Statistical Feature Ordering for Neural-based Incremental Attribute Learning

Thesis submitted in accordance with the requirements of the

University of Liverpool for the degree of

Doctor in Philosophy

by

Ting Wang

June 2013

Abstract

In pattern recognition, better classification or regression results usually depend on highly discriminative features (also known as attributes) of datasets. Machine learning plays a significant role in the performance improvement for classification and regression. Different from the conventional machine learning approaches which train all features in one batch by some predictive algorithms like neural networks and genetic algorithms, Incremental Attribute Learning (IAL) is a novel supervised machine learning approach which gradually trains one or more features step by step. Such a strategy enables features with greater discrimination abilities to be trained in an earlier step, and avoids interference among relevant features. Previous studies have confirmed that IAL is able to generate accurate results with lower error rates. If features with different discrimination abilities are sorted in different training order, the final results may be strongly influenced. Therefore, the way to sequentially sort features with some orderings and simultaneously reduce the pattern recognition error rates based on IAL inevitably becomes an important issue in this study.

Compared with the applicable yet time-consuming contribution-based feature ordering methods which were derived in previous studies, more efficient feature ordering approaches for IAL are presented to tackle classification problems in this study. In the first approach, feature orderings are calculated by statistical correlations between input and output. The second approach is based on mutual information, which employs minimal-redundancy-maximal-relevance criterion (mRMR), a well-known feature selection method, for feature ordering. The third method is improved by Fisher's Linear Discriminant (FLD). Firstly, Single Discriminability (SD) of features is presented based on FLD, which can cope with both univariate and multivariate output classification problems. Secondly, a new feature ordering metric called Accumulative Discriminability (AD) is developed based on SD. This metric is designed for IAL classification with dynamic feature dimensions. It computes the multidimensional feature discrimination ability in each step for all imported features including those imported in previous steps during the IAL training. AD can be treated as a metric for accumulative effect, while SD only measures the one-dimensional feature discrimination ability in each step. Experimental results show that all these three approaches can exhibit better performance than the conventional one-batch training method. Furthermore, the results of AD are the best of the three, because AD is much fitter for the properties of IAL, where feature number in IAL is increasing.

Moreover, studies on the combination use of feature ordering and selection in IAL is also presented in this thesis. As a pre-process of machine learning for pattern recognition, sometimes feature orderings are inevitably employed together with feature selection. Experimental results show that at times these integrated approaches can obtain a better performance than non-integrated approaches yet sometimes not. Additionally, feature ordering approaches for solving regression problems are also demonstrated in this study. Experimental results show that a proper feature ordering is also one of the key elements to enhance the accuracy of the results obtained.

Acknowledgements

First of all, I want to thank my primary supervisor, Prof. Steven Sheng-Uei Guan. With his guidance, I made a rapid progress in my research on machine learning and pattern recognition in the past several years. His insistence on academic rigour and innovative thinking inspires me along the way of my research. I can hardly complete my PhD thesis without his close supervision.

Furthermore, I want to express my appreciation for the support and guidance from my co-supervisor, Dr. David Liu, Department of Mathematical Sciences, Xi'an Jiaotong-Liverpool University (XJTLU), and Dr. Alexei Lisitsa, Computer Science Department, University of Liverpool (UoL). Also, many thanks are expressed to Dr. Fei Liu (La Trobe University, Australia), Dr. Prudence W. H. Wong (UoL) and Dr. Sadasivan Puthusserypady (Technical University of Denmark).

Moreover, my heartfelt thanks go to all the people at XJTLU. As an off-site PhD student of UoL, I spent four years in this international university campus. During this period, many people from XJTLU helped me in my research. Those lecturers, classmates and colleagues that worth to be mentioned include Prof. Yuhui Shi, Dr. Bailing Zhang, Dr. Ka Lok Man, Dr. Kaiyu Wan, Dr. Emmanuel M. Tadjouddine, Dr. Nan Zhang, Dr. Tiew On Ting, Mr. Olan Walsh, Ms. Liying Liu, Mr. Yungang Zhang, Mr. Jieming Ma, Miss Meihua Li, Mr. Shang Yang, and Miss Shujuan Guo for their support. The academic discussion with them was very helpful for my research works.

In addition, I am thankful to XJTLU to grant me a scholarship. Also, my appreciation also goes to the National Natural Science Foundation of China for the project with Grant No. 61070085; and financial assistance to my research from the Government of Wuxi Binhu District and the Government of China-Singapore Suzhou Industrial Park District.

Finally, I want to thank my family, especially my wife and my parents for the continuing support and the encouragement given in completing my PhD study. Without their support and patience, the completion of this PhD study would merely be a formidable task in my life.

Declaration

I confirm that the thesis is my own work, that I have not presented anyone else's work as my own and that full and appropriate acknowledgement has been given where reference has been made to the work of others.

Ting Wang
June 2013

List of Publications

● Journal Papers

[1].WANG Ting, GUAN Sheng-Uei, and LIU Fei. "Ordered Incremental Attribute Learning based on mRMR and Neural Networks". 86-90, International Journal of Design, Analysis and Tools for Integrated Circuits and Systems, Vol. 2, no 2, 2011.

[2].WANG Ting, GUAN Sheng-Uei, and LIU Fei. "Correlation-based Feature Ordering for Classification based on Neural Incremental Attribute Learning". 807-811 International Journal of Machine Learning and Computing, Vol. 2, No. 6, December 2012.

[3].WANG Ting, GUAN Sheng-Uei, PUTHUSSERYPADY Sadasivan, WONG Prudence. Statistical Discriminability Estimation for Pattern Classification based on Neural Incremental Attribute Learning. International Journal of Applied Evolutionary Computation. (ACCEPTED)

● Conference Papers

[1].WANG Ting, WANG Yuanqian. Pattern Classification with Ordered Features using mRMR and Neural Networks. (in):Jiang Hui. ICINA2010: 2010 International Conf. on Information, Networking and Automation. Kunming, China.: IEEE and IACSIT, 2010, vol2, 128-131

[2].WANG Ting, GUAN Sheng-Uei, LIU Fei. Feature discriminability for pattern classification based on neural incremental attribute learning. Foundations of Intelligent Systems: Proceedings of the Sixth International Conference on Intelligent Systems and Knowledge Engineering, Shanghai, China, Dec 2011 (ISKE2011), Tiergartenstrasse 17, Heidelberg, D-69121, Germany, Springer Verlag.

[3]. WANG Ting, GUAN Sheng-Uei, LIU Fei. Entropic feature discrimination ability for pattern classification based on neural IAL. 9th International Symposium on Neural Networks, ISNN 2012, July 11, 2012 - July 14, 2012, Shenyang, China, Springer Verlag.

[4]. WANG Ting, GUAN Sheng-Uei, TING T.O., MAN Ka Lok, LIU Fei.. Evolving linear discriminant in a continuously growing dimensional space for incremental attribute learning. 9th IFIP International Conference on Network and Parallel Computing, NPC 2012, September 6, 2012 - September 8, 2012, Gwangju, Korea, Republic of, Springer Verlag.

[5]. WANG Ting, GUAN Sheng-Uei, editors. Feature Ordering for Neural Incremental Attribute Learning based on Fisher's Linear Discriminant. 5th International Conference on Intelligent Human-Machine Systems and Cybernetics, IHMSC 2013, August 26-27, Hangzhou, China, IEEE.（ACCEPTED）

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem Statement

How to solve a big and difficult problem? Many people may answer, “divide it and conquer it”. Yes, “Divide and conquer” is a very basic way in solving problems. Prof. Donald Ervin Knuth took post office as an example for “divide and conquer”, “ letters are sorted into separate bags for different geographical areas, each of these bags is itself sorted into batches for smaller sub-regions, and so on until they are delivered.” [1] However, only “divide and conquer” is insufficient. If the postman is an internship student, who is not very familiar with the place of the mail delivery, what can this fresh postman do? An effective and easy way is that this internship postman can start from the place he had known, and search the unknown place later. Therefore, he can not only save time, but also gradually get familiar with the mail delivery zone step by step. Eventually, he can finish his work along with the learning.

During the process, apart from “divide and conquer”, there are some other important elements which guarantee the final success of this fresh postman's first day work. Firstly, ranking sub-problem from easy to difficult should be done in the first place. For example, the postman ranks all the subarea and knows which place can be visited early and the one that can be visited later. Secondly, the ordering plays an essential role after ranking. The easier the sub-problem is, the earlier it will be solved. After the postman finished ranking all the subarea, he gets an ordering of his work. Mail delivery in a familiar place is easier than in a strange place. Thirdly, is the feedback component. Results should receive feedback in each step for the final decision making. When the postman finished delivery in one subarea, he should mark on his map on the

finished areas. His marking on his map is a feedback to his entire decision, so that he will not repeat the same area. Likewise, to solve a big and difficult problem, we can firstly divide it, then sort it from easy to hard, and lastly, solve the easy part and output the feedback in every step.

For intelligent machines, things are similar. In pattern recognition, to obtain a highly accurate classification and regression result is an extremely difficult problem. Sometimes, due to the large number of patterns or features, the problem also appears to be huge. Solving such a big and difficult problem usually depends on discriminative features which are also known as attributes of datasets. In another aspect, as an approach for pattern recognition, machine learning plays a significant role in the improvement of classification and regression performance. Therefore, the question is, can we improve the pattern recognition performance based on the approach we mentioned above, where features are divided one by one or group by group firstly, then sorted from high discriminative to low discriminative by division and trained by each division with feedbacks in the last stage?

The machine learning strategy which firstly sorts features into some orderings based on some criteria, and then gradually trains and tests features one by one or group by group based on the feature orderings is called Feature-based Incremental Learning, also can be interpreted as **Incremental Attribute Learning (IAL)** [2].

Different from conventional machine learning approaches that train all features in one batch by some predictive algorithms like **Neural Network (NN)** and **Genetic Algorithm (GA)**, IAL is a novel supervised machine learning approach which gradually trains one or more input features step by step. Such a strategy makes the features which have greater discrimination abilities be trained in an earlier step than others, and get rid of interference between features during classification. Therefore, in IAL, it is necessary to know which feature is good during classification, and what kinds of machine learning approaches can cope with these problems. Features with a greater discrimination ability should be trained and tested in an earlier stage.

Previous studies showed that IAL can be independently employed and successfully applied based on many machine learning approaches, such as Neural Networks [2-11], Genetic Algorithms [11-16], Bayesian classifier [17], **Decision Trees (DT)** [18], **Particle Swarm Optimizations (PSO)** [11, 19] and **Support Vector Machines (SVM)** [20]. Apart from this, the final results produced by IAL also exhibit better performance in pattern recognition compared

with conventional approaches which import all features into pattern recognition systems for training in one batch. One of the most important reasons why IAL can succeed in pattern recognition is that it trains features with greater discrimination ability at an earlier stage in the process which can successfully avoid interference from other features whose discrimination abilities are weaker [8]. Therefore, feature ordering should be regarded as one of the most important preprocesses of sequential feature training in IAL.

