A User Driven Data Mining Process Model and Learning System

Esther Ge, Richi Nayak, Yue Xu and Yuefeng Li

CRC for Construction Innovations
Faculty of Information technology
Queensland University of Technology, Brisbane, Australia
r.nayak@qut.edu.au

Abstract. This paper deals with the problem of using the data mining models in a real-world situation where the user can not provide all the inputs with which the predictive model is built. A learning system framework, Query Based Learning System (QBLS), is developed for improving the performance of the predictive models in practice where not all inputs are available for querying to the system. The automatic feature selection algorithm called Query Based Feature Selection (QBFS) is developed for selecting features to obtain a balance between the relative minimum subset of features and the relative maximum classification accuracy. Performance of the QBLS system and the QBFS algorithm is successfully demonstrated with a real-world application.

Keywords: Data Mining, Learning System, predictive model, lifetime prediction, corrosion prediction, feature selection, civil engineering

1 Introduction

Data Mining (DM) has been driven by the need to solve practical problems since its inception [20]. In order to achieve a greater usability of DM models, there are three main phases in the lifecycle of a data mining project: (1) training of the model, (2) evaluation (or testing) of the model and (3) using the final trained model in practice. The third phase is usually carried out by business managers or a typical user of the system. A number of Knowledge Discovery and Data Mining (KDDM) process models have been established to organize the lifecycle of a DM project within a common framework. The existing KDDM process models end up with the deployment phase in which rules and patterns inferred from the trained models are utilized in decision making [6]. These process models, however, do not consider the utilization of trained model as a prediction tool in real use for prediction purpose.

DM has been successfully applied in many areas such as marketing, medical and financial [13]. Civil engineering is also one of the areas where a variety of successful real-world data mining applications are reported in building construction [5, 9, 12, 14, 17-19, 21-23, 26]. One such application is metallic corrosion prediction in buildings. These applications can be classified into two main categories: 1) building the predictive models using various data mining techniques [5, 9, 12, 17, 18, 21-23, 26];

and 2) improving the prediction accuracy using new hybrid methods [14, 19]. All of these predictive models assume that the inputs that users will provide in *using* the model are the same as the input features used in *training* the models. However, if users have information of limited inputs only, the predicted results will not be as good as they were during the training and evaluation phases of the data mining system. In other words, the performance of the predictive model degrades due to the absence of many input values. A major problem that still needs to be solved is how to select appropriate features to build the model for a real situation when users have information on limited inputs only.

A considerable body of research has emerged to address the problems of selecting the relevant features from a set of features. The existing feature selection algorithms can be grouped into two broad categories: filters and wrappers [2, 15, 27] for finding the best subset for maximizing classification accuracy [7] [11] [16] [1]. Filters are independent of the inductive algorithm and evaluate features based on the general characteristics of the data, whereas wrappers evaluate features using the inductive algorithm that is finally employed for learning as well. However, these algorithms do not suffice to address our problem. These algorithms can not be used to remove relevant features while the classification accuracy is still acceptable and, in particular, it is known that users in practice would not be familiar with these features while using the system in predictive modeling.

This paper deals with the problem of using the data mining models in a real-world situation where the user can not provide all the inputs with which the model is built. We have developed a learning system framework, called as Query Based Learning System (QBLS), for improving the performance of the predictive models in practice where not all inputs are available for querying to the system. The automatic feature selection algorithm, called as Query Based Feature Selection (QBFS) is developed for selecting features to obtain a balance between the relative minimum subset of features and the relative maximum classification accuracy. This algorithm is evaluated on a number of synthetic data sets and a real-world application of the lifetime prediction of metallic components in buildings.

2 Query Based Learning System

The proposed Query Based Learning System (QBLS) (presented in Figure 1) is a data mining process model based on the most popular industry standard model, CRISP-DM (Cross Industry Standard Process for Data Mining)[6]. The QBLS model consists of nine phases structured as sequences of predefined steps. Three procedures that are different from the CRISP-DM are highlighted. These three procedures - Query Based Feature Selection (QBFS), Results Post-processing and Model in Use - are critical for the success of the proposed QBLS model. The QBFS is separated from the data pre-processing step as it has the involvement of users or domain experts and hence is different from the usual feature selection. The basic idea of the QBFS is to select a minimum subset of relevant features with which the predictive model provides an acceptable performance, as well as, to make the selected features available to users when the model is used in practice. Section 3 will discuss this further.

