Designing an Intelligent Decision Support System for Effective Negotiation Pricing

DOI:
10.1016/j.dss.2017.02.003

Document Version
Accepted author manuscript

Citation for published version (APA):
Fu, X., Zeng, X., Luo, X., Wang, D., Xu, D., & Fan, Q-L. (2017). Designing an Intelligent Decision Support System for Effective Negotiation Pricing: A Systematic and Learning Approach. *Decision Support Systems*, *96*, 49-66. https://doi.org/10.1016/j.dss.2017.02.003

Published in:
Decision Support Systems

Designing an Intelligent Decision Support System for Effective Negotiation Pricing: A Systematic and Learning Approach

Xin Fu[a], Xiao-Jun Zeng[b], Xin(Robert) Luo[c], Di Wang[d], Di Xu[a,*], Qing-Liang Fan[e]

[a] *Department of Management Science, School of Management, Xiamen University, Xiamen, 361005, China*
[b] School *of Computer Science, University of Manchester, M13 9PL, U.K*
[c] *Anderson School of Management, The University of New Mexico, 1924 Las Lomas NE, MSC05 3090, Albuquerque, NM 87131, United States*
[d] *EBTIC, Khalifa University, Abu Dhabi*
[e] *Wang Yanan Institute for Studies in Economics (WISE), Xiamen University, Xiamen, 361005, China*

Abstract

Automatic negotiation pricing and differential pricing aim to provide different customers with products/services that adequately meet their requirements at the "right" price. This often takes place with the purchase of expensive products/services and in the business-to-business context. Effective negotiation pricing can help enhance a company's profitability, balance supply and demand, and improve the customer satisfaction. However, determining the "right" price is a rather complex decision-making problem that puzzles pricing managers, as it needs to consider information from many constituents of the purchase channel. To further advance this line of research, this study proposes a systematic and learning approach that consists of three different types of fuzzy systems (FSs) to provide intelligent decision support for negotiation pricing. More specifically, the three FSs include: 1) a standard FS, which is a typical multiple inputs and single output FS that forms a mathematical mapping from the input space to the output space; 2) an SFS-SISOM, which is a linear fuzzy inference model with a single input and a single output module; and 3) a hierarchical FS, which consists of several FSs in a hierarchical manner to perform fuzzy inference. To address the existing problem of a standard FS suffering from the high-dimensional problem with a large number of influential factors, a generalized type of FS (named hierarchical FS), including its mathematical models and suitability for tackling the negotiation pricing problem, is introduced. In particular, a proof-of-concept prototype system that integrates these three FSs is also developed and presented. From a system design perspective, this artifact provides immense potential and flexibility for end users to choose the most suitable model for the given problem. The utility and effectiveness of this proposed system is illustrated and examined by three experimental datasets that vary from dimensionality and data coverage. Moreover, the performances of three different approaches are compared and discussed with respect to some important properties of decision support systems (DSSs).

Keywords:
Decision Support Systems, Hierarchical Fuzzy Systems, Negotiation Pricing

1. Introduction

Intelligent negotiation pricing and differential pricing are prevalent in retailing and business-to-business (B2B), and it is playing an increasingly important role in electronic businesses. In traditional retailing, it is natural to provide standard products and services to all customers at a standard price. In recent years, with the rapid development of marketing science, it is well recognized that different marketing and pricing strategies should be applied to different segments of customers, because differential pricing can substantially enhance organizational profitability and improve customer satisfaction. This emerging phenomenon often occurs in the purchase of expensive products/services (e.g., cars, houses, and systems). In particular, tailor-made products/services require a sales process to negotiate and settle the final price with customers individually. Furthermore, the outcomes of negotiation pricing and differential pricing often have a long-term impact on the organizational supply chain relationship and the reputation of business in the B2B arena. In supply chain management, for instance, price negotiation takes place during annual price reviews, thereby providing an opportunity for suppliers to adjust prices in response to recent changes in costs, as well as the customer relationships [20, 21]. Many examples of this scenario can be found in the industry. For example, broadband wireless service providers may provide differentiated services to their customers with a variety of customized prices [9]; theaters provide personalized prices of last-minute tickets depending on the time remaining and the customer's location [15].

To increase the effectiveness and efficiency of negotiation pricing, companies often delegate certain degrees of pricing authority to sales representatives who have direct contact with and better knowledge of customers. The agreed price will then be approved by the pricing manager. However, the salespeople may have different sales skills and preferences, as empirical findings have revealed that sales representatives might offer too many price concessions in order to ensure the order [30]. Therefore, from the perspective of pricing managers, disclosing the reservation price to all human agents is unfavorable. In essence, decision-making for negotiation pricing is a rather complicated process because it needs to consider information from a plethora of organizational dimensions to identify the right price for the company. It is worth noting that negotiation pricing does not support the entire dynamic and interactive process of price negotiation. Instead, it supports the most fundamental problem in the price negotiation process, which helps pricing managers to identify the right price when offering the unique product/service to each individual customer.

Prior research efforts have been devoted to providing decision support for negotiation pricing through different techniques, including game theory (GT) [9, 34], neural networks [2, 20], expert systems [4], case-based reasoning [15], and fuzzy logic [12, 17, 18]. Notwithstanding, certain drawbacks and challenges still exist with these approaches because some hypothetical assumptions are difficult to achieve in real scenarios, and the assessment of utility functions is not feasible in many studies due to heterogeneity and incidental

[*] Corresponding author, Email: dxu@xmu.edu.cn

parameters problems. Expert systems are heavily reliant on expert knowledge and/or static negotiation strategies. Hence, capturing knowledge in manual ways for the resultant system would be less flexible and inefficient to handle the dynamic changes and new cases in negotiations. Similarly, the neural network approach [2, 20] is limited in its interpretability, and its derived results are questionable for the end users.

These extant approaches should resort to adequate and precise information provided by negotiation parties for decision making. However, uncertain information is often inherent in the dynamic negotiation environments of real negotiation scenarios. The involvement of uncertain information is an extremely imperative but often under-addressed issue in negotiation pricing. Fuzzy set theory [32] is well regarded as a useful tool for handling uncertain information, and preserving transparency and interpretability in modeling. Recently, several research efforts [12, 17, 18, 34] have employed this technique to provide decision support for negotiation pricing. In essence, the majority of the existing approaches either focus on the representation of involved uncertain attributes by using linguistic terms, or employ expert knowledge to build static fuzzy rule-based systems for reasoning. Yet, slight changes in negotiation conditions may necessitate substantial expert interventions to modify the corresponding rules to reflect the new conditions. Moreover, it is difficult to validate and assess the quality of knowledge captured from experts. Therefore, it is more desirable to build a fuzzy system (FS) that would automatically learn from historical records to generate the fuzzy rules, rather than completely depending on external knowledge. Additionally, when the number of influential factors is large, a standard FS easily suffers from the problem of dimensionality, since the number of required modeling parameters and fuzzy rules exponentially increases with the number of involved attributes. As such, dealing with uncertain information within negotiation pricing is of particular interest to researchers and practitioners. Given the different features (e.g., dimensionality and data coverage) of the available historical dataset, choosing the most appropriate FSs is another crucial task faced by the end users of decision support systems (DSSs).

In an effort to remedy these pressing issues and challenges, this study substantially extends the initial work of [8], and proposes a systematic and learning approach to provide decision support for negotiation pricing through FS theory. In essence, this study concerns bilateral negations on the price, and considers the negotiation pricing problem particularly from the seller's point of view to provide intelligent decision support for pricing managers. Given a set of historical records, mathematical relationships between influential factors and the proposed price will be built by both learning from the data itself and integrating expert knowledge. It is believed that the proposed model can be leveraged to better predict the will-to-pay and other reference prices (e.g., reservation price, target price, and initial price) for unforeseeable transactions. Beyond the simplified FS with a single input and a single output module (SFS-SISOM) that has been presented in [8], this work employs the hierarchical fuzzy system (HFS) approach, which is effective for tackling the dimensionality problem to build predictive models for negotiation pricing. The performances of three approaches (i.e., standard FS, SFS-SISOM, and HFS) are further compared and discussed from different perspectives, including interpretability, accuracy, generality, computational cost, and applicability. Moreover, a prototype of an intelligent DSS for negotiation pricing is designed and developed with an integration of these three fuzzy approaches. The IT artifact provides substantial potential and flexibility for end users to choose the most suitable model for the existing negotiation pricing problem.

In the general context of DSS, it is worth distinguishing the features of the proposed negotiation pricing DSS. Most extant studies have been devoted to finding, from a set of known feasible decisions, the best decision to fit with the given set of decision criteria or maximizing the known utility functions. Therefore, these studies can be regarded as DSS with complete and certain information, and the dominant approaches in DSS are decision analysis, ranking, and optimization methods. For the negotiation pricing DSS, however, the situation is very different because the best decision is attainable, and it is the highest price a customer is willing to pay. Yet, the actual problem is that the highest willing-to-pay price is unknown. Therefore, this type of decision-making problem requires DSS with uncertain information. As the distinguishing feature here is the information's uncertainty, the dominant approaches in most existing DSS are no longer applicable and a new approach is needed.

This paper is organized as follows: Section 2 reviews the related work of negotiation pricing decision support, and introduces the HFS and its challenges. In Section 3, a systematic approach employing three different types of FS is proposed and presented to provide decision support for intelligent negotiation pricing. The applicability and utility of the proposed approach is demonstrated and tested against three datasets in Section 4, and the derived results are compared and discussed in Section 5. In Section 6, a prototype system is developed and presented to provide a proof-of-concept for the proposed work. The final section concludes this paper and suggests further work directions.

2. Related work

2.1. Negotiation pricing decision support

Negotiation is a crucial activity in business, and is a complex, time consuming, and iterative process which might involve intensive information exchange and processing. In most business negotiations, price is the most important attribute. According to utility theory, in a multi-issue negotiation problems, a utility function can be employed to model price, such that multi-criteria can be converted and evaluated by one dimension [13]. Negotiation pricing aims to identify a mutually beneficial price for both seller and buyer. This can often be achieved by iterative one-to-one interactions. An effective negotiation pricing DSS may offer many benefits, such as maximizing the company's profitability, maintaining good customer relations, balancing supply and demand, and improving inventory management. Much work has been done recently on negotiation pricing decision support in the areas of influential factor identification, modeling negotiation behavior, user utility elicitation, and price prediction by employing various techniques. A summary of recent works on decision support for negotiation pricing is presented in Table 1.

Table 1: Review of literature drawing on negotiation pricing decision support

Author(s) & Year	Method(s)	Application domain and datasets	Aims and results
Giacomazzi et al. [9] (2012)	Game-theoretical approach	Simulated dataset in broadband wireless access	A bilateral negotiation algorithm is proposed to identify the price of the wireless bandwidth services.
Zhao and Wang [34] (2015)	Game-theoretical approach	Simulated dataset	This work investigates the pricing problem and service decisions in a supply chain under fuzzy uncertainty conditions. Expected value models are presented to identify the optimal pricing and service strategies under three different scenarios.
Carbonneau et al. [2] (2008)	Neural networks	Data obtained from conducting bilateral negotiation experiments in the Inspire system	This work presents a neural network approach to estimate the opponents' responses in negotiations.
Chan et al. [4] (2011)	Customer segmentation & expert systems (generate pricing rules by experts) & empirical analysis	Computers and peripherals online shop	This work applies customer segmentation, and offers more price discounts to more valuable customers. The experiment results reveal that the use of differential pricing and promotion strategies can improve the total sales without greatly reducing the company's revenues.
Lee et al. [15] (2012)	Case based reasoning & fuzzy cognitive map	Simulation experiments	This paper proposes an agent-based mobile negotiation framework for personalized pricing of last minutes theater tickets whose values are dependent on the time remaining until the performance and the locations of potential customers.
Moosmayer et al. [20] (2013)	Three layer neural network & questionnaire to collect data	Business-to-business contexts	This work explores the importances and significance of the influential factors of negotiation pricing. The obtained order of significance is: target price, initial price, walk-away price and the size of the relationships.
Wilken et al. [30] (2010)	Empirical method	Questionnaire survey	This work presents how pricing managers can influence salespeople's pricing behaviors through information control. A theoretical model is proposed and tested against the collected dataset.
Kolomvatsos et al. [12] (2014)	Fuzzy inference	Simulation experiments	This work proposes a fuzzy logic (FL) based approach for negotiation decision support for sellers. Initially, the seller can use FL reasoning to estimate negotiation time and rounds, and then reason based on user-defined FL rules to accept/reject decisions.
Lin et al. [17] (2011)	Fuzzy expert systems (user-defined fuzzy rules)	Internet auction	An agent-based price negotiation system is proposed in this work. Within the system, end users are allowed to customize their negotiation pricing strategies through user pre-defined fuzzy rules.
Lin and Chang [18] (2008)	Fuzzy approach and fixed/flexible quantity mixed integer programming (MIP) models	Numerical examples	This work develops a fuzzy approach to evaluating buyers. The results are employed to support the order selection and final pricing decisions. The more valuable buyers receive more discounts.

For identifying and analyzing the underlying relationships between influential factors and the negotiation price, the pioneering work of [10] investigates the relationship between car price and buyer attributes (e.g., income, race, and gender). Recently, more efforts have been conducted to explore how reference prices influence the final price in negotiations and analyze their importance (e.g., [1, 13, 20, 21, 22]). The reference prices can refer to the initial price, participants' reservation prices (i.e., the walk-away prices), and participants' target prices. Having recognized the importance of these reference prices, some works aim to employ both parties' references prices to accelerate and improve the negotiation process. A common approach is to require both the buyer and the seller to report their reference prices to a third-party before negotiations. If the zone of potential agreement is empty, the negotiation will not be launched (e.g., [13]). A classic approach to model the negotiation behavior of participants is the well-known GT, which emphasizes the interactions between buyers and sellers, and has been widely applied in bargaining problems [9, 24, 34]. Some of these game-theoretical models often assume that participants would behave rationally and strategically to optimize their negotiation outcome based on symmetrically available information. However, such assumptions are difficult to achieve in real scenarios, and limit the applicability of GT approaches for solving realistic negotiation problems. In principle, GT is a utility-based approach that relies on a mathematical concept of optimal convergence where both parties' utility functions are defined and optimized. However, the assessment of utility functions is a time-consuming and error-prone process.

