Application of Machine Learning Techniques to Design Style Classification

Aruna Lorensuhewa
s.lorensuhewa@qut.edu.au

Binh Pham
Center for Information Technology Innovation,
Faculty of Information Technology, QUT
Brisbane, Australia.
b.pham@qut.edu.au

Shlomo Geva
s.geva@qut.edu.au

Abstract

Fuzzy knowledge exists in domains such as medicine, law and design as it is difficult to retrieve all knowledge and experience from experts. Machine learning and data mining techniques can be used to automatically extract knowledge from unstructured information sources. The aim of this research is to develop a generic framework and methodologies that will facilitate knowledge extraction automatically by using machine learning and data mining techniques and integrating with expert knowledge. We show that classification accuracy can be improved by integrating expert knowledge with other machine learning classifiers, SVM and Nearest Neighbour.

Keywords
Data Mining, Feature selection, Classification, Support Vector Machine, Decision Trees, C4.5, Design Style.

1 INTRODUCTION

Fuzzy knowledge exists in domains where it is difficult to retrieve all knowledge and experience from experts. For instance, expertise may not be easily expressible, it may be incomplete, or may exist at a subconscious level. Experts have often gained or improved their expertise from experience, by dealing with concrete cases, reading literature. Sometimes no one has a complete understanding of the domain. In such cases we can gather information from experts through questionnaires, domain specific databases and literature. The challenge is to derive structured knowledge in an automatic fashion from unstructured sources and augment this with available expert knowledge.

Machine learning and data mining techniques have been used to automatically extract knowledge from unstructured information sources. Commonly used algorithms are C4.5 [1], Support Vector machine (SVM) [2], Nearest Neighbour [5] and Neural Networks.

On Design, some experts believe that design style is an intangible concept and that its knowledge is difficult to present in a formal way. So far, there has been no computer supported automatic technique to assist novice designers in learning to distinguish design styles or judge how similar a design is to a specific style.

The aim of this research is to develop a generic framework and methodologies that will enable knowledge extraction in an automatic fashion by using machine learning and data mining techniques, then integrating expert knowledge with extracted knowledge. Furniture Design style has been selected as the domain for the evaluation of the framework.

Data is collected for seven different styles: Chippendale, Classical, Jacobean, Early Victorian and Queen Anne. A Web based questionnaire (Figure. 1) is used to collect data from users and domain experts. In total fifteen different features were examined, including appearance, chair arms, back shape, leg type, seat shape, etc. Most of the features are categorical. For example, Foot can have options such as lion, ball, pad, drake and block etc. The Connectedline Furniture Design Style Guide database [3] is commercially available software for the Windows platform. This guide identifies and dates about 20 furniture styles and their distinctive features. This database is used to create an expert classifier for this research.

The dataset collected from the experiment has categorical data fields characterized by a number of distinct values. On the other hand, the dataset we collected has characteristics such as uncertainty, incompleteness and imprecision.

The total framework is divided into three main phases: selecting an encoding scheme and classifiers, feature reduction and weight assignment, and integrating expert knowledge with knowledge extracted from machine learning techniques.

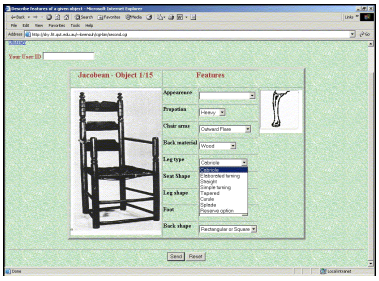

Figure 1. Web based questionnaire of the experiment

The remainder of the paper is organized as follows: Section 2 provides the methodology for selecting a suitable coding scheme and suitable classifiers for the selected encoding scheme. Section 3 provides the process of selecting a suitable feature reduction and weight assignment scheme. The method of integrating expert knowledge by combning multiple classifiers to raise total accuracy is presented in section 4. The conclusion is given in the final section.

