MCE 1” tab is for the clustering based cost estimation approach. It has two main parts. The first part has three inputs, namely the minimum number of clusters, the maximum number of clusters and a red dot to mark the selected number of clusters on the graphs. The second part has two inputs, the best number of clusters and the polynomial regression model degree: linear, quadratic or a higher degree. A screenshot of the clustering based cost estimation tab is given in Figure 4.31. The interface passes the given information to the server and the server side application renders the C-index, Gamma and silhouette width graphs based on the minimum and maximum number of clusters. The user is required to enter the best number of clusters to proceed to the cost estimation step. The number of clusters can be determined by following the simple heuristic described in section 3.1.2. When the best number of clusters is entered, the application builds the final cluster contents and cluster specific estimation models and then produces the actual cost vs. predicted cost graph along with a table of predicted values (the column name is y_hat) for each data point. In this table, there is an extra column called “cluster” that shows in which cluster the specific data point is classified. A screenshot of the MCE 1 tab after solving a cost estimation problem is given in Figure 4.32.

The “MCE 2” tab is for the spline based cost estimation approach. It has seven main input elements as discussed in section 4.3. The spline model inputs are maximum and minimum spline degrees, maximum and minimum number of segments, optimization complexity, knot placement strategy, spline basis, optimization algorithm, and the cross-validation function. A screenshot of the spline based cost estimation tab is given in Figure 4.33. All inputs are passed to the “crs” package and then a categorical spline regression model is constructed to predict manufacturing costs. The output is similar to the “MCE 1” tab’s output. It generates a graph of the actual vs. predicted costs and also a table of predicted values. A screenshot of the MCE 2 tab after solving a cost estimation problem is given in Figure 4.34.

The last tab, “MCE 3”, represents the traditional cost estimation approach, a single polynomial regression model. It only has a single input for the regression degree. A screenshot of the single regression cost estimation tab is provided in Figure 4.35. Once the regression degree is determined (selected) a similar output is generated where the actual vs. predicted cost graph and the table of predicted values are shown. A screenshot of the MCE 3 tab after solving a cost estimation problem is given in Figure 4.36.



Manufacturing Cost Estimation

Upload File

Choose CSV File

No file chosen

☑ Header

Separator

● Comma
○ Semicolon
○ Tab

Quote

○ None
● Double Quote
○ Single Quote

Enter Variable Type

Enter the type of variable for each attribute with a comma between values

1: Continuous | Interval Scaled

2: Continuous | Ratio Scaled

3: Categorical | Nominal

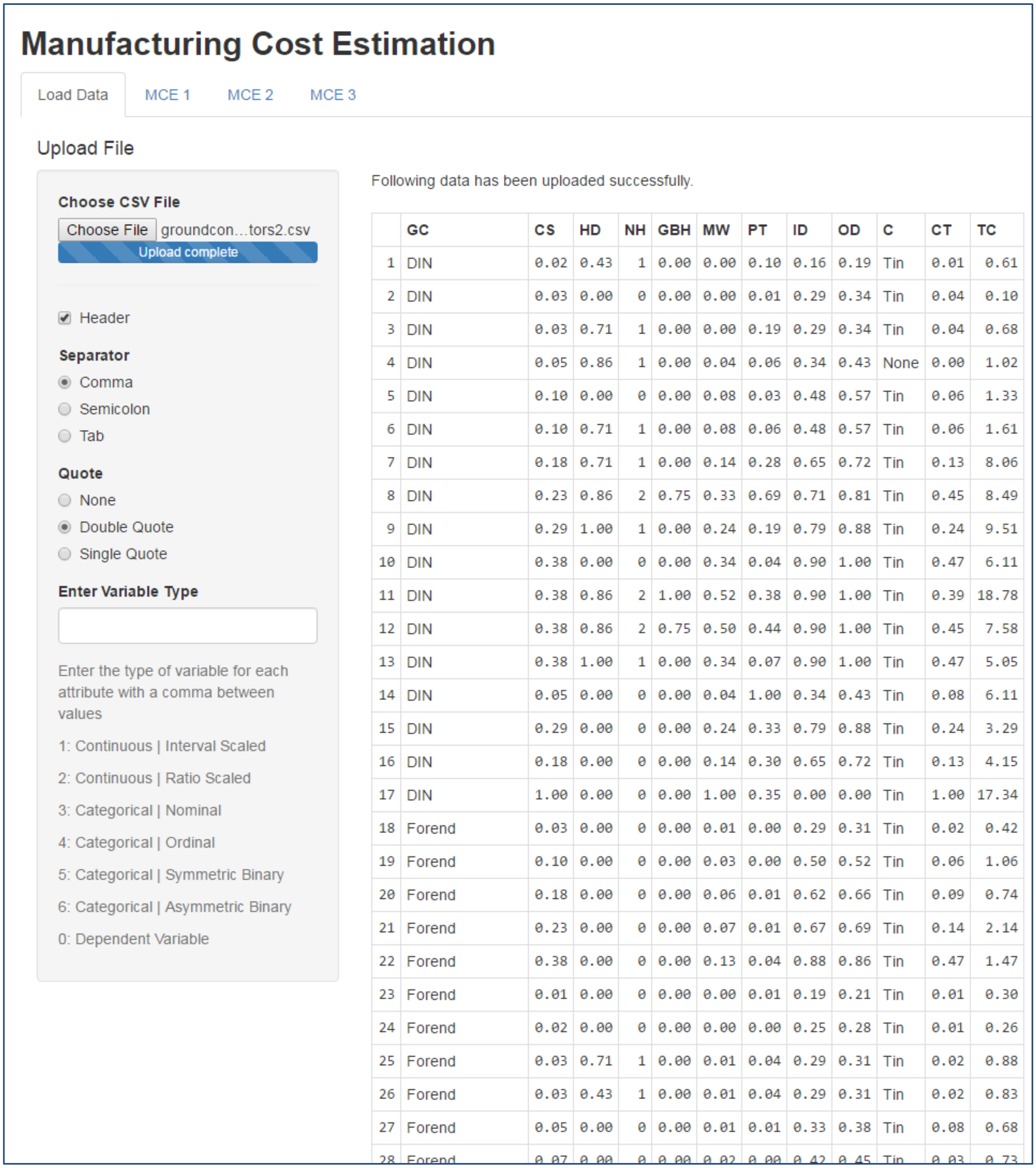
4: Categorical | Ordinal

5: Categorical | Symmetric Binary

6: Categorical | Asymmetric Binary

0: Dependent Variable

Figure 4.29: Data loading tab of the manufacturing cost estimation user interface