The above approaches mainly rely on adequate and precise information for negotiation pricing decision support. In recent years, researchers have become increasingly recognizant of the importance of the uncertain information inherent in dynamic negotiation environments. For instance, it is not easy to precisely measure certain influential factors of negotiation pricing, such as customer values, customer's knowledge about the product, and the degree of product popularity. In literature, FSs have been widely regarded as appropriate tools for handling uncertain information, and they have been employed to deal with the negotiation pricing problem [12, 17, 18, 34]. The majority of the existing systems generally either employs linguistic terms to represent and fuzzify the involved uncertain attributes, or rely on expert knowledge and/or a static strategy to generate decision-making rules. However, such expert-defined and static fuzzy rules are rather limited. It has been reported in [26] that only when the problem domain is certain, small, and loosely coupled, can the knowledge be captured through manual methods such as interviews and observations. Also, these static fuzzy rules are inefficient to response the changes of negotiation conditions.

To sum up, providing efficient and intelligent decision support for negotiation pricing is not an easy task. This is evident in the aforementioned drawbacks and challenges of existing approaches. Applying computational intelligent methods to support negotiation pricing decision making is gaining increased interest. In particular, the approach, which employs computational intelligence and soft computing to discover negotiation pricing patterns and behaviors from historical data and human knowledge, seems appropriate to overcome the existing limitations and handle the problem at hand.

2.2. Hierarchical fuzzy systems

The hierarchical fuzzy system (HFS) was firstly proposed to control a steam generator's drum level [23]. It has been proved that the HFS is capable of approximating any non-linear function on a compact set to arbitrary accuracy [29]. As such, HFS has been applied, together with other machine learning techniques, to a variety of application areas (e.g., [14, 16, 19]). Besides its universal approximation, HFS is also capable of relieving the high-dimensional problem by reducing the number of required fuzzy rules [11, 29]. More specifically, a special case of an incremental hierarchical structure [5] and the Takagi-Sugeno-Kang (TSK) FS [25] was proposed in [29], and has been proved that the number of required fuzzy rules in HFS increases linearly with the number of involved input attributes, rather than exponential as in a standard FS. However, since the THEN part of TSK FS is a polynomial of the input attributes, more parameters are needed to achieve a good universal approximation capability. In [11], the outputs of the lower level sub-FSs are fed as the THEN part into the connected higher level sub-FS, such that more parameters are required to compute the THEN part of each sub-FS. Since both the above methods move the dimensional complexity of fuzzy rules from the IF part to the THEN part, the proposed hierarchical structures in [11, 29] still suffer from the curse of dimensionality in terms of the number of parameters.

To overcome the parameter dimensionality problems and enhance the transparency and interpretability, a standard FS is used in [33] to analyze the universal approximation of HFS. The results show that the HFS is superior to a standard FS by using fewer parameters and fuzzy rules in terms of achieving the same approximation accuracy. Hence, the proposed HFS in [33] is employed in this work. Unlike a standard FS, which uses a flat high-dimensional FS to model the given problem, the underlying mechanism of HFS is to group several low-dimensional sub-FSs (in the form of a standard FS) in a hierarchical manner to model the problem. A general structure of an HFS consists of several levels that jointly contribute to computing the final output. Each level can consist of sub-FSs and/or original input attributes. In general, the lowest level's sub-FSs receive the original input attributes, and then produce the outputs that feed to their upper level. Note that, the sub-FSs in the upper level not only can receive the outputs from its lower level sub-FSs, but also the original input attributes. The outputs from the same level's sub-FSs are then propagated as inputs to the upper level sub-FSs, until they reach the highest level to derive the final output.

It has been attested theoretically that any continuous function with a natural hierarchical structure can be modelled by HFSs with fewer fuzzy rules and parameters [33]. This indicates that the task can be divided into several sub-tasks, so it is easier to construct a matching hierarchical structure for the problem if it has a natural hierarchical structure. As such, it is conceived that HFSs are more suitable and effective to construct the system with a natural hierarchical structure. For the problem at hand, the pricing manager's advice and experience can be applied to decompose the influential factors into several categories. Figure 1 illustrates a possible way to group the sample influential factors. As depicted in Figure 1, a natural hierarchical structure exists in the negotiation pricing problem and the influential factors can be structurally classified. The sub-FSs in this hierarchical structure produce meaningful intermediate variables, such as customer-related, product-related, service-related, and company-related factors. The hierarchical structure can help pricing managers make a better decision, especially when the number of influential factors is large. The HFS aggregates the common influential factors into the same sub-FS, and generates a lower number of meaningful intermediate attributes. In so doing, the generated top-level fuzzy rules are more interpretable by pricing managers. Note that there is no supporting theory behind the classification of influential factors for price negotiation in the current work. The classification of influential factors is actually case driven, and it relies on expert knowledge to construct the hierarchical structure, rather than automatically generating the structure from any classification theory.

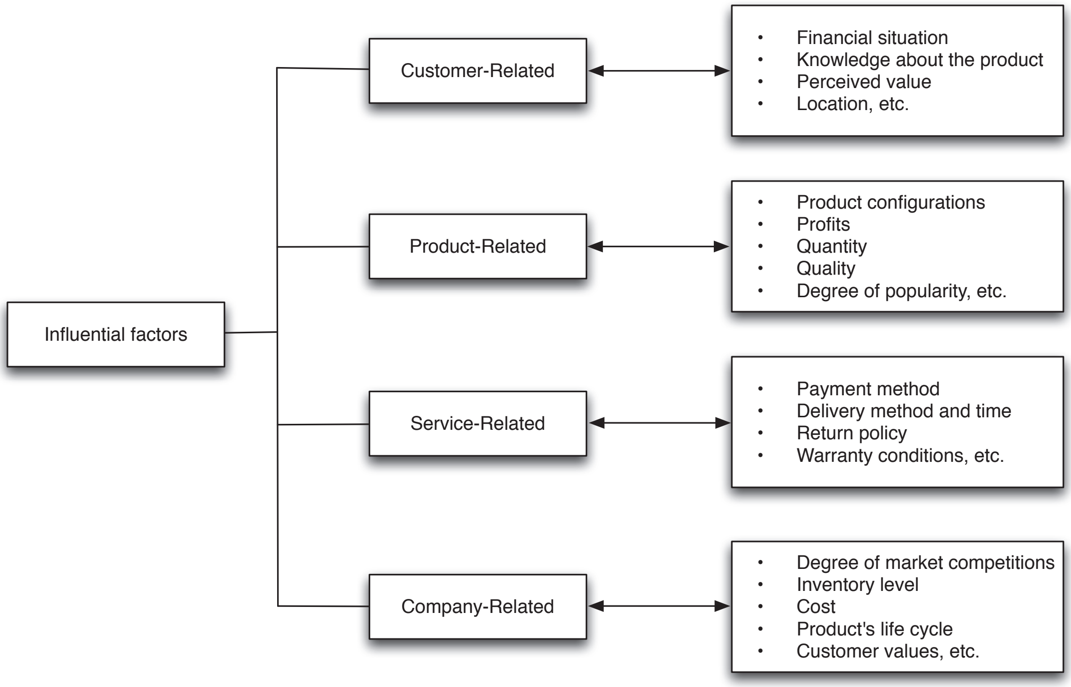


Figure 1: The factors influencing negotiation pricing

3. The proposed approach

Fuzzy set theory has become an increasingly prevalent methodology for representing and dealing with uncertain information, and has been successfully applied to many IT contexts, such as control engineering, soft computing, and intelligent DSSs [6, 7]. The merits of utilizing fuzzy sets for representing subjective expertise/knowledge, handling uncertainty, and modeling reasoning processes have been widely discussed and verified [31]. This study proposes a systematic and learning approach based on FSs to support negotiation pricing. One of the major contributions of this study is shedding light on investigating the features of three different types of FSs, namely standard FS, SFS-SISOM, and HFS, especially on their applicabilities in handling different negotiation pricing problems. As illustrated in Figure 2, the proposed approach is a data-driven method that aims to learn from historical transactions. These three types of FSs are thereby integrated into a single prototype system for pricing managers. This section presents the mathematical models of these FSs, as well as their associated learning algorithms. Since the standard FS and the SFS-SISOM have been reported in [8], this paper places a particular focus on the technical details of the HFS.

This study advances the understanding of the bilateral negotiation which involves the one-to-one pricing problem. The model aims to offer the right price to the right customer. Suppose that a series of influential factors is $X = (x_1, \cdots, x_n)$, and the offered price is y; then the given problem is to build the mathematical relationship between price y and the vector of influencing factors X. So it is denoted as:

$$y = P(X) = P(x_1, \cdots, x_n) \tag{1}$$

If the relationship $P(X)$ can be learned from historical data, then for each new customer, the right negotiation price can be estimated by using $P(X)$ based on this customer's values of the influential factors. As uncertainties are inherent in negotiation pricing, they may exist in the measure of influential factors, and/or the relationships between influential factors and the proposed price. Thus an FS approach becomes suitable and useful herein.

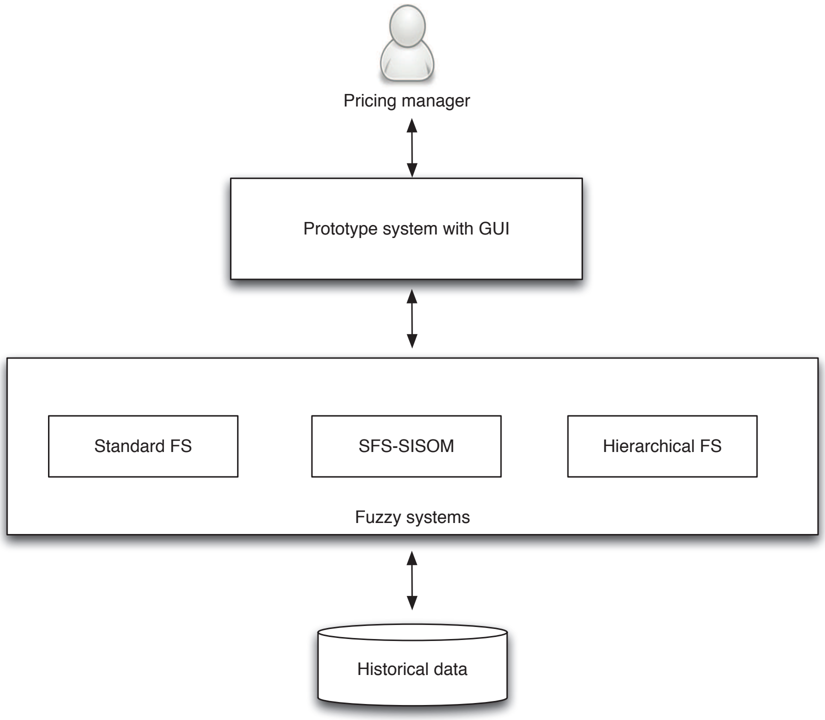


Figure 2: The proposed approach for negotiation pricing decision support

Note that the current study employs expert knowledge to manually group the input attributes in the HFS. It is possible to develop algorithms to allow for more automatic grouping of the input attributes, which may improve the prediction accuracy in some cases. Also, clustering is another plausible and useful approach to grouping the input attributes. One main reason for utilizing the manual grouping of the input attributes is due to the consideration of the application. Since most of the automatic grouping algorithms are driven by historical data, they are likely to lead to less relevant attributes being selected in the same group. Consequently, the intermediate variables will become meaningless and the resulting HFS will lose its interpretability. For example, to consider a DSS for house negotiation pricing, if the floor number of the apartment and local crime rate are classified in the same group by an algorithm, the corresponding intermediate variables will not be meaningful and the resulting HFS will just be a numerical model. On the other hand, by allowing a pricing manager to manually group input attributes, he/she may select the floor number and condition of an apartment in the same group, and include crime rate and closeness to the shopping center in the same group. In this manner, the corresponding intermediate variables can be regarded as the house facility index and location index, which are meaningful. Further, the resulting HFS will show the impacts of the house facility index and location index on the price of an apartment. As such, the manual grouping of input attributes is very useful from an application point of view. For this reason, it is perhaps a better choice than an algorithm grouping, since it enables a user to build a negotiation pricing model that fits his/her thinking and ensures interpretability and understandability.

3.1. Standard fuzzy system

A MISO standard FS (i.e., $f(X)$) builds a mathematical relationship between n input variables ($x_j \in U_j (j = 1, 2, \cdots, n)$, and $U_j \subset \mathbb{R}$) and one output variable ($y \in V$, and $V \subset \mathbb{R}$). Each variable consists of several fuzzy sets, in the inference process, fuzzy intersection operators are employed on a full combination of all input variables. Consequently, given n input variables , its complete fuzzy rule base requires $K = \prod_{j=1}^{n} N_j$ fuzzy rules, where N_j is the number of fuzzy sets of the j^{th} input variable. In order to enhance the model's transparency and reduce the number of required fuzzy rules and modeling parameters, the triangular membership functions that used in [8] and the center-average defuzzifier are employed in all three types of FSs in this study. Given a historical transaction with numeric inputs $X = (x_1, x_2, \cdots, x_n)$ and the output $\hat{y}$, the standard FS can be derived and represented as:

$$\hat{y} = f(X) = \sum_{i_1 \cdots i_n \in I} (\prod_{j=1}^{n} \mu_{i_j}^{j}(x_j)) y_{i_1 \cdots i_n} \tag{2}$$

where I is the index set, and it is represented as $I = \{i_1, \cdots, i_n | i_j = 1, 2, \cdots, N_j; j = 1, 2, \cdots, n\}$, $i_1 i_2 \cdots i_n$ is the index of fuzzy rule, and $\mu_{i_j}^{j}(x_j)$ is the membership degree of the i_j^{th} fuzzy set in the j^{th} input variable fired by the input value x_j.

In Equation (2), the parameters $y_{i_1 \cdots i_n}$ are supposed to be learned through a learning algorithm. The recursive least square (RLS) learning algorithm [28] is used in both standard FSs and SFS-SISOMs, technical details for implementing the RLS can be found in [8]. To start the learning process, a disturbance parameter (i.e., σ) needs to be identified, and it is often a large number (e.g., 100000). This parameter is introduced to tune and balance the fitting to the historical data and to the initial parameters. During the learning process, the parameters are iteratively updated with the aim to minimize the sum of errors. In the end, the recursive learning process terminates either by achieving a satisfied error rate (i.e., ε) or reaching the maximum iteration number (i.e., T).

3.2. SFS-SISOM

To address the problem of high dimensionality in standard FSs, a novel SFS-SISOM has been proposed in [8]. Rather than modeling a complete combination of all input variables and the output variable at a time, the SFS-SISOM breaks the complex model into several simple sub-models. The SFS-SISOM builds the linear relationship between the individual input variable and the output variable, respectively. Therefore, a fuzzy rule in the SFS-SISOM only contains one input variable and the output variable, and it only requires $S = \sum_{j=1}^{n} N_j$ (i.e., N_j is the number of fuzzy sets in the j^{th} input variable) fuzzy rules to cover the whole input space.