2 ENCODING SCHEME AND CLASSIFIERS

The collected furniture design style dataset consists of Categorical data fields characterized by a large number of distinct values. It represents a serious challenge for many classification and regression algorithms that require numerical inputs. Classification and pattern recognition techniques such as neural networks, linear regression and support vector machines (SVM) [2] require numerical input. However some machine learning algorithms, like decision trees and other rule-induction methods (CART, C4.5, etc.), can handle high cardinality categorical attributes without the need for external pre-processing. We tested different coding schemes with different classifiers and selected the most promising encoding and best classifiers for the selected encoding scheme.

Different Encoding Schemes

The Following encoding schemes were found to be suitable for converting our original categorical dataset for use with regression type classifiers.

Binary Scheme: For low cardinality categorical attributes the most widely used numerical representation method is N binary derived inputs, one for each possible value of the original attribute. This scheme represents each value of the original categorical feature by a binary vector with the i^{th} component set to one, and the rest set to zero. When N is relatively small, this 1 to N mapping can be used. This technique is not suitable for a data set with attributes having hundreds of distinct values.

Categorical Encoding using target statistics scheme: This scheme was introduced by Daniele [4]. The basic idea is to map individual values of the high-cardinality categorical independent attribute to an estimate of the probability of the dependent attribute. In the case of binary target attribute $Y \in \{0,1\}$, the transformation maps individual values X_i of a high-cardinality categorical attribute X to a scalar S_i representing and estimating of the probability of Y=1 given that X= X_i:

$$X_i \rightarrow S_i \equiv P(Y|X = X_i) \qquad (1)$$

This scheme can be extended to m-valued categorical targets, $Y \in [y_1, y_2, ,.. y_m]$ in the following way. For each possible value Y_j of the dependent attribute a derived input attribute X_j^* is created in substitution of the original high cardinality categorical independent attribute X. Each derived attribute X_j^* will represent an estimate of $P(Y = Y_j \,|\, X = X_i)$ using the formula 1.

Different Classification Techniques

The original categorical dataset, binary coded dataset and statistical based coded dataset were tested with C4.5 [1], SVM [2] and Nearest Neighbour [5] classifiers. Brief descriptions of each of the methods are given below.

Nearest Neighbour: In the instance based learning, all the real work is done when the time comes to classify new instances, rather than when a training set is processed. In instance-based learning, each new instance is compared with existing ones using a distance metric, and the closest existing instance is used to assign the class to the new one. Sometimes more than one nearest neighbour is used, and the majority class of the closest k neighbours (or the distance-weighted average, if the class is numeric) is assigned to the new instance: this is termed the k-nearest neighbour method. With nearest neighbour we can use different distance metrics to measure similarity between two vectors. Exact matching, hamming distance and Euclidian distance matrices were used for the three encoding schemes.

Support Vector Machines (SVM): SVM [4] recently gained popularity in the learning community. SVM techniques for classification and regression provide powerful tools for learning models that generalize well even in sparse, high dimensional settings. It is directly applicable for binary classification tasks. Multi class categorisation has to be treated as a series of dichotomous classification problems [2].

The SVM method is defined over a vector space. In its simplest linear form, SVM is a hyperplane that separates a set of positive and examples from a set of negative examples with maximum interclass distance, the margin. The hyperplane is determined by only the training instances on the margin, the support vectors. The SVM is extended to nonlinear models by mapping the input space into a high dimensional feature space. In this space, an optimal separating hyperplane is constructed. After the optimal hyperplane is found, new examples can be classified by checking which side of the hyperplane they on.

Decision tree based classifier (C4.5): Quinlan's C4.5 [1] is a modified version of ID3 which addresses many of the deficiencies of ID3. These includes pruning, ability to deal with continuous data and built in facility to extract rules by tracing a path from the root to leaf. C4.5 is an algorithm that summarises the training data in the form of a decision tree. To build a decision tree from data, C4.5 employs a greedy approach that uses an information theoretic measure (gain ratio) as its guide. Choosing an attribute for the root of the tree divides the training instances into subsets corresponding to the values of the attribute. If the entropy of the class labels in the subsets is less than the entropy of the

class labels in the full training set, then information has been gained through splitting on the attribute. C4.5 chooses the attribute that gains the most information to be at the root of the tree. The algorithm is applied recursively to form sub-trees, terminating when a given subset contains instances of only one class.