Manufacturing Cost Estimation

Upload File

Choose CSV File

groundcon...tors2.csv

☑ Header

Separator

● Comma
○ Semicolon
○ Tab

Quote

○ None
● Double Quote
○ Single Quote

Enter Variable Type

Enter the type of variable for each attribute with a comma between values

1: Continuous | Interval Scaled

2: Continuous | Ratio Scaled

3: Categorical | Nominal

4: Categorical | Ordinal

5: Categorical | Symmetric Binary

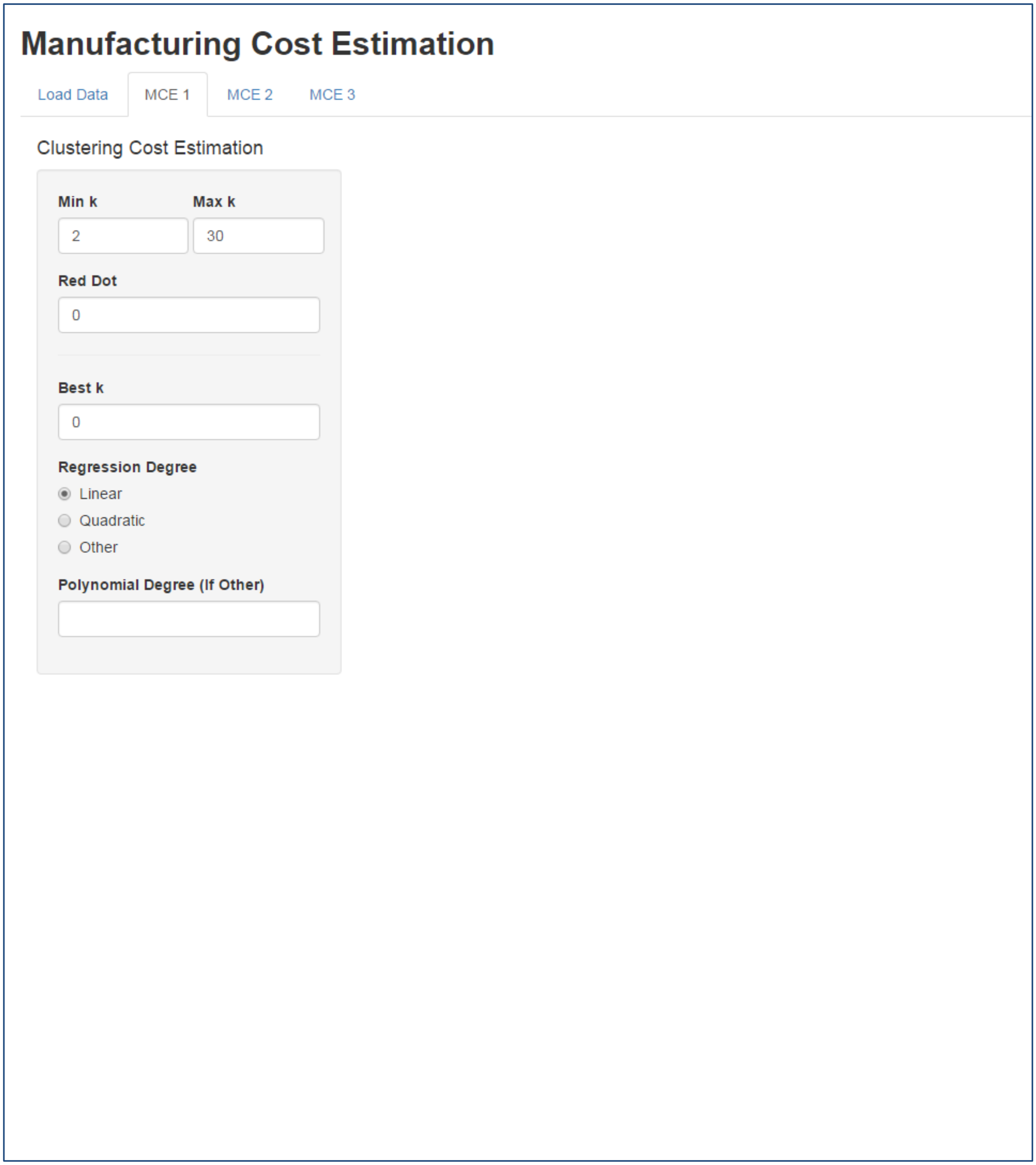
6: Categorical | Asymmetric Binary

0: Dependent Variable

Following data has been uploaded successfully.

	GC	CS	HD	NH	GBH	MW	PT	ID	OD	C	CT	TC
1	DIN	0.02	0.43	1	0.00	0.00	0.10	0.16	0.19	Tin	0.01	0.61
2	DIN	0.03	0.00	0	0.00	0.00	0.01	0.29	0.34	Tin	0.04	0.10
3	DIN	0.03	0.71	1	0.00	0.00	0.19	0.29	0.34	Tin	0.04	0.68
4	DIN	0.05	0.86	1	0.00	0.04	0.06	0.34	0.43	None	0.00	1.02
5	DIN	0.10	0.00	0	0.00	0.08	0.03	0.48	0.57	Tin	0.06	1.33
6	DIN	0.10	0.71	1	0.00	0.08	0.06	0.48	0.57	Tin	0.06	1.61
7	DIN	0.18	0.71	1	0.00	0.14	0.28	0.65	0.72	Tin	0.13	8.06
8	DIN	0.23	0.86	2	0.75	0.33	0.69	0.71	0.81	Tin	0.45	8.49
9	DIN	0.29	1.00	1	0.00	0.24	0.19	0.79	0.88	Tin	0.24	9.51
10	DIN	0.38	0.00	0	0.00	0.34	0.04	0.90	1.00	Tin	0.47	6.11
11	DIN	0.38	0.86	2	1.00	0.52	0.38	0.90	1.00	Tin	0.39	18.78
12	DIN	0.38	0.86	2	0.75	0.50	0.44	0.90	1.00	Tin	0.45	7.58
13	DIN	0.38	1.00	1	0.00	0.34	0.07	0.90	1.00	Tin	0.47	5.05
14	DIN	0.05	0.00	0	0.00	0.04	1.00	0.34	0.43	Tin	0.08	6.11
15	DIN	0.29	0.00	0	0.00	0.24	0.33	0.79	0.88	Tin	0.24	3.29
16	DIN	0.18	0.00	0	0.00	0.14	0.30	0.65	0.72	Tin	0.13	4.15
17	DIN	1.00	0.00	0	0.00	1.00	0.35	0.00	0.00	Tin	1.00	17.34
18	Forend	0.03	0.00	0	0.00	0.01	0.00	0.29	0.31	Tin	0.02	0.42
19	Forend	0.10	0.00	0	0.00	0.03	0.00	0.50	0.52	Tin	0.06	1.06
20	Forend	0.18	0.00	0	0.00	0.06	0.01	0.62	0.66	Tin	0.09	0.74
21	Forend	0.23	0.00	0	0.00	0.07	0.01	0.67	0.69	Tin	0.14	2.14
22	Forend	0.38	0.00	0	0.00	0.13	0.04	0.88	0.86	Tin	0.47	1.47
23	Forend	0.01	0.00	0	0.00	0.00	0.01	0.19	0.21	Tin	0.01	0.30
24	Forend	0.02	0.00	0	0.00	0.00	0.00	0.25	0.28	Tin	0.01	0.26
25	Forend	0.03	0.71	1	0.00	0.01	0.04	0.29	0.31	Tin	0.02	0.88
26	Forend	0.03	0.43	1	0.00	0.01	0.04	0.29	0.31	Tin	0.02	0.83
27	Forend	0.05	0.00	0	0.00	0.01	0.01	0.33	0.38	Tin	0.08	0.68
28	Forend	0.07	0.00	0	0.00	0.02	0.00	0.42	0.45	Tin	0.03	0.73

Figure 4.30: Data loading tab of the user interface after uploading a comma separated data file



Manufacturing Cost Estimation

Clustering Cost Estimation

Min k: 2

Max k: 30

Red Dot: 0

Best k: 0

Regression Degree

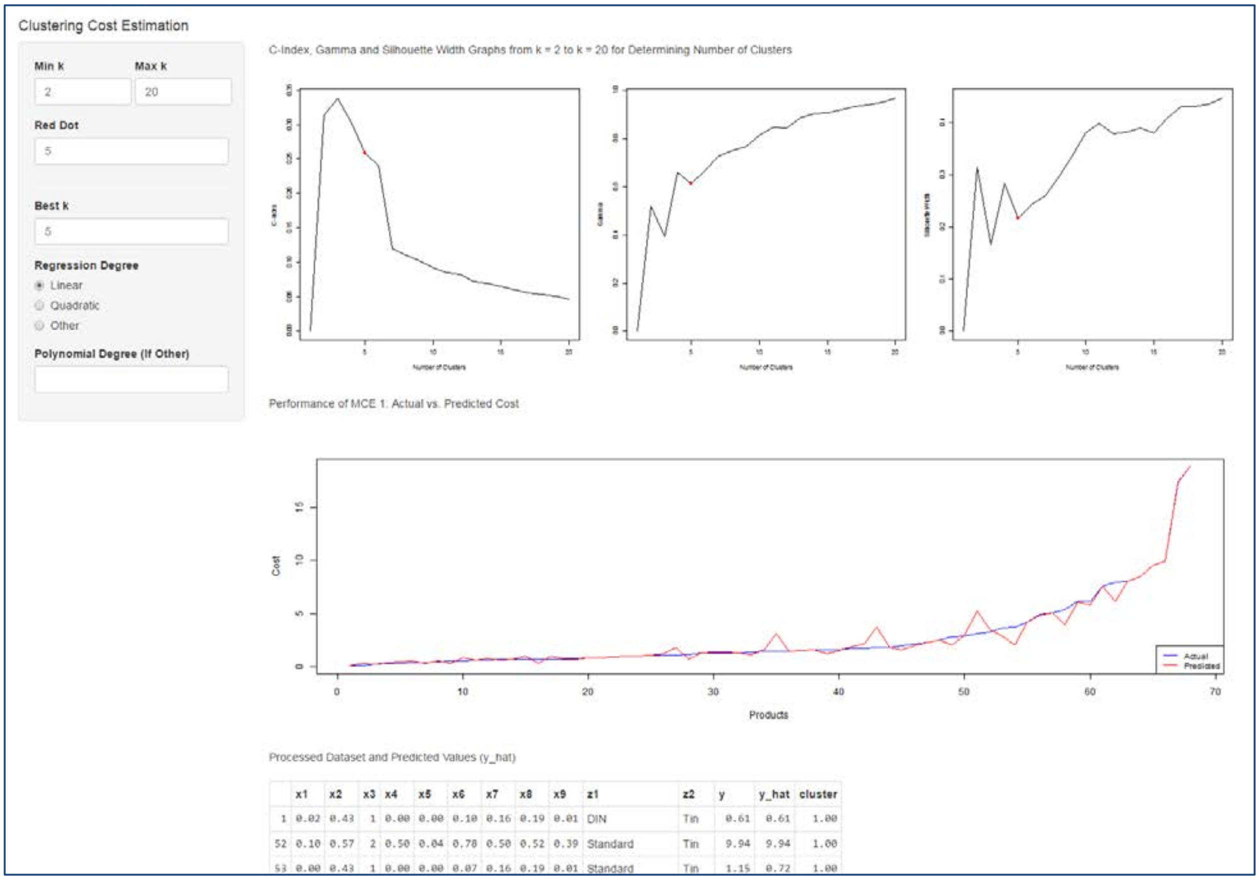
● Linear

○ Quadratic

○ Other

Polynomial Degree (If Other):