Mathematically speaking, the SFS-SISOM is formed by several SISO standard FSs, and the outputs of such SISO FSs contribute to the final output of the SFS-SISOM by carrying different important weights. Thus, the SFS-SISOM can be derived and represented as:

$$\hat{y} = \sum_{j=1}^{n} w_j \left(\sum_{i_j=1}^{N_j} \mu_{i_j}^{j}(x_j) y_{i_j}^{j} \right) = \sum_{j=1}^{n} \sum_{i_j=1}^{N_j} w_j y_{i_j}^{j} \mu_{i_j}^{j}(x_j) = \sum_{j=1}^{n} \sum_{i_j=1}^{N_j} C_{i_j}^{j} \mu_{i_j}^{j}(x_j), \quad (3)$$

where w_j represents the important weight of the j^{th} input variable/SISO standard FS, and $y_{i_j}^{j}$ is the central point of the corresponding fuzzy set of the output variable when given the i_j^{th} fuzzy set in the j^{th} input variable, x_j. These two parameters form a new parameter $C_{i_j}^{j}$ which is supposed to be learned from the RLS learning algorithm. The main steps of RLS are similar to the standard FS, with the main differences exist in the construction of input matrix, parameter vector, and output vector.

3.3. Hierarchical fuzzy system

3.3.1. The model of the hierarchical fuzzy system

The mathematical model of the standard FS has been introduced in Section 3.1, and the HFS consists of several standard FSs in a hierarchical manner. As shown in Figure 3, the a^{th} sub-FS in the h^{th} level (i.e., SFS_a^h) contains several intermediate attributes (i.e., $y_1^{h,a} \cdots, y_{n_{h,a}}^{h,a}$), and original attributes (i.e., $x_1^{h,a}, \cdots, x_{m_{h,a}}^{h,a}$). In this model, x refers to the original input attributes, while y refers to the output of the sub-FS. The output of SFS_a^h, denoted as $o^{h,a}$, is used as an intermediate attribute (i.e., $y_1^{h+1,b}$) of its upper level's standard FS, so that the SFS_a^h can be represented as:

$$o^{h,a} = y_1^{h+1,b} = f_{h,a}(y_1^{h,a}, \cdots, y_{n_{h,a}}^{h,a}, x_1^{h,a}, \cdots, x_{m_{h,a}}^{h,a})$$

$$= \frac{\sum_{j_1 j_2 \cdots j_{n_{h,a}} i_1 i_2 \cdots i_{m_{h,a}}} y^{h,a}_{j_1 j_2 \cdots j_{n_{h,a}} i_1 i_2 \cdots i_{m_{h,a}}} \prod_{k=1}^{n_{h,a}} \mu^{h,a,k}_{j_k}(y^{h,a}_k) \prod_{k=1}^{m_{h,a}} w^{h,a,k}_{i_k}(x^{h,a}_k)}{\sum_{j_1 j_2 \cdots j_{n_{h,a}} i_1 i_2 \cdots i_{m_{h,a}}} \prod_{k=1}^{n_{h,a}} \mu^{h,a,k}_{j_k}(y^{h,a}_k) \prod_{k=1}^{m_{h,a}} w^{h,a,k}_{i_k}(x^{h,a}_k)}. \tag{4}$$

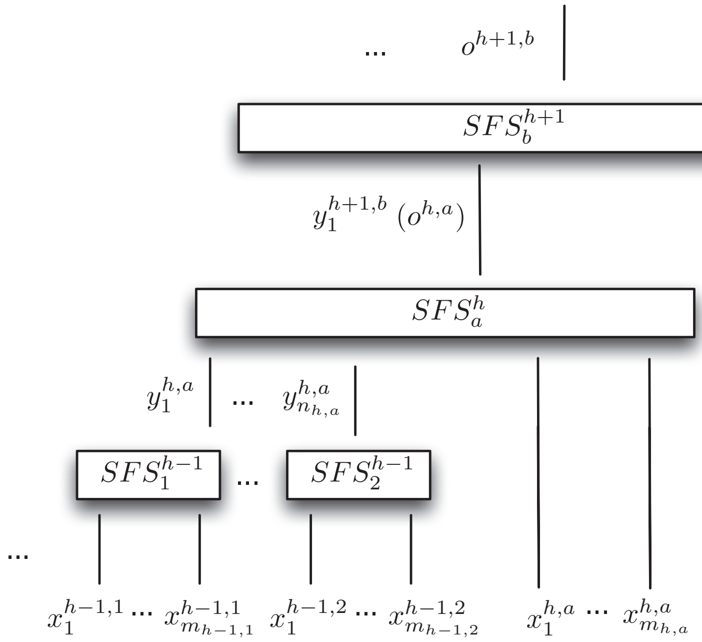

...
$o^{h+1,b}$
SFS_b^{h+1}
$y_1^{h+1,b}$ $(o^{h,a})$
SFS_a^h
$y_1^{h,a}$ ... $y_{n_{h,a}}^{h,a}$
SFS_1^{h-1} ... SFS_2^{h-1}
...
$x_1^{h-1,1}$ ... $x_{m_{h-1,1}}^{h-1,1}$ $x_1^{h-1,2}$ ... $x_{m_{h-1,2}}^{h-1,2}$ $x_1^{h,a}$ ... $x_{m_{h,a}}^{h,a}$

Figure 3: Mathematical model of HFS

In Equation (4), $n_{h,a}$ is the number of intermediate attributes of SFS_a^h and $m_{h,a}$ represents the number of the SFS_a^h's original input attributes. $y^{h,a}_{n_{h,a}}$ is the output of the sub-FS in the $h-1$ level, and it is used as the $n^{th}_{h,a}$ intermediate input attribute of the SFS_a^h; and $x^{h,a}_{m_{h,a}}$ is the $m^{th}_{h,a}$ original input attribute of the SFS_a^h. Also, $j_1 j_2 \cdots j_{n_{h,a}} i_1 i_2 \cdots i_{m_{h,a}}$ is the index of fuzzy rules in the SFS_a^h, and $\mu^{h,a,k}_{j_k}(y^{h,a}_k)$ stands for the corresponding output of the j_k^{th} fuzzy set of the k^{th} intermediate attribute of the SFS_a^h, while $w^{h,a,k}_{i_k}(x^{h,a}_k)$ is the output of the i_k^{th} fuzzy set in the k^{th} original input attribute of the SFS_a^h.

By employing the triangular membership functions in the HFS, Equation (4) can derive that:

$$\sum_{j_1 j_2 \cdots j_{n_{h,a}} i_1 i_2 \cdots i_{m_{h,a}}} \prod_{k=1}^{n_{h,a}} \mu^{h,a,k}_{j_k}(y^{h,a}_k) \prod_{k=1}^{m_{h,a}} w^{h,a,k}_{i_k}(x^{h,a}_k) = 1. \tag{5}$$

Equation (4) can be rewritten as:

$$o^{h,a} = \sum_{j_1 j_2 \cdots j_{n_{h,a}} i_1 i_2 \cdots i_{m_{h,a}}} \left[\prod_{k=1}^{n_{h,a}} \mu^{h,a,k}_{j_k}(y^{h,a}_k) \prod_{k=1}^{m_{h,a}} w^{h,a,k}_{i_k}(x^{h,a}_k) \right] y^{h,a}_{j_1 j_2 \cdots j_{n_{h,a}} i_1 i_2 \cdots i_{m_{h,a}}}. \tag{6}$$

The number of fuzzy rules in the HFS can be represented as:

$$H = \sum_{i=1}^{L} \left[\sum_{j=1}^{S_i} \left(\prod_{k=1}^{n_{i,j}} N_{i,j}^k \prod_{k=1}^{m_{i,j}} N_{i,j}^k \right) \right] \tag{7}$$

where L is the number of levels in the hierarchical structure and S_i is the number of sub-FSs in the i^{th} level. $n_{i,j}$ and $m_{i,j}$, respectively, represent the number of intermediate attributes and the number of original input attributes of the j^{th} sub-FS in the i^{th} level (i.e., SFS_j^i). $N_{i,j}^k$ is the number of fuzzy sets in the k^{th} input attribute (can be either an intermediate attribute or an original input attribute) of the SFS_j^i.

3.3.2. Gradient descent learning algorithm

Different from the standard FS and SFS-SISOM, the HFS employs the gradient descent learning algorithm to learn from historical records in this work. Although RLS is capable of finding the global optima, it is not applicable to HFS due to the existence of intermediate attributes. Since an HFS consists of several lower-level sub-FSs, the final output is nonlinearly dependent on the non-top sub-FSs, and the the gradient descent learning algorithm is employed herein to optimize the HFS's parameters. The objective function of the gradient decent algorithm only minimizes the error based on the current data instance (i.e., r):

$$e^r = \frac{1}{2}[f(X_r) - y_r]^2. \tag{8}$$

In [27], an improved gradient descent algorithm is proposed and it is adapted in this work. This algorithm aims to minimize the local error by updating the parameters and integrates a normalization step to handle the intermediate attributes in the HFS. Given an HFS, the final error back propagates from the top level to the lower levels, and the error derived from the upper level is used to update the parameters of the neighboring lower level. Assume t is the learning iteration index; the final error of a HFS is written as:

$$e_{L,1}(t) = o^{L,1}(t) - y(t), \tag{9}$$

where L is the total number of levels in an HFS, and $o^{L,1}(t)$ is the predicted output of the top-level sub-FS in the t^{th} learning iteration, while $y(t)$ is the target output provided in the given dataset. Given a hierarchical structure as shown in Figure 3, the error of SFS_b^{h+1} is propagated to SFS_a^h in the following way:

$$e_{h,a}(t) = e_{h+1,b}(t) \times \frac{\partial o^{h+1,b}(t)}{\partial o^{h,a}(t)}. \tag{10}$$

For simplicity, the level index (e.g., h and $h+1$) is omitted. Hence, Equation (10) is rewritten as:

$$e_a(t) = e_b(t) \times \frac{\partial o^b(t)}{\partial o^a(t)}. \tag{11}$$

Also, the HFS as shown in Equation (6) can be simplified as:

$$o^a = \sum_{j_1 j_2 \cdots j_{n_a} i_1 i_2 \cdots i_{m_a}} \left[\prod_{k=1}^{n_a} \mu_{j_k}^{a,k}(y_k^a) \prod_{k=1}^{m_a} w_{i_k}^{a,k}(x_k^a) \right] y^a_{j_1 j_2 \cdots j_{n_a} i_1 i_2 \cdots i_{m_a}}. \tag{12}$$

The fuzzy rule index $j_1 j_2 \cdots j_{n_a} i_1 i_2 \cdots i_{m_a}$ of the SFS_a^h is denoted as I_a, and $\prod_{k=1}^{n_a} \mu_{j_k}^{a,k}(y_k^a) \prod_{k=1}^{m_a} w_{i_k}^{a,k}(x_k^a)$ is denoted as $A_{I_a}^a$. Therefore, the output of the SFS_a^h is represented as:

$$o^a = \sum_{I_a} A_{I_a}^a y_{I_a}^a. \tag{13}$$

Equation (11) can then be represented as:

$$\begin{aligned} e_a(t) &= e_b(t) \times \frac{\partial \sum_{I_b} A_{I_b}^b(t) y_{I_b}^b(t)}{\partial o^a(t)} \\ &= e_b(t) \times \sum_{I_b} y_{I_b}^b(t) \frac{\partial A_{I_b}^b(t)}{\partial o^a(t)}, \end{aligned} \tag{14}$$

in which $\frac{\partial A_{I_b}^b(t)}{\partial o^a(t)} = \frac{A_{I_b}^b(t)}{\mu_{j_1}^{b,1}(y_1^b,t)} \times \frac{\partial \mu_{j_1}^{b,1}(y_1^b,t)}{\partial o^a(t)}$. As shown in Figure 3, the output of SFS_a^h (i.e., o^a) is the first intermediate attribute (i.e., y_1^b) of SFS_b^{h+1}. In addition, since $o^a = y_1^b$, Equation (14) can be written as:

$$\frac{\partial A_{I_b}^b(t)}{\partial o^a(t)} = \frac{A_{I_b}^b(t)}{\mu_{j_1}^{b,1}(o^a,t)} \times \frac{\partial \mu_{j_1}^{b,1}(o^a,t)}{\partial o^a(t)}. \tag{15}$$

According to the definition of triangular membership functions, it is derived that:

$$\frac{\partial \mu_{j_1}^{b,1}(o^a,t)}{\partial o^a(t)} = \begin{cases} 1 & o^a \in [c^{j_k-1}, c^{j_k}] \\ -1 & o^a \in [c^{j_k}, c^{j_k+1}] \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

where c^{j_k} is the central point of the j_k^{th} fuzzy set of the k^{th} attribute in SFS_b. This results in that the propagated error can be represented as:

$$e_a(t) = \begin{cases} e_b(t) \times \sum_{I_b} \frac{y^b_{I_b}(t)A^b_{I_b}(t)}{\mu^{b,1}_{j_1}(o^a,t)} & o^a \in [c^{j_k-1}, c^{j_k}] \\ -e_b(t) \times \sum_{I_b} \frac{y^b_{I_b}(t)A^b_{I_b}(t)}{\mu^{b,1}_{j_1}(o^a,t)} & o^a \in [c^{j_k}, c^{j_k+1}] \\ 0 & \text{otherwise.} \end{cases} \quad (17)$$

Given the previous iteration results (e.g., $y^a_{I_a}(t)$), the gradient descent learning algorithm iteratively updates the parameters of the current iteration by:

$$y^a_{I_a}(t+1) = y^a_{I_a}(t) - \lambda \times \frac{\partial o^a(t)}{\partial y^a_{I_a}(t)} \times e_a(t) \quad (18)$$

where λ is the learning rate parameter, and according to Equation (13),

$$\frac{\partial o^a(t)}{\partial y^a_{I_a}(t)} = \frac{\partial(\sum_{I_a} A^a_{I_a}(t) y^a_{I_a}(t))}{\partial y^a_{I_a}(t)} = A^a_{I_a}(t). \quad (19)$$

Therefore, Equation (18) can be rewritten as:

$$y^a_{I_a}(t+1) = y^a_{I_a}(t) - \lambda \times A^a_{I_a}(t) \times e_a(t). \quad (20)$$

To sum up, according to the mathematical model of the HFS (i.e., Equation (6)) and gradient descent algorithm (i.e., Equations (17) and (20)), the learning process of the HFS can be described as:

Step 1: Construction of the hierarchical structure

This task involves identifying the number of hierarchical levels, the number of sub-FSs in each level, the allocation of original input attributes, and the segmentation of the associated input attributes to construct a sub-FS. This is currently identified by the end user.