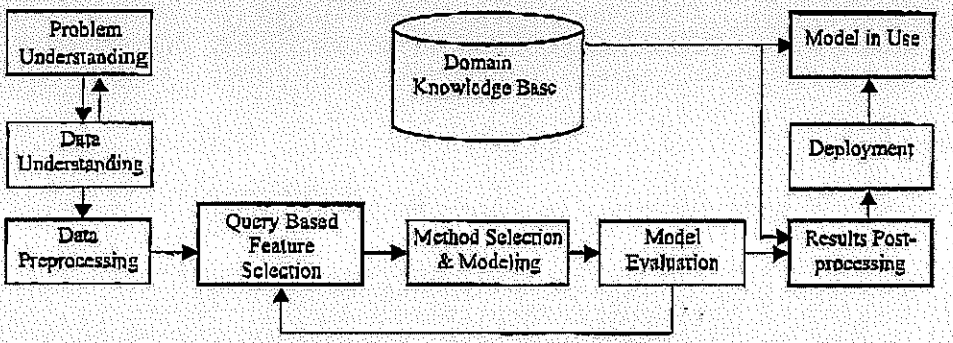

Fig. 1. Query based learning system

In the proposed process model QBLS, a domain knowledge base assists to deal with the vague queries in the "Model in Use" phase and with eliminating illogical outcomes in post-processing. Some features included in the final model may not be directly provided by users but can be inferred by the domain knowledge base. For example, "annual rainfall" is an important factor in determining the service life of building components in civil engineering. While using the DM model to predict the service life of a building component, the user will most likely provide the location and material as an input. The user may not be aware of the exact value of rainfall in the area. However, a domain knowledge base will have such information. This information can now be treated as one of the input values for the model. Furthermore, the domain knowledge base can be used in reinforcing the outputs inferred by the predictive model. Since the real-world DM models are for solving practical problems, the final result is critical to users. However, mining errors are inevitable even for a perfect model. The domain knowledge base is used to confirm that the results predicted by the data mining system do abide by the rules of the domain and/or domain experts. For example, it is domain knowledge in civil engineering that (1) a roof in a severe marine location will not last longer than one in a benign environment, and (2) a stainless steel roof should last longer than one with galvanized steel. Such in-built rules are checked to ensure the correctness of the results processed by models.

3 Query Based Feature Selection (QBFS)

The basic idea of QBFS is to obtain the minimum subset of features that will produce an accurate predictor (numerical prediction) by improving the performance of the QBLS when used in practice where not all inputs are available for querying.

3.1 Algorithms

The first step of QBFS involves removing the features with no mining value such as identification features. The remaining features, $A = \{a_1, a_2, \ldots, a_k, a_{k+1}, \ldots, a_m, a_{m+1}, \ldots, a_n\}$ are clustered into three groups according to their easy availability to users.

- Group 1 (a_1 - a_l): Features that user can easily provide while using the model.
- Group 2 (a_{l+1} - a_m): Features that can not be provided by the user but can be obtained from the external domain knowledge.
- Group 3 (a_{m+1} - a_n): Features that can not be provided by the user or obtained from domain knowledge.

Group 1 will be included in the final model because features in Group 1 are not only useful in mining but can also be provided by users while they are using the model. Group 3 will be rejected because they can not be provided in model use although they may have mining value. If we include the features of Group 3 in the final model, their values in new data will be missing. As a result, the generalization accuracy will decrease. A decision has to be made for features in Group 2, as they can not be provided by users but they can be obtained from external domain knowledge. If we include all the features of Group 2, the measurements to obtain some of these values may be too complex and computationally expensive. If we exclude those features, the performance of the model may not be accepted by users.

For QBFS to be commonly used or to be used in applications in which the expertise is not available to guide the categorisation of features into three groups, the size of Group 1 and Group 3 is reduced to zero so that all available features become the Group 2 members and can now be tested for their relevancy to the target feature. The three variations of QBFS - QBFS-F, QBFS-W, and QBFS-FW (described below) - are applied to the features of Group 2 for selecting a minimum subset.

QBFS-F. QBFS-F is a correlation-based filter algorithm based on the concept that a feature is good if it is highly correlated to the target feature but not highly correlated to any of the other features [28]. Features in Group 1 are already chosen so they become a starting subset to remove the features of Group 2. A Group 2 feature is removed if the level of correlation between any of the Group 1's features and the Group 2 feature is high enough to cause redundancy. Moreover, if the features in Group 2 are highly correlated to each other, one of them, which is more redundant to Group 1's features, is removed. The Pearson's correlation (R) is used to estimate the level of correlation between two features as it is a symmetrical measure and there is no notion of one feature being the "class". When two features (X and Y) are continuous, the standard linear (Pearson's) correlation is calculated (Equation 1) where $\bar{x}_i$ is the mean of X and $\bar{y}_i$ is the mean of Y features.

$$R_{XY} = \frac{\sum_i (x_i - \bar{x}_i)(y_i - \bar{y}_i)}{\sqrt{\sum_i (x_i - \bar{x}_i)^2} \sqrt{\sum_i (y_i - \bar{y}_i)^2}} \qquad (1)$$

When one feature is continuous and the other is discrete, a weighted Pearson's correlation is calculated (Equation 2) where p is the prior probability that X takes value x_i and X_{hi} is a binary feature that takes value 1 when X has value x_i and 0 otherwise.

algorithms maintain the accuracy while the CFS and ReliefF algorithms perform very poorly in a practical situation. Even in some cases (e.g. CFS and ReliefF for Holistic-III_Zi and ReliefF for Holistic-II), the accuracy is reduced to zero or minus. This indicates that the good generalization accuracy can be obtained in real situations with the use of QBFS, while the existing algorithms can not deal with the problem of selecting appropriate features for building the model in a real-world situation in which the user can not provide all the information with which the model is being trained.