Figure 4.31: Clustering based cost estimation (MCE 1) application tab in the user interface



Clustering Cost Estimation

Min k: 2

Max k: 20

Red Dot: 5

Best k: 5

Regression Degree
● Linear
○ Quadratic
○ Other

Polynomial Degree (If Other):

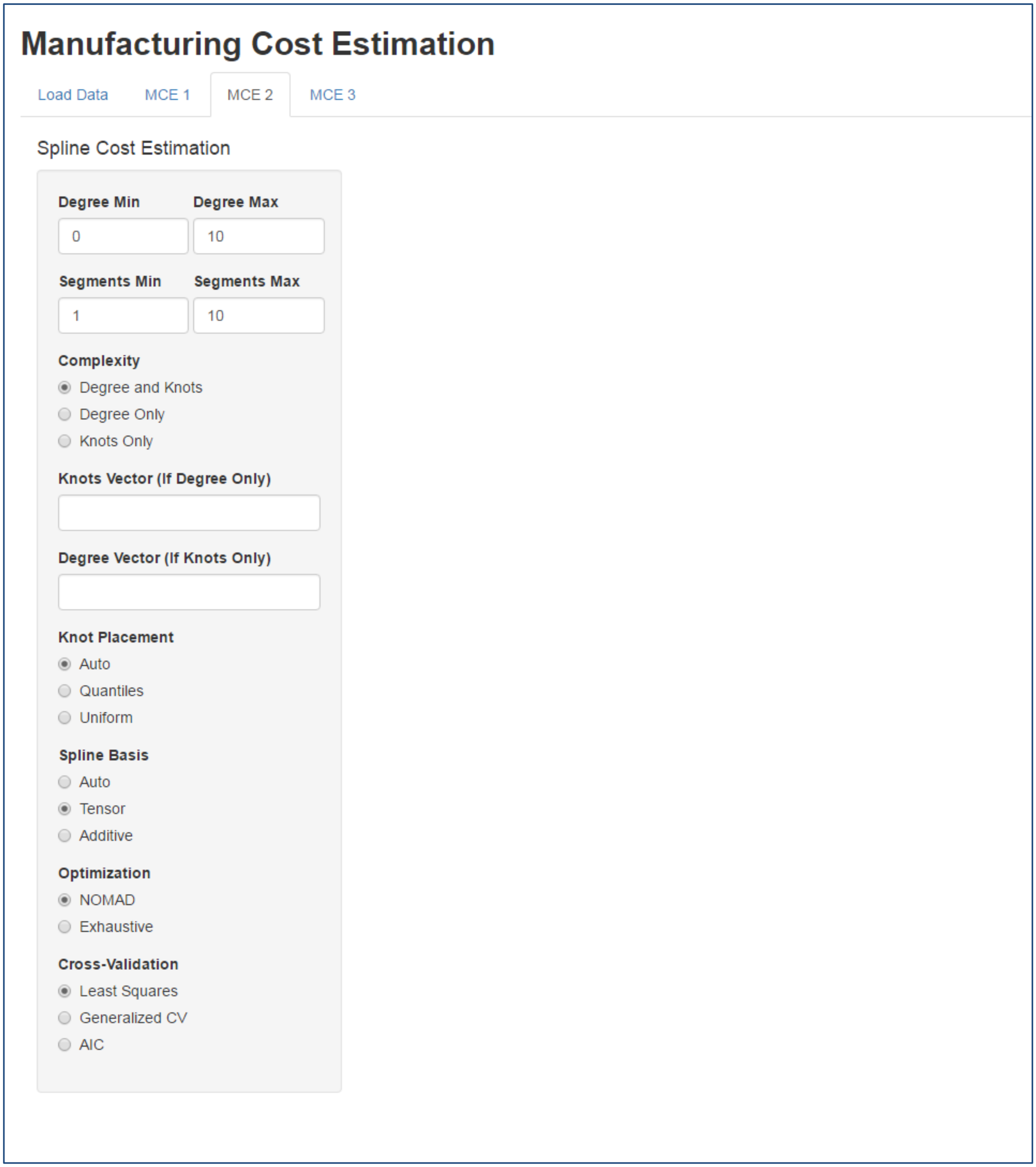
C-Index, Gamma and Silhouette Width Graphs from k = 2 to k = 20 for Determining Number of Clusters

Performance of MCE 1: Actual vs. Predicted Cost

Processed Dataset and Predicted Values (y_hat)

	x1	x2	x3	x4	x5	x6	x7	x8	x9	z1	z2	y	y_hat	cluster
1	0.02	0.43	1	0.00	0.00	0.10	0.16	0.19	0.01	DIN	Tin	0.61	0.61	1.00
52	0.10	0.57	2	0.50	0.04	0.78	0.50	0.52	0.39	Standard	Tin	9.94	9.94	1.00
53	0.00	0.43	1	0.00	0.00	0.07	0.16	0.19	0.01	Standard	Tin	1.15	0.72	1.00

Figure 4.32: Clustering based cost estimation (MCE 1) application tab after analysis



Manufacturing Cost Estimation

Load Data | MCE 1 | MCE 2 | MCE 3

Spline Cost Estimation

Degree Min: 0
Degree Max: 10

Segments Min: 1
Segments Max: 10

Complexity
- ● Degree and Knots
- ○ Degree Only
- ○ Knots Only

Knots Vector (If Degree Only):

Degree Vector (If Knots Only):

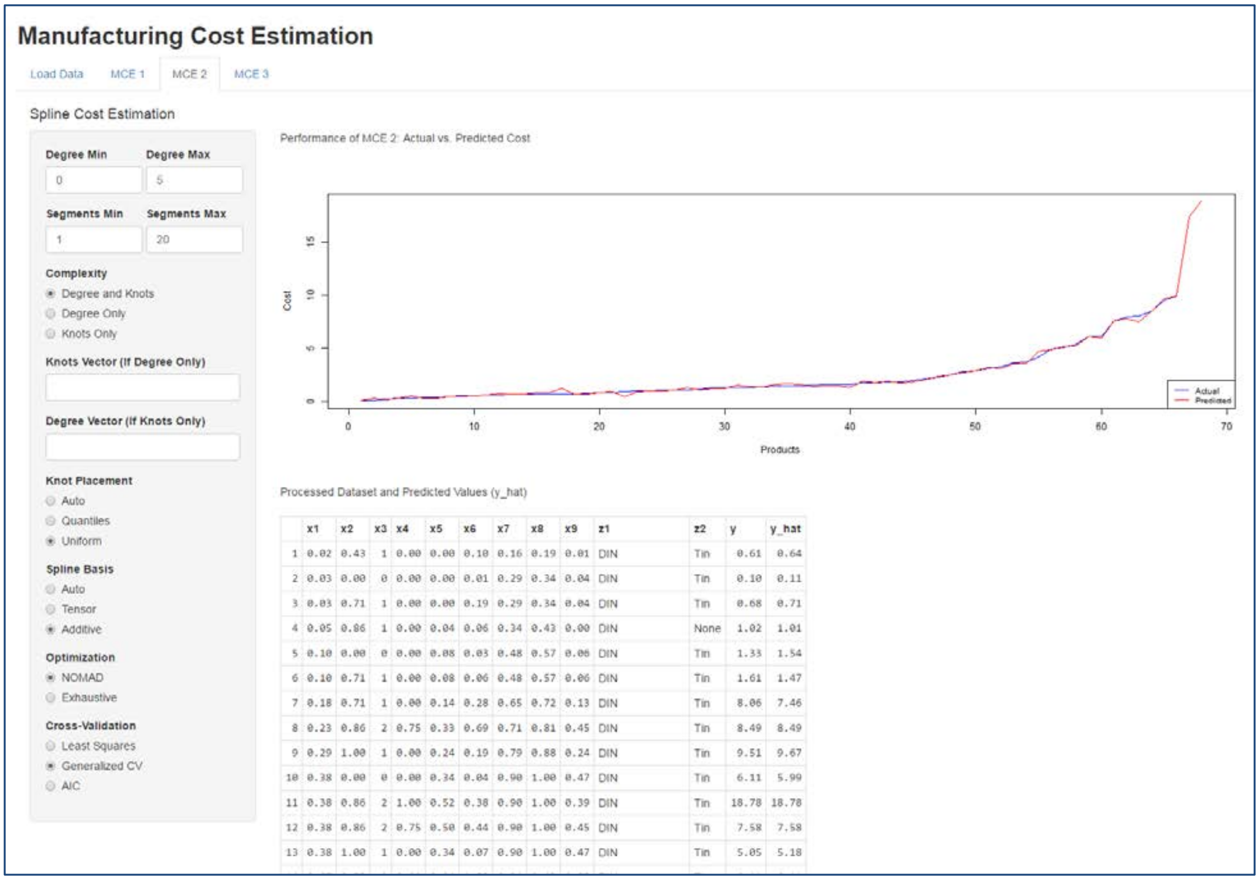
Knot Placement
- ● Auto
- ○ Quantiles
- ○ Uniform