Step 2: Initialization of the learning parameters

The initial parameters for each sub-FS (e.g., $y^{h,a}_{I_a}(0)$ for SFS^h_a) can either be initialized by experts or use default values. These parameters will be iteratively updated in the learning process. Since the gradient descent learning algorithm may converge to a local optima, the choice of initial parameters becomes essential. If the initial parameters are close to the global optima, the algorithm stands a good chance of finding the global optima. Also, the learning rate parameter (i.e., λ) and two termination parameters (i.e., the maximum number of iteration T, and the error threshold ξ) need to be initialized herein.

Step 3: Update of the parameters by using the recursive gradient descent algorithm

Based on the given initial parameters, calculate the final output of the HFS and the error. After that, the error is back propagated from the top level to the lowest level by using Equation (17), and the error for each sub-FS (e.g., $e_{h,a}$) can be calculated, respectively. Then apply Equation (20) to update the parameters for each sub-FS.

Step 4: Normalization of the intermediate parameters

Intermediate attributes are a newly introduced problem in the HFS. In many cases, the intermediate attributes do not have semantical meanings. Thus, this may result in the universe of discourse of the derived intermediate attribute being unpredictable and difficult to pre-define. It has been proved in [27] that the approximation accuracy of the HFS is independent of the definition domain of intermediate attributes. Therefore, a method is proposed in [27] to overcome this problem by normalizing the domains of intermediate attributes into $[0, 1]$. The normalization process can be described as:

$$y_{I_a}^{h,a} = \frac{y_{I_a}^{h,a} - \min(y_I)}{\max(y_I) - \min(y_I)} \tag{21}$$

where $\min(y_I)$ is the minimal value of the intermediate attribute outputs within all non-top level sub-FSs, while $\max(y_I)$ is the maximum value. This ensures that the value of $y_{I_a}^{h,a}$ always lies within $[0, 1]$.

Step 5: Termination of the learning process

Similar to RLS, the recursive learning terminates either by reaching the maximum iteration number T, or reaching a predefined error threshold ξ. Calculate the overall error $E(t) = \frac{1}{2} \times \sum_{r=1}^{M}(o_r^{L,1}(t) - y_r(t))$; if $E(t) < \xi$ or $t > T$, the learning terminates. Otherwise, go back to Step 3 with $t = t + 1$ and set $r = 1$ to start a new iteration.

4. Empirical results

4.1. Experimental data

Three negotiation pricing related datasets, which vary from the number of instances to dimensionalities, were used in this study. A summary of these three datasets are presented in Table 2.

Table 2: Summary of experimental datasets

Dataset	No. of Input Attributes	No. of Instances	Input Attributes	Output Attribute
DS1 - MP3 Player Dataset	4	248	1. Usefulness; 2. Importance; 3. Budget; 4. Knowledge	Discount
DS2 - Boston House Dataset	11	499	1. Crime; 2. Education; 3. Population; 4. Air Quality; 5. Zone Proportion; 6. No. of Rooms; 7. Age; 8. Tax; 9. Convenience; 10. Distance; 11. Accessibility to Highway	Property value
DS3 - California Dataset	8	20,640	1. Median Income; 2. Housing Median Age; 3. Total Rooms; 4. Total Bedrooms; 5. Population; 6. Households; 7. Latitude; 8. Longitude	ln(median house value)

4.1.1. DS1 - MP3 player dataset

A questionnaire survey was conducted at a major British university to collect this dataset. The questionnaire consists of two parts: Part 1 introduces the background scenario, in which a MP3 company conducts a promotion to provide potential buyers with customized discounts, with the aim to boost the product sales; Part 2 contains two questions that capture participants' demographic attributes and four sets of questions that are associated with the four influential attributes (i.e., importance, budget, usefulness, and knowledge about the product) of product purchase. Participants were asked to answer the questions along a five-point Likert-type scale. Furthermore, if the product was deemed too expensive at its current price, the participants were asked to suggest a minimal numeric discount that would persuade them to reconsider of the purchase, and the specified discount was also regarded as the reservation price for pricing managers. The questionnaires were sent out by hand or electronically distributed via a mailing list to students and academic staff at the university. A total of 500 questionnaires were send out and 248 valid responses were returned and processed for further analysis.

4.1.2. DS2 - Boston House Dataset

This is a publicly available dataset that captures the residential property price for house sales in Boston in 1976.[1] In this dataset, 11 influential factors (see Table 2) were considered when identifying the house price. Given the house properties, the right negotiation price can be predicted by employing the proposed models. This dataset, in particular, was chosen to testify to the utility of the proposed approach in handling relatively high-dimensional problems.

4.1.3. DS3 - California Dataset

This dataset collects house information from the 1990 Census in California, and it is publicly available. [2] The dataset includes all block groups in California and each block group on average includes $1,425.5$ individuals living in a geographically compact area. The distances among the centroids of each block group as measured in latitude and longitude were computed. In addition, all the block groups reporting zero entities for the observed attributes were filtered out. In total, this dataset records 20,640 observations on eight independent attributes (i.e., median income, housing median age, total rooms, total bedrooms, population, households, latitude, and longitude) and the dependent variable is ln(median house value). Note that, this dataset does not report individual house information, as it instead captures the aggregated information for all block groups.

[1] http://lib.stat.cmu.edu/datasets/boston

[2] http://lib.stat.cmu.edu/datasets/houses.zip

4.2. Experimental results

The selection of initial values of learning parameters is a common problem in literature of machine learning. In this work, the initializations of learning parameters are mainly determined by the combination of expertise and pre-experiments with the goal to achieve the optimal prediction results (i.e., cross validation based on least APE and RMSE). In different FSs, different sets of initial values of learning parameters were tested in pre-experiments, and only the learning parameters that produce the optimal results are reported. More specifically, the initializations of the different learning parameters are explained as follows: 1) Error threshold: in order to monitor the best performance of different FSs, the error threshold is set to be very small (i.e., 0.01) for all experiments; 2) Number of iterations: within the pre-experiment, how the error changes with the increase of iterations can be observed and plotted. The iteration parameter that produces the stable prediction performance is selected; 3) Learning rate: it is taken within the range of $[0, 1]$ to avoid missing out the optima, as a small value (e.g., 0.02) of learning rate is selected in this work. A set of the combinations of iteration and learning rate parameters were tested in the pre-experiments; 4) Disturbance parameter: this parameter is used to construct the disturbance matrix in the RLS learning. As suggested by [28], a real large number (e.g., 100,000) is often used for the initial value of this parameter. The performances standard FSs and SFS-SISOMs are quite sensitive to this parameter. Therefore, several pre-experiments were conducted to select the appropriate disturbance parameter for different FSs.

4.2.1. DS1 - MP3 player dataset

In this experiment, three FSs were all employed to predict diverse situations, which varied in the number of training/testing instances (i.e., 50, 100, and 248 samples were used in different situations, respectively) and fuzzy attribute partitions. In the HFS, a hierarchical structure with 2 sub-systems and no free parameters are employed for all experiments. More specially, *Usefulness* and *importance* are grouped to construct a sub-standard FS, whereas *financial capability* and *knowledge* are grouped to construct another standard FS. The learning rate is set to 0.02 and the error threshold is set to 0.01. The number of iterations for the standard FS and the SFS-SISOM is set to 100. Since the input matrixes of the SFS-SISOM and the standard FS are different, the selection of the disturbance parameter (σ) value could be different. In this study, pre-experiments revealed that the standard FS and SFS-SISOM achieve optimal results when setting the $\sigma = 9999999$ and 99999, respectively. The reason for setting a larger σ for the standard FS is that it has more parameters, so a smaller $1/\sigma$ is needed to weigh down the impact of the initial parameters of the standard FS model to fit the data better.

The obtained results, as listed in Tables 3 through 5, reveal that all three FSs perform well when the divided sub-spaces are well covered by training samples. The derived results mirror some interesting findings: First, the standard FS suffers from the over-fitting problem when there is an insufficient number of training data samples (e.g., Exp_3, Exp_{10}, and Exp_{12} in Table 3). However, by increasing the number of training data samples to include all 248 instances being used, the over-fitting problem for the standard FS is removed gradually. Second, when the number of training data samples is sufficient to cover the divided sub-spaces, the standard FS, among the three FSs, performs the best in the sense of goodness of fitting (i.e., better accuracy for training data or smaller training error). However, the SFS-SISOM and the HFS both obtain better testing accuracy (i.e., smaller error for testing data) than the standard FS. This is verified, respectively, by the Exp_1, Exp_7, and Exp_{13} in Tables 3 and 4. Moreover, compared with the HFS, the SFS-SISOM can obtain similar testing accuracy with fewer fuzzy rules and a lower number of learning iterations. Thus, it can be concluded that the SFS-SISOM performs the best in this experiment.

Table 3: Results of using the standard FS in the MP3 player dataset [8]

Experiment	Training dataset(#)	Testing dataset(#)	# of rules	Partitions	Training APE(%)	Training RMSE	Testing APE(%)	Testing RMSE	Running time (second)
Exp_1	31 - 100 (70)	1 - 30 (30)	36	1,2,1,2	6.6283	1.1305	15.7317	1.7482	1.81
Exp_2	1 - 70 (70)	71 - 100 (30)	36	1,2,1,2	6.5245	0.7582	14.7936	2.4849	1.75
Exp_3	1 - 40 (40)	41 - 100 (60)	36	1,2,1,2	1.6495	0.1739	80.3334	7.8112	1.30
Exp_4	11 - 50 (40)	1 - 10 (10)	36	1,2,1,2	3.2771	0.3826	21.4055	2.5766	1.28
Exp_5	21 - 50 (30)	1 - 20 (20)	36	1,2,1,2	1.1137	0.2591	18.9492	2.8863	1.13
Exp_6	1 - 5; 21 - 50 (35)	6 - 20 (15)	36	1,2,1,2	1.2440	0.2680	21.3402	3.4829	1.20
Exp_7	51 - 248 (198)	1 - 50 (50)	36	1,2,1,2	8.3482	1.2396	12.6061	1.2643	3.73
Exp_8	51 - 248 (198)	1 - 50 (50)	144	2,3,2,3	3.4679	0.8347	42.5941	3.2094	107.39
Exp_9	1 - 30; 91 - 248 (188)	31 - 90 (60)	36	1,2,1,2	8.0517	1.0080	14.5664	2.0378	2.08
Exp_{10}	1 - 30; 91 - 248 (188)	31 - 90 (60)	144	2,3,2,3	2.5822	0.3812	63.6023	12.2093	101.11
Exp_{11}	1 - 120 (120)	121 - 248 (128)	36	1,2,1,2	8.6280	1.1464	12.5673	1.7509	2.57
Exp_{12}	1 - 120 (120)	121 - 248 (128)	144	2,3,2,3	1.7404	0.2948	24.8623	4.6594	111.01
Exp_{13}	1 - 150; 241 - 248 (158)	151 - 240 (90)	36	1,2,1,2	9.1615	1.2624	10.2230	1.4241	3.11
Exp_{14}	1 - 150; 241 - 248 (158)	151 - 240 (90)	144	2,3,2,3	3.3248	0.7825	24.8890	4.9452	120.21

Table 4: Results of using the SFS-SISOM in the MP3 player dataset [8]

Experiment	Training dataset(#)	Testing dataset(#)	# of rules	Partitions	Training APE(%)	Training RMSE	Testing APE(%)	Testing RMSE	Running time (second)
Exp_1	31 - 100 (70)	1 - 30 (30)	10	1,2,1,2	16.2000	1.7012	12.3035	1.2875	0.47
Exp_2	1 - 70 (70)	71 - 100 (30)	10	1,2,1,2	16.2004	1.6322	12.5001	1.5594	0.42
Exp_3	1 - 40 (40)	41 - 100 (60)	10	1,2,1,2	12.7127	1.0876	16.1389	1.9536	0.29
Exp_4	11 - 50 (40)	1 - 10 (10)	10	1,2,1,2	16.6560	1.3656	6.1107	1.0769	0.22
Exp_5	21 - 50 (30)	1 - 20 (20)	10	1,2,1,2	21.3886	1.4440	7.4940	1.1040	0.19
Exp_6	1 - 5; 21 - 50 (35)	6 - 20 (15)	10	1,2,1,2	20.4390	1.3806	8.5950	1.1521	0.20
Exp_7	51 - 248 (198)	1 - 50 (50)	10	1,2,1,2	9.8862	1.3894	12.1184	1.4428	0.77
Exp_8	51 - 248 (198)	1 - 50 (50)	14	2,3,2,3	7.9622	1.2033	13.0766	1.2628	0.94
Exp_9	1 - 30; 91 - 248 (188)	31 - 90 (60)	10	1,2,1,2	9.7417	1.2073	11.7267	1.8978	0.77
Exp_{10}	1 - 30; 91 - 248 (188)	31 - 90 (60)	14	2,3,2,3	7.8455	0.9967	10.8846	1.7643	0.88
Exp_{11}	1 - 120 (120)	121 - 248 (128)	10	1,2,1,2	13.4239	1.5468	10.4996	1.3152	0.61
Exp_{12}	1 - 120 (120)	121 - 248 (128)	14	2,3,2,3	10.7789	1.3717	8.5867	1.2005	0.74
Exp_{13}	1 - 150; 241 - 248 (158)	151 - 240 (90)	10	1,2,1,2	11.1983	1.4857	9.1089	1.3065	0.66
Exp_{14}	1 - 150; 241 - 248 (158)	151 - 240 (90)	14	2,3,2,3	10.1013	1.3206	7.6733	1.0936	0.88

4.2.2. DS2 - Boston house dataset

In this dataset, there were 499 historical samples with 11 input attributes. For such a relatively high-dimensional but limited sample dataset, the standard FS was not usable as it requires $2^{11} = 2,048$ fuzzy rules, even when applying the simplest partition in which each attribute contains only two fuzzy sets.