Generalisation

Cross validation is often used to estimate the generalization ability of classifiers (i.e. performance on previously unseen data) where the amount of available data is insufficient to form the training, validation and test partitions. Under cross validation, the available data is divided into k disjointed sets, k models are then trained, each one with a different combination of k-1 partitions and tested on the remaining partition. The k-fold cross-validation estimate of a given performance statistic is then simply the mean of that statistic evaluated for each of the k models over the corresponding test partitions of the data. Cross validation thus makes good use of the available data as each pattern used both as training and test data. The most extreme form of cross-validation, where k is equal to the number of training patterns, is known as leave-one-out cross validation and has been used widely when the data set is very small..

Experimental Details

The experiments were conducted with three encoding schemes: the original categorical dataset with numerical labels, binary and statistically encoded dataset. Results are given in Table 1.

Original dataset (nominal dataset): The original data set with nominal attributes was tested with Nearest Neighbour and C4.5 classifiers. Nearest neighbour classifier used simple attribute wise exact matching as a distance metric. A Windows based software implementation of C4.5 (See 5) was used as decision tree classifier. In both cases ten-fold cross validation is used for validation.

Binary coded dataset: The original categorical (nominal) dataset with 16 attributes is converted to a binary dataset with 100 binary attributes excluding class attribute which has 7 different class numbers. Nearest Neighbour and SVM classifiers are used to classify design style. In this case, Nearest Neighbour classifier used Hamming distance as the distance metric because of the binary dataset. In the case of SVM, seven different classifiers were trained to classify seven different styles separately. Each classifier determined only if a given style attributes belonged to the corresponding style or not (binary classification). The classification decision for the entire ensemble of classifiers was based on the classifier giving the maximum output value (largest margin). In both cases ten-fold cross validation was used for validation

Table 1. Summary of Results for different encoding schemes and classification techniques

Method used	Categorical data	Binary encoded data	Statistically Encoded data
SVM	Not applicable	88.75±5.73	83.30±8.10
NN	85.59±9.49	85.68±9.47	77.5±12.24
C4.5	76.50±3.70	Not applicable	Not applicable

Statistically encoded dataset: The original dataset is converted to dataset with numerical attributes using the technique "categorical encoding using target statistics for multicasts datasets" discussed earlier. Nearest Neighbour and SVM classifiers are used to classify design style. In Nearest Neighbour classifier, the distance between input and an instance of a training data is measured by using Euclidian distance metric. Ten-fold cross validation is used for validation in both cases.

Conclusion

On average, the binary coded dataset gave higher level of overall accuracy when compared to the original categorical dataset and statistically encoded dataset. Nearest Neighbour and SVM classifiers gave the best classification accuracy over C4.5. In addition, the binary coded and statistically encoded datasets can be used with neural networks, regression type techniques and support vector machines. Therefore, it is useful to proceed to further analysis using the binary encoded dataset and Nearest Neighbour and SVM classifiers.

3 FEATURE REDUCTION & WEIGHT ASSIGNMENT

Feature selection is a useful process when dealing with high dimensional input patterns. When the numbers of features are potentially quite large classification is very expensive computationally. Also features may sometimes have certain amount of noise that leads to low accuracy of classification. So it is better to eliminate such features or set low weights.

The problem of feature selection is to take a set of candidate features and select a subset that performs the best under some classification system. An important question in the field of machine learning, pattern recognition and knowledge discovery is how to select the best subset of features. A good set of features may not only help to improve performance accuracy, but also to find a small model for data, resulting in better understanding and interpretation of the data. It also reduces the cost of extracting features.

When selecting a best subset of attributes, there are two fundamentally different approaches:

Filter (Scheme-independent selection): Make an independent assessment based on general characteristics of data before learning commences.

Wrapper (Scheme-dependent selection): Evaluate subset of features using machine learning algorithm that will ultimately be employed for learning. This is called wrapper because learning algorithm is wrapped into the selection scheme. We experimented with two approaches, one from

the each specific scheme: discriminative analysis based and genetic algorithm based.