Spline Basis
- ○ Auto
- ● Tensor
- ○ Additive

Optimization
- ● NOMAD
- ○ Exhaustive

Cross-Validation
- ● Least Squares
- ○ Generalized CV
- ○ AIC

Figure 4.33: Splines based cost estimation (MCE 2) application tab in the user interface



Manufacturing Cost Estimation

Spline Cost Estimation

Degree Min: 0
Degree Max: 5

Segments Min: 1
Segments Max: 20

Complexity
● Degree and Knots
○ Degree Only
○ Knots Only

Knots Vector (If Degree Only)

Degree Vector (If Knots Only)

Knot Placement
○ Auto
○ Quantiles
● Uniform

Spline Basis
○ Auto
○ Tensor
● Additive

Optimization
● NOMAD
○ Exhaustive

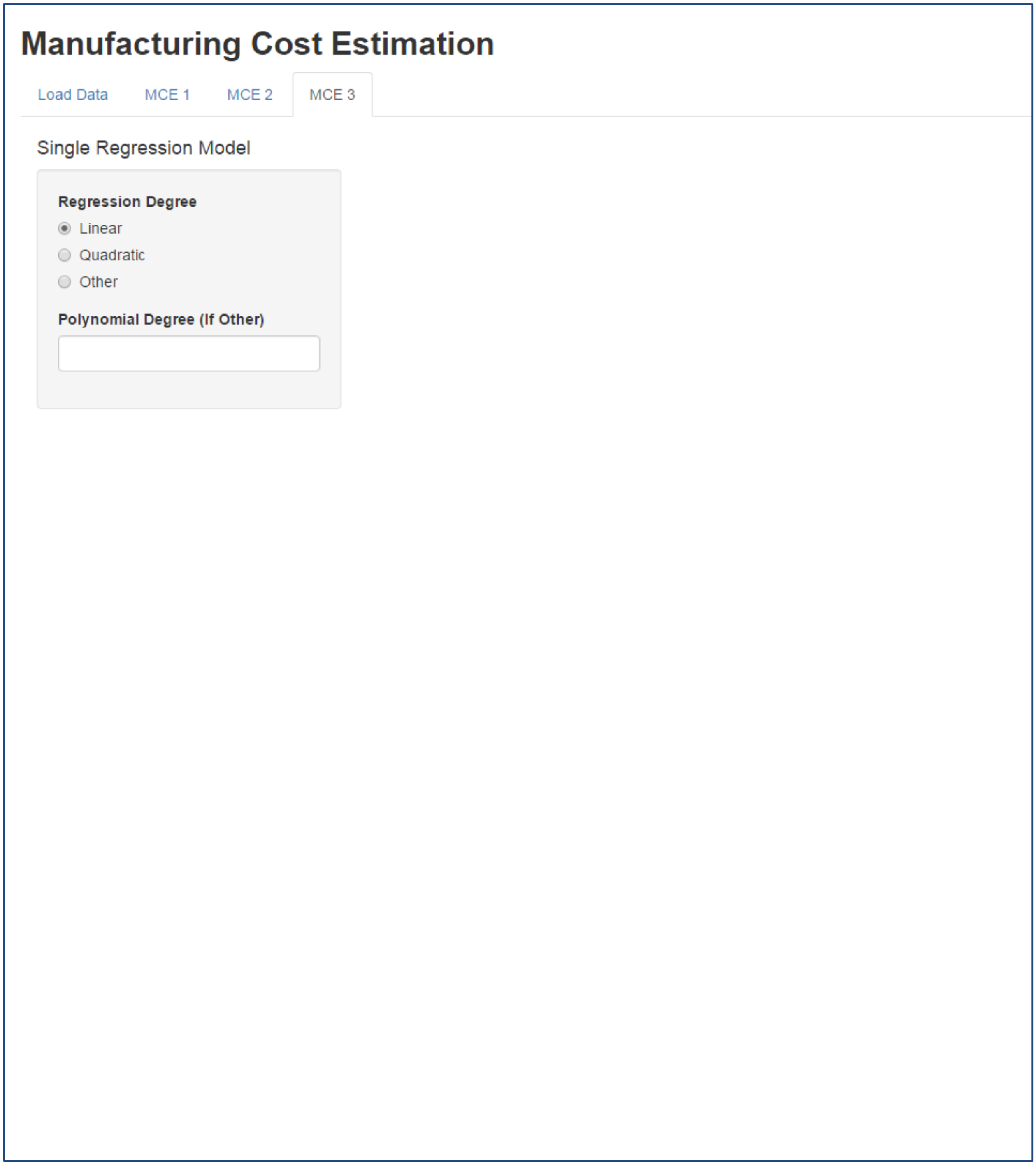
Cross-Validation
○ Least Squares
● Generalized CV
○ AIC

Performance of MCE 2: Actual vs. Predicted Cost

Processed Dataset and Predicted Values (y_hat)

	x1	x2	x3	x4	x5	x6	x7	x8	x9	z1	z2	y	y_hat
1	0.02	0.43	1	0.00	0.00	0.10	0.16	0.19	0.01	DIN	Tin	0.61	0.64
2	0.03	0.00	0	0.00	0.00	0.01	0.29	0.34	0.04	DIN	Tin	0.10	0.11
3	0.03	0.71	1	0.00	0.00	0.19	0.29	0.34	0.04	DIN	Tin	0.68	0.71
4	0.05	0.86	1	0.00	0.04	0.06	0.34	0.43	0.00	DIN	None	1.02	1.01
5	0.10	0.00	0	0.00	0.08	0.03	0.48	0.57	0.06	DIN	Tin	1.33	1.54
6	0.10	0.71	1	0.00	0.08	0.06	0.48	0.57	0.06	DIN	Tin	1.61	1.47
7	0.18	0.71	1	0.00	0.14	0.28	0.65	0.72	0.13	DIN	Tin	8.06	7.46
8	0.23	0.86	2	0.75	0.33	0.69	0.71	0.81	0.45	DIN	Tin	8.49	8.49
9	0.29	1.00	1	0.00	0.24	0.19	0.79	0.88	0.24	DIN	Tin	9.51	9.67
10	0.38	0.00	0	0.00	0.34	0.04	0.90	1.00	0.47	DIN	Tin	6.11	5.99
11	0.38	0.86	2	1.00	0.52	0.38	0.90	1.00	0.39	DIN	Tin	18.78	18.78
12	0.38	0.86	2	0.75	0.50	0.44	0.90	1.00	0.45	DIN	Tin	7.58	7.58
13	0.38	1.00	1	0.00	0.34	0.07	0.90	1.00	0.47	DIN	Tin	5.05	5.18

Figure 4.34: Splines based cost estimation (MCE 2) application tab after analysis