Table 5: Results of using the HFS in the MP3 player dataset

Experiment	Training dataset(#)	Testing dataset(#)	# of rules	Partitions	Iterations	Training APE(%)	Training RMSE	Testing APE(%)	Testing RMSE	Running time (second)
Exp_1	31 - 100 (70)	1 - 30 (30)	16	1 2; 1 2; 1 1	10000	9.9741	0.9404	12.7680	1.8076	2.12
Exp_2	31 - 100 (70)	1 - 30 (30)	21	1 2; 1 2; 2 2	6000	10.8963	1.7663	11.7490	1.6266	1.36
Exp_3	31 - 100 (70)	1 - 30 (30)	28	1 2; 1 2; 3 3	10000	10.4902	1.5118	12.8370	1.8318	2.06
Exp_4	1 - 70 (70)	71 - 100 (30)	16	1 2; 1 2; 1 1	10000	10.1038	1.2613	12.6070	1.8312	2.06
Exp_5	1 - 70 (70)	71 - 100 (30)	21	1 2; 1 2; 2 2	10000	9.5615	1.1274	12.2300	1.9679	2.08
Exp_6	1 - 70 (70)	71 - 100 (30)	28	1 2; 1 2; 3 3	10000	10.5901	1.2619	12.8150	1.9052	2.03
Exp_7	1 - 40 (40)	41 - 100 (60)	16	1 2; 1 2; 1 1	10000	10.3302	1.5111	12.5770	1.8197	1.24
Exp_8	1 - 40 (40)	41 - 100 (60)	21	1 2; 1 2; 2 2	10000	11.4601	1.0948	14.9430	2.9411	1.27
Exp_9	1 - 40 (40)	41 - 100 (60)	28	1 2; 1 2; 3 3	10000	13.8669	1.7309	25.7410	5.6944	1.26
Exp_{10}	11 - 50 (40)	1 - 10 (10)	16	1 2; 1 2; 1 1	10000	11.2670	1.2055	6.6900	0.7983	1.27
Exp_{11}	11 - 50 (40)	1 - 10 (10)	16	1 2; 1 2; 1 1	3000	14.4835	1.4092	10.6896	1.6482	0.53
Exp_{12}	11 - 50 (40)	1 - 10 (10)	21	1 2; 1 2; 2 2	10000	16.1273	1.6353	12.9130	1.6091	1.29
Exp_{13}	11 - 50 (40)	1 - 10 (10)	28	1 2; 1 2; 3 3	10000	11.8673	1.0891	15.6870	3.7580	1.31
Exp_{14}	21 - 50 (30)	1 - 20 (20)	16	1 2; 1 2; 1 1	10000	13.7768	1.1744	8.7620	1.2906	0.99
Exp_{15}	21 - 50 (30)	1 - 20 (20)	16	1 2; 1 2; 1 1	3000	13.4583	1.3571	12.3371	1.8762	0.45
Exp_{16}	21 - 50 (30)	1 - 20 (20)	21	1 2; 1 2; 2 2	10000	12.0021	1.0689	13.0840	2.2542	1.02
Exp_{17}	21 - 50 (30)	1 - 20 (20)	28	1 2; 1 2; 3 3	10000	10.8501	0.9221	14.5210	4.8472	1.04
Exp_{18}	51 - 248 (198)	1 - 50 (50)	16	1 2; 1 2; 1 1	10000	8.8865	1.3633	12.5860	1.4644	5.40
Exp_{19}	51 - 248 (198)	1 - 50 (50)	22	2 2; 2 2; 1 1	6000	8.4084	1.3067	13.1501	1.3202	3,40
Exp_{20}	51 - 248 (198)	1 - 50 (50)	21	1 2; 1 2; 2 2	10000	8.5953	1.3225	13.9480	1.4482	5,38
Exp_{21}	51 - 248 (198)	1 - 50 (50)	28	1 2; 1 2; 3 3	10000	8.9526	1.3174	15.3150	1.3774	5.47
Exp_{22}	1 - 30; 91 - 248 (188)	31 - 90 (60)	16	1 2; 1 2; 1 1	6000	9.1456	1.2401	11.1345	1.8376	3.14
Exp_{23}	1 - 30; 91 - 248 (188)	31 - 90 (60)	22	2 2; 2 2; 1 1	6000	8.6851	1.1210	12.7091	1.9002	3.30
Exp_{24}	1 - 30; 91 - 248 (188)	31 - 90 (60)	21	1 2; 1 2; 2 2	6000	8.0572	1.1163	15.2720	1.8671	3.20
Exp_{25}	1 - 30; 91 - 248 (188)	31 - 90 (60)	28	1 2; 1 2; 3 3	6000	8.1971	1.1077	16.0640	1.9446	3.21
Exp_{26}	1 - 120 (120)	121 - 248 (128)	16	1 2; 1 2; 1 1	6000	10.6254	1.5046	9.9050	1.6352	2.11
Exp_{27}	1 - 120 (120)	121 - 248 (128)	22	2 2; 2 2; 1 1	6000	10.7058	1.4687	9.9592	1.6921	2.15
Exp_{28}	1 - 120 (120)	121 - 248 (128)	21	1 2; 1 2; 2 2	6000	11.4216	1.7332	11.6220	2.0507	2.12
Exp_{29}	1 - 120 (120)	121 - 248 (128)	28	1 2; 1 2; 3 3	6000	10.5594	1.4292	11.5960	1.8939	2.10
Exp_{30}	1 - 150; 241 - 248 (158)	151 - 240 (90)	16	1 2; 1 2; 1 1	6000	10.2264	1.4780	9.9200	1.5209	2.70
Exp_{31}	1 - 150; 241 - 248 (158)	151 - 240 (90)	22	2 2; 2 2; 1 1	6000	9.9087	1.4225	9.1745	1.3747	2.74
Exp_{32}	1 - 150; 241 - 248 (158)	151 - 240 (90)	21	1 2; 1 2; 2 2	6000	10.6577	1.5578	10.1140	1.5967	2.69
Exp_{33}	1 - 150; 241 - 248 (158)	151 - 240 (90)	28	1 2; 1 2; 3 3	6000	10.3626	1.5206	10.4050	1.6123	2.74

Therefore, only the SFS-SISOM and the HFS were applied in this experiment. The available dataset was randomly decomposed into training/testing datasets, in which 414 samples were randomly selected to form the training dataset and the other 85 records were used as the testing dataset. A natural hierarchical structure (as depicted in Figure 4) is associated with the given problem. Three meaningful intermediate attributes, namely *living environments, house property,* and *convenience*, construct the top level standard FS. Hence, the experiments in HFS all employ the same hierarchical structure which consists of three sub-systems with no free parameters. In addition, the learning rate is set to 0.02 and error threshold is set to 0.01 for all experiment in the HFS. In the SFS-SISOM, the disturbance parameter is set to be $99,999$ and the number of iterations is set to 100. The pre-experiments show that the obtained results are very similar when employing different number of iterations (i.e., 10, 100, and 1000), therefore, only the results of running the 100 iterations are reported. The results of using the SFS-SISOM and the HFS, together with the employed parameters, are reported in Tables 6 and 7. Note that, in Table 7, when the partition parameter is set to "n", this indicates that the original and intermediate attributes were all divided into n sub-spaces (e.g., Exp_2); when the partition parameter is set to "n, m", the original attributes were partitioned into n sub-spaces, and the intermediate attributes were divided into m sub-spaces (e.g., Exp_1 and Exp_3).

It can be concluded from the experimental results that both the SFS-SISOM and the HFS performed well in this dataset, and can be used as effective decision support models. Both approaches achieved approximately 84% – 88% prediction accuracy in the testing dataset. In general, the HFS performs slightly better than the SFS-SISOM in terms of predictability (i.e., Testing APE(%)). On the other hand, the standard FS was incapable of dealing with this dataset due to the curse of dimensionality.

Table 6: Results of using the SFS-SISOM in the Boston house dataset [8]

Experiment	# of sub-spaces in Attributes											# of rules	Training APE(%)	Training RMSE	Testing APE(%)	Testing RMSE	Running time (second)
	1	2	3	4	5	6	7	8	9	10	11						
Exp_1	2	2	3	2	3	2	3	2	2	2	2	25	12.4817	3.6516	14.3455	5.3367	6.53
Exp_2	3	3	5	3	5	3	5	3	3	3	3	90	12.4064	3.3852	13.1579	5.1074	13.22
Exp_3	3	3	3	3	3	3	3	3	3	3	3	44	12.0596	3.6064	14.0818	5.1709	9.98
Exp_4	1	1	2	1	2	1	2	1	1	1	1	25	13.0903	3.7723	13.1299	5.2278	3.62
Exp_5	2	2	2	2	2	2	2	2	2	2	2	33	14.9134	4.1978	16.5419	6.0427	5.53

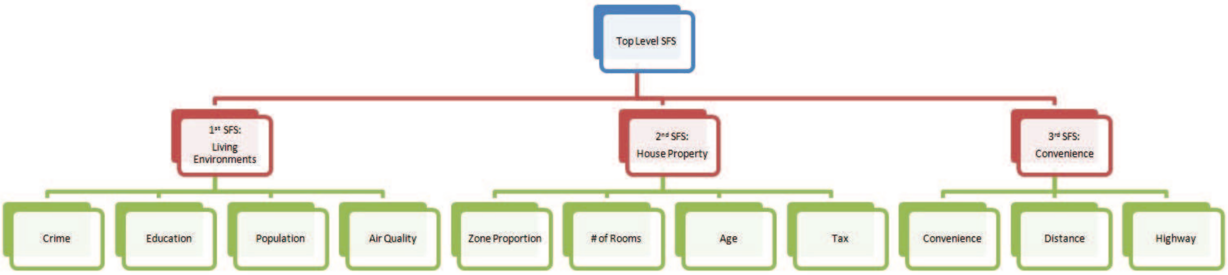


Figure 4: Hierarchical structure for Boston house dataset

Table 7: Results of using the HFS in the Boston house dataset

Experiment	# of rules	Partitions	Iterations	Training APE(%)	Training RMSE	Testing APE(%)	Testing RMSE	Running time (second)
Exp_1	253	2,3	6000	13.6606	4.1415	12.8858	5.5013	66.93
Exp_2	48	1	6000	15.3197	4.5951	15.2067	6.7762	59.48
Exp_3	104	1,3	6000	14.0024	4.8466	13.4519	6.5981	61.96
Exp_4	116	1 3 1 3; 1 2 1 2; 1 1 1; 1 1 1	6000	12.2817	3.6230	13.9064	5.9200	60.25
Exp_5	212	1 3 1 3; 1 2 1 2; 3 3 3; 3 3 2	6000	14.1919	3.6930	13.7923	5.5817	63.53
Exp_6	212	1 3 1 3; 1 2 1 2; 3 3 3; 3 3 2	3000	13.1382	3.4611	14.4630	6.0849	32.92

4.2.3. DS3 - California Dataset

In this dataset, 20,640 observations were randomly partitioned into two subsets, in which 70% (i.e., 14,448) were used for training and 30% (i.e., 6,192) were used for testing. The eight input attributes can be naturally grouped into two standard sub-FSs: the median income, population and households reflect the *block group properties*, while the housing median age, total rooms, total bedrooms, latitude, and longitude represent the *house properties*. Then these two meaningful intermediate attributes (i.e., *block group properties* and the *house properties*) construct the top-level standard FS. Thus, the sub-FSs construction setting is: 1 5 6;2 3 4 7 8. The experiments of the HFS in this dataset all employ the above hierarchical structure that consists of two sub-FSs with no free parameters. In addition, the learning parameters of the HFS were set to: learning rate = 0.02, error threshold = 0.01, and iterations = 1000. In the standard FS, in order to obtain the result within an hour, the number of iteration is set to 1 because the computational cost is very high (see more details in Section 5.4). The results of using the three FSs, together with the used parameters, are reported in Tables 8 through 10.

Table 8: Results of using the standard FS in the California dataset

Experiment	# of rules	Partitions	Disturbance parameter	Training APE(%)	Training RMSE	Testing APE(%)	Testing RMSE	Running time (second)
Exp_1	256	1,1,1,1,1,1,1,1	999	1.9407	0.3120	1.9653	0.3156	1,013.79
Exp_2	256	1,1,1,1,1,1,1,1	99999	1.8963	0.3057	1.9432	0.3322	511.17
Exp_3	256	1,1,1,1,1,1,1,1	9999999	14.6492	8.1259	15.5764	10.5383	543.34
Exp_4	384	2,1,1,1,1,1,1,1	999	1.8679	0.3022	1.8922	0.3040	2,754.45
Exp_5	384	2,1,1,1,1,1,1,1	99999	1.8248	0.2963	1.8728	0.3185	2,785.24

The results reveal that in general all three FSs can achieve very good performances in terms of training and testing prediction accuracy (i.e., 96.3% - 98.2%). However, the standard FS and the SFS-SISOM are quite sensitive to the disturbance parameter. If an inappropriate disturbance parameter was employed, the performance drops dramatically and even fails to derive the results (e.g., Exp_3 in Table 8; Exp_4 and Exp_8 in Table 9). In addition, although the standard FS can produce good prediction results, the flexibility of dividing the input space is quite limited due to its high computational complexity. As shown in Table 8, when 384 fuzzy rules are involved, it takes over 45 minutes to derive the results. It is time consuming and even impractical to employ the standard FS when some attributes are required to be finely divided. The HFS produces slightly better prediction performance than the SFS-SISOM, but with far more fuzzy rules. Although the number of required fuzzy rules is large, the running time to derive the result is short because the HFS requires much less space computational cost. Moreover, the HFS is less sensitive to the learning parameters, it produces more stable prediction performances in this experiment. This dataset involves relatively high dimensions, and the HFS outperforms the other two FSs with respect to the interpretability for the following reasons: 1) the standard FS jointly considers eight attributes in a fuzzy rule, and the derived fuzzy rule might be too complicated for end users to understand. Also, only a small number of fuzzy sets (i.e., 1 or 2) is employed to describe an attribute due to the curse of dimensionality; 2) the SFS-SISOM only considers one attribute in a fuzzy rule. Instead, the HFS decomposes the high-dimensional problem into three sub-SFSs, therefore, the negotiation behaviors can be better explained and understood by considering diverse modeling granularities. Further from Exp_5 in Table 10, it shows that by uneven partitions for different attributes, the number of rules in the HFS can be reduced significantly. The distinguishing merit of the SFS-SISOM is the modeling efficiency, as it can produce good prediction accuracy by using fewer fuzzy rules and iterations as long as the right learning parameters are identified.

In summary, this experiment shows that both the SFS-SISOM and the HFS can handle high dimensional and large data effectively. The dataset is based on the block group rather than individual house, as it is less complicated and less non-linear, therefore the linear model (i.e., the SFS-SISOM) performs well. However, it is expected that, if the dataset with the individual houses is available, the HFS can be more useful and effective. Therefore, the proposed three FSs provide a comprehensive set of model choices and are able to handle various needs to accurately model the negotiation pricing behavior.