Table 3. Correlation Coefficient (CC) of M5 on selected features of real-world datasets

Dataset	QBFS-F		QBFS-W/FW		CFS		ReliefF		Full Set	
	10-CV	test	10-CV	test	10-CV	test	10-CV	test	10-CV	test
Delphi Survey	0.9198	0.9016	0.9198	0.9016	0.8577	0.8850	0.6812	0.6581	0.9198	0.9016
Holistic-I	0.9790	0.9746	0.9764	0.9672	0.9081	0.9568	0.6543	0.7709	0.9790	0.9746
Holistic-II	0.8421	0.8796	0.9973	0.9978	0.9994	0.3037	1	-0.7067	1	-0.8349
Holistic-III_Ga	0.9393	0.9321	0.9393	0.9321	0.9948	-0.633	0.9913	0.3403	0.9883	0.4756
Holistic-III_Zi	0.8801	0.9449	0.8801	0.9449	0.9969	0	0.9971	0	0.9971	0
Average	**0.9121**	**0.9266**	**0.9426**	**0.9487**	**0.9514**	**0.3025**	**0.8648**	**0.2125**	**0.9768**	**0.3034**

* 10-CV denotes CC on 10-CV, test denotes CC on a specified test set

The UCI Datasets. The details of the datasets taken from the UCI repository are presented in Table 4. Since we do not have domain experts in these datasets to guide the categorization of three groups, the size of Group 1 and Group 3 are reduced to zero. All available features are kept in Group 2.

Table 5 shows the total number of features selected by the algorithms for each dataset and their selected feature ID. Table 6 presents the prediction accuracy using M5 on 10-CV. From Table 5, we observe that on average, both of the QBFS-F and QBFS-W algorithms reduce more than half of the features (especially for QBFS-W, which reduces features to 2.5) while CFS and ReliefF select more features (4.75 and 4.25). From Table 6, we observe that the CC of all algorithms on average is very close to the CC when calculating with the full set. That means all algorithms can maintain the accuracy after feature reduction. QBFS-FW has achieved 0.9198 of CC in spite of using 4.5 features only. On the contrary, CFS only achieves 0.9175 of CC in spite of using 4.75 features and ReliefF achieves 0.8796 of CC in spite of using 4.25 features. QBFS-F and QBFS-W select relatively fewer features and achieves relatively higher accuracy. The above experimental results suggest that the QBFS algorithm is a practical solution to the domain driven data mining problems. Moreover, for all datasets, QBFS performs comparably with two representative feature selection algorithms CFS and ReliefF.

ReliefF selects more features. There are some overlapping between the selected features by QBFS, CFS and ReliefF while for QBFS-F, QBFS-W and QBFS-FW, the selected features are almost the same.

Table 1. Details of real-world datasets

Dataset	I	F	BC	BM	T
Delphi Survey	683	10	Roofs / Gutters / Others	Galvanized Steel / Zincalume / Colorbond / Others	Mean
Holistic-I	9640	11	Gutters	Galvanized Steel / Zincalume	MLannual
Holistic-II	4780	20	Gutters	Colorbond	Life of gutter at 600um
Holistic-III	1297	18	Roofs	Galvanized Steel / Zincalume	Zincalume Life
					Galvanized Life

I: number of instances; F: number of features; BC: building component;
BM: building material; T: target feature

Table 2. Total number and feature ID of selected features on real-world datasets

Data Set	QBFS			CFS	ReliefF	Full Set
	QBFS-F	QBFS-W	QBFS-FW			
Delphi Survey	5 (1,2,4,5,6)	5 (1,2,4,5,6)	5 (1,2,4,5,6)	3 (2,4,6)	1 (4)	5
Holistic-I	6 (2,3,6,8,9,10)	5 (2,3,8,9,10)	5 (2,3,8,9,10)	4 (3,6,9,10)	4 (2,3,6,10)	6
Holistic-II	4 (2,3,7,12)	5 (2,3,7,12,14)	5 (2,3,7,12,14)	3 (2,20,21)	10 (2,3,4,13, 14,15,18-21)	14
Holistic-III_Ga	3 (3,4,9)	3 (3,4,9)	3 (3,4,9)	6 (3,9,10,11,13,15)	10 (3,4,6,7,10, 11,12,13,15,16)	12
Holistic-III_Zi	3 (3,4,9)	3 (3,4,9)	3 (3,4,9)	4 (3,8,9,14)	5 (3,4,8,9,14)	6
Average	**4.2**	**4.2**	**4.2**	**4**	**6**	**8.6**

In order to examine the effect of these selections on accuracy, we apply M5 on the selected features. Table 3 shows the Correlation Coefficient (*CC*) using M5 on 10-CV and a specified test set. This specified test set only includes those features whose values can be provided by users and leaving other features as missing values. For example, CFS chose 4 features (Longitude, ZincalumeMassLoss, Marine and N) for Holistic-III_Zi to build the model. When users utilise this model, they only can input "Longitude" and "Marine", leaving "ZincalumeMassLoss" and "N" as missing values. Therefore, the values of "ZincalumeMassLoss" and "N" in the specified test set for CFS should be omitted. Such test set reflects predictive performance of the models on unseen cases in a practical scenario.