In this study of IAL, feature ordering is investigated as a significant and independent machine learning preprocess. Whether feature ordering is valid and useful with the strategy of IAL in obtaining better performance than the other approaches and whether such a new preprocessing can be used for classification, regression and some other preprocessing are two main and basic objectives. To achieve these objectives, the following sub-objectives are derived based on the problems raised above:

- To find feature ordering approaches for IAL;
- To discover some ranking metrics for feature ordering;
- To find the optimum feature ordering for IAL, and get the solution;
- To check whether ordered IAL can exhibit better performance and to find the reasons;
- When a new feature is imported, try to confirm whether it will influence the whole prediction system or not. If so, try to find the approach to measure the influence;
- To check whether feature ordering can be used in classification problems;
- To check whether feature ordering can be employed in regression problems;
- Try to find out whether feature ordering is able to be employed with feature selection;
- Try to make sure whether there are some new algorithms and criteria existing for optimum feature ordering and IAL.

1.2 Topic Overview

The work presented in this thesis has three main motivations which are related to each other. Firstly, this thesis aims to make feature ordering as an independent and necessary preprocessing in IAL-based pattern recognition, because in the last decade, feature ordering was seldom studied

independently. This was often researched as a by-product of IAL, which is often ignored in the research on IAL. For example, in Guan's experiments [21], all the features were imported in the original order. In the research on the incremental decision tree learning methodology [18] and the Incremental Feature Learning of SVM [20], researchers have not arranged new feature orderings for the training. The original feature ordering was directly employed in their experiments.

Secondly, some more efficient feature ordering approaches need to be developed, because in the previous studies, most feature ordering methods are time-consuming. For example, in Guan's work [5], feature ordering was derived by a contribution-based method which is similar to wrapper feature selection approaches. In this method, each feature is solely employed in training and testing in the first step, and then all features are sorted according to the individual testing accuracy. Guan et al. presented three types of criteria for feature ordering: ascending order, descending order and random order [2]. Obviously, no matter what type of ordering is employed, such a method often costs a great deal of time in preprocessing, especially when the number of features is large [4]. Furthermore, Zhu employed Guan's contribution-based feature ordering method in his **Ordered Incremental Genetic Algorithm (OIGA)**, where attributes are arranged in order by evaluating their individual discriminating ability [13]. This work also indicates that feature ordering approaches are independent of machine learning algorithms. One feature ordering method can be employed, no matter what kinds of predictive pattern recognition methods are used later.

Thirdly, this research aims to study the influence of new features according to a given feature ordering in IAL. In previous research, feature ordering was derived by each feature's single contribution in a stable feature space. However, the feature space dimension is increasing in IAL, thus whether feature discrimination ability should be jointly calculated with the newly imported feature instead of computing them independently is an issue needed to be studied. In previous studies of feature ordering, features were usually ranked by individual contribution, such as Jun Liu's contribution-based feature ordering. However, when new features are imported, the final result will be influenced by not only the old features but also the new features. Thus, a combined feature discrimination ability is more important than a pre-computed single feature discrimination ability. Therefore, feature discrimination ability should be accumulatively calculated and thus the corresponding approaches will be developed in this study.

Therefore, to cope with the problems mentioned in the first section, this study begins with the research on IAL feature ordering based on some applicable methodologies. A number of metrics and criteria should be developed for optimum feature ordering seeking. Thus some different feature ordering approaches are required to be developed. In the meanwhile, the reasons why the optimum feature ordering is feasible should be investigated. Also, the problem whether newly developed feature ordering approaches are applicable to be employed with some other pattern recognition preprocessing like feature selection is an important issue in this research. In addition, some corresponding algorithms for feature ordering is another significant topic in this study.

1.3 Thesis Structure

In this thesis, the introduction to IAL is given in Chapter 2 with corresponding literature review. Methodologies about data sampling and experiments in this study are presented in Chapter 3. Further, three different filter feature ordering approaches are developed based on statistical correlations, mutual information and linear discriminant. These are demonstrated with theories and experiments from Chapters 4 to 6. Then, the experimental results of these newly developed approaches are compared with each other and some results from previous studies with analysis in Chapter 7. In Chapter 8, feature ordering is implemented and combined with feature selection, another very important pattern recognition process. In addition, the application of feature ordering in regression and function approximation is illustrated in Chapter 9. Lastly, the conclusions on the studies concerning IAL feature ordering are drawn in the last Chapter.

Furthermore, this thesis also contains four appendices. Appendix A gives a brief introduction to all the dataset used in the experiments. Appendix B demonstrates the results of Contribution-based feature ordering approach, which was presented by Guan and Liu[2, 4]. Information about parameter setting and stopping criterion for neural network training used in our study is presented in Appendix C, and an introduction to the neural networks program software is given in Appendix D.

Chapter 2

Literature Review

2.1 Incremental Attribute Learning

Incremental Attribute Learning (IAL) is a “divide-and-conquer” machine learning strategy which gradually trains input features one after another. There are two main objectives for implementing such a novel approach. One is to solve easy problems at the early stage of the process. Due to the fact that each feature has a different ability in classification for different output, IAL aims to, firstly, solve easy pattern recognition problems by using several corresponding features and, secondly, leave difficult problems to the next round using some other different features. The other objective is to avoid dimensional disasters. The “divide-and-conquer” character of IAL has the capability to reduce the complexity of computing as not all features will be imported for calculation at the same time. Such a process is effective to avoid the curse of dimensionality in computing where the problem has a high-dimensional feature space. Therefore, as a new approach, IAL not only can cope with problems which can be solved by existing methods, it is also applicable for problems which have newly imported features or problems whose number of features is large. Figure 2.1 shows the main structure of IAL. The left side of the Prediction Method is the input, while the part on the right side is the output. Generally, IAL focuses on the input aspect, while the output aspect is concentrated by another similar incremental approach called **Hierarchical Incremental Class Learning (HICL)** [22-25], which is not a research topic in this study.

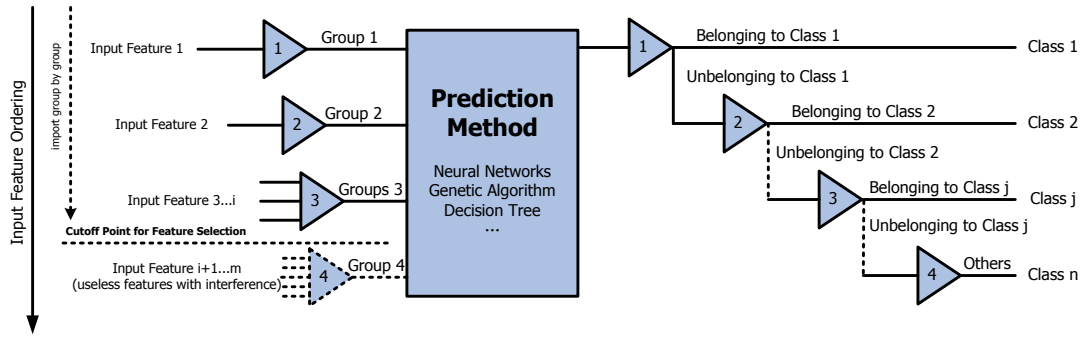


Figure 2.1: The process of IAL with ordered input features and output

A number of experiments and studies have shown that IAL often exhibits better performance than other conventional machine learning techniques that train data in one batch. For example, based on datasets from University of California at Irvine (UCI) Machine Learning Repository, Guan et al. employed IAL to solve several classification and regression problems by NN [2-5, 7, 8, 21, 26-28], PSO [19] and GA [13, 15]. Almost all of their results using IAL were better than those derived from traditional methods. Specifically, based on **Incremental Learning in terms of Input Attributes (ILIA)** [21] and **Incremental neural network Training with an Increasing input Dimension (ITID)** [4], two effective algorithms were developed on the basis of IAL, and as a result, classification errors obtained by incremental neural networks for input feature learning of Diabetes, Thyroid and Glass datasets reduced by 8.2%, 14.6% and 12.6%, respectively [2, 4]. Furthermore, based on OIGA, the testing error rates derived by incremental genetic algorithms of Yeast, Glass and Wine declined by 25.9%, 19.4% and 10.8% [14], respectively, in classification. Further, Ang et al. proposed interference-less networks in his paper. He divided features into several groups without interference in the same group. Such an approach led to more acceptable results from the experiments [8].

Moreover, Chao et al. used a decision tree to implement IAL, and presented **Intelligent, Incremental and Interactive Learning (i$^+$Learning)** and **i$^+$Learning regarding attributes (i$^+$LRA)** in their paper [18]. These algorithms were employed to run in 16 different datasets supplied by UCI. The results indicated that the algorithms based on IAL performed better than ITI in 14 of the 16 datasets. Furthermore, Agrawal and Bala presented an incremental Bayesian classification approach for multivariate normal distribution data. In their experiments, features

are imported one by one into Bayesian classifier. Their experimental results also demonstrates that feature-based incremental Bayesian classifier is computationally efficient over batch Bayesian classifier in terms of time, although both of the results derived by these two methods are equivalent [17]. In addition, successful research on incremental SVM extended IAL to a wider application field [20]. All of these previous IAL studies showed that IAL can indeed improve the performance of pattern recognition. These studies denoted that different feature ordering can exhibit different pattern recognition results and feature ordering is gradually recognized as a formal preprocessing step of IAL.

Recently, IAL has been employed into real-world application. Kankuekul et al. developed a new online incremental zero-shot learning method based on **self-organizing and incremental neural networks (SOINN)** for applications in robotics and mobile communications. Comparing the conventional method with their proposed approach, this novel approach can learn new attributes and update existing attributes in an online incremental manner in a more effective way [10]. Moreover, Kawewong, A. and O. Hasegawa presented a new approach called **Attribute Transferring based on SOINN (AT-SOINN)** for learning and classifying object's attribute in an online incremental manner. Comparing with some state-of-the-art methods, AT-SOINN performs a fast attribute learning, transferring and classification while at the same time retaining the high accuracy of attribute classification [9].

Some IAL approaches are listed and compared in Table 2.1. According to the table, IAL is definitely can be implemented based on a number of intelligent predictive methods.

The achievements of IAL contribute to the characteristics of this novel machine learning strategy. Compared with other machine learning strategies for pattern recognition, some of these characteristics are similar to the existing methodologies while others are not. For example, there is another well-known “divide-and-conquer” strategy called **Incremental Learning (IL)**, which concentrates on the number increase with respect to training samples [29]. Nevertheless, IAL is different. It focuses on an increase in the number of features. In addition, IAL utilizes features one by one, or group by group, which is different from conventional machine learning techniques that always train data in one batch. Last but not least, apart from feature reduction in preprocessing, IAL has another unique data preparation process called feature ordering that is required for almost all problems solved by IAL.