Manufacturing Cost Estimation

Single Regression Model

Regression Degree

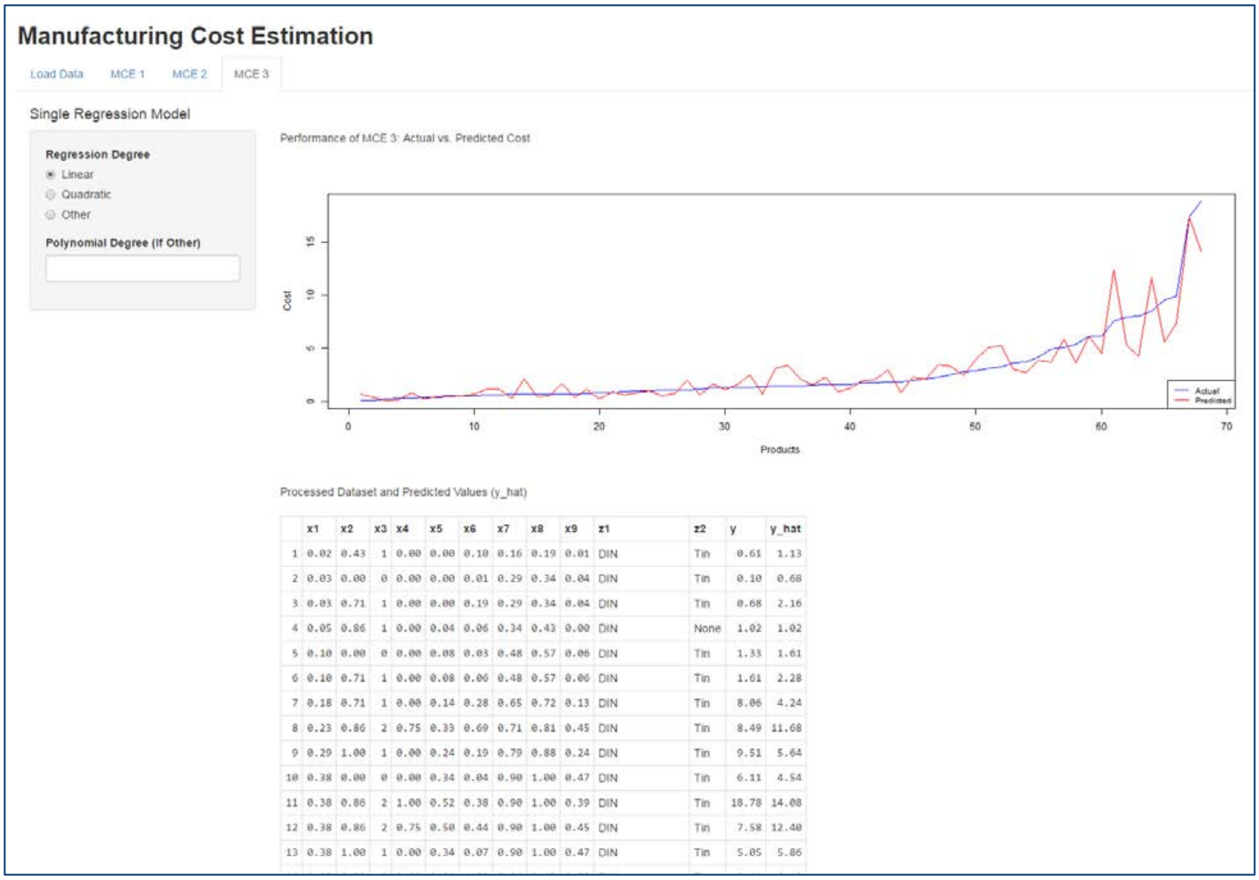
● Linear

○ Quadratic

○ Other

Polynomial Degree (If Other)

Figure 4.35: Polynomial regression based cost estimation (MCE 3) application tab in the user interface



Regression Degree

● Linear

○ Quadratic

○ Other

Polynomial Degree (If Other)

Processed Dataset and Predicted Values (y_hat)

	x1	x2	x3	x4	x5	x6	x7	x8	x9	z1	z2	y	y_hat
1	0.02	0.43	1	0.00	0.00	0.10	0.16	0.19	0.01	DIN	Tin	0.61	1.13
2	0.03	0.00	0	0.00	0.00	0.01	0.29	0.34	0.04	DIN	Tin	0.10	0.68
3	0.03	0.71	1	0.00	0.00	0.19	0.29	0.34	0.04	DIN	Tin	0.68	2.16
4	0.05	0.86	1	0.00	0.04	0.06	0.34	0.43	0.00	DIN	None	1.02	1.02
5	0.10	0.00	0	0.00	0.08	0.03	0.48	0.57	0.06	DIN	Tin	1.33	1.61
6	0.10	0.71	1	0.00	0.08	0.06	0.48	0.57	0.06	DIN	Tin	1.61	2.28
7	0.18	0.71	1	0.00	0.14	0.28	0.65	0.72	0.13	DIN	Tin	8.06	4.24
8	0.23	0.86	2	0.75	0.33	0.69	0.71	0.81	0.45	DIN	Tin	8.49	11.68
9	0.29	1.00	1	0.00	0.24	0.19	0.79	0.88	0.24	DIN	Tin	9.51	5.64
10	0.38	0.00	0	0.00	0.34	0.04	0.90	1.00	0.47	DIN	Tin	6.11	4.54
11	0.38	0.86	2	1.00	0.52	0.38	0.90	1.00	0.39	DIN	Tin	18.78	14.08
12	0.38	0.86	2	0.75	0.50	0.44	0.90	1.00	0.45	DIN	Tin	7.58	12.40
13	0.38	1.00	1	0.00	0.34	0.07	0.90	1.00	0.47	DIN	Tin	5.05	5.86

Figure 4.36: Polynomial regression based cost estimation (MCE 3) application tab after analysis

Chapter 5

Conclusions and Future Research

5.1 Conclusions

In this research, we investigated ways of using clustering methods and splines to predict the manufacturing cost of a product without actually manufacturing it. The accuracies of the two methodologies presented in this work are assessed in comparison to each other and also a simple regression model with the absence of clustering approaches. The main concerns behind this research are to predict the manufacturing cost of a product without dealing with arbitrary assignments of statistical distributions to cost related attributes and without stating strong assumptions about parametric distributions. In real production systems often a variety of products are being manufactured under a single facility roof. Therefore, over a diverse product family, establishing only a simple accurate estimation model is challenging and even questionable. This motivated us grouping products according to their design features, common manufacturing operations or some other factors by dividing the whole database of products into neighborhoods. Then for each group of products (clusters), a cost estimation model is developed to predict the manufacturing cost of new product with using the cluster specific model. Also, we investigated whether implementation of a spline approach provides accurate predictions of manufacturing costs where splines constitute a reasonable approach for the nonparametric estimation of manufacturing cost functions.

In real applications, the most likely scenario is to have a set of data about products and their cost related attributes where these attributes are mixed categorical and numerical. Unfortunately, not every clustering algorithm is compatible with all types of data. As a result, we used the best possible clustering algorithm that can fulfill the requirements, namely k-medoids, and implemented an approach to handle mixed type of data. We developed cluster-specific regression models to predict the manufacturing cost of a new unique product design. We also suggested using another methodology, namely non-parametric regression splines, to overcome the limitations of cost estimation efforts for continuous predictors. In this process, implementation of kernel weighting in regression splines as suggested by Racine et al. [98] helped us to establish robust piecewise spline models. Finally, the leave-one-out cross-validation is integrated into the cost estimation module to assess the quality of these two methodologies for predicting the cost of prospective products.

Our research intends to make the following contributions to practice:

Contribution 1. *We implemented the first manufacturing cost estimation approach using clustering techniques.*

We are the first to introduce a manufacturing costs estimation approach for mixed type of variables using clustering methods. To distinguish our work from others, this research is the first attempt in the manufacturing cost estimation literature that investigates the possibility of using partitioning methods to establish the price of a product before it is actually manufactured. Over a diverse product family, establishing only a single and simple but accurate estimation model is challenging. This motivated us to divide the whole database of products into neighborhoods until these neighborhoods become sufficiently homogenous. Using statistical terminology, we call these neighborhoods, groups or clusters. We developed cost estimation regression models for each cluster. Since every current and historical product can be represented as points in multidimensional space with respect to their design attributes, we investigated in which cluster a new product falls. After assigning the new product to the best cluster, we used the cluster-specific estimation model to predict its manufacturing cost.

Contribution 2. *We implemented the first manufacturing cost estimation approach using splines.*

In the literature, there are a number of different statistical models devised to predict the manufacturing cost of a product. However, using spline models has not been done since there is a limitation for building models based on only continuous predictors. However, in reality it is very likely to encounter mixed categorical and numeric design attributes in manufacturing processes. Categorical design attributes are inseparable or, in other terms, integral to estimation efforts due to their important contribution to the actual cost. In this research, we captured the complex relationships of categorical and numeric design attributes using categorical regression splines proposed by Racine et al. [98]. The recent developments in kernel weighting for categorical predictors along with tensor product regression splines have enabled us to deploy an efficient spline model to accommodate the existence of mixed categorical and numeric variables. It is expected that spline models will be used with accelerated frequency as a better substitute for linear regression by the non-parametric cost estimation community.