Based on the CC on 10-CV of each algorithm, we observe that the learning accuracy is very close to when using the full set. This indicates the ability of QBFS, CFS and ReliefF to identify redundant features. On the other hand, based on the CC on the specified test set of each algorithm, we find that only the proposed QBFS

$$R_{XY} = \sum_{i=1}^{k} p(X = x_i) R_{X_i Y} \qquad (2)$$

When both features are discrete, all weighted correlations are calculared for all combinations [11] (Equation 3).

$$R_{XY} = \sum_{i=1}^{k} \sum_{j=1}^{l} p(X = x_i, Y = y_j) R_{x_i y_j} \qquad (3)$$

The QBFS-F algorithm as shown in Figure 2 consists of two major parts. The first part removes features in Group 2 that are highly correlated to features in Group 1 (Step 3) and the second part removes features in Group 2 that are highly correlated to each other (Step 6). The correlation level between a feature a_i and a class C, denoted as Ra_i,c is used as reference to determine whether the feature is highly correlated. For each feature a_i in Group 2, if its correlation to any features a_j in Group 1 (Ra_j,a_i) is equal to or greater than Ra_i,c, it will be removed. The remaining features in Group 2 are sorted in ascending order based on their Ra_j,a_i values so the first one is the least correlated to any features in Group 1. If any two features are highly correlated to each other and one of them needs to be removed, the one with bigger Ra_j,a_i values is removed. The bigger Ra_j,a_i value indicates the higher possibility of this feature to be redundant to Group 1's features.

If the size of group 1 is zero, the algorithm is simplified as only the second part. In such a case, the feanures in Group 2 are sorted in descending order based on the Ra_i,c values. If any two features are highly correlated to each other, the one with the smaller Ra_i,c value is removed.

QBFS-W. As the QBFS-F algorithm does not take into account the inductive algorithms, the features it has chosen might not be appropriate for a learning method. The proposed wrapper algorithm, called as QBFS-W involves an inductive algorithm as the evaluation function. As a wrapper method includes an iterative step for retraining the model, the run time for QBFS-W is inevitably long. The forward selection heuristic search strategy [2, 7, 8] is employed to reduce the time complexity. Forward selection starts from the empty set, and each iteration generates new subsets by adding a feature selected by some evaluation function [7].

The algorithm (Figure 3) begins a search at the Group 1's features and adds the p 2's features one at a time. If the size of Group 1 is zero, it begins the search with the empty set of features. Each time a feature with the best performance is chosen and then this feature is removed from Group 2. The loop is terminated when an acceptable model with the performance $\geq \delta$ is built with the minimum set of features or all Group 2's features are chosen. The stopping criterion δ is a predefined threshold ($0 < \delta < 1$) using Correlation Coefficient (CC) as performance measures. A small δ is associated with a high probability of removing relevant features. This parameter is fine-tuned empirically to provide good performance. The worst case is that in which all features in Group 2 are included in the feature set. Suppose the size of Group 2 is N, the search space of this algorithm is $O(N!)$ for the worst case.

Input:
D: the whole data set
A_1: the features of Group 1 $\{a_1, \ldots, a_k\}$
A_2: the features of Group 2 $\{a_{k+1}, \ldots, a_m\}$
C: the target feature
Output:
S_{min}: the minimum subset

1. **For** $i = k+1$ **to** m
   Calculate Ra_i,c for a_i
   End
2. $A_2' = \emptyset$
3. **For** $i = k+1$ **to** m
   For $j = 1$ **to** k
   Calculate Ra_j,a_i for a_i
   If $Ra_j,a_i \geq Ra_i,c$
   Remove a_i
   Exit For
   End
   End
   If a_i is not removed
   Append a_i to A_2'
   Keep the biggest Ra_j,a_i for a_i
   End
4. Sort A_2' in ascending values of Ra_j,a_i
5. $a_p \leftarrow$ get the first element of A_2'
6. **Do begin**
   $a_q \leftarrow$ get the next element of A_2' following a_p
   Do begin
   Calculate Ra_p,a_q
   If $Ra_p,a_q \geq Ra_q,c$
   Remove a_q from A_2'
   $a_q \leftarrow$ get the next element of A_2' following a_q
   Until $a_q = NULL$
   $a_p \leftarrow$ get the next element of A_2' following a_p
   Until $a_p = NULL$
7. $S_{min} = A_2'$

Fig. 2. QBFS-F algorithm

QBFS-FW. Wrappers and filters can complement each other to propose a better method, in that filters search through the feature space efficiently while the wrappers provide good accuracy [29]. The algorithms combining filters and wrappers usually choose some best subsets using a goodness measure and then exploit cross validation to decide a final best subset across different cardinalities [28]. The proposed QBFS-FW algorithm (shown in Figure 4) tries to combine the advantages of both methods.