Table 2.1: A Comparison of Different IAL Approaches

	Approach	Predictive Method	Descriptions
1	HICL [22-25]	Neural Networks	IAL in the output dimensions
2	ILIA [21]	Neural Networks	IAL with fixed feature ordering in the input dimensions
3	ITID [4]	Neural Networks	IAL with changeable feature ordering in the input dimensions
4	OIGA [13]	Genetic Algorithms	IAL in the input dimensions based on GA
5	i^+Learning / i^+LRA [18]	Decision Trees	Incremental Decision Tree Learning algorithms by concerning new available attributes in addition to the new incoming instances
6	Incremental Bayesian classifier [17]	Bayesian Classifiers	Incremental Bayesian classifier trains features following multivariate normal distribution one by one
7	Incremental Feature Learning [20]	Least Square Support Vector Machines	An incremental feature learning algorithm which can tackle with incremental learning problems with new features
8	IAPSO [19]	Particle Swarm Optimizations	IAL using PSO
9	SOINN [9] / AT-SOINN [10]	Neural Networks	IAL Application Algorithms

2.2 Algorithms of Neural IAL

ITID [2, 4] is an incremental neural network training approach derived from ILIA [21]. It divides the whole input dimensions into several sub dimensions each of which corresponds to an input feature as shown in Figure 2.2 and 2.3. As shown in Figure 2.2, instead of learning input features altogether as an input vector in a training instance, Neural Networks learn input features one after another through their corresponding sub-networks and the structure of Neural Networks is grown incrementally with an increasing input dimension. During training, the information obtained from a new sub-network is merged together with the information obtained from the old ones to refine the current Neural Networks structure. Such architecture is based on ILIA1. After the training with the structure as implemented in Figure 2.2, the outputs of Neural Networks are collapsed. An additional network sitting on the top with links to the collapsed output units, and all the input units are built to collect more information from the inputs as shown in Figure 2.3. This structure

is based on ILIA2. Finally, a pruning technique is adopted to find out the appropriate network architecture. With less internal interference among input features, ITID achieves higher generalization accuracy than conventional methods [4].

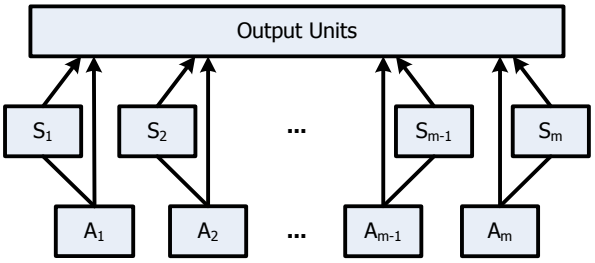


Figure 2.2: ITID based on ILIA1 with the basic network structure

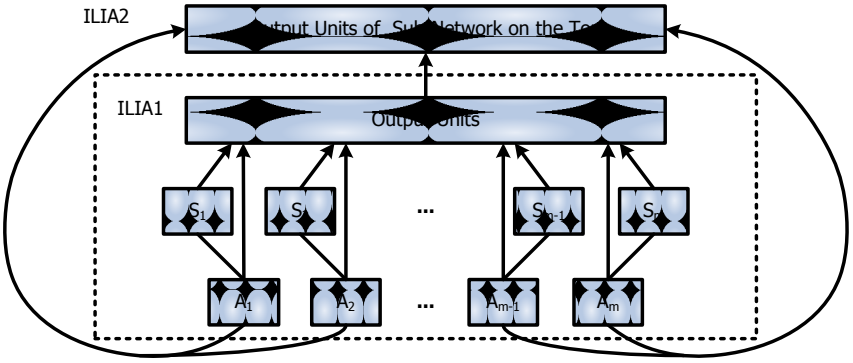


Figure 2.3: The network structure of ITID based on ILIA2

2.3 Preprocessing of IAL

Preprocessing aims to standardize the structure of raw data collected in the data preparation stage. As IAL has both similarities and differences to conventional methods, preprocessing of feature-based incremental learning strategy should not only include traditional preprocessing techniques such as normalization, data smoothing and feature selection, but also contain some special methods, such as feature ordering. In this subsection, traditional preprocessing of data transformation is firstly reviewed. , This is then followed by feature ordering, the very special step of IAL. Other preprocessing works such as feature selections are reviewed in the last step.

2.3.1 Data Transformations

A transformation from raw data to data with standard form is propitious to data mining and

prediction. Data transformation is not a novel technique for data preparations. Generally, there are three steps of data transformation: normalization, data smoothing, and a complement of missing values.

- Normalization

Normalization is very useful for classification and regression problems which are based on neural networks or statistical distance. Normalization can speed up training for neural networks and avoid features with large scale outweighing features with small scale by using statistical distance. In normalization, raw data will be scaled to a special range, such as from -1 to 1 or from 0 to 1. There are three general normalization methods: Min-max normalization, z-score normalization and Normalization by decimal scaling [30].

- Data smoothing

Data smoothing aims to get rid of noisy data which have ambiguous meanings from raw dataset. For example, it is ambiguous for a number when it can be categorized into two classes at the same time. Thus, these noisy data will bring interference to the solution. Discarding noise before prediction will enhance accuracy in final results. At present, binning, regression and clustering are three popular data smoothing techniques.

- Missing values

A number of practical problems have missing data in the datasets. These missing data are sometimes indispensable for solving problems. Therefore, people cannot simply ignore these missing data in datasets. A naive way for dealing with missing values is to fill them with a constant or a mean of its class. A more precise way for coping with missing values is using prediction, such as regression or classification. In addition, missing values do not mean that the data are wrong. For example, in some application form, applicants are often demanded to put their occupations onto the table. However, if the applicant is jobless, data on the blank of job will be a missing value.

2.3.2 Sequential Feature Ordering

Feature Ordering is a very important and unique property of IAL. It not only determines which feature should be imported into the prediction system in the first place, which one should be trained next, and which feature should be introduced to the training process in the last place, but also aims to get rid of interference from some other features during the process of training. Such a property is crucial to the improvement of final results in pattern recognition like classification and regression. Therefore, the method to produce an optimum feature ordering for different problems is an important question in IAL. Furthermore, a good strategy to measure whether a feature ordering is suitable or not for classification and regression is the key to obtain the optimum feature ordering.

In our daily life, people often employ different metrics to measure different things. For example, meters and feet are used to measure the length of an object; while kilograms and pounds are employed for the weight of different objects; in addition, Newton for force, Ampere for electric current, Celsius for temperature, and so on. Similar to this, feature ordering also requires some metrics for measurement.

The essence of the measurement for feature ordering is the capacity of feature discrimination ability for classification and regression. Such a capacity was researched in a number of previous studies on feature selection, where a subset of features with good discrimination ability is selected for pattern recognition. It is believed that such a process can reduce the interference from some other features whose discrimination abilities are not as strong as those selected features. Therefore, previous achievements on feature discrimination ability are studied again for feature ordering in our research on IAL. At least four kinds of different measurements of feature discrimination abilities have been studied: the first one is based on feature's single contribution [5, 7], the second one employs the correlation between input and output, the third one uses mutual information, a concept in information theory, and the last one is developed by pattern distribution in each dataset.

2.3.3 Feature Ordering based on Single Contribution

Contribution-based feature ordering aims to detect the contribution of each feature to outputs in the first place, and sort all the features according to the contribution of each feature in the next place [4]. In supervised machine learning process, training is a very important step, where the structure of predictive system is adjusted in this step. After training, testing will be implemented whereby the adjusted predictive system will be employed for prediction. Sometimes, between the steps of training and testing, there is an independent step called validation. This is where parameters of the predictive system will be adjusted. The contribution-based feature ordering should be employed in all of these three steps. In short, in all three steps, all the features should be sorted according to the ordering obtained by their contribution.

Here, the definition of feature's contribution is given as below:

Definition 2.1: ***Feature's Single Contribution*** refers to the error rate of one feature when it is the only feature which has been employed to predict the final results in pattern recognition.

According to Definition 2.1, an approach of contribution-based feature ordering is shown below:

- **Step 1:** Features are solely used for final result prediction one by one;
- **Step 2:** Features are sorted according to their single predictive error rate with ascending order obtained in step 1;
- **Step 3:** To rebuild the dataset using new sorted features and employ this new sorted dataset with IAL in training, validation, and testing.

Obviously, feature ordering here is based on the predictive error rate of each single feature.

In previous studies on IAL, contribution-based feature ordering is widely researched. It can cope with both classification and regression problems efficiently. However, there exists a great disadvantage, which is the time spent on the calculation about feature ordering is quite long [31]. The reason for this is, firstly, the problem will be totally computed for n+1 times, where n is the number of input features; secondly, most of the work between two rounds of single feature

training only can be done manually which also make contribution-based feature ordering approach time-consuming and incomparable in the aspect of time cost. Besides, some experiments also found that contribution-based feature ordering cannot always obtain better performance than conventional approaches which train all features in one batch. Some examples of these experimental results using some benchmark problems will be demonstrated in Appendix B.

2.3.4 Feature Selection and Dimensional Reduction

Apart from feature ordering, feature selection is another important preprocessing work of IAL. Feature Selection also can be called as variable selection, attribute selection or variable subset selection. It is now a very hot research area in the disciplines of Statistics and Machine Learning. Feature Selection assumes that there are relevancy and redundancy existing between input features and output categories or values, and not all the features are useful to solve pattern recognition problems. Therefore, features with relevancy and redundancy are discarded in the preprocessing work and a selected subset of features will be employed for further prediction. As a result of that, the feature number decreases and the number of feature dimensions also reduces. If the number of feature dimensions is regarded as a measurement of the complexity of a pattern recognition problem, feature dimensional reduction derived by feature selection can also be treated as an approach for the pattern recognition problem simplification. Obviously, feature selection can provide three main benefits to predictive models: firstly, improved model interpretability; secondly, shorter training time; and lastly, enhance generalization by reducing overfitting.

There are two main different feature selection technical types: Filters and Wrappers. The former uses metrics to rank all features according to some criteria. Mutual Information, Correlation and Distance are common metrics in filter technique. However, wrapper is quite different from filters, where predictive models are used to score features based on their single error rates. Commonly used predictive models are NN, GA, SVM etc. Therefore, it is obvious that filters depend on feature's own data properties whereas wrappers are based on each feature's contribution to the entire outputs. These two kinds of feature selection methods are usually

compared with each other for a long time. Generally, filter technique is faster than wrappers while wrappers often get more accurate results than filters [32].

In previous studies, feature selection of IAL is mainly based on the contribution based wrapper approaches [5]. Experimental results demonstrated feature selection can be used together with feature ordering in IAL, and feature selection also can improve the final results in IAL. However, whether feature selection in IAL can be implemented by some other methods such as filter methods and whether new feature selection technique is useful in IAL are new problems in this study.

2.3.5 Feature Partitioning and Grouping

Feature partitioning also known as feature grouping is a very important preprocessing step in IAL. It can reduce the interference between features and integrates rational features into one powerful feature. For example, Ang et al. divided features into two groups: significant and insignificant features. He not only used batch training within two groups, but also employs IAL between two groups. He developed **Incremental Discriminatory Batch and Individual Training (ID-BIT)** approach to cope with classification and regression problems [6]. Moreover, they also employed a feature grouping approach to find a way to get rid of interference between features. Their works improved the theory of IAL. Experimental results showed that, sometimes, a proper feature grouping training may improve the pattern recognition performance [8]. Generally, Ang's research focused on how to merge existing features into one group. In fact, apart from this feature partition, there exists some other types of feature groups such as natural grouped features.