Table 9: Results of using the SFS-SISOM in the California dataset

Experiment	# of sub-spaces in Attributes								# of rules	Disturbance parameter	Iterations	Training APE(%)	Training RMSE	Testing APE(%)	Testing RMSE	Running time (second)
	1	2	3	4	5	6	7	8								
Exp_1	1	1	1	1	1	1	1	1	16	999	10	2.2763	0.3913	2.2711	0.3653	9.13
Exp_2	1	1	1	1	1	1	1	1	16	999	1000	2.2699	0.3574	2.2657	0.3591	519.16
Exp_3	1	1	1	1	1	1	1	1	16	9999	10	2.4640	0.3837	2.4566	0.3848	9.01
Exp_4	1	1	1	1	1	1	1	1	16	99999	10	N/A	N/A	N/A	N/A	8.89
Exp_5	2	2	2	2	2	2	2	2	24	999	10	2.1638	0.3446	2.1531	0.3419	13.04
Exp_6	2	2	2	2	2	2	2	2	24	999	1000	2.1590	0.3436	2.1498	0.3413	1,409.67
Exp_7	2	2	2	2	2	2	2	2	24	9999	10	2.2233	0.3508	2.2170	0.3487	13.23
Exp_8	2	2	2	2	2	2	2	2	24	99999	10	18050.908	2838.5105	18183.354	2866.5483	13.05
Exp_9	3	3	3	3	3	3	3	3	32	999	10	2.0386	0.3263	2.0354	0.3247	21.06
Exp_{10}	3	3	3	3	3	3	3	3	32	9999	10	2.1547	0.3414	2.1586	0.3413	20.62
Exp_{11}	3	3	3	3	3	3	3	3	32	99999	10	N/A	N/A	N/A	N/A	20.39
Exp_{12}	4	4	4	4	4	4	4	4	40	999	10	2.0918	0.3340	2.0752	0.3296	30.35
Exp_{13}	4	4	4	4	4	4	4	4	40	9999	10	2.2093	0.3486	2.1936	0.3448	30.68
Exp_{14}	4	4	4	4	4	4	4	4	40	99999	10	8.9838	1.6173	9.1614	1.6408	30.74

Table 10: Results of using the HFS in the California dataset

Experiment	# of rules	Partitions	Iterations	Training APE(%)	Training RMSE	Testing APE(%)	Testing RMSE	Running time (second)
Exp_1	76	1 1 1;1 1 1 1 1;5 5	1000	2.5533	0.3977	2.6233	0.4107	226.16
Exp_2	306	2 2 2;2 2 2 2 2;5 5	1000	2.1122	0.3413	2.1939	0.3581	328.81
Exp_3	1,124	3 3 3;3 3 3 3 3;5 5	1000	1.9511	0.3160	2.039	0.3320	395.49
Exp_4	3,286	4 4 4;4 4 4 4 4;5 5	1000	1.866	0.3084	1.9342	0.3160	636.48
Exp_5	780	5 1 1;1 4 1 5 5;5 5	1000	1.9082	0.3111	1.9908	0.3257	355.71

5. Discussion

5.1. Interpretability and transparency

In regard to the development of DSSs, interpretability/transparency is one of the most crucially desirable features. It is of paramount importance for researchers to further apply this insight to the predicament of negotiation pricing. The proposed FS approach not only enables effective knowledge discovery, but also efficient knowledge representation in the form of linguistic IF-THEN fuzzy rules, which are understandable and interpretable. When the dimension is low, the standard FS has the best interpretability, because it can explain the complete conditions of input attributes within one fuzzy rule. On the contrary, the SFS-SISOM can merely represent one influential aspect and the outcome in the derived fuzzy rules. However, the standard FS and the SFS-SISOM offer a flat view in the sense that all attributes are listed at the same level, and the impacts of different attributes become less apparent when a large number of attributes and rules are involved. In the HFS, individual and less important attributes are aggregated (by the lower level sub-FS) into the higher-level indexes, which are combined with important attributes to form a top-level sub-FS to derive the system output (i.e., the negotiation price). For example, in DS2, the higher-level sub-FS shows the aggregated impacts (in terms of intermediate variables) of living environments, house property, and convenience on the house price. In this manner, the top-level sub-FS provides a high-level overview and interpretation (i.e., from a forest point of view), while the lower-level sub-FS represents a more detailed view of how each index is formed or formulated (i.e., how the forest is formed from trees). In other words, the fuzzy rules derived from the HFS present comprehensive multi-views to understand negotiation pricing behaviors. Such a feature of HFSs, which can provide both tree and forest views, is very useful for complicated negotiation pricing problems to ensure and enable the interpretability and transparency.

5.2. Accuracy

The APE (%) and the RMSE were employed in this study to measure performance accuracy. Such performance is dependent on how well the partitioned sub-spaces are covered by the training data. In order to analyse this point, consider the case where n input attributes exist and the input space of each attribute is partitioned by m sub-spaces (i.e., $m+1$ fuzzy sets). As such, there are m^n and $m \times n$ divided sub-spaces in the standard FS and the SFS-SISOM, respectively. In the HFS, the total number of the divided sub-spaces is the same as the number of the divided sub-spaces in the hierarchical level that contains the maximum number of attributes. Suppose there are L levels in the structure and n_i attributes in the i^{th} level; then the total number of divided sub-spaces can be represented as: $\sum_{i=1}^{L} m^{n_i}$. Since $n = \sum_{i=1}^{L} n_i$, in most cases, the number of the divided sub-spaces in the SFS-SISOM (i.e., $m \times n$) and the number in the HFS (i.e., $\sum_{i=1}^{L} m^{n_i}$) is smaller than that of the standard FS (i.e., m^n). Thus, the divided sub-spaces in the SFS-SISOM and the HFS stands a better chance of being covered when given the same training dataset and partitions. This explains why the testing accuracy of the standard FS is lower than the other two FSs in some cases. Besides the number of required data samples, the selected learning algorithms also contribute to the model's performance. The RLS learning algorithms for standard FSs and SFS-SISOMs ensure that the global optima are found. However, the gradient decent algorithm used in HFSs may lead to a local optima. If the selected initial parameters diverge to the global optima, the performance of the HFS can be affected.

5.3. Generality

Generality relates not only to the capability of handling wide problem domains, but also the capability of satisfying various modeling requirements. In general, compared to the SFS-SISOM and the HFS, the standard FS is less generic in handling high-dimensional problems due to the large number of learning parameters and fuzzy rules it requires. In order to cover the divided input space, the number of required training samples should be at least as many as the number of required fuzzy rules; otherwise, the standard FS may suffer from the high-dimensional problem. This drawback is clearly evident in some experiments in DS1 (e.g., Exp_{12} in Table 3) and the inability to build usable models on DS2. In addition, from the application point of view, in some cases the user would like to partition the input variable into a specified number of sub-spaces, such that the negotiation behavior in a certain sub-space could be observed. The capability of the standard FS to satisfy various modeling requirements is also bounded by the dimensionality problem. For instance, in the DS1, the standard FS fails to produce acceptable predictive results when the input attributes need to be partitioned into "$2, 3, 2, 2$" sub-spaces (i.e., Exp_8, Exp_{10}, and Exp_{12} in Table 3). Under the same circumstance, SFS-SISOM and HFS are still capable of accomplishing such tasks. As such, the SFS-SISOM and the HFS are superior to the standard FS in terms of model generality. Further, HFSs are universal approximators and, therefore, can represent any nonlinear negotiation pricing models, whereas SFS-SISOMs are not universal approximators and cannot accurately model some highly nonlinear scenarios.

5.4. Computational cost

The computational cost concerns both the space and time complexity in this study. For space complexity, both standard FSs and SFS-SISOMs need to construct the input matrix. Hence, the required space for standard FSs is $\mathbf{O}(MK)$, where M is the number of instances, $K = \prod_{j=1}^{n} N_j$ is required number of fuzzy rules, and N_j is the number of fuzzy sets of the j^{th} attribute, while the required space for SFS-SISOMs is $\mathbf{O}(MS)$, where $S = \sum_{j=1}^{n} N_j$ is the required number of fuzzy rules for SFS-SISOMs. Since HFSs employ the gradient decent algorithm for learning, the learning parameters are updated by instances. Therefore, the required computational space of HFSs is much less than the other two FSs, and it is $\mathbf{O}(H)$, where H (see Equation (7)) is the number of required fuzzy rule of HFSs.

In the proposed approach, triangular membership functions are selected in order to achieve the interpretability and computing effectiveness. Triangular membership functions are basically piecewise linear functions, and their derivatives are simple to calculate. Given T iterations, the time complexity of standard FSs is either $\mathbf{O}(KMT)$ or $\mathbf{O}(KM\xi^{-2})$ (depends on which termination condition reaches first), where ξ is the error threshold. The time complexity of SFS-SISOMs is either $\mathbf{O}(SMT)$ or $\mathbf{O}(SM\xi^{-2})$. In HFSs, although there are multi-level sub-FSs, all computations only involve the derivatives of triangular membership functions and some simple algebraic manipulations. The time complexity of HFS can be represented as either $\mathbf{O}(HMT)$ or $\mathbf{O}(HM\xi^{-2})$ [3]. In this study, all the experiments were conducted on a laptop with

Mac OS X 10.7.4, processor 1.7 GHz Intel Core i5, 4GB memory, and the running time of different FSs on different datasets are summarized in Table 11. Table 11 shows that the proposed models can obtain the results within a few seconds in most cases, and the standard FSs take the longest time to derive the result, which is consistent with the above analyses. In DS3, the standard FS takes over 45 minutes when 384 rules are involved. For SFS-SISOMs, if the number of iteration is small, it is quick to derive the result. For HFSs, although they involve a large number of fuzzy rules in DS3, the space complexity is small, therefore, the running time of each iteration of the HFSs is actually less than the standard FSs and SFS-SISOMs.

Table 11: Summary of running time

	Standard FS		SFS-SISOM		HFS	
	No. of rules (iterations)	running time (second)	No. of rules (iterations)	running time (second)	No. of rules (iterations)	running time (second)
DS1	36 (100)	1.99	10 (100)	0.46	16-28 (6000)	2.56
	114 (100)	109.93	14 (100)	0.86	16-28 (10000)	2.19
DS2	N/A		25-90 (10)	7.78	48-212 (6000)	62.43
DS3	256 (1)	689.43	16-48 (100)	23.59	76-1,124 (1000)	326.54
	384 (1)	2,769.84	16-48 (1000)	964.42	3,286-8,028 (1000)	875.30

5.5. Applicability

One of the main contributions of this work is examining the applicability of three FSs in dealing with negotiation pricing. In this study, no approach can always outperform the others. The above experimental results reveal that the applicability of three FSs can differ, so understanding their applicabilities would be beneficial to select the appropriate approach for the given problem. Generally speaking, the standard FS is a better approach for low-dimensional problems with sufficient data samples (i.e., dense data coverage). The main reason behind this is that the required number of partitioned sub-spaces in the input space is relatively small when modeling low-dimensional problems, and such sub-spaces are more likely to be well covered by training samples. This would help to improve prediction performance. Moreover, in the standard FS, complete combinations of input conditions are linked through fuzzy intersection operators and are represented in the form of fuzzy rules. As such, it is more suitable to select the standard FS when it is required to emulate and observe different aspects of negotiation pricing behaviors. On the contrary, the SFS-SISOM is a better approach for high-dimensional problems with insufficient data samples (i.e., sparse data coverage). In the SFS-SISOM, the mathematical relationship between the input attributes and the output attributes is individually emulated. This results in requiring far fewer fuzzy rules when modeling the high-dimensional problems. The applicability of three FSs are depicted in Figure 5.

The performance of the HFS lies between the standard FS and the SFS-SISOM, as both the standard and the SFS-SISOM FSs can be represented as special cases of HFS (as shown in Figure 6). When there are no free parameters in the hierarchical structure and each original input attribute constructs a sub-FS

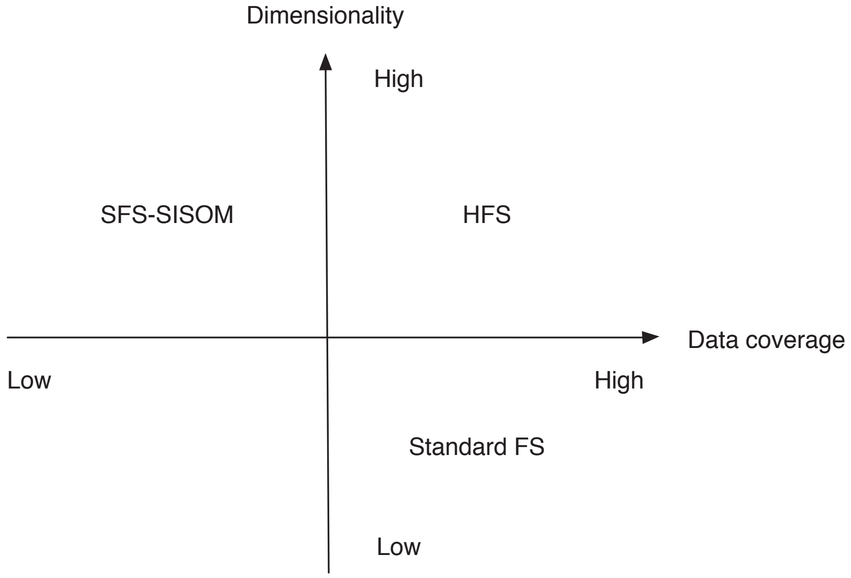


Figure 5: Applicability of different FSs

(e.g., SFS_1^1, SFS_2^1, and SFS_3^1 in Figure 6), the HFS becomes a SFS-SISOM. On the other hand, when only one sub-FS exists (i.e., the top level sub-FS) in the hierarchical structure, and all original input attributes contribute to the top level sub-FS, the HFS can be treated as a standard FS. Consequently, HFSs provide more flexibility in building the predictive model, as the hierarchical structure can be modified according to the features of the available datasets. Compared with the other two FSs, HFSs can be more easily adapted to meet various modeling requirements. The discussions of the properties of three FSs are summarised in Table 12 in which 'A' indicates the best performance and 'C' indicates the worst.