QBFS-FW first calculates the correlation level (Ra_i,c) between each feature a_i in Group 2 and target feature C and sorts Group 2's features in descending order based on their Ra_i,c so that the most relevant feature is positioned at the beginning of the list. Then it employs backward elimination [7, 8] to remove the Group 2's features one at a time. The algorithm attempts to keep the features that are strongly relevant to the target feature in the selected subset. It also reduces the time complexity to be linear. Since the time for calculating Ra_i,c can be ignored compared to the model training time, the search space of this algorithm is $O(N)$ for the worst case where N is the size of Group 2.

Input:
D: the whole data set
A_1: the features of Group 1 $\{a_1, \ldots, a_k\}$
A_2: the features of Group 2 $\{a_{k+1}, \ldots, a_m\}$
C: the target feature
δ: a predefined threshold
Output:
S_{min}: the minimum subset

1. $S_{min} = A_1$
2. Train the model with S_{min} and keep the performance P
3. If $P \geq \delta$, **Return** S_{min}
4. $\underline{Q} = A_2$
5. While $\underline{Q} \neq \phi$
   1) For each $q \in \underline{Q}$
      Set $S' \leftarrow \{q\} \cup S_{min}$
      Train the model with S' and note the performance P
   2) Set $S_{min} \leftarrow \{q^*\} \cup S_{min}$ where q^* corresponds to the best P obtained in step 5.1
   3) If $P \geq \delta$, **Return** S_{min}
      Else
      Set $\underline{Q} \leftarrow \underline{Q} \setminus \{q^*\}$
      End

Fig. 3. QBFS-W algorithm

3.2 Evaluation

The proposed algorithms are evaluated in terms of number of selected features and prediction accuracy. The experiments were performed on a real-world dataset for predicting lifetime of metallic components. Other datasets from the UCI collection [3] are also used in experiments for comparison purposes. The model tree algorithm (M5) [24] is chosen as inductive algorithms for QBFS-W and QBFS-FW algorithm. Two representative feature selection methods - CFS [11] and ReliefF [16] - are chosen for

comparison as they are leading algorithms. For ReliefF, we set $m = 250$ (the number of instances sampled), $k = 10$ (number of nearest neighbors) for discrete class data and 200 for numeric class data, $\sigma = 20$ (a parameter that controls the influence of nearest neighbors) and $\delta = 0.01$ (a threshold by which features can be discarded). Tenfold cross validation (10-CV) was used throughout the experiments.

Input:
D: the whole data set
A_1: the features of Group 1 $\{a_1, \ldots, a_k\}$
A_2: the features of Group 2 $\{a_{k+1}, \ldots, a_m\}$
C: the target feature
δ: a predefined threshold
Output:
S_{min}: the minimum subset

1. $S_{min} = A_1 + A_2$
2. **For** $i = k+1$ **to** m
   Calculate Ra_i,c for a_i
   End
3. Sort A_2 in descending values of Ra_i,c
4. $a_p \leftarrow$ get the last element of A_2
5. **Do begin**
   Set $S' \leftarrow S_{min} \setminus \{a_p\}$
   Train the model with S'
   If the performance $P \geq \delta$, remove a_p from S_{min}
   $a_p \leftarrow$ get the next element of A_2 preceding a_p
   Until $a_p == NULL$ or $P < \delta$
6. **Return** S_{min}

Fig. 4. QBFS-FW algorithm

The Real-world Dataset. The objective is to predict the service life of metallic components in Queensland school buildings based on the information from multiple sources. The details of these input datasets are presented in Table 1. The datasets include four different sources of service life information from the Delphi Survey, Holistic- I, -II and –III, where Holistic-III was divided into two parts in terms of different target features. The multiple sources are independent but complementary to each other. These sources can not be combined and the models are required to be constructed independently from each of them. Holistic-I, -II and -III relate to different component types with different materials while Delphi contains all component types with all materials. Each data source contains completely different features in which some can not be provided by users or obtained from domain knowledge. Features of each data source are divided into three groups (as defined in section 3.1) after the consultation with domain experts before applying QBFS.

For each of the data sources, we run all the three algorithms of QBFS, CFS and ReliefF. Table 2 presents the total number of features selected by the three algorithms for each of the datasets and their selected feature ID. The results in Table 2 reveal that on an average, both QBFS and CFS reduce more than half of the features while

Table 4. Details of UCI datasets

ID	Dataset	Instances	Features	Target Feature
1	autoMPG	398	9	mpg
2	CPU	209	10	PRP
3	housing	506	14	medv
4	servo	167	5	class

Table 5. Total number and feature ID of selected features on UCI datasets

Dataset	QBFS			CFS	ReliefF	Full Set
	QBFS-F	QBFS-W	QBFS-FW			
autoMPG	3 (5,6,7)	2 (3,7)	5 (2,3,4,5,7)	7 (2-8)	6 (2,3,4,5,7,8)	7
CPU	5 (4,5,6,7,8)	2 (5,6)	3 (4,5,6)	5 (4,5,6,7,8)	5 (4,5,6,7,8)	7
housing	4 (4,6,11,13)	3 (6,9,13)	6	4 (4,6,11,13)	5 (1,4,6,11,13)	13
servo	3 (1,2,3)	3 (1,2,3)	4 (1,2,3,4)	3 (1,2,3)	1 (3)	4
Average	**3.75**	**2.5**	**4.5**	**4.75**	**4.25**	**7.75**