Natural grouped feature is the feature containing more than one sub-attribute. For example, colour in computer science often employs RGB for description. RGB refers to three value in Red, Green and Blue. Thus, the feature colour has three attributes, which are Red, Green and Blue. It is obvious that all attributes in one natural grouped feature should be computed simultaneously.

2.4 Summary

This chapter gave an introduction to IAL, and surveyed literatures of algorithms of IAL based on

neural networks and approaches of preprocessing in IAL. This has been taken as a basis of proposed research in this thesis, which will be presented in the forthcoming chapters. Apart from some conventional preprocessing works like data transformation, the investigation in this chapter concluded some important points of feature ordering, which is a unique preprocessing in IAL studies. Following items are the key points within this field:

- **IAL Sequential Feature Ordering.** Due to the fact that IAL gradually imports and trains input features in one or more size, whether different feature ordering will impact on the final results of pattern recognition is an issue need to be discussed. Previous research has confirmed that feature ordering is crucial to classification and regression, thus excepting the current existing feature ordering approaches, whether there are some metrics or approaches existing for IAL is necessary to be studied. Therefore, feature ordering should be regarded as an independent preprocessing phase in IAL for pattern recognition. In the next chapter, some basic methodology will be introduced with experimental data descriptions, which is an important and indispensable part in the whole study.
- **Existing Feature Ordering Approaches.** Currently, feature ordering is often calculated by wrapper methods, which are similar to wrappers in feature selection and based on each feature's single contribution to the entire problem solutions. However, such a method is time-consuming, and not applicable when feature number is very large. Therefore, it is necessary to find some novel metrics and fast approaches for feature ordering measurement in preprocessing stage. Chapters 4, 5 and 6 cover some novel metrics based on statistical correlation, mutual information, and linear discriminant for feature ordering, corresponding approaches will also demonstrated in these chapters. The performances of different feature ordering approaches are compared with each other in Chapter 7 with the analysis about the data properties of the best feature ordering approach.
- **Feature Ordering with Feature Selection.** As we know, feature selection is a very useful preprocessing work in pattern classification and regression. Thus when feature ordering is employed in IAL, it is inevitable to use feature selection at the same time. How to merge these two different preprocessing works together and avoid conflicts

during the process are important issues needed to be studied in this study. Feature ordering with feature selection is illustrated in detailed in Chapter 8 with some pattern classification experiments.

- **Feature Ordering with Feature Grouping.** Feature grouping is a special preprocessing stage that need to be implemented when one feature has more than one attribute in IAL. Due to the fact that features are gradually imported into predictive systems in IAL, when a feature contains more than one attribute, whether all the attributes in a multi-attribute feature should be imported simultaneously or should be imported individually, is very important to the predictive mechanism settings. Moreover, relative influence brought by feature grouping is also necessary to be studied. Feature ordering with grouping will be researched in Chapter 9.

Each of these areas plays an important part in the impending chapters for the preprocessing work of IAL.

Chapter 3

Methodology

3.1 Overview

As in many studies on pattern recognition, this research is also carried out with a quantitative analysis methodology. Both hypotheses about the elements which can improve final classification or regression results and theories on how to solve and confirm those hypotheses are implemented by mathematical deduction and experimental confirmation. It is obvious that mathematical deduction and experiments always play an important role in the discovery of metrics for relevant element measurement and the design of algorithm for final result prediction.

The main structure of this study can be divided into five different, but continuous and relevant parts: data preparation, preprocessing, IAL algorithms, simulation and application. The main relevant techniques of these five components have been described in Figure 3.1. More specifically, data preparation, the first part, aims to collect data from some real-world problems. However, some raw data directly collected from the real-world problems are too rough to be employed by the predictive systems or algorithms. All the raw data should be normalized in the preprocessing, so that the natural interference brought by raw data can be discarded, and all the data can be put in a united range for further calculation. Besides normalization, data smoothing and missing value estimation are also essential for the precise level of final results. Due to different reasons, some patterns may contain a few missing values in different features. These missing data may reduce the accuracy of final predictive results. Thus, a feasible approach to collect data is to try to select data from some benchmark problems that have no missing value. In addition, if missing values are inevitable in the study, it is necessary to estimate these missing

data based on some existing approaches such as binning, regression and clustering. Different from those common preprocessing works like data normalization, data smoothing, and missing value estimation, the special preprocessing works in IAL like feature ordering are unique and only exist in IAL preprocess. As the main part of this research, special preprocessing works are very important in final prediction. Feature ordering and some related calculations must be completed before formal prediction using IAL algorithms. Moreover, some ordinary advanced preprocessing works such as feature selection also can be done in line with feature ordering, although there should be some integrative approaches to make a fusion of these different preprocessing works.

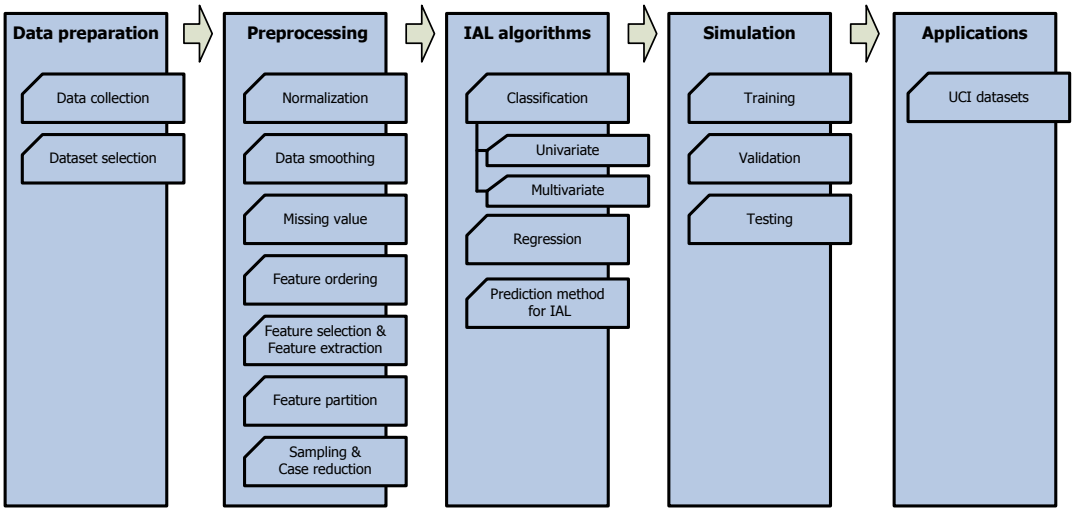


Figure 3.1: A methodology sketch for IAL

After preprocessing, IAL strategy is employed with mature predictive algorithms like neural networks and genetic algorithms. These predictive algorithms can cope with pattern recognition problems, both classification and regression. Almost all these problems have more than one input features. Similarly, many classification problems also have more than one output categories. The outputs of some regression problems also have more than one dimension. If the prediction methods belong to supervised machine learning, all the data should be divided into three groups according to some ratios. The three groups of data are training set, validation set and testing set. Usually, such a process can be applied in some real-world fields. In this study, all the datasets are collected from UCI machine learning repository which was derived from real-world dataset.

This study mainly focuses on feature ordering and its relevant works. Thus, if the feasibility of metrics and algorithms about feature ordering is needed to be precisely evaluated, a proper way is to change metrics and algorithms relating to feature ordering and stabilize all other parts of the IAL process. Figure 3.2 illustrates the changeable parts and stable parts in IAL preprocessing research. For example, when feature ordering is being studied, all parts except feature ordering itself should be kept stable, so that influence from predictive methods or from datasets can be isolated and discarded. More specifically, datasets collection should insure that all experiments choose the same benchmark problems for comparison. Besides that, the prediction methods should be the same one. By taking this study as an example, no other prediction algorithms are employed except ITID. As a result of that, no interference is brought into the final result in the aspect of the prediction algorithms. According to Figure 3.2, if datasets and predictive systems are stable and feature ordering and selection are changeable, once the final results will be changed, it is obvious that the influence is brought by different feature ordering or subset selection.

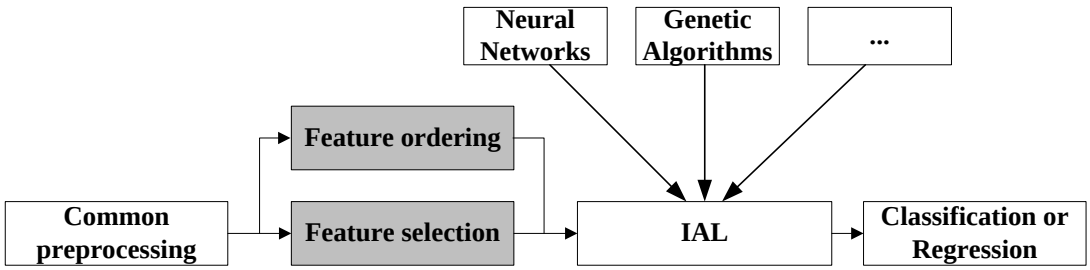


Figure 3.2: Changeable and stable parts in IAL preprocessing research

This chapter focuses on the unchangeable parts in this IAL preprocessing research. In Section 3.2, experimental datasets are introduced. The data sampling and some related works are introduced in Section 3.3. Section 3.4 concentrates on the prediction methods and algorithms of IAL classification and regression.

3.2 Experimental Data

3.2.1 Dataset Descriptions

Data preparation is the first phase in the studies, where data for experiments or applications will be collected. A well-known dataset for machine learning is UCI Machine Learning Repository [33], which have been widely employed in a great number of experiments for a long time. For comparing these previous studies, datasets for simulation are recommended to use UCI machine learning repository such as Diabetes, Glass, Thyroid, Flare, Cancer and so on. These data are frequently used data, which have the superiority for comparing with final results from the same datasets. Here is a brief introduction of some datasets which will be employed in the following experiments. This introduction is given with regards to machine learning and pattern recognition. Further information about the practical significance of these datasets is listed in Appendix A.

Table 3.1 shows the experimental datasets which have been employed in this IAL research for classification and regression. All of these datasets are donated by researchers in professional academic field. These datasets are Diabetes, Cancer [34], Glass, Thyroid, Semeion, Flare, Building, Hearta, and Housing [35]. It is necessary to make it clear that the collection of some datasets is from the benchmark also well-known as Proben1. Proben1 was a set of benchmark problems for neural network studies [36]. It not only provided standard benchmark datasets for neural network study but also made a standard experimental benchmark rules for neural network research. Thus, in the study presented in this thesis, methodology and experimental methods strictly follow the Proben1's rules.

Table 3.1: Experimental Data for IAL Research

#	Dataset	Input Number	Output Number	Pattern Number	Type	Missing Value	Proben1
1	Diabetes	8	2	768	Univariate Classification	Yes	Yes
2	Cancer	9	2	699	Univariate Classification	Yes	Yes
3	Glass	9	6	214	Multivariate Classification	No	Yes
4	Thyroid	21	3	7200	Multivariate Classification	No	Yes
5	Flare	24	3	1066	Multivariate Regression	No	Yes
6	Building	14	3	4208	Multivariate Regression	No	Yes
7	Hearta[1]	35	1	920	Univariate Regression	Yes	Yes
8	Housing	13	1	506	Univariate Regression	No	No
9	Semeion	256	10	1593	Multivariate Classification	No	No

3.2.2 Solutions to Missing Values

In these datasets, some of them, for example, Diabetes, Cancer and Hearta have some missing values. The treatment of missing value for these dataset is different.