Discriminative Analysis Based Scheme

We have used a scheme-independent approach to select the best subset of features and weights to improve the accuracy and efficiency of the design style classification. In this method, the discriminative power of features in the data set itself was used to construct a feature subset, where the discriminative power E (Entropy) for a particular feature is calculated from the following equation

$$E_{feature} = \frac{-\sum_{i=1}^{i=n}\left(\frac{x_i}{\sum_{i=1}^{i=n} x_i} \times \log\left(\frac{x_i}{\sum_{i=1}^{i=n} x_i}\right)\right)}{\log_2(n)} \quad (2)$$

where n is the number of styles and x_i is the number of times a feature occur in a style i.

High E values (close to 1) mean that features uniformly occur among all styles. Low E values mean features are more selective. This scheme allows you to find the correct subset of features for a given problem and set different weights for each attribute based on the relevant E value of each feature.

Calculating E value: The dataset is split into two parts to remove bias of E calculation from the classification process. The first part is used for calculating E and the other half is used for training and validating a classifier. As the dataset is limited (˜100 instances), it is important to know the splitting ratio to obtain maximum accuracy and highest feature reduction. The size of the dataset for training and validation is reduced due to splitting and this will decrease the accuracy of the classifier. Figure 3 shows how the Nearest Neighbour classifier is sensitive to the size of the dataset. According to Figure 3, if the dataset is in the range of 50% or less then the accuracy drastically decreases.

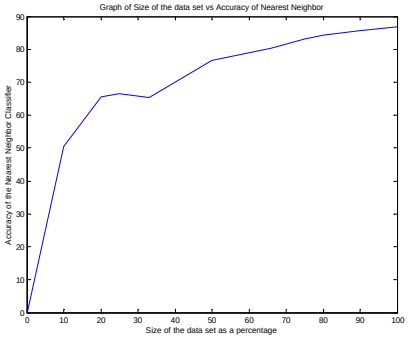

Figure 2 Graph of size of the dataset as a % Vs Accuracy of the nearest neighbour classifier as a %

Figure 3 shows a comparison of Nearest Neighbour classifier and Nearest Neighbour classifier with different weighting for the feature based on E value. The X axis of Figure 4 represents data proportion used for training and validation. If the training proportion is x%, the data proportion for calculating E is (100-x)%. Leave one out cross validation is used because of the smaller training dataset. Figure 4 also shows, when the training data proportion is in the range between 25% and 60% (proportion for E 75% and 40%) the weighted Nearest Neighbour gives significant higher accuracy over Nearest Neighbour.

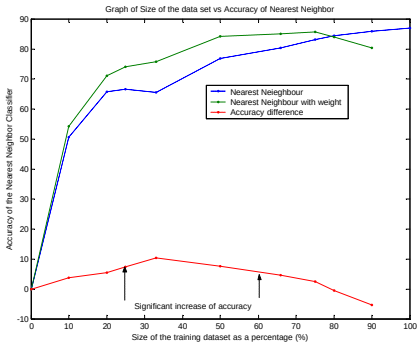

Figure 3 The Graph of size of the dataset as a percentage Vs Accuracies of Nearest Neighbour classifiers

Figure 4 shows how the accuracy of the Nearest Neighbour classifier changes when low significant attributes are dropped, one at a time. These low attributes are selected based on E value. The experiment is repeated for three selected cases and three different curves represent the results of the experiment.

Accuracies of Nearest Neighbour Classifiers: According to the results in Figure 4, maximum accuracy goes beyond 85% when 10% of the dataset is used for E calculation. However, only 10-20 features (low significant) are dropped based on the E value. Maximum feature reduction (about 50-60 features) and classification accuracy around 85% can be achieved when 30% of the dataset is used for E calculation and 70% of the dataset for validation and training

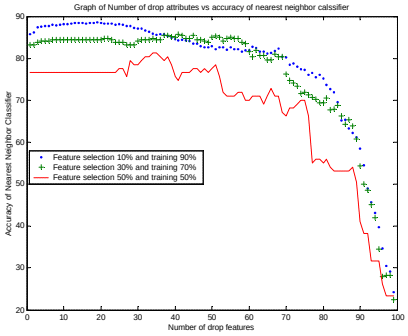

Figure 4 The Graph of numbers of drop features Vs Accuracies of Nearest Neighbour clssifires

Feature Selection Using GA

Genetic Algorithm is an optimization tool which can be used to solve various optimization problems. GA maintains

a population of members, usually called "genotypes" and classically represented by binary strings, which can be mutated and combined according to a measure of their "fitness", as measured by a task-dependent evaluation function.