Contribution 3. *In the presence of categorical and numeric design attributes, we implemented a simple heuristic procedure to determine the appropriate number of clusters when there is no prior knowledge about the number of product groups.*

Existence of categorical and numeric design attributes in a dataset also constrains the applicability of a very common and effective procedure described in the user manual of SAS [79]. The procedure aims to derive the appropriate number of clusters by monitoring the consensus among three statistics, namely Sarle's cubic clustering criterion, pseudo T^2, and Calinski and Harabasz's pseudo F. In the case of mixed design attributes, this approach is incompatible due to its limitation to continuous variables only. For k-medoids clustering, Rousseeuw's silhouette width is recommended to check the novelty of the clustering content and the number of clusters. Since there is no definitive rule for the number of clusters, restricting the study only using one statistic may be inferior. Implementing a heuristic which is similar to the one devised in SAS is a logical approach because a consensus among multiple powerful statistics gives more leverage to a single value.

Recall Milligan and Cooper's investigation that assesses the performances of 30 statistics, the top five statistics are pseudo F, pseudo T^2, C-index, Gamma and F-ratio, respectively [90]. In this list, only two out of five statistics are applicable for our general cost estimation approach. We combined these two statistics along with the silhouette width and implemented a procedure to establish the appropriate number of clusters. We monitor plots of the C-index, Gamma and silhouette width, where local peaks of the Gamma and silhouette width combined with local troughs of the C-index is our choice for the number of clusters.

5.2 Future Work

As we discussed earlier in the literature review section, most clustering techniques require a similarity measure to operate. The selection of similarity measure is important since a poorly chosen one may not locate the clusters correctly. Assignment of the similarity measure is usually made based on expertise in the application area where the requirements of the specific real-world problem is considered. Higher discrimination power is desired by practitioners to isolate the clusters that eventually maximize the inter-cluster variability relative to the within-cluster variability. No existing similarity measure can operate with the presence of mixed numeric and categorical design attributes in the same dataset. To overcome this issue, a robust alternative approach has been illustrated recently with the *k*-prototypes clustering algorithm where a weighted summation of the Euclidean distance (for continuous variables) and the simple matching coefficient (for categorical variables) is used as the objective function [26]. Unfortunately, the objective function attempts to integrate a quadratic expression with a linear expression based on a weighting factor assigned according to an ad hoc logic. Instead, consolidating two linear or two quadratic dissimilarity expressions might be mathematically more meaningful. Our approach combines numeric and categorical design attributes with a linear expression, namely Gower's index. Even though using linear expressions for each variable type seems consistent, there might be a lack of discrimination power. A direction of future research should be in developing a comprehensive similarity measure that demonstrates high inter-cluster variability (high discrimination power) while being able to handle mixed categorical and numeric design attributes.

The ultimate goal of the clustering algorithms is to find the optimum cluster contents. However, most of these algorithms are heuristics and may end up finding effective but suboptimal groups. A deterministic model such as a mixed integer programming model can be implemented to obtain the optimal cluster results for the cost estimation models building phase. Furthermore, an adaptive heuristic such as simulated annealing or genetic algorithm may be used for clustering to improve the clustering analysis. Additionally, extending this research to the *k*-prototypes algorithm mentioned in the literature review chapter is a promising alternative to handle the existence of both numeric and categorical design attributes.

Another possible future research is to focus on using regression trees. A regression tree is a variant of decision trees where real-valued functions are approximated. The regression tree methodology may be generalized to manufacturing cost estimation since it is not limited to continuous predictors only. That is, using mixed numeric and categorical data is allowed in the regression tree building process.

In this research, irrelevant predictors are removed from the MCE 1, MCE 2 and MCE 3 models as described in Chapters 3 and 4. Future research may consider the information gain criterion when deciding on the inclusion of a candidate predictor in the cost estimation model. This approach could yield an information rich but parsimonious set of cost drivers to be used in predicting cost using our clustering or spline approach.

As a final point, a more efficient splines approach should be developed to handle larger datasets since calculating tensor products for the piecewise spline functions and optimization of bandwidths are computationally costly operations. According to our observations on the application problems, the current approach might have difficulties for high dimensional datasets running on computers with lower memory and processor specifications. To fulfill the expectations about improved accuracy in cost estimation with an effective quick price quote delivery, a compressive but less resource consuming splines approach could be developed.

Bibliography

[1] B. Efron, "The jackknife, the bootstrap and other resampling plans," in *CBMS-NSF Regional Conference Series in Applied Mathematics*, Philadelphia, 1983.

[2] A. C. Davison and D. V. Hinkley, Bootstrap Methods and their Applications, Cambridge: Cambridge University Press, 1998.

[3] "Oxford English Dictionary," [Online]. Available: http://www.oed.com/view/Entry/187182?rskey=gSsiYv&result=1#eid. [Accessed 12 5 2016].

[4] J. Epperson, "History of splines," *NA Digest*, vol. 98, no. 26, 1998.

[5] A. K. Jain and R. C. Dubes, Algorithms for Clustering Data, Upper Saddle River: Prentice Hall, 1988.

[6] R. A. Johnson and D. W. Wichern, Applied Multivariate Statistical Analysis, Upper Saddle River: Pearson, 2002.

[7] V. G. Sprindzhuk, "Dirichlet box principle," Encyclopedia of Mathematics, [Online]. Available: https://www.encyclopediaofmath.org/index.php/Dirichlet_box_principle. [Accessed 13 5 2016].

[8] J. S. Racine, Z. Nie and B. D. Ripley, "Package 'crs': Categorical regression splines," R Package version 0.15-24, https://github.com/JeffreyRacine/R-Package-crs/, 2014.

[9] L. Kaufmann and P. J. Rousseeuw, Finding Groups in Data, New York: John Wiley & Sons, 1990.

[10] A. Layer, E. T. Brinke, F. V. Houten, H. Kals and S. Haasis, "Recent and future trends in cost estimation," *International Journal of Computer Integrated Manufacturing,* vol. 15, no. 6, pp. 499-510, 2002.

[11] J. S. Dai, A. Niazi, S. Balabani and L. Seneviratne, "Product cost estimation: Technique classification and methodology review," *Journal of Manufacturing Science and Engineering,* vol. 128, no. 2, pp. 563-575, 2006.

[12] S. Cavalieri, P. Maccarrone and R. Pinto, "Parametric vs. neural network models for the estimation of production costs: A case study in the automotive industry," *International Journal of Production Economics,* vol. 91, no. 2, pp. 165-177, 2004.

[13] A. K. Jain, M. N. Murty and P. J. Flynn, "Data clustering: A review," *ACM Computing Surveys (CSUR),* vol. 31, no. 3, pp. 264-323, 1999.

[14] J. H. Ward Jr, "Hierarchical grouping to optimize an objective function," *Journal of the American Statistical Association,* vol. 58, no. 301, pp. 236-244, 1996.

[15] P. H. A. Sneath, "The application of computers to taxonomy," *Microbiology,* vol. 17, no. 1, pp. 201-226, 1957.

[16] L. L. McQuitty, "Hierarchical linkage analysis for the isolation of types," *Educational and Psychological Measurement,* vol. 20, no. 1, pp. 55-67, 1960.

[17] R. R. Sokal and P. H. A. Sneath, Principles of Numerical Taxonomy, San Francisco: W. H. Freeman, 1963.

[18] T. Sørenson, "A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons," *Biologiske Skrifter,* vol. 5, pp. 1-34, 1948.

[19] R. R. Sokal and C. D. Michener, "A statistical method for evaluating systematic relationships," *University of Kansas Scientific Bulletin,* vol. 38, pp. 1409-1438, 1958.

[20] J. MacQueen, "Some methods for classification and analysis of multivariate observations," *Proceedings of Fifth Berkeley Symposium on Mathematical Statistics and Probability,* vol. 1, pp. 281-297, 1967.

[21] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *Journal of Cybernetics,* vol. 3, no. 3, pp. 32-57, 1973.

[22] J. C. Bezdek, R. Ehrlich and W. Full, "FCM: The fuzzy c-means clustering algorithm," *Computers & Geosciences,* vol. 10, no. 2, pp. 191-203, 1984.

[23] L. Kaufmann and P. Rousseeuw, "Clustering by means of medoids," in *Statistical Data Analysis Based on the L1-norm and Related Methods,* Amsterdam, Springer, 1987, pp. 405-416.

[24] Z. Huang, "Extensions to the k-means algorithm for clustering large data sets with categorical values," *Data Mining and Knowledge Discovery,* vol. 2, no. 3, pp. 283-304, 1998.

[25] Z. He, X. Xu and S. Deng, "Attribute value weighting in k-modes clustering," *Expert Systems with Applications,* vol. 38, no. 12, pp. 15365-15369, 2011.

[26] Z. Huang, "Clustering large data sets with mixed numeric and categorical values," *Proceedings of the 1st Pacific-Asia Conference on Knowledge Discovery and Data Mining Conference,* pp. 21-34, 1997.

[27] N. R. Pal, J. C. Bezdek and E. C.-K. Tsao, "Generalized clustering networks and Kohonen's self-organizing scheme," *IEEE Transactions on Neural Networks*, vol. 4, no. 4, pp. 549-557, 1993.