The missing values in Diabetes are smoothed by zeros. Diabetes is a special dataset. The website of Diabetes do not indicate that this dataset has missing values before Feb. 28, 2011. A user of this dataset reported that this cannot be true because there are a number of zeros in the dataset that are impossible in biology. Thus missing values in Diabetes have already been replaced by zeros before it was donated [33].

In Cancer, there are 16 missing values in the dataset. All these missing values are existing in the sixth feature. In previous studies of IAL [2, 4, 5, 37-39], these missing values are replaced by the average number of all the values in the sixth feature. Thus in this study, such an approach will be maintained.

In comparison with Diabetes and Cancer, Hearta uses another method to solve the missing value problems. The number of missing values in Hearta is very large. Feature 10, 12, and 11 have 309, 486 and 611 values missing respectively. Furthermore, most other features have around 60 missing values. Additional Boolean input features are employed to mark the status of these missing values. For example, feature 24 has a number of missing values. To deal with such a problem, a new feature is created as feature 25 to give a mark to these missing values. If a value

[1] Hearta is the Heart dataset for function approximation.

in feature 24 is missing, it will be replaced with a "0", and in the meanwhile, at the corresponding place in the newly created feature, it is marked as "1". Otherwise, if there is no missing value in one pattern in feature 24, the corresponding value in feature 25 is a "0". Such a Boolean input feature can effectively give a mark to missing values. In order to coincide with previous research, these missing-value strategies will be maintained in this study.

3.3 Data Sampling and Case Reduction

In statistical classification and regression, there are two useful machine learning meta-algorithms: Boosting [40, 41] and Bagging [42]. The former reduces bias in supervised learning using a kind of weighted voting based on the previous performance created classifiers, while the latter aims to improve the stability and accuracy with an equal weight voting as a combining method. Generally, bagging is more consistent. It increases the error of the base learner less frequently than boosting [43]. Therefore, in this study, bagging is employed.

In order to obtain the optimum structure of IAL in this research, the sampling data for experiments should be consistent. Otherwise, different instance sampling will bring interference into systems, which may be confusing. People will not know what caused the difference between the results, the sampling or the feature ordering. Although a stochastic sampling is crucial in obtaining a more precise experimental result, it also introduces interference into experimental result if the stochastic results of sampling are different. Therefore, to compare conventional approaches, all experiments of IAL should take sampling with similar cases.

Apart from features, the number of samples is another vital factor of the complexity of problems. The number of rows in the spreadsheet of large-dimensional data refers to the number of cases. Because there are always some cases belonging to noise in the datasets, more cases are not always better than fewer cases. Occasionally, the more cases are stored in the datasets, the more interference will be brought into the system. Furthermore, redundant cases often cause over-fitting in some prediction methods such as neural networks and genetic algorithms.

There are three types of problems tending to take more cases than others: multivariate classification problems, regression problems and low-prevalence classification problems. For

multivariate classification problems, each class needs at least one feature to discriminate. Therefore, the more classes in a problem, the more features this problem needs. Moreover, a feature for discriminating a class should be unique or different from others. Thus, more features will bring more cases. A similar situation also exists in regression problems because regression problems can be regarded as a multivariate classification problem with an extremely large number of classes. For low-prevalence classification problems, the number of cases for larger class and that for small class are not in balance. The larger a class is, the more cases this class will take. Thus, this will be harmful in describing the smaller class. Consequently, seeking a balance point of case reduction is crucial for obtaining a precise result.

In this study, supervised machine learning approaches are employed for IAL research. Therefore, the whole machine learning process is divided into two parts: training and testing. Moreover, in order to adjust the parameters of the neural networks to avoid overfitting, it is suggested to add a step called validation, where neural networks can achieve good generalization [44]. Obviously, training is a basic machine learning process, where training data are imported to fit some parameters of the system, such as weight. The validation is a tuned process, where validation data are imported to adjust the architecture of system, for example to choose the number of hidden units in a neural network. Testing is a checking process, where final results will be investigated as a system performance. The reasons why validation datasets are separated from testing datasets are, firstly, estimation of error rate of the model with final structure on validation data is biased, usually smaller than the true error rate, because the validation dataset is used to compose the final architecture of the model; secondly, the structure of the system cannot be changed any further after assessing the final architecture with the relevant test set. Therefore, the dataset should be subdivided into three parts: training set, validation set and testing set, where training set is a set of patterns used for learning, which is to fit the parameters of the classifier; validation set is a set of patterns used to tune the architecture of a classifier; and testing set is a set of patterns used only to assess the performance of a fully specified classifier.

The machine learning process with training, validation and testing can also be called as Online machine learning, which is a basic traditional machine learning method, where the system often learns one pattern at a time. The objective of online machine learning is to label the category for classification problems or estimate a value for function approximate regression

problems. The learning process of an online machine learning approach can be divided into three steps: firstly, receive the instance, secondly, predict the label of the instance, lastly, compare the predictive label with the true label. The last step is the most significant that the predictive algorithm can use this label feedback to check its hypothesis for future trials [45]. Online machine learning has been successfully implemented by a number of predictive algorithms, such as perceptron and neural networks. Previous studies have confirmed that online machine learning are able to adapt in difficult situations. It also performs well when a hyperplane exists that splits the data into two or more categories.

In another aspect, the training process in machine learning can be divided into three different types: Hold-out Validation, K-fold Cross Validation, and Leave One Out Cross Validation.

More specifically, hold-out validation is a simple validation approach, which randomly divides all patterns into three datasets: training, validation and testing. Because the pattern selection process is random, the data distribution of the subdivided datasets will be similar to that of the original dataset. To guarantee the applicability and discard some random impact factors of the predictive algorithms, it is required to repeat the hold-out process for a number of times. The average results of each hold-out result in the repetition can be treated as the formal final results of the prediction method. Whether the prediction algorithm is better than others, this can be estimated based on this average results.

It is obvious that hold-out is easy to carry out. However, the results derived by hold-out is less convincing than those from K-fold cross validation [46]. Cross validation usually segment all the data into K folds, and make the first to K/2 folds as the training set, the K/2+1 to 3K/4 folds as the validation set, and the rest folds are treated as the testing set. Such a segmentation is regarded as the first round. After the first round, folds get rotation. In the second round, the second to K/2+1 folds as the training set, the K/2+2 to 3K/4+1 folds as the validation set, and the rest and the first folds are treated as the testing set. The rotation repeats round and round. Lastly, there are totally K rounds, and K final results are produced. Similar to hold-out, an average number is needed, which is regarded as the representative of the results in the whole machine learning process. Cross validation methods is more stable to exhibit better performance than some other approaches [47]. It also overcomes the disadvantages of hold-out validation. For

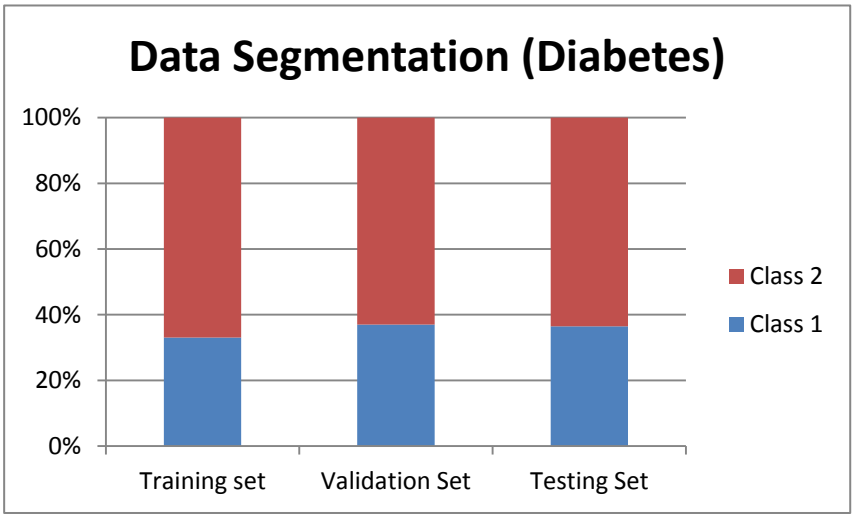
example, hold-out validation is much more arbitrary. it is necessary to determine how many times should be repeated during the hold-out training process. The number of iteration times is difficult to be decided. However, in cross validation, it is only necessary to consider how many rotations should be made, which directly impacts the fold number. Actually, the more fold number, the more accurate result will be obtained. An extreme case of cross validation is leave one out cross validation.

The leave one out cross validation is improved from k-fold cross validation. Similar to ordinary k-fold cross validation, leave one out cross validation also divides data into training set, validation set and testing set, also rotate the division, also use the average number to measure whether the predictive method is overwhelming or not. However, different from ordinary k-fold cross validation, leave one out cross validation always leave one instance as the testing set. As such, the number of testing pattern is always one. It is believed that the leave one out cross validation can produce more accurate final results than ordinary k-fold cross validation. However, it is obvious that leave one out cross validation is quite time-consuming.

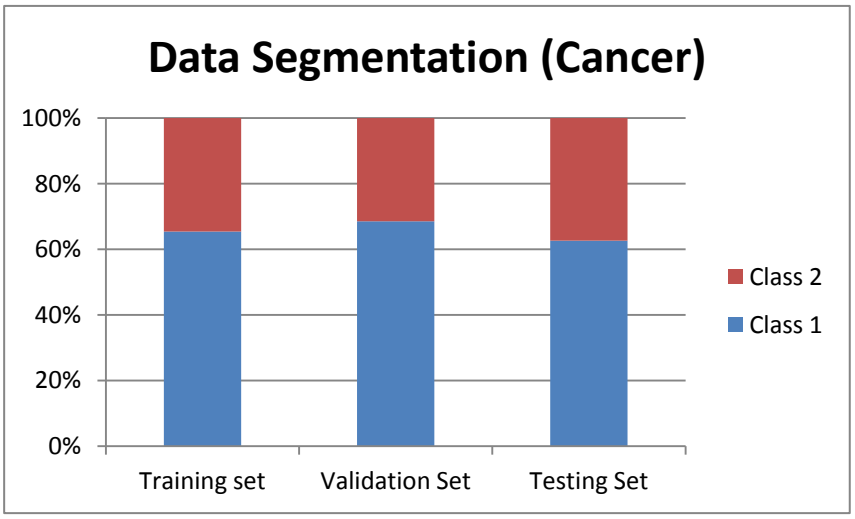
Therefore, no matter which kind of validation is employed in this study, all the data should be segmented into three subsets. However, in this research, the main work concentrates on the preprocessing of IAL. Hence, the key of the experimental result comparison is not in the aspect of predictive methods, but in the aspect of preprocessing like feature ordering. In order to make the comparison of preprocessing be clearer, it is necessary to keep the prediction methods to be stable. Once the prediction methods and datasets is fixed, the only reason why final results are different is due to preprocessing such as feature ordering. Then, it is easy to observe which kind of preprocessing approach is better. In this way, as it is necessary to keep the prediction method stable, the type of validation that should be employed in this research also becomes insignificant. Therefore, for easier calculation, hold-out validation is employed. Moreover, it is also unnecessary to random segment patterns into training, validation and testing datasets multiple times. On the contrary, one time random segment is adequate for this IAL preprocessing research. According to the benchmark rules made in Proben 1, the proportion of the number of training data, validation data and test data is 50%, 25% and 25%. The data segmentation of UCI classification datasets used in the experiments are shown in Table 3.2 and Figure 3.3, and data segmentation of UCI regression datasets are shown in Table 3.3.