GA can be used for feature selection and weight assignment.

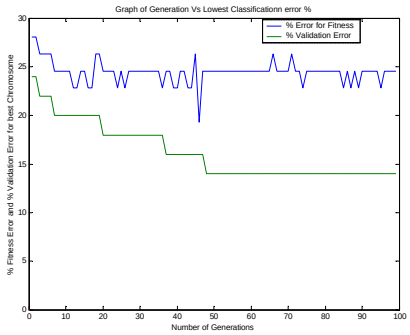

Fig. 5: The Graph of the number of generation Vs Fitness value and Validation error for best chromosome

We used GA-based technique for feature selection and weight selection for furniture style classification system. This is achieved by using GA to select binary/real number weight feature vectors for the population and Nearest Neighbour algorithm to find fitness values for each chromosome in the population based on the classification error. The binary weights are used for optimum feature selection and real weight chromosomes are used for best weight vector selection. The complete data set split into two segments, for training and validation. The training dataset is used to find a fitness values for a population in each cycle. The chromosome with best fitness value (lowest classification error) is tested with the validation set. By comparing the validation and training error for each generation the termination point and the best solution is found. Leave one out cross validation was used throughout the experiment.

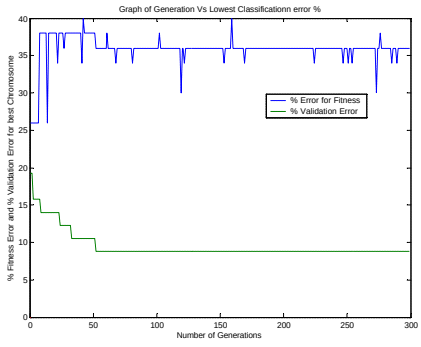

Fig. 6: The Graph of the number of generation Vs Fitness value and Validation error for best chromosome

Figure 5 shows the experimental results for binary feature selection using GA. According to the graph, the validation error stabilises around 24% after approximately ten generations. The best chromosome after ten generations is selected as the best binary weight vector. This chromosome is used for feature selection. The features relevant to 1s in the binary stream are selected and others are dropped.

Figure 6 shows the experimental results for real number feature selection using GA. According to the graph, validation error becomes stable at around 37% after fifty generations. The best chromosome after fifty generations is sleeted as the best weight vector for weight assignment.

Conclusion

We have experimented with two different schemes: scheme independent and scheme specific. The discriminative analysis based (scheme independent) outperformed the GA based scheme (scheme specific) for feature selection and weight assignment for furniture design domain. Discriminative analysis based technique presented the best classification accuracy of (80-85) % for feature selection and weight assignment. The GA based technique presented the best accuracy of (70-75) % for feature selection and (60-65) % accuracy for weight selection.

4 INTEGRATING EXPERT KNOWLEDGE

Multiple approaches have been developed for improving predictive performance of a system by creating and combining various learned models. D. Bahler and L. Navarro [6] are proposed that the combination of classifiers has long been proposed as a method to improve the accuracy achieved in isolation by a single classifier. There are two main approaches to creating model ensembles. The first is to create a set of learned models by applying an algorithm repeatedly to different training sample data; the second applies various learning algorithms to the same sample data. The predictions of the models are then combined according to a particular scheme. The reason for combining the outputs of multiple classifiers are compelling, because different classifiers may implicitly represent different useful aspects of a problem, or of the input data, which no one classifier represents all useful aspects.

Combining Multiple Classifiers

There are several methods of combining multiple classifiers. Two most commonly used techniques are voting and Bayesian ensemble.

Voting: Generally speaking, the voting principle is just what we know as majority voting. Several variations of this idea have been proposed: unanimity, majority and threshold plurality. Combining classifiers with this method is simple; it does not require any previous knowledge of the behaviour of the classifier. It only counts the number of classifiers that agree in their decision and accordingly decided the class to which the input pattern belongs. This simplicity has a drawback, however: the weight of the decision of all the classifiers is equal, even when some of the classifiers are much more accurate than others.