Table 6. Correlation Coefficient (CC) of M5 on selected features of UCI datasets

Dataset	QBFS-F	QBFS-W	QBFS-FW	CFS	ReliefF	Full Set
autoMPG	0.9213	0.9033	0.9228	0.9258	0.9230	0.9258
CPU	0.9420	0.9149	0.9248	0.9420	0.9420	0.9242
Housing	0.8767	0.8756	0.8961	0.8767	0.8977	0.9131
Servo	0.9254	0.9254	0.9356	0.9254	0.7558	0.9356
Average	**0.9164**	**0.9048**	**0.9198**	**0.9175**	**0.8796**	**0.9247**

4 Implementation of QBLS

In this section, we deploy the Query Based Learning System in a real-world application of the lifetime prediction of metallic components in buildings.

4.1 Overview of the System

The proposed QBLS system (illustrated in Figure 5) basically consists of three main parts: feature selection, predictors and domain knowledge. The QBFS algorithm is first applied to the datasets to select a minimum subset of features which can be provided by users. A data mining method (we discuss the method selection in next section) is applied on selected features to build the predictors for all of the datasets. The predictors are used to carry out prediction for user input queries. The domain knowledge base consists of three parts: salt deposition knowledge, rainfall knowledge

and generalized rules extracted from domain expert opinions. Because the features selected to build the predictors include features of "Salt Deposition" and "Rainfall Annual", the salt deposition and rainfall database is included in the knowledge base, which is for pre-processing user inputs. Generalized rules are used in post-processing the predicted results, for example, solving the inconsistency in predicted results.

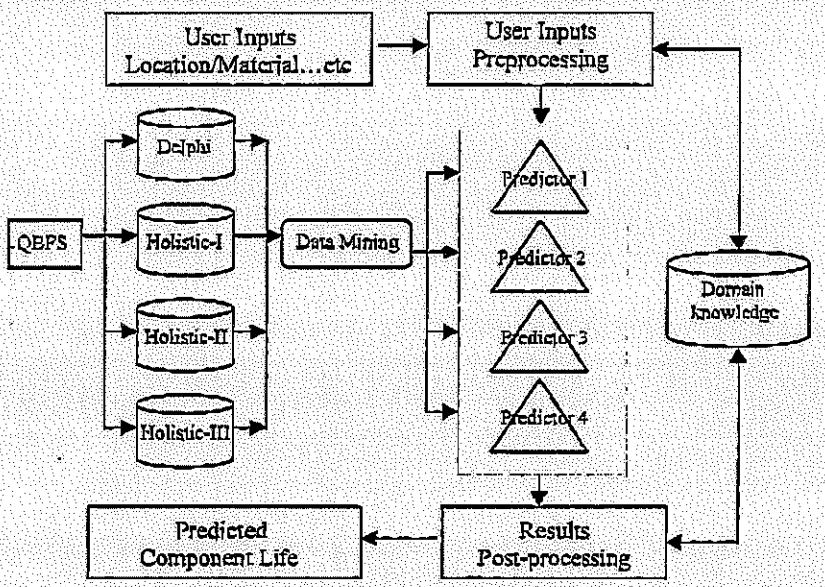

Fig. 5. Overview of the system

4.2 Representation of Knowledge Base

The knowledge base is implemented as relational database in which the knowledge is represented as items in the database. Examples of the salt deposition knowledge and rainfall knowledge in the generated knowledge base are presented in Tables 7 and 8 respectively. There are total 18 generalized rules in the generated knowledge base and some of them are presented in Table 9.

As the location (longitude and latitude) that users input in the queries may not exactly match the salt deposition and rainfall knowledge, a similarity principle is employed to obtain the value of salt deposition and rainfall. The similarity principle means that the nearest geographic location will have the most similar value for salt deposition and annual rainfall. Once the user inputs longitude and latitude, the system finds the nearest location from the knowledge base and then gets the value of salt deposition and rainfall. These values can then be treated as user inputs for the predictors.

In terms of the predicted results, the system checks them with the generalised rules. If the component, material and environment are matched and the predicted service life is in the range, the results are considered reasonable and presented to users. Otherwise the system gives a message that the result does not abide by the

generalised rules with presenting the values instead of removing this unreasonable result.