[28] D. R. Jones and M. A. Beltramo, "Solving partitioning problems with genetic algorithms," *Proceedings of the Fourth International Conference on Genetic Algorithms*, pp. 442-449, 1991.

[29] S. Z. Selim and K. S. Al-Sultan, "A simulated annealing algorithm for the clustering problem," *Pattern Recognition,* vol. 24, no. 10, pp. 1003-1008, 1991.

[30] M. Omran, A. Salman and A. P. Engelbrecht, "Image classification using particle swarm optimization," *Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning,* vol. 2002, pp. 370-374, 2002.

[31] N. Monmarché, M. Slimane and G. Venturini, "On improving clustering in numerical databases with artificial ants," in *Advances in Artificial Life,* Heidelberg, Springer, 1999, pp. 626-635.

[32] K. S. Al-Sultan, "A tabu search approach to the clustering problem," *Pattern Recognition,* vol. 28, no. 9, pp. 1443-1451, 1995.

[33] C. B. Lucasius, A. D. Dane and G. Kateman, "On k-medoid clustering of large data sets with the aid of a genetic algorithm: background, feasiblity and comparison," *Analytica Chimica Acta,* vol. 282, no. 3, pp. 647-669, 1993.

[34] M. K. Ng and J. C. Wong, "Clustering categorical data sets using tabu search techniques," *Pattern Recognition,* vol. 35, no. 12, pp. 2783-2790, 2002.

[35] U. Boryczka, "Ant clustering algorithm," *Intelligent Information Systems, Kluwer Academic Publishers,* pp. 377-386, 2008.

[36] W. L. G. Koontz, P. M. Narendra and K. Fukunaga, "A branch and bound clustering algorithm," *IEEE Transactions on Computers,* vol. 24, no. 9, pp. 908-915, 1975.

[37] C. T. Zahn, "Graph-theoretical methods for detecting and describing gestalt clusters," *IEEE Transactions on Computers,* vol. 100, no. 1, pp. 68-86, 1971.

[38] W. B. Frakes and R. Baeza-Yates, Information Retrieval: Data Structures and Algorithms, Englewoods Cliff: Prentice Hall, 1992.

[39] J. H. Wolfe, "Pattern clustering by multivariate mixture analysis," *Multivariate Behavioral Research,* vol. 5, no. 3, pp. 329-350, 1970.

[40] B. S. Everitt, "A finite mixture model for the clustering of mixed-mode data," *Statistics & Probability Letters,* vol. 6, no. 5, pp. 305-309, 1988.

[41] I. Moustaki and I. Papageorgiou, "Latent class models for mixed variables with applications in Archaeometry," *Computational Statistics and Data Analysis,* vol. 48, no. 3, pp. 659-675, 2005.

[42] A. Strehl and J. Ghosh, "Cluster ensembles - A knowledge reuse framework for combining multiple partitions," *The Journal of Machine Learning Research,* vol. 3, pp. 583-617, 2003.

[43] Z. He, X. Xu and S. Deng, "Clustering mixed numeric and categorical data: A cluster ensemble approach". Patent CoRR abs/cs/0509011, 2005.

[44] J. Rubin, "Optimal classification into groups: an approach for solving the taxonomy problem," *Journal of Theoretical Biology,* vol. 15, no. 1, pp. 103-144, 1967.

[45] S. C. Ducker, W. T. Williams and G. N. Lance, "Numerical classification of the Pacific forms of Chlorodesmis (Chlorophyta)," *Australian Journal of Botany,* vol. 13, no. 3, pp. 489-499, 1965.

[46] D. H. Colless, "An examination of certain concepts in phenetic taxonomy," *Systematic Biology,* vol. 16, no. 1, pp. 6-27, 1967.

[47] J. C. Gower, "A general coefficient of similarity and some of its properties," *Biometrics,* vol. 27, no. 4, pp. 857-871, 1971.

[48] S. M. Emran and N. Ye, "Robustness of canberra metric in computer intrusion detection," in *Proceedings of IEEE Workshop on Information Assurance and Security,* West Point, 2001.

[49] R. J. Bray and J. T. Curtis, "An ordination of the upland forest communities of southern Wisconsin," *Ecological Monographs,* vol. 27, no. 4, pp. 325-349, 1957.

[50] L. Bobrowski and J. C. Bezdek, "c-means clustering with the L1 and L∞ norms," *IEEE Transactions on Systems, Man and Cybernetics,* vol. 21, no. 3, pp. 545-554, 1991.

[51] M. B. Eisen, P. O. Brown, P. T. Spellman and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proceedings of the National Academy of Sciences,* vol. 95, no. 25, pp. 14863-14868, 1998.

[52] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks,* vol. 16, no. 3, pp. 645-678, 2005.

[53] P. Karl, "Notes on regression and inheritance in the case of two parents," *Proceedings of the Royal Society of London,* vol. 58, no. 347-352, pp. 240-242, 1895.

[54] J. Zubin, "A technique for measuring like-mindedness," *The Journal of Abnormal and Social Psychology,* vol. 33, no. 4, pp. 508-516, 1938.

[55] D. J. Rogers and T. T. Tanimoto, "A computer program for classifying plants," *Science,* vol. 132, no. 3434, pp. 1115-1118, 1960.

[56] P. Jaccard, "Nouvelles recherches sur la distribution florale," *Société Vaudoise des Sciences Naturelles Bulletin,* vol. 44, pp. 223-270, 1908.

[57] L. R. Dice, "Measures of the amount of ecologic association between species," *Ecology,* vol. 26, no. 3, pp. 297-302, 1945.

[58] I. J. Schoenberg, "Contributions to the problem of approximation of equidistant data by analytic functions," *Quarterly Applied Mathematics,* vol. 4, no. 2, pp. 112-141, 1946.

[59] I. J. Schoenberg, Cardinal Spline Interpolation, Philadelphia: Society for Industrial and Applied Mathematics, 1973.

[60] R. Levien and C. H. Séquin, "Interpolating splines: Which is the fairest of them all?," *Computer-Aided Design and Applications,* vol. 6, no. 1, pp. 91-102, 2009.

[61] E. V. Shikin and A. I. Plis, Handbook on Splines for the User, Boca Raton: CRC Press, 1995.

[62] C. H. Reinsch, "Smoothing by spline functions," *Numerische Mathematik*, vol. 10, no. 2, pp. 177-183, 1967.

[63] C. H. Reinsch, "Smoothing by spline functions. II," *Numerische Mathematik*, vol. 16, no. 5, pp. 451-454, 1971.

[64] H. B. Curry and I. J. Schoenberg, "On spline distributions and their limits: The Polya distribution functions," *Bulletin of the American Mathematical Society,* vol. 53, no. 1114, 1947.

[65] C. de Boor, A Practical Guide to Splines, New York: Springer-Verlag, 1976.

[66] P. H. C. Eilers and B. D. Marx, "Flexible smoothing with B-splines and penalties," *Statistical Science,* vol. 11, no. 2, pp. 89-102, 1996.

[67] L. L. Schumaker, Spline Functions: Basic Theory, New York: Cambridge University Press, 2007.

[68] T. Hastie and R. Tibshirani, "Generalized additive models," *Statistical Science,* vol. 1, no. 3, pp. 297-318, 1986.

[69] J. H. Friedman, "Multivariate adaptive regression splines," *The Annals of Statistics,* vol. 19, no. 1, pp. 1-67, 1991.

[70] J. H. Friedman, "Estimating functions of mixed ordinal and categorical variables using adaptive splines, Technical Report No: 108," Department of Statistics, Stanford University, 1991.

[71] L. Angelis and I. Stamelos, "A simulation tool for efficient analogy based cost estimation," *Empirical Software Engineering,* vol. 5, no. 1, pp. 35-68, 2000.

[72] A. Lee, C. H. Cheng and J. Balakrishnan, "Software development cost estimation: integrating neural network with cluster analysis," *Information & Management,* vol. 34, no. 1, pp. 1-9, 1998.

[73] Z. Xu and T. M. Khoshgoftaar, "Identification of fuzzy models of software cost estimation," *Fuzzy Sets and Systems,* vol. 145, no. 1, pp. 141-163, 2004.

[74] J. S. Pahariya, V. Ravi and M. Carr, "Software cost estimation using computational intelligence techniques," *World Congress on Nature and Biologically Inspired Computing*, pp. 849-854, 2009.

[75] K. Michaud, J. Messer, H. K. Choi and F. Wolfe, "Direct medical costs and their predictors in patients with rheumatoid arthritis," *Arthritis and Rheumatism*, vol. 48, no. 10, pp. 2750-2762, 2003.