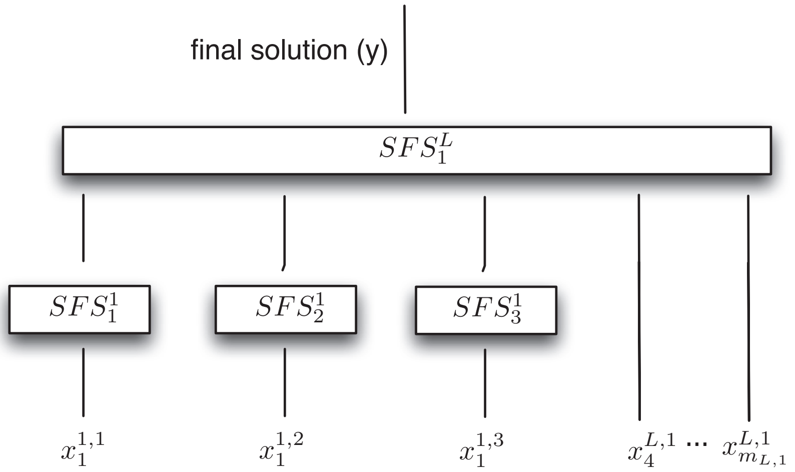


Figure 6: Special cases of HFSs

Table 12: Summary of the properties of three FS approaches

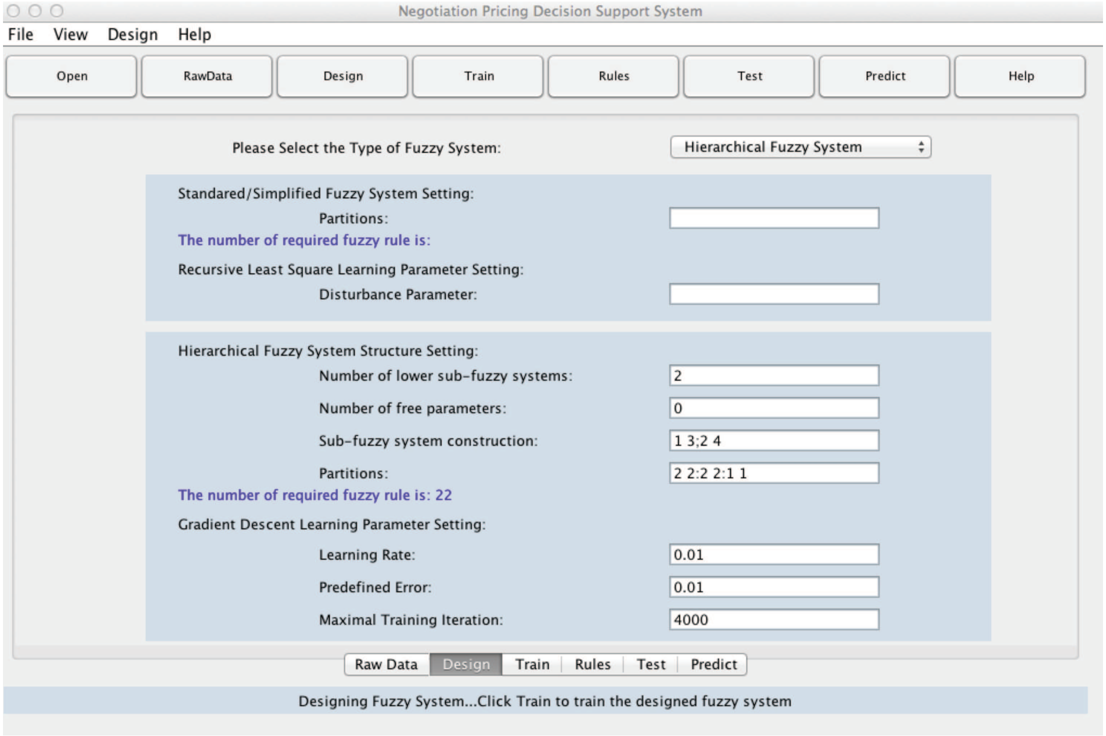
Approach	Interpretablity		Computational cost	Generality	Flexibility
	Low-dimensional	high-dimensional			
Standard FS	A	C	C	C	C
SFS-SISOM	C	B	A	B	B
HFS	B	A	B	A	A

The newly introduced HFS is important for the following reasons: First, compared to standard FSs, which can represent any continuous function to any degree of accuracy (i.e., universal approximators) but suffer from the problem of dimensionality, HFSs are universal approximators and are capable of overcoming the problem of dimensionality. Second, from the learning mechanism point of view, although both SFS-SISOMs and HFSs can handle the high-dimensional data, the mechanism behind is vastly different. The SFS-SISOM is a simplified approach by omitting certain impacts of pricing attributes and more complicated behaviors (in particular, the aggregated cross or $1 + 1 > 2$ effect) , so that it can be learned from less data. However, the SFS-SISOM is unable to represent some complicated systems or behaviors (i.e., not universal approximators) due to their inherent structure. On the other hand, the HFS does not omit pricing component, but employs the transfer learning mechanism to handle data coverage problem. Take the DS2 for an example, as long as there are sufficient data to learn the relationship between the three intermediate variables and the house price, as well as how each of the above three intermediate variables are formed, the knowledge from one data sample of good living environment, property, and convenience can be transferred to represent other cases of good living environment, property, and convenience even if these other cases are good due to different reasons (i.e., the values of the attributes in these cases are different from the available data sample). In short, rather than using the simplification or omitting attributes as SFS-SISOMs, the transfer mechanism enables HFSs to work effectively when training samples don't densely cover the input space. Thus, HFSs as universal approximators are capable of representing any complicated systems and providing a flexible structure to represent any given negotiation pricing system with the right complexity and desired accuracy if designed properly. Third, in the case of a large number of negotiation attributes, HFSs can be designed to develop interpretable systems where the lower-level sub-systems represent the impact from the raw attributes, and the higher-level sub-systems provide a small number of human-understandable indexes which aggregate the impacts from a set of raw attributes.

6. Developing a Prototype System

In this section, a prototype system has been developed and presented. The system is a standalone Java application that provides intelligent decision support for a series of negotiation pricing tasks. The system allows pricing managers to *load dataset*, *design*, *train*, and *test* the fuzzy model, and then *predict* the negotiation price/discount for a new transaction. The graphic user interface (GUI) consists of five components: menu, toolbar panel, display panel, tab control panel, and status display panel (see Figure 7). The menu groups the basic functions of the system, and the toolbar panel provides a more convenient way to access these functions. Once a user clicks the functional buttons on the toolbar panel, the tab control panel will automatically switch to the corresponding tab. When the system is in use, the current status and instructions are displayed on the status display panel. The basic functions are presented below:

- **Open and view dataset:** At the beginning, a pricing manager can select the dataset which will be modeled from the *Open* button. Before the system is in use, the user should ensure the training dataset and the corresponding testing dataset will be readily available and are named correctly in the predefined way (i.e., filename + "-Test" indicates the testing dataset). The loaded dataset, including the training dataset and testing dataset, can then be displayed in the display panel.



Please Select the Type of Fuzzy System: Hierarchical Fuzzy System

Standard/Simplified Fuzzy System Setting:

Partitions:

The number of required fuzzy rule is:

Recursive Least Square Learning Parameter Setting:

Disturbance Parameter:

Hierarchical Fuzzy System Structure Setting:

Number of lower sub-fuzzy systems: 2

Number of free parameters: 0

Sub-fuzzy system construction: 1 3;2 4

Partitions: 2 2:2 2:1 1

The number of required fuzzy rule is: 22

Gradient Descent Learning Parameter Setting:

Learning Rate: 0.01

Predefined Error: 0.01

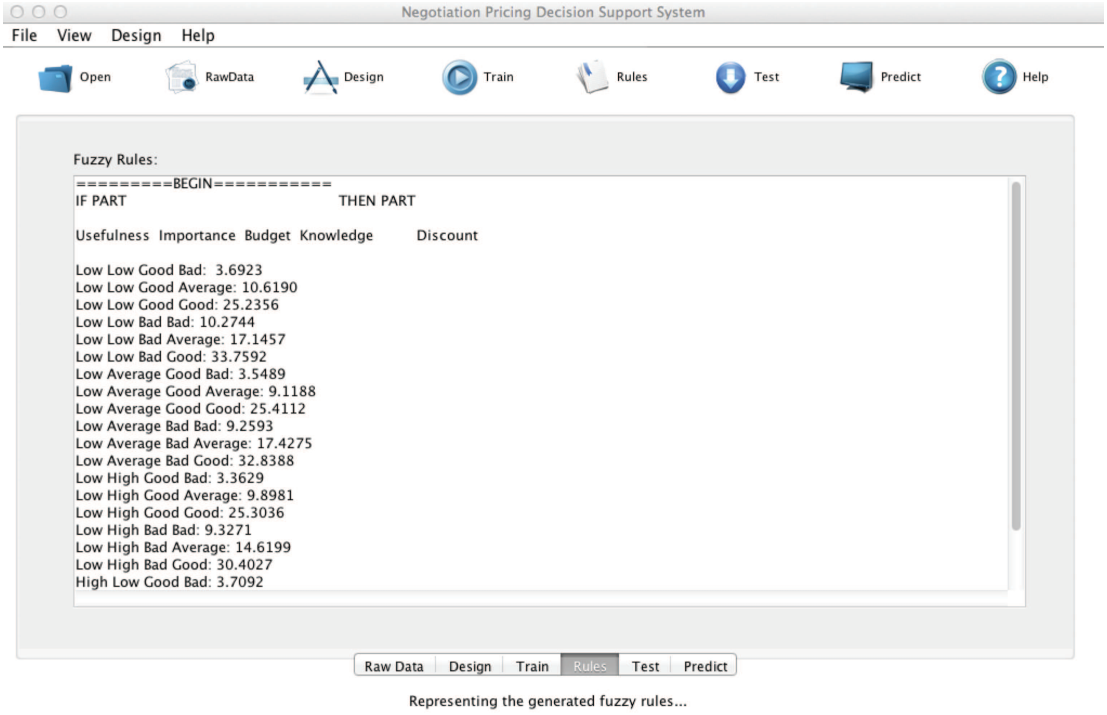
Maximal Training Iteration: 4000

Designing Fuzzy System...Click Train to train the designed fuzzy system

Figure 7: Prototype system - model design

- **Design the model:** When the user clicks the *Design* button on the toolbar panel, a design panel, as shown in Figure 7, is provided and allows the user to identify the model type and specify different learning parameters. The design panel contains two parts: the SFS panel for the design of standard and SFS-SISOM FSs, and the HFS panel for the design of HFSs. Once the model type is selected, only the corresponding design panel is enabled. The learning parameters and the hierarchical structure of

HFSs must be entered in a predefined format. A tooltip with the format instructions is also provided for each parameter. The raw dataset will be reconstructed according to the defined structure (e.g., attributes that in the same sub-FS will be grouped together). The reconstructed dataset can be checked via the "Processed Data" option in the "View" menu.



Fuzzy Rules:

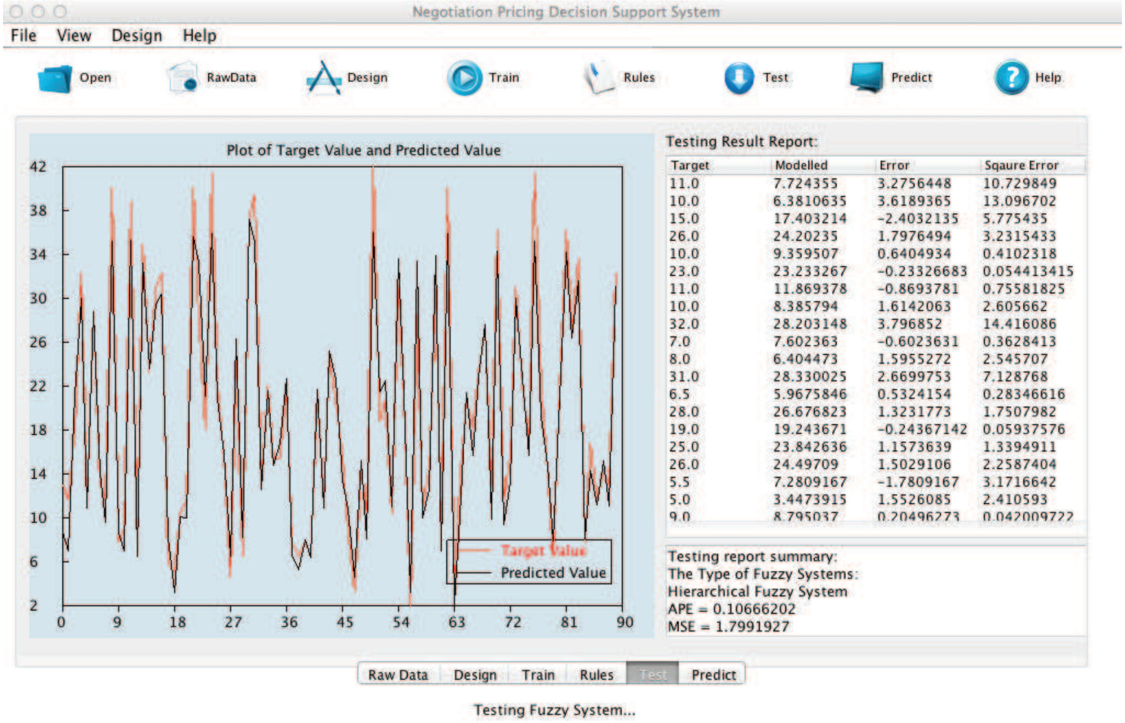
```
=========BEGIN===========
IF PART                                    THEN PART

Usefulness  Importance  Budget  Knowledge      Discount

Low Low Good Bad:  3.6923
Low Low Good Average: 10.6190
Low Low Good Good: 25.2356
Low Low Bad Bad: 10.2744
Low Low Bad Average: 17.1457
Low Low Bad Good: 33.7592
Low Average Good Bad: 3.5489
Low Average Good Average: 9.1188
Low Average Good Good: 25.4112
Low Average Bad Bad: 9.2593
Low Average Bad Average: 17.4275
Low Average Bad Good: 32.8388
Low High Good Bad: 3.3629
Low High Good Average: 9.8981
Low High Good Good: 25.3036
Low High Bad Bad: 9.3271
Low High Bad Average: 14.6199
Low High Bad Good: 30.4027
High Low Good Bad: 3.7092
```

Representing the generated fuzzy rules...

Figure 8: Prototype system - generated fuzzy rules

- **Train and test the model:** Once the design of the predictive model is accomplished, the user can click the *Train* button to train the designed model. The learning algorithm and defined parameters are employed to learn the fuzzy rules from the training dataset. The training results and generated rules will then be displayed. Consequently, the generated fuzzy rules (see Figure 8) are used in conjunction with the testing dataset to further validate the built model. The testing results, including target values, predicted value, error, APE, and RMSE, are shown in Figure 9. The plot of the target value and the predicted value provides the user with a clear and intuitive view of model performance.