Table 7. Salt Deposition Knowledge

XLong	YLat	Salt Deposition
151.986	-28.0373	3.80842

Table 8. Rainfall Knowledge

XLong	YLat	Rain Annual (mm)
151.986	-28.0373	1595

Table 9. Generalised Rules

Component	Environment	Material	Min (years)	Max (years)
Gutters	Marine	Galvanised Steel	5	15
Gutters	Benign	Colorbond	20	50

4.3 Method Selection and Modeling

There are various data mining methods like Naïve Bayes, K-Nearest Neighbors (K-NN), Decision Tree (DT) [28] and Neural Network (NN) that can be considered to do prediction tasks. Ge et al. [10] have reported the detailed method to choose the best data mining algorithm for building predictors. The results show that the DT and naive bayes methods on the discretised output perform poorly [10]. Three best methods are M5 for Delphi Survey, KNN for Holistic-I and NN and M5 for Holistic-II and III. Considering the balance between accuracy and comprehensibility of predictors, M5 is chosen as the learning method in QBLS.

Since the notion of QBLS is user query based, we combine the model-based learning (M5) with the instance-based learning [25] to improve the performance. This method first uses the instance-based approach to find a set of instances similar to the target instance. Then the class values of similar instances are adjusted using the value predicted by the model tree before they are combined. We use the KNN (k=3) for the instance-based method.

Therefore, the final predictors are built using M5+KNN on the features selected by QBFS. The performance of the predictors is presented graphically in Figures 7 and 8. The performance of this M5+KNN combined model is compared with the M5 model and the ensemble model with bagging [4]. Figures 7 and 8 show that the better correlation coefficient and lower mean absolute error are obtained by combining the M5 and KNN learning methods. The method seems to provide significant improvement for relatively weaker models like Holistic-II and Holistic-III_Zi, whereas the improvement for the near-perfect models such as Holistic-I, is not so obvious. The combined M5+KNN model also outperforms the ensemble model with bagging.

D: Delphi H-I: Holistic-I H-II: Holistic-II H-III_G: Holistic-III for Galvanized Steel H-III_Z: Holistic-III for Zincalume

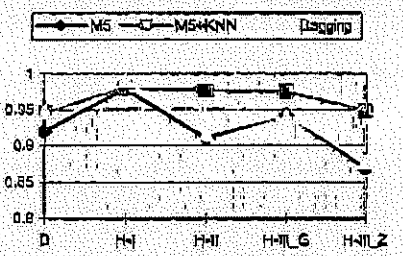

Fig. 7. Correlation Coefficient of M5, M5+KNN and bagging

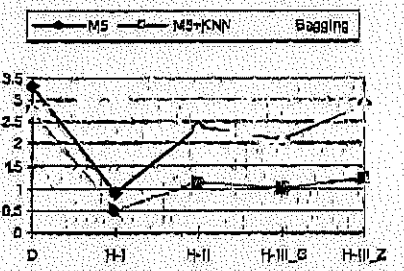

Fig. 8. Mean Absolute Error of M5, M5+KNN and bagging

5 Conclusions

This paper develops a new learning system framework, called as Query Based Learning System (QBLS) for improving the performance of the predictive models in practice where not all inputs are available for querying to the system. This paper also presents a new feature selection algorithm, called as Query Based Feature Selection (QBFS) for selecting the features according to the interest of domain expert or user, while maintaining the accuracy of the predictive model. The available features can all be grouped into three types: 1) that can be provided by users 2) that can be obtained from domain knowledge 3) neither 1) nor 2). External domain knowledge is successfully used for dealing with incomplete and vague queries in pre-processing. It is also used in dealing with inconsistency of the predicted service life from different predictors in post-processing. This novel use of domain knowledge improves the prediction accuracy when users can not provide all inputs.

The proposed Query Based Feature Selection algorithm is compared with two representative feature selection methods CFS and ReliefF. The results prove that QBFS outperforms the existing methods in selecting the features of domain-driven and expert-driven data mining problems. The use of query based feature selection procedure in QBLS indicates that feature selection is not only used for removing irrelevant features but also greatly assists in choosing features according to the use in practice. If a feature, which belongs to type 3 that is it can not be available for user, is included in the final model, the performance of model reduces significantly in using the model. Therefore, such features, even if they are useful in mining, should be rejected. The proposed Query Based feature selection may result in some useful features being rejected. This may reduce the performance of the predictive models. We show that the integrated method combining M5 (model-based) and KNN (instance-based) is successfully applied in such cases for improving performance.

Acknowledgement. Authors would sincerely like to thank the WEKA developers and owners to make use of WEKA codes in this project. We would like to thank CRC-CI to provide us the financial support, and to Penny Corrigan and Michael Ball to provide us the data and feedback during intermittent progress.