[76] D. Almond, K. Y. Chay and D. S. Lee, "The cost of low birth weight," *The Quarterly Journal of Economics*, vol. 120, no. 3, pp. 1031-1084, 2005.

[77] G. W. Carides, J. F. Heyse and B. Iglewicz, "A regression-based method for estimating mean treatment cost in presence of right-censoring," *Biostatistics*, vol. 1, no. 3, pp. 299-313, 2000.

[78] S. C. Valverde and D. B. Humphrey, "Predicted and actual costs from individual bank mergers," *Journal of Economics and Business*, vol. 56, pp. 137-157, 2004.

[79] "SAS/STAT 9.2 User's Guide," SAS Institute Inc., Cary, 2008.

[80] G. W. Milligan and M. C. Cooper, "An examination of procedures for determining the number of clusters in a data set," *Psychometrika*, vol. 50, no. 2, pp. 159-179, 1985.

[81] T. Calinski and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics-theory and Methods*, vol. 3, no. 1, pp. 1-27, 1974.

[82] R. O. Duda, P. E. Hart and D. G. Stork, Pattern Classification, New York: John Wiley & Sons, 2012.

[83] E. C. Dalrymple-Alford, "Measurement of clustering in free recall," *Psychological Bulletin*, vol. 74, no. 1, pp. 32-34, 1970.

[84] F. B. Baker and L. J. Hubert, "Measuring the power of hierarchical cluster analysis," *Journal of the American Statistical Association,* vol. 70, no. 349, pp. 31-38, 1975.

[85] L. A. Goodman and W. H. Kruskal, "Measures of association for cross classifications," *Journal of the American Statistical Association,* vol. 49, no. 268, pp. 732-764, 1954.

[86] E. M. L. Beale, Cluster Analysis, London: Scientific Control Systems, 1969.

[87] W. S. Sarle, "Cubic Clustering Criterion (SAS Technical Report A-108)," SAS Institute Inc., Cary, 1983.

[88] B. S. Everitt, S. Landau, M. Leese and D. Stahl, Cluster Analysis, Chichester: John Wiley & Sons, 2010.

[89] J. M. Lattin, J. D. Carroll and P. E. Green, Analyzing Multivariate Data, Pacific Grove: Thomson Brooks/Cole, 2003.

[90] M. C. Cooper and G. W. Milligan, The Effect of Measurement Error on Determining the Number of Clusters in Cluster Analysis, Berlin: Springer, 1988.

[91] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics,* vol. 20, pp. 53-65, 1987.

[92] E. Dimitriadou, S. Dolnicar and A. Weingessel, "An examination of indexes for determining the number of clusters in binary data sets," *Psychometrika,* vol. 67, no. 1, pp. 137-159, 2002.

[93] D. L. Davies and D. W. Bouldin, "A cluster seperation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vols. PAMI-1, no. 2, pp. 224-227, 1979.

[94] R. Tibshirani, G. Walther and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *Journal of the Royal Statistical Society,* vol. 63, no. 2, pp. 411-423, 2001.

[95] W. J. Krzanowski and Y. T. Lai, "A criterion for determining the number of groups in a data set using sum-of-squares clustering," *Biometrics,* vol. 44, no. 1, pp. 23-34, 1988.

[96] J. A. Hartigan, Clustering Algorithms, New York: John Wiley & Sons, 1975.

[97] A. P. Reynolds, G. Richards, B. de la Iglesia and V. J. Rayward-Smith, "Clustering rules: A comparison of partitioning and hierarchical clustering algorithms," *Journal of Mathematical Modelling and Algorithms,* vol. 5, no. 4, pp. 475-504, 2006.

[98] S. Ma, J. S. Racine and L. Yang, "Spline regression in the presence of categorical design predictors," *Journal of Applied Econometrics,* vol. 10, no. 5, pp. 705-717, 2014.

[99] Q. Li and R. J. S., Nonparametric Econometrics: Theory and Practice, Princeton, NJ: Princeton University Press, 2007.

[100] J. Aitchison and C. Aitken, "Multivariate binary discrimination by the kernal method," *Biometrica,* vol. 63, no. 3, pp. 413-420, 1976.

[101] Q. Li and J. Racine, "Cross-validated local linear nonparametric regression," *Statistica Sinica,* vol. 14, no. 2, pp. 485-512, 2004.

[102] Z. Nie and J. S. Racine, "The crs package: Nonparametric regression splines for continuous and categorical predictors," *The R Journal 4.2,* pp. 48-56, 2012.

[103] I. DiMatteo, C. R. Genovese and R. E. Kass, "Bayesian curve-fitting with free-knot splines," *Biometrika,* vol. 88, no. 4, pp. 1055-1071, 2001.

[104] D. Ruppert, "Selecting the number of knots for penalized splines," *Journal of Computational and Graphical Statistics,* vol. 11, no. 4, pp. 735-757, 2012.

[105] M. Stone, "An asymptotic equivalence of choice of model by cross-validation and Akaike's criterion," *Journal of the Royal Statistical Society, Series B,* vol. 39, no. 1, pp. 44-47, 1977.

[106] G. Wahba, Spline Models for Observational Data, Society for Industrial and Applied Mathematics: Philadelphia, PA, 1990.

[107] S. Ma and J. S. Racine, "Additive regression splines with irrelevant categorical and continuous regressors," *Statistica Sinica,* vol. 23, no. 2, pp. 515-541, 2013.

[108] P. G. Hall and J. S. Racine, "Infinite order cross-validated local polynomial regression," *Journal of Econometrics,* vol. 185, no. 2, pp. 510-525, 2015.

[109] C. Audet, S. Le Digabel and C. Tribes, "NOMAD user guide, Technical Report G-2009-37," Les cahiers du GERAD, 2009.

[110] P. Craven and G. Wahba, "Smoothing noisy data with spline functions," *Numerische Mathematik,* vol. 31, no. 4, pp. 377-403, 1978.

[111] C. M. Hurvich, J. S. Simonoff and C. Tsai, "Smoothing parameter selection in nonparametric regression using an improved Akaike information criterion," *Journal of the Royal Statistical Society, Series B,* vol. 60, no. 2, pp. 271-293, 1998.

[112] M. Quenouille, "Approximate tests of correlation in time-series," *Journal of the Royal Statistical Society,* vol. 11, no. 1, pp. 68-84, 1949.

[113] M. R. Chernick, Bootstrap Methods: A Practitioner's Guide, New York: John Wiley & Sons, 1999.

[114] B. Efron and R. J. Tibshirani, An Introduction to the Bootstrap, New York: Chapman & Hall, 1993.

[115] B. Efron, "Nonparametric estimates of standard error: the jack-knife, the bootstrap, and other methods," *Biometrika,* vol. 68, pp. 589-599, 1981.

[116] D. W. K. Andrews and M. Buchinsky, "A three-step method for choosing the number of bootstrap repetitions," *Econometrica,* vol. 68, no. 1, pp. 23-51, 2000.

[117] J. W. Tukey, "Bias and confidence in not quite large samples," *Annals of Mathematical Statistics,* vol. 29, no. 2, pp. 614-614, 1958.

[118] A. E. Smith and A. K. Mason, "Cost estimation predictive modeling: regression versus neural network," *The Engineering Economist,* vol. 42, no. 2, pp. 137-161, 1997.

[119] W. Chang, "Package 'shiny': Web application framework for R," R Package version 0.13.2, https://github.com/rstudio/shiny/, 2016.

[120] L. V. Fausett, Fundamentals of Neural Networks, Englewood Cliffs: Prentice Hall, 1994.

[121] C.-H. Cheng, "A branch and bound clustering algorithm," *IEEE Transactions on Systems, Man and Cybernetics,* vol. 25, no. 5, pp. 895-898, 1995.

[122] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller and E. Teller, "Equation of state calculations by fast computing machines," *The Journal of Chemical Physics,* vol. 21, no. 6, pp. 1087-1092, 1953.

[123] B. Widrow and M. E. Hoff, "Adaptive switching circuits," *1960 WESCON Convention Record Part IV,* pp. 96-104, 1960.

[124] Z. Huang and M. K. Ng, "A fuzzy k-modes algorithm for clustering categorical data," *IEEE Transactions on Fuzzy Systems,* vol. 7, no. 4, pp. 446-452, 1999.

[125] Z. Wang, J. Ji, W. Pang, C. Zhou and X. Han, "A fuzzy k-prototype clustering algorithm for mixed numeric and categorical data," *Knowledge-Based Systems*, vol. 30, pp. 129-135, 2012.

[126] M. R. Anderberg, Cluster Analysis for Applications, New York: Academic Press, 1973.

[127] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi, "Optimization by simulated anealing," *Science*, vol. 220, no. 4598, pp. 671-680, 1983.