- **Prediction:** A suggested negotiation price/discount can be provided by the system to assist the pricing manager when given a new transaction. In the prediction panel, the user can manually enter the information of a new transaction, and then click the *Predict* button. The predicted price will be displayed (an example is shown in Figure 10). This proposed price can be used as a baseline or resistance point in the price negotiation.



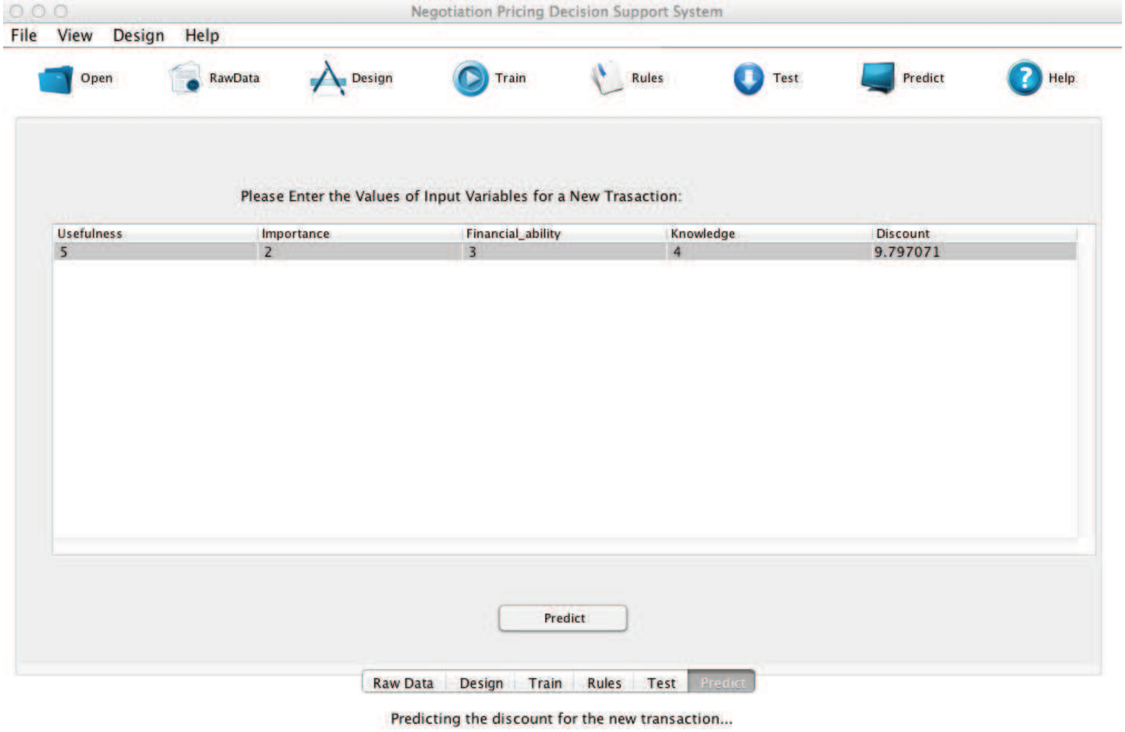
Testing Result Report:

Target	Modelled	Error	Sqaure Error
11.0	7.724355	3.2756448	10.729849
10.0	6.3810635	3.6189365	13.096702
15.0	17.403214	-2.4032135	5.775435
26.0	24.20235	1.7976494	3.2315433
10.0	9.359507	0.6404934	0.4102318
23.0	23.233267	-0.23326683	0.054413415
11.0	11.869378	-0.8693781	0.75581825
10.0	8.385794	1.6142063	2.605662
32.0	28.203148	3.796852	14.416086
7.0	7.602363	-0.6023631	0.3628413
8.0	6.404473	1.5955272	2.545707
31.0	28.330025	2.6699753	7.128768
6.5	5.9675846	0.5324154	0.28346616
28.0	26.676823	1.3231773	1.7507982
19.0	19.243671	-0.24367142	0.05937576
25.0	23.842636	1.1573639	1.3394911
26.0	24.49709	1.5029106	2.2587404
5.5	7.2809167	-1.7809167	3.1716642
5.0	3.4473915	1.5526085	2.410593
9.0	8.795037	0.20496273	0.042009722

Testing report summary:
The Type of Fuzzy Systems:
Hierarchical Fuzzy System
APE = 0.10666202
MSE = 1.7991927

Testing Fuzzy System...

Figure 9: Prototype system - model testing



Please Enter the Values of Input Variables for a New Trasaction:

Usefulness	Importance	Financial_ability	Knowledge	Discount
5	2	3	4	9.797071

Predicting the discount for the new transaction...

Figure 10: Prototype system - model prediction

7. Conclusions

This study proposes a systematic and learning approach consisting of three different FSs (i.e., standard FS, SFS-SISOM, and HFS) to provide intelligent decision support for negotiation pricing, in particular under the high-dimensional and uncertain scenarios. The effectiveness and applicability of the proposed approaches are demonstrated by three experimental datasets, varying from dimensionality to data coverage. The main contributions of this work include: 1) Instead of tackling the difficult and unrealistic tasks to identify and maximize the utility functions in negotiation pricing, this work proposes a new learning approach that concentrates on identifying the right negotiation pricing by learning from historical data. The proposed approach also is capable of learning the different influential factors for different customers, gaining a good customer relationship by offering the right price to the right customer; 2) To handle the uncertain information in negotiation pricing, a systematic and comprehensive set of fuzzy models is proposed, developed, and compared. Their advantages and capabilities under different application scenarios are analyzed and discussed with respect to the crucial properties of DSSs: interpretability and transparency, accuracy, generality, computational cost, and applicability; and 3) A DSS prototype integrating the three FSs is developed and validated. The system illustrates how the negotiation pricing DSS based on the proposed approach can be designed and implemented for practical uses.

This research has several significant implications for research and practice. First, interpretability and transparency is one of the most desirable features for DSSs. Compared with other data mining techniques (e.g., neural networks), the FS-based approach is capable of transforming the acquired negotiation pricing knowledge into human understandable IF-THEN fuzzy rules. The derived fuzzy rules can serve as unified guidance in negotiation pricing and also be employed to train new sales representatives. Second, this study endeavors to draw more attention to the dimensionality problem in negotiation pricing, which has yet to be thoroughly investigated in the literature. It has been proven both theoretically and practically that the standard FS is more susceptible to the problem of dimensionality. This study has shown that the other two types of FSs can be used as a means to complement the standard FS in handling high-dimensional problems. Third, this study reveals the properties and features of the three FSs. The proposed approach and the integrated platform can provide immense potential and flexibility for end users to choose the most suitable model for any particular problem.

Like many prior empirical studies, this research inevitably has several limitations that may trigger further research. With the rapid development and promising results of e-negotiations, the proposed approach can be integrated into e-negotiation platforms. In addition, the current prototype is not role-based and it does not distinguish different groups of targeted users (e.g., data operators, model trainers, salespeople, and managers). It is an important direction to improve the prototype by separating different tasks for different groups of users, and providing clear visuals that guide users to the appropriate and applicable parts of the system. Moreover, since different users may have different preferences on the model properties, it is difficult for the intelligent DSS to determine the most appropriate FS model on behalf of the decision maker. One plausible way to automate the model selection is to allow the user to specify his/her model preferences in the prototype system, then an overall performance score can be derived. Finally, it will be interesting to extend the current model to support multi-issue and bilateral negotiation decision support for future studies. If the historical data of bidding and offering during the price negotiation process were available, it will be possible to build two additional models to learn the bidding behavior and successful offering strategy. Then the system will be not only able to predict the final agreement prices proposed in this study, but also guide the offering strategy based on the bidding behavior.

Acknowledgments

This work is supported by the National Nature Science Foundation of China (No. 71301133, 71572166, and 71671149) and the Fundamental Research Funds for the Central Universities (Project No. 20720161044). Thanks also go to all participants who helped to complete the MP3 dataset questionnaire.

References

[1] Cao, M., Luo, X., Luo, X. R., Dai, X., 2015. Automated negotiation for e-commerce decision making: A goal deliberated agent architecture for multi-strategy selection. Decision Support Systems 73, 1 – 14.

[2] Carbonneau, R., Kersten, G. E., Vahidov, R., 2008. Predicting opponent's moves in electronic negotiations using neural networks. Expert Systems with Applications 34 (2), 1266 – 1273.

[3] Cartis, C., Gould, N. I. M., Toint, P. L., 2010. On the complexity of steepest descent, newton's and regularized newton's methods for nonconvex unconstrained optimization problems. SIAM Journal on Optimization 20 (6), 2833 – 2852.

[4] Chan, C.-C. H., Cheng, C.-B., Hsien, W.-C., 2011. Pricing and promotion strategies of an online shop based on customer segmentation and multiple objective decision making. Expert Systems with Applications 38, 14585 – 14591.

[5] Chung, F.-L., Duan, J.-C., 2000. On multistage fuzzy neural network modeling. IEEE Transactions on Fuzzy Systems 8 (2), 125–142.

[6] Fu, X., Shen, Q., 2010. Fuzzy compositional modelling. IEEE Transactions on Fuzzy Systems 18 (4), 823–840.

[7] Fu, X., Shen, Q., 2011. Fuzzy complex numbers and their application for classifiers performance evaluation. Pattern Recognition 44 (7), 1403–1417.

[8] Fu, X., Zeng, X., Wang, D., Xu, D., Yang, L., 2015. Fuzzy system approaches to negotiation pricing decision support. Journal of Intelligent & Fuzzy Systems 29 (2), 685 – 699.

[9] Giacomazzi, P., Stanojev, I., Verticale, G., 2012. A negotiation-based scheme for service level pricing for wireless access. Computer Communications 35 (4), 444 – 453.

[10] Goldberg, P. K., June 1996. Dealer price discrimination in new car purchases: Evidence from the consumer expenditure survey. Journal of Political Economy 104 (3), 622–54.

[11] Joo, M., Lee, J., 2002. Universal approximation by hierarchical fuzzy system with constraints on the fuzzy rule. Fuzzy Sets and Systems 130, 175 – 188.

[12] Kolomvatsos, K., Anagnostopoulos, C., Hadjiefthymiades, S., 2014. Sellers in e-marketplaces: A fuzzy logic based decision support system. Information Sciences 278, 267 – 284.

[13] Kwon, S., Yoo, B., Kim, J., Shang, W., Lee, G., 2009. Reservation price reporting mechanisms for online negotiations. Decision Support Systems 46 (4), 755 – 762.

[14] L*ó*pez, V., Fern*á*ndez, A., del Jesus, M. J., Herrera, F., 2013. A hierarchical genetic fuzzy system based on genetic programming for addressing classification with highly imbalanced and borderline data-sets. Knowledge-Based Systems 38, 85 – 104.

[15] Lee, K. C., Lee, H., Lee, N., 2012. Agent based mobile negotiation for personalized pricing of last minute theatre tickets. Expert Systems with Applications 39 (10), 9255 – 9263.

[16] Li, I.-H., Lee, L.-W., 2011. A hierarchical structure of observer-based adaptive fuzzy-neural controller for {MIMO} systems. Fuzzy Sets and Systems 185 (1), 52 – 82.

[17] Lin, C.-C., Chen, S.-C., Chu, Y.-M., 2011. Automatic price negotiation on the web: An agent-based web application using fuzzy expert system. Expert Systems with Applications 38 (5), 5090 – 5100.

[18] Lin, H.-T., Chang, W.-L., 2008. Order selection and pricing methods using flexible quantity and fuzzy approach for buyer evaluation. European Journal of Operational Research 187 (2), 415 – 428.

[19] Linkens, D., Shieh, J., Peacock, J., 1996. Hierarchical fuzzy modelling for monitoring depth of anaesthesia. Fuzzy Sets and Systems 79 (1), 43 – 57.

[20] Moosmayer, D. C., Chong, A. Y.-L., Liu, M. J., Schuppar, B., 2013. A neural network approach to predicting price negotiation outcomes in business-to-business contexts. Expert Systems with Applications 40 (8), 3028 – 3035.

[21] Moosmayer, D. C., Schuppar, B., Siems, F. U., 2012. Reference prices as determinants of business-to-business price negotiation outcomes: An empirical perspective from the chemical industry. Journal of Supply Chain Management 48 (1), 92–106.

[22] Park, S., Bolton, G. E., Rothrock, L., Brosig, J., 2010. Towards an interdisciplinary perspective of training intervention for negotiations: Developing strategic negotiation support contents. Decision Support Systems 49, 213–221.

[23] Raju, G. V. S., Zhou, J., Kisner, R., 1991. Hierarchical fuzzy control. International Journal of Control 54 (5), 1201 – 1216.

[24] Rubinstein, A., 1982. Perfect equilibrium in a bargaining model. Econometrica 50 (1), 97 – 109.

[25] Takagi, T., Sugeno, M., 1985. Fuzzy identification of systems and its applications to modeling and control. IEEE Transactions on Systems, Man and Cybernetics SMC-15 (1), 116 – 132.

[26] Turban, E., Aronson, J. E., Liang, T.-P., 2004. Decision Support Systems and Intelligent Systems (7th Edition). Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

[27] Wang, D., Zeng, X., Keane, J. A., 2009. Intermediate variable normalization for gradient descent learning for hierarchical fuzzy system. IEEE Transactions on Fuzzy Systems 17 (2), 468–476.

[28] Wang, L.-X., 1997. A course in fuzzy systems and control. Prentice Hall PTR.

[29] Wang, L.-X., 1998. Universal approximation by hierarchical fuzzy systems. Fuzzy Sets and Systems 93 (2), 223 – 230.

[30] Wilken, R., Cornelien, M., Backhaus, K., Schmitz, C., 2010. Steering sales reps through cost information: An investigation into the black box of cognitive references and negotiation behavior. International Journal of Research in Marketing 27 (1), 69 – 82.

[31] Zadeh, L., Kacprzyk, J., 1999. Computing with Words in Information/intelligent Systems: Foundations. Studies in Fuzziness and Soft Computing. Physica-Verlag.

[32] Zadeh, L. A., 1965. Fuzzy sets. Information and control 8, 338 – 353.

[33] Zeng, X., Keane, J. A., 2005. Approximation capabilities of hierarchical fuzzy systems. IEEE Transactions on Fuzzy Systems 13 (5), 659–672.

[34] Zhao, J., Wang, L., 2015. Pricing and retail service decisions in fuzzy uncertainty environments. Applied Mathematics and Computation 250 (0), 580 – 592.