References

1. Almuallim, H. and Dietterich, T.G., *Learning Boolean Concepts in the Presence of Many Irrelevant Features.* Artificial Intelligence, 1994. 69(1-2): p. 279-305.
2. Bengio, S., *Statistical Machine Learning from Data Feature Selection.* 2006: Matigny, Switzerland.
3. Blake, C., Keogh, E., and Merz, C.J., *UCI Repository of Machine Learning Data Bases.* 1998: Irvine, CA: University of California, Department of Information and Computer Science.
4. Breiman, L., *Bagging Predictors.* Machine Learning, 1996. 24(2): p. 123-140.
5. Brence, J.R. and Brown, D.E., *Data mining corrosion from eddy current non-destructive tests.* Computers & Industrial Engineering, 2002. 43(4): p. 821-840.
6. CRISP-DM, *Cross Industry Standard Process for Data Mining.* 2003.
7. Dash, M., & Liu, H, *Feature selection for classification.* Intelligent Data Analysis, 1997(1): p. 131-156.
8. Devijver, P.A. and Kittler, J., *Pattern Recognition: A Statistical Approach.* 1982: Prentice Hall.
9. Furuta, H., Deguchi, T., and Kushida, M. *Neural network analysis of structural damage due to corrosion.* in *Proceedings of ISUMA - NAFIPS '95 The Third International Symposium on Uncertainty Modeling and Analysis and Annual Conference of the North American Fuzzy Information Processing Society.* 1995.
10. Ge, E., Nayak, R., Xu, Y., and Li, Y. *Data Mining for Lifetime Prediction of Metallic Components (AusDM 2006).* in *The Australasian Data Mining Conference.* 2006. Sydney.
11. Hall, M.A. *Correlation-based Feature Selection for Discrete and Numeric Class Machine Learning.* in *Proc. 17th International Conf. on Machine Learning.* 2000: Morgan Kaufmann, San Francisco, CA.
12. KamrunNahar, M. and Urquidi-Macdonald, M. *Data mining of experimental corrosion data using Neural Network.* in *208th Meeting of the Electrochemical Society, Oct 16-21 2005.* 2005. Los Angeles, CA, United States: Electrochemical Society Inc., Pennington, NJ 08534-2896, United States.
13. Kantardzic, M. and Zurada, J., *Next Generation of Data-Mining Applications.* 2005: Wiley-IEEE Press.
14. Kessler, W., Kessler, R.W., Kraus, M., Kubler, R., and Weinberger, K. *Improved prediction of the corrosion behaviour of car body steel using a Kohonen self organising map.* in *Advances in Neural Networks for Control and Systems, IEE Colloquium on.* 1994.
15. Kohavi, R. and John, G.H., *Wrappers for Feature Subset Selection.* Artificial Intelligence, 1997. 97(1-2): p. 273-324.

16. Kononenko, I. *Estimating Attributes: Analysis and Extensions of RELIEF*. in *European Conference on Machine Learning*. 1994.
17. Leu, S.-S., Chen, C.-N., and Chang, S.-L., *Data mining for tunnel support stability: neural network approach*. Automation in Construction, 2001. **10**(4): p. 429-441.
18. Melhem, H.G. and Cheng, Y., *Prediction of remaining service life of bridge decks using machine learning*. Journal of Computing in Civil Engineering, 2003. **17**(1): p. 1-9.
19. Melhem, H.G., Cheng, Y., Kossler, D., and Scherschligt, D., *Wrapper Methods for Inductive Learning: Example Application to Bridge Decks*. Journal of Computing in Civil Engineering, 2003. **17**(1): p. 46-57.
20. Melli, G., Zaiane, O.R., and Kitts, B., *Introduction to the special issue on successful real-world data mining applications*. SIGKDD Explor. Newsl., 2006. **8**(1): p. 1-2.
21. Mita, A. and Hagiwara, H. *Damage Diagnosis of a Building Structure Using Support Vector Machine and Modal Frequency Patterns*. in *Smart Structures and Materials 2003: Smart Systems and Nondestructive Evaluation for Civil Infrastructures, Mar 3-6 2003*. 2003. San Diego, CA, United States: The International Society for Optical Engineering.
22. Morcous, G., Rivard, H., A.M.ASCE, Hanna, A.M., and F.ASCE, *Modeling Bridge Deterioration Using Case-based Reasoning*. Journal of Infrastructure Systems, 2002. **8**(3): p. 86-95.
23. Morcous, G., Rivard, H., and Hanna, A.M., *Case-Based Reasoning System for Modeling Infrastructure Deterioration*. Journal of Computing in Civil Engineering, 2002. **16**(2): p. 104-114.
24. Quinlan, J.R. *Learning with Continuous Classes*. in *5th Australian Joint Conference on Artificial Intelligence*. 1992.
25. Quinlan, J.R. *Combining instance-based and model-based learning*. in *Proceedings of the Tenth International Conference on Machine Learning*. 1993. Amherst, Massachusetts: Morgan Kaufmann.
26. Skomorokhov, A.O. *A knowledge discovery method - APL implementation and application*. in *Proceedings of the APL Berlin 2000 Conference, Jul 24-27 2000*. 2000. Berlin, Germany: Association for Computing Machinery.
27. Tang, W. and Mao, K., *Feature Selection Algorithm for Data with Both Nominal and Continuous Features*, in *Advances in Knowledge Discovery and Data Mining*. 2005, Springer Berlin / Heidelberg. p. 683-688.
28. Yu, L. and Liu, H. *Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution*. in *Proceedings of The Twentieth International Conference on Machine Leaning (ICML-03)*. 2003. Washington, D.C.
29. Zexuan, Z., Ong, Y.-S., and Dash, M., *Wrapper-Filter Feature Selection Algorithm Using A Memetic Framework*. IEEE Transactions on System, Man, and Cybernetics, Part B.