Table 3.2: Data Segmentation of Classification Datasets

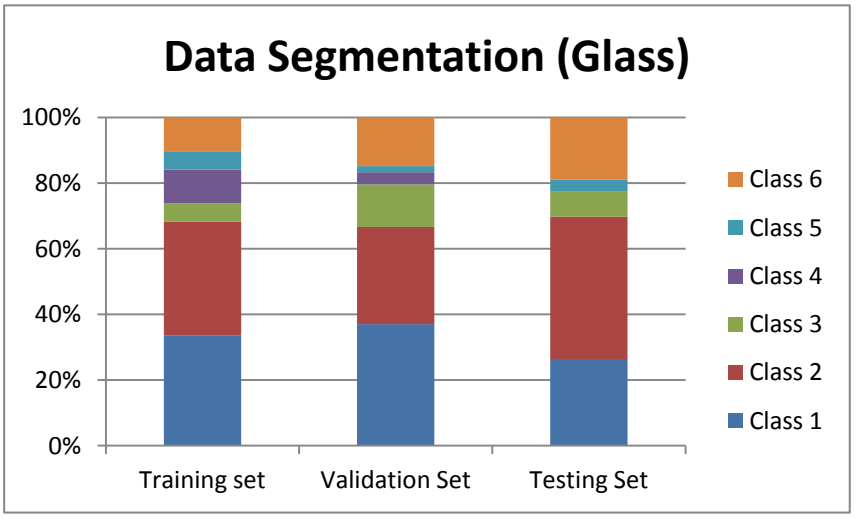
#	Name	Training set	Validation Set	Testing Set
1	Diabetes	Total: 384 Class 1: 127 (33.07%) Class 2: 257 (66.93%)	Total: 192 Class 1: 71 (36.98%) Class 2: 121 (63.02%)	Total: 192 Class 1: 70 (36.46%) Class 2: 122 (63.54%)
2	Cancer	Total: 350 Class 1: 229 (65.43%) Class 2: 121 (34.57%)	Total: 175 Class 1: 120 (68.57%) Class 2: 55 (31.43%)	Total: 174 Class 1: 109 (62.64%) Class 2: 65 (37.36%)
3	Glass	Total: 107 Class 1: 36 (33.64%) Class 2: 37 (34.58%) Class 3: 6 (5.61%) Class 4: 11 (10.28%) Class 5: 6 (5.61%) Class 6: 11 (10.28%)	Total: 54 Class 1: 20 (37.04%) Class 2: 16 (29.63%) Class 3: 7 (12.96%) Class 4: 2 (3.7%) Class 5: 1 (1.85%) Class 6: 8 (14.81%)	Total: 53 Class 1: 14 (26.42%) Class 2: 23 (43.40%) Class 3: 4 (7.55%) Class 4: 0 (0.00%) Class 5: 2 (3.77%) Class 6: 10 (18.87%)
4	Thyroid	Total: 3600 Class 1: 91 (2.53%) Class 2: 181 (5.03%) Class 3: 3328 (92.44%)	Total: 1800 Class 1: 35 (1.94%) Class 2: 96 (5.33%) Class 3: 1669 (92.72%)	Total: 1800 Class 1: 40 (2.22%) Class 2: 91 (5.06%) Class 3: 1669 (92.72%)
5	Semeion	Total: 796 Class 1: 80 (10.05%) Class 2: 82 (10.30%) Class 3: 79 (9.92%) Class 4: 79 (9.92%) Class 5: 81 (10.18%) Class 6: 79 (9.92%) Class 7: 81 (10.18%) Class 8: 79 (9.92%) Class 9: 78 (9.92%) Class 10: 78 (9.92%	Total: 399 Class 1: 41 (10.28%) Class 2: 40 (10.03%) Class 3: 40 (10.03%) Class 4: 40 (10.03%) Class 5: 40 (10.03%) Class 6: 40 (10.03%) Class 7: 40 (10.03%) Class 8: 39 (9.77%) Class 9: 39 (9.77%) Class 10: 40 (10.03%)	Total: 398 Class 1: 40 (10.05%) Class 2: 40 (10.05%) Class 3: 40 (10.05%) Class 4: 40 (10.05%) Class 5: 40 (10.05%) Class 6: 40 (10.05%) Class 7: 40 (10.05%) Class 8: 40 (10.05%) Class 9: 38 (9.55%) Class 10: 40 (10.05%)



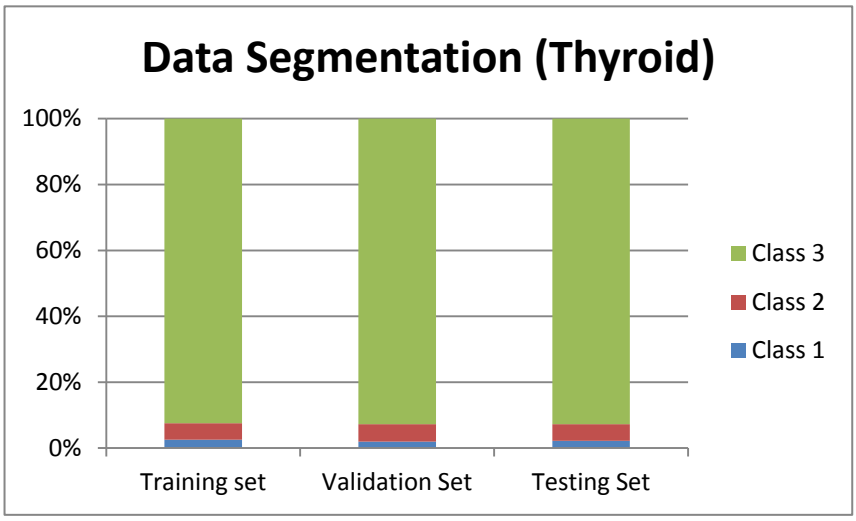
(a). Data Segmentation of Diabetes



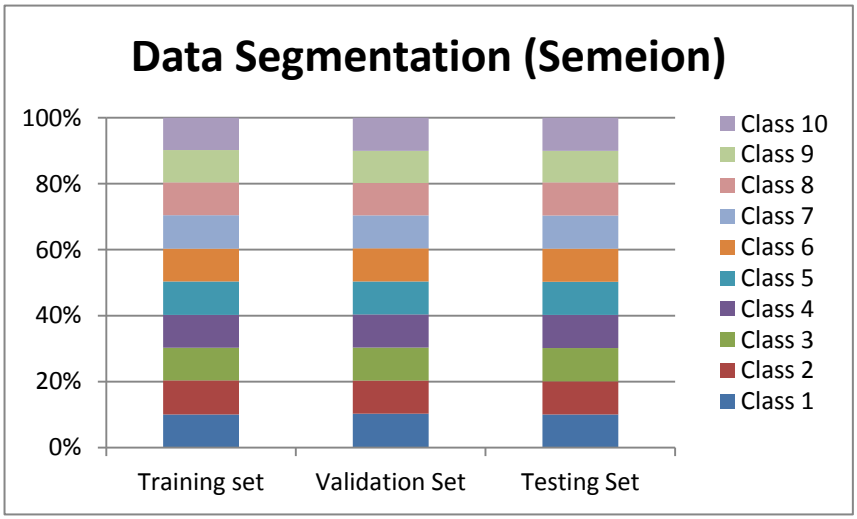
(b). Data Segmentation of Cancer



(c). Data Segmentation of Glass



(d). Data Segmentation of Thyroid



(e). Data Segmentation of Semeion

Figure 3.3: Data Segmentation of Classification Datasets

Table 3.3: Data Segmentation of Regression Datasets

#	Name	Training set	Validation Set	Testing Set
1	Flare	Total: 533	Total: 267	Total: 266
2	Building	Total: 2104	Total: 1052	Total: 1052
3	Hearta	Total: 460	Total: 230	Total: 230
4	Housing	Total: 253	Total: 127	Total: 126

3.4 Experimental Methodology

This subsection is about the neural network algorithms and software programs used in this study. Although this study aims to emphasize significance of the preprocessing of IAL, it is still necessary to give a brief introduction on the kind of predictive method and the kind of programs that have been employed in this research. The experimental methodology used here should be the same as the one in previous studies, so that the results impacted by the preprocessing in this study and those derived in previous research can be acceptable and comparable. Therefore, the working procedure of neural network and software programs should be fixed and unchangeable during the whole pattern recognition process.

3.4.1 Neural Networks and Prediction Methods

As we know, neural networks have exhibited a wonderful performance in pattern recognition. It usually employs supervised machine learning strategy to cope with classification and regression problems. As neural networks can work in an evolutionary way [48, 49] which matches the characters of IAL where features are trained incrementally, this study will also employ neural networks to tackle with prediction tasks.

Neural networks have a number of variants. In this study, **Resilient Backpropagation (RPROP)** algorithm is employed with ITID [4], as this has been adopted in most previous IAL studies for many times, and it is also necessary to keep the consistency between new predictive approach and previous approaches. More specifically, features of datasets in classification or regression problems are arranged to be imported into the prediction system like neural networks one by one or group by group according to the strategy of ITID. When one or some new features are imported, they will be trained in a constructive way using RPROP. This constructive way in our research is based on ILIA1 or ILIA2 [21], and RPROP in such a constructive way also can be regarded as a variant of **Constructive Backpropagation (CBP)**. In previous research, CBP has played a successful role in pattern classification [39, 50].

RPROP is a heuristic supervised machine learning approach in feedforward artificial neural networks. It is a first-order optimization algorithm, also, one of the fastest weight update

mechanisms. It was firstly developed by Martin Riedmiller and Heinrich Braun in 1992 [51]. In this study, all the parameters of RPROP are set to be the same as those in previous IAL studies [4, 21]. Moreover, the stopping criteria are also the same as those presented in previous studies [36]. In this study, an early stopping criteria is employed. As we know, constructive learning algorithms have many advantages [52-57], however, they are very sensitive to change in the stopping criteria [21]. Neural networks may not generate good results, if training is too short. However, if training is too long, it will spend much computation time and may get overfitting and poor generalization. By referring to [36, 52], the method of early stopping using a validation set to prevent overfitting is adopted. The detailed introduction to parameter setting and stopping criteria is presented in Appendix C.