Bayesian Ensemble method: Voting methods are based solely on the output label computed by each classifier. No expertise or accuracy is considered. In these methods the decision of each classifier is treated as one vote, but what happens if one of the classifiers is much more accurate than any other? To address this problem we can establish weights proportional to each expert's accuracy, so each classifier's output is considered according to its past performance and combining them using Bayes' theorem.

Combining SVM, Nearest Neighbour and Expert classifiers using simple rules

The main purpose of this experiment is to determine how we can combine expert knowledge acquired from Connected line database to improve the overall accuracy of the individual classifiers. Three different classifiers were used in this phase. The first two classifiers were based on Nearest Neighbour and SVM. The third one was created from the Connectedline database.

Expert Classifier: Expert classifier is built from knowledge available in the Connected line database. All the features in a specific style were represented in one specific rule. For all seven styles (classes), seven different binary rules were constructed based on information of Connectedline database. The expert classifier is evaluated with the binary encoded dataset. In this approach, the distance between instance of the dataset and each of the expert rules was measured using exact matching metric. The rule provides maximum overlap is the best rule to explain the data instance. The style that belongs to this rule is the predicted style or class.

Table 2. Class wise accuracies for three classifiers

Class	% Accuracy of classifiers			Selected Classifier
	SVM	N-Neighbour	Expert	
1	84.21	73.68	5.26	SVM
2	85.71	85.71	14.29	SVM/NN
3	95.65	95.65	17.39	SVM/NN
4	96.55	96.55	79.31	SVM/NN
5	82.35	82.35	100.00	Expert
6	90.00	80.00	10.00	SVM
7	0.00	0.00	0.00	None

```
If (SVM Prediction=NN Prediction)
        Final Prediction=SVM Prediction
Else If (SVM Prediction~=NN Prediction) & (Expert Prediction=5)
        Final Prediction=Expert Prediction
Else If (SVM Prediction~=NN Prediction) & (Expert Prediction~=5)
        Final Prediction=SVM Prediction
End
```

Rule set 1: Simple rule set to combine classifiers

According to results in Table 2, different styles in furniture design domain are giving different accuracy for different classifiers (SVM, Nearest Neighbour and Expert). SVM classifier provides highest accuracy for class 1 and class 6. SVM and Nearest Neighbour classifiers are giving similar accuracy for class 2, 3 and 4. Expert rule classifier is giving highest accuracy for class 5. Based on the results on the Table 2 (passed performance), we have constructed simple rules to combine three different classifiers to get maximum accuracy. The set of rules is given in Rule set 1.

By combining three different classifiers with simple rules we have achieved overall accuracy of design style recognition around 90% which is higher than the individual accuracy of each of the different classifiers

FINAL CONCLUSION

We have found that furniture design style can be recognized by using SVM and Nearest Neighbour classifiers with an accuracy above 85%. The classification accuracy has been further increased (90%) by integrating expert knowledge with other data driven classifiers: SVM and Nearest Neighbour. The binary encoding scheme is more suitable for encoding for the selected furniture style domain. Accuracy of the furniture design classifier has been further improved though the use of feature reduction and weight assignment. The discriminative power of feature analysis based scheme gave more accurate results compared to GA based scheme for feature selection and weight assignment. This total frame work can be used in other similar domains.

BIBILIOGRAPHY

[1] Quinlan, J.R., C4.5: Programs for Machine Learning. 1993: Morgan Kaufmann.

[2] Schlkopf, B. and A.J. Smola, Learning with kernels 2002, Cambridge: MIT University Press.

[3] Connectedlines, The On-Line Furniture Style Guide. 1998, http://www.connectedlines.com/styleguide/index.htm.

[4].Daniele, M.B., A Pre-processing Scheme for High-Cardinality Categorical Attributes in Classification and Prediction Problems. SIGKDD Explorations, 2001. 3(1): p. 27-32.

[5] Theodoridis, S. and K. Koutroumbas, Pattern Recognition. 1999, England: Academic Press.

[6] Bahler, D. and L. Navarr. Methods for Combining Heterogeneous Sets of Classifiers. in 17th Natl. Conf. on AI (AAAI 2000), Workshop on New Research Problems for Machine Learning. 2000. Austin, Texas.