3.4.2 Programs of IAL based on Neural Networks

In this study, the prediction system based on neural networks was developed by C++. Some basic codes such as foundation classes Base_Node, Base_Link, is from the book "*Object-oriented neural networks in C++*" [58]. Other advanced classes were developed based on these basic classes. Figure 3.3 shows the class hierarchy in RPROP IAL code. Our program uses "winner-takes-all" to predict the final results. When the program is launched, the initial network is firstly built. After that, if the initial networks cannot get the acceptable generalization performance, the hidden units will be added. When a hidden unit is added, a pool of candidates should be trained and the best one is selected.

In our experiments, error rates derived by ITID (ILIA1) and ITID (ILIA2) will be used as final results for comparison. This IAL program can produce five different prediction results, which are derived from ITID with ILIA1, 2, 3, 4, and 5, respectively. The reasons why results from ITID (ILIA1) and ITID (ILIA2) are chosen are: firstly, referring to [21], "ILIA2 algorithm is better than the other ILIA algorithms. " Secondly, previous research often employs results from ILIA1 and ILIA2 as the formal final results for comparison [4]. Therefore, error rates derived by ITID (ILIA1) and ITID (ILIA2) are chosen for comparison in the last stage. Further, in this thesis, except error rates derived by ITID (ILIA1) and ITID (ILIA2), the average value of ITID (ILIA1) and ITID (ILIA2) will also be taken as a comparison metric. This is because these two metrics

are related to each other and thus the average value can be a representative to measure the stability of the performance of feature ordering with ITID-ILIA algorithms. Some further information about the program operation used in this study is presented in Appendix D.

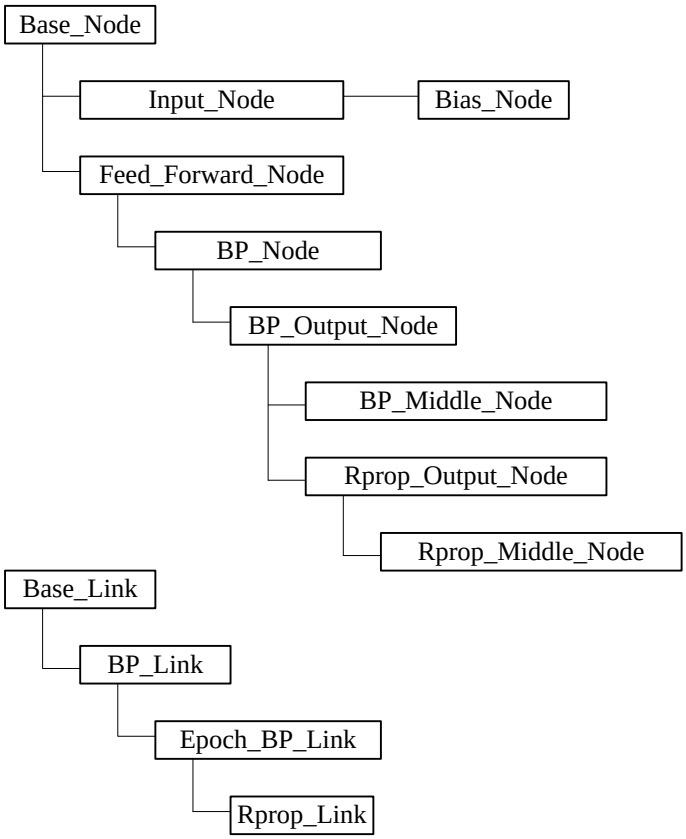


Figure 3.4: Class Hierarchy in RPROP IAL

3.5 Summary

This chapter mainly presented the methodologies about experimental data descriptions and IAL predictive approaches used in this study. To make the results derived from this study can be compared with those from previous studies, datasets used in this study are identically collected from UCI Machine Learning Repository. Moreover, based on UCI datasets, conventional data preparation like data sampling and case reduction is carried out. All the datasets are divided into three parts: training, validation and testing. Based on these division of datasets, IAL neural networks like ITID can be employed for pattern classification and regression. During the experimental process, it is necessary to keep dataset segmentations and neural network structures

stable, so that the stage of preprocessing can be treated as the only source for all the fluctuations derived from final results by different approaches. In the forthcoming chapters, novel feature ordering metrics and approaches will be presented and compared. All of the later works will be based on the methodologies showed in this chapter.

Chapter 4

Feature Ordering based on Correlations

4.1 Correlations

In previous studies on features, correlation has been regarded as a significant measure for feature discrimination ability that can be employed to rank features. Such a function of correlation has already been applied in feature selection [59, 60]. In this section, correlation will be employed for feature ordering.

Mathematically, correlation is a measurement that is used to calculate the relation between two vectors. A correlation between two vectors is a kind of simple correlation, which can be marked by **Correlation Coefficient**, also known as **Pearson Product-moment Correlation Coefficient**. In Statistics, Correlation Coefficient gives a value in [-1, 1] to measure the correlation (linear dependence) between two variables *X* and *Y*. Correlation Coefficient between *X* and *Y* is defined as the covariance of the two variables divided by the product of their standard deviations, which is given in Eq.(4.1) for a population and Eq.(4.2) for a sample:

For a population:

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y} \tag{4.1}$$

where σ is the standard deviation, cov is the covariance, and μ is the mean.

For a sample:

$$r_{X,Y} = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}} \tag{4.2}$$

Some of the Correlation Coefficients are positive, while the others are negative or zero. If the Correlation Coefficient is zero, it means that there is no correlation between these two variables. The closer the Correlation Coefficient to 1, the stronger of the positive correlation is; and the closer of the Correlation Coefficient to -1, the stronger of the negative correlation is. No matter what kinds of correlations between two vectors are, the stronger of the correlation. Hence, it seems that the absolute value of Correlation Coefficient is more important than the coefficient itself. The absolute value of Correlation Coefficient can be used as a metric to measure the correlation between input and output in pattern recognition problems.

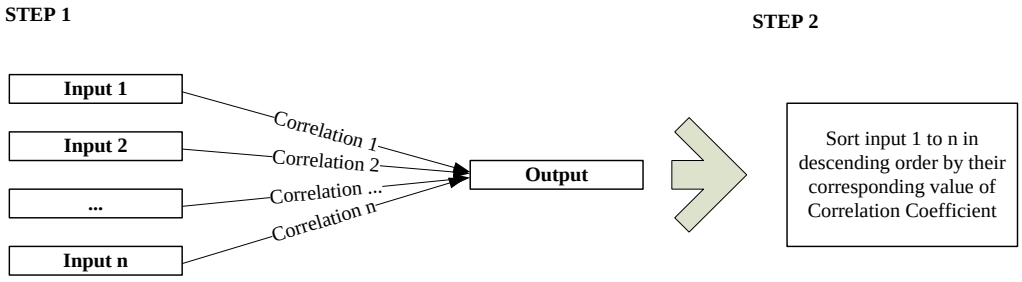
4.2 Input-Output Correlation-based Feature Ordering

In pattern recognition problems such as classification and regression, most of the input features are strongly or weakly correlated to the output attributes. According to the value, if the Correlation Coefficient between input and output is close to 1, it means that the input can seriously impact on the output, while once the coefficient is close to 0, which means that there is little correlation between input and output.

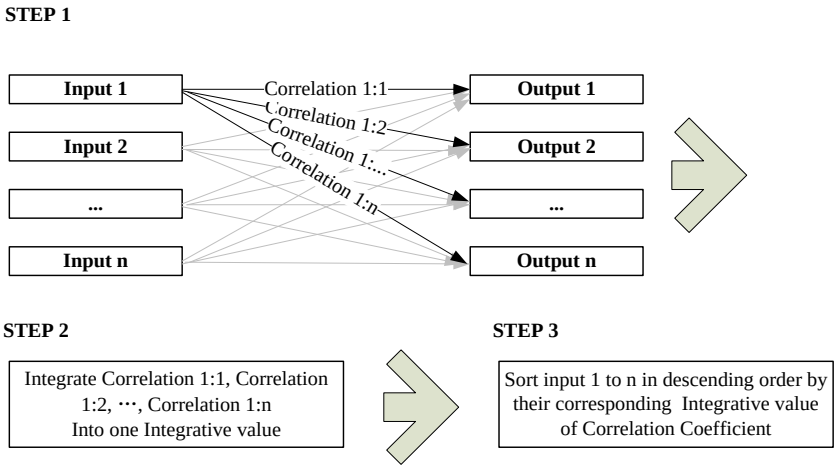
Obviously, the input features which can greatly influence the output attributes absolutely have greater discrimination ability than those that cannot. If all the features should be sorted in some orderings for IAL pattern classification or regression, the feature which have greater discrimination ability should be imported into the predictive system in an earlier stage. Therefore, we can employ Correlation Coefficients between each input feature and output to rank features and sort them. For univariate output problems, feature ordering can be directly sorted according to the input-output correlation. However, for multivariate output problems, it is necessary to integrate the multiple output attributes to one output in the first place. Similarly, this is also to integrate partial input-output correlations into one correlation value before sorting. Figure 4.1

illustrates different ways to sort features by input-output correlation according to different situations. More specifically, Figure 4.1(a) shows the approach of feature sorting by input-output correlation for univariate problems, whereas (b) and (c) demonstrate methods of feature ordering by input-output correlation multivariate output problems. It is manifested that the approach shown in (a) can be used in both classification and regression problems. However, for multivariate output problems, only approach (b) can be used for feature ordering calculation in both classification and regression problems. Approach (c) is not suitable for regression problems, especially for multivariate output regression problems, because the output estimation value in multivariate regression problems is a multidimensional value, which is presented in more than one dimension. This is quite different from multivariate classification problems as no patterns can belong to more than one classes at a time. The output attributes for multivariate classification problems are always in the "one-yes, others-no" style, while all the output attributes for multivariate regression problems are usually have their own values at the same time. Therefore, approach (c), which makes all output convert to one output, is inappropriate for the multivariate regression problems. This only can be employed in multivariate classification problems.

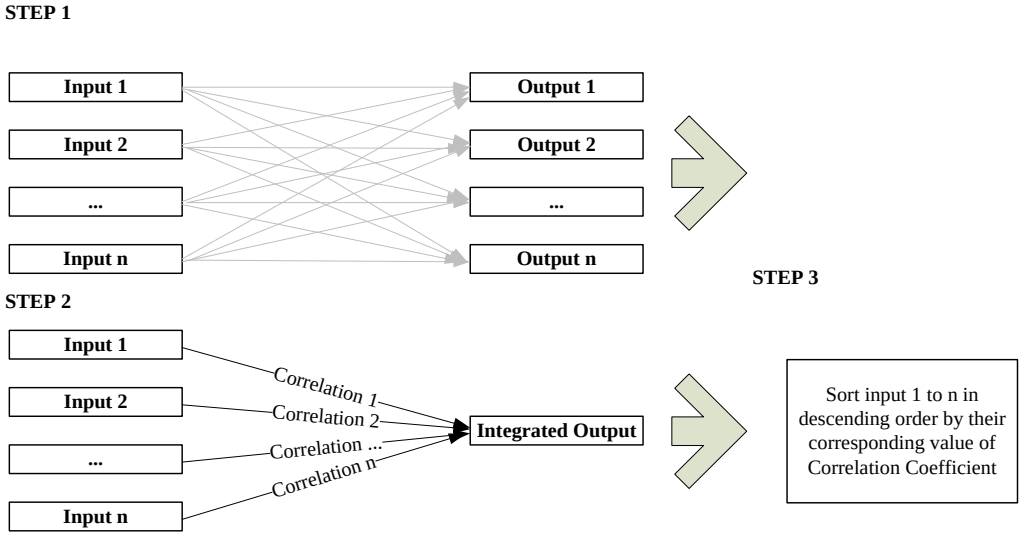
For classification problems, before approach (b) is employed, if the output attributes are categorical, it should be transformed into "0-1" style in order to get rid of the bias introduced by different values of output. In the case of Glass, a UCI dataset, it has 6 output attributes. If they are all marked as univariate, they should be marked with "1, 2, 3, 4, 5, 6". However, it is difficult to determine which class should be marked as 1, which should be marked as 6. Otherwise, if all the classes are marked with "1, 2, 3, 4, 5, 6" in one vector, it is hard to find the reason why one class should be marked as 1, and not 6. Therefore, simply marking outputs with "1, 2, 3, 4, 5, 6" in one vector is not suitable. The output in Glass should be marked in a six-dimension vector. If one pattern belongs to the first class, the first variate should be marked as 1, while others in the same pattern output should be marked as 0. Thus "1, 0, 0, 0, 0, 0" denotes that the pattern belongs to the first class, while "0, 1, 0, 0, 0, 0" indicates that the pattern belongs to the second class, and so on. Such a marking process is unbiased, and the input-output correlation for these outputs can be calculated by their average of each input-output correlation coefficients, or their weighted average according to their number in every class.



(a). Sorting features by input-output correlation in univariate output problems



(b). Sorting features by input-output correlation in multiple output problems with correlation integration



(c). Sorting features by input-output correlation in multiple output problems with output integration

Figure 4.1: Different ways to sort features by input-output correlations

For multivariate regression problems, before approach (b) is applied, they have no output marking process like classification as each output has its own meaning, and their multiple output

attributes are equally important to each other, no one else is more significant than the other. Therefore, they cannot be integrated into one output, and they cannot be marked in one vector. Furthermore, because all output are equally important, weighted correlation-base feature ordering cannot be employed in the multivariate regression problems. They must use the average of each input-output correlation coefficients for feature ordering.

4.3 Feature Ordering based on Integrated Correlation

Conventionally, there are three types of correlations in the datasets of classification problems: correlation between input features, correlation between input features and output classes, and correlation between output classes. Classification has an either-or property, thus no patterns belong to two or more classes simultaneously. Consequently, it is only necessary to check the first two types of correlation for one dataset. Therefore, the relation between input and output, and the relation among each features should be investigated in IAL.

As a matter of fact, the study on correlation in pattern recognition is not something new. Previous research on correlation in pattern recognition aim to develop feature selection approaches that can be used to alleviate the effect of the curse of dimensionality, enhance generalization capability, speed up learning process and to improve model interpretability [59]. Furthermore, most of the previous research in this area focused on feature selection. In order to achieve the above mentioned objectives, feature selection approaches are divided into two categories: feature subset selection and feature ranking. The former searches a set of possible features for the optimal subset while the latter ranks features by a metric and discards all features whose score is under the threshold according to some criteria. When correlation analysis is employed in the feature selection process, both feature ranking and subset searching can be employed for classification. Previous research confirmed that good feature subsets often contain high-correlated features with the classification, but they are uncorrelated to each other [59].

Therefore, in the process of feature selection, for feature ranking, we should select features which not only have high correlation with outputs, but also have low correlation with each other; for feature subset selection, we should search the optimal feature subset that has high correlations

with classification outputs and low correlations among themselves. Thus, for feature subset selection and feature ranking, no matter which type is selected for feature selection, these two correlation analysis approaches for classification are the same in essence.

Moreover, in IAL, data preparation is quite different from conventional machine learning approaches, where features are trained by batch. Feature ordering, a new data preprocessing stage, is deemed as a requirement before training. Due to the fact that feature ranks have different values which can be employed as a measurement to arrange features in some order, feature ranking is more useful in data preprocessing phase than subset searching. Accordingly, features should be trained one by one according to the order derived by the fusion of correlations between input features and that with input and output together.

Correlation-based feature ordering can be calculated by **Correlation Index**. Correlation Index of *i*-th feature can be computed by

$$CorrelationIndex_i = \frac{|Correlation(Input_i, Output)|}{\left(\sum_{j=1}^{n}|Correlation(Input_i, Input_j)|\right)/n} \tag{4.3}$$

which is the ratio between correlation of *i*-th input and all output, and the average correlation between *i*-th feature and all other input features. Furthermore, correlation can be calculated by Pearson Correlation Coefficient or Covariance Matrix by Eq.(4.2) and *n* is the number of input features. Similar to correlation-based feature selection, it is obvious that the greater the correlation index in Eq.(4.3), the earlier the feature should be trained.

It is obvious that Eq.(4.3) can cope with the problem which only has one output, or all its outputs are integrated before using Eq.(4.3). If the problem has more than one outputs, and it is necessary to calculate the Correlation Index of *i*-th feature to *k*-th output, Eq.(4.3) can be rewrite as:

$$CorrelationIndex_{i,k} = \frac{|Correlation(Input_i, Output_k)|}{\left(\sum_{j=1}^{n}|Correlation(Input_i, Input_j)|\right)/n} \tag{4.4}$$

Such a formula can respectively compute all Correlation Indices for multivariate classification problems.

4.4 Weighted Correlation-based Feature Ordering

In multivariate pattern classification problems, different classes usually have different numbers of patterns. In most of the problems, the numbers of patterns belonging to different classes are unequal and out of a proportion. Intuitively, classes which have more patterns may have stronger influence than those which have fewer patterns. Thus, whether the number of different patterns is an element which may impact on the final results is necessary to be studied.

In multivariate problems, each output has one Correlation Coefficient with input features. If we do not employ output integration approach for feature ordering, then according to Figure 4.1(b), correlation of different output attributes must be integrated by some calculation formulae. A potential approach is to summarize all these correlation coefficients by their influence in the entire problem. Mathematically, the influence can be computed by the weight of categories.

Supposing that p is the number of total patterns, p_i is the number of patterns belonging to j-th class, the influence, also the weight, brought by j-th class can be calculated by

$$\omega_j = \frac{p_j}{p} \tag{4.5}$$

Therefore, base on Eq.(4.5), if there is an n-category classification problem, the **Integrated Correlation Coefficient (ICC)** of i-th input feature and outputs can be given by

$$ICC_i = \omega_1 r_1 + \omega_2 r_2 + \cdots + \omega_n r_n \tag{4.6}$$

where r is the sample correlation coefficient between each output and i-th input feature given in Eq.(4.2). According to Eq.(4.6), the input-output correlation coefficients of one feature in a multivariate problem can be integrated into one value, and a multivariate classification problem with m input features and n output attributes, which should have an $m \times n$ matrix about input-output correlation coefficients, will has an $m \times 1$ vector about integrated input-output correlation coefficients eventually. Therefore, a multivariate classification problem is simply converted to a univariate classification problem.

4.5 Experiments

Experiments on correlation-based feature ordering are carried out in two aspects: simple

input-output correlation and integrated correlation. For multivariate output problems, apart from solely computing the simple input-output correlation and integrated correlation, weighted correlation is also employed for these two kinds of correlations, so that we can find whether pattern numbers of one category will impact on the final results. These experiments aim to check the feasibility of the approaches on correlation-based feature ordering.

1. Diabetes

Diabetes is a univariate output problem. Thus it is unnecessary to merge the weights into correlations. Table 4.1 shows correlations of features and output. Feature ordering based on simple input-output correlation is obtained in descending order of the absolute value of correlation coefficients of each input feature and output. Based on the input-input and input-output correlation coefficients shown in Table 4.2, correlation index is calculated based on Eq.(4.3), which is also presented in Table 4.2, and feature ordering can be derived by the descending order of correlation index. Table 4.3 compares two kinds of correlation-based feature orderings with conventional approach which trains all features in one batch. The results show correlation-based feature ordering can exhibit a better performance.

Table 4.1: Correlations of Features and Output (Diabetes)

Feature	1	2	3	4	5	6	7	8
1	1							
2	0.0978	1						
3	0.1148	0.1604	1					
4	-0.1361	0.0969	0.2582	1				
5	-0.1456	0.3270	0.0888	0.4526	1			
6	-0.0309	0.2700	0.2610	0.4437	0.2432	1		
7	-0.0180	0.1652	0.0745	0.2280	0.2298	0.1615	1	
8	0.6014	0.2048	0.2379	-0.1181	-0.1538	0.0667	0.0192	1
Output	0.1931	0.4480	0.0691	0.1416	0.1300	0.3363	0.2126	0.2373
Ordering	5	1	8	6	7	2	4	3

Table 4.2: Correlation Index and Feature Ordering (Diabetes)

	$\sum_{j=1}^{n}\lvert Correlation(Input_i, Input_j)\rvert$	$\lvert Correlation(Input_i, Output)\rvert$	Feature	Correlation Index
1	1.7701	0.4480	2	0.031637
2	1.1088	0.2126	7	0.023967
3	1.8133	0.3363	6	0.023183
4	1.6392	0.2373	8	0.018096
5	1.3377	0.1931	1	0.018044
6	1.8752	0.1416	4	0.009439
7	1.7708	0.1300	5	0.009177
8	1.2647	0.0691	3	0.006830

Table 4.3: Classification Error of Correlation-based Feature Ordering(Diabetes)

	Approach	Feature Ordering	Classification Error (%)		
			ITID (ILIA1)	ITID (ILIA2)	AVG[2]
1	Input-Output Correlation-based	2-6-8-7-1-4-5-3	21.84896	22.39583	22.12240
2	Integrated Correlation-based	2-7-6-8-1-4-5-3	21.32812	22.47396	21.90104
3	Conventional Method	No Feature Ordering	23.93229		

2. Cancer

Cancer is a univariate output problems. Similar to Diabetes, weights have not been fused into correlations. Correlations of features and output are shown in Table 4.4 . Feature ordering based on a simple input-output correlation is obtained according the descending order of the absolute value of correlation coefficients of each input feature and output. Based on the input-input and input-output correlation coefficients shown in Table 4.4, correlation index is calculated according to Eq.(4.3) in Table 4.5, and feature ordering can be derived by the descending order of correlation index. Table 4.6 compares two kinds of correlation-based feature orderings with conventional one-batch training approach. The results show that simple input-output correlation-based feature ordering can exhibit a better performance, whereas the results of integrated correlation-based approach are not better than the conventional one. It indicates that correlation-based feature ordering approach is unstable. Correlation is not the only element which may influence the final classification results.

[2] AVG stands for an average number.

Table 4.4: Correlations of Features and Output (Cancer)

Feature	1	2	3	4	5	6	7	8	9
1	1								
2	0.6284	1							
3	0.6493	0.9083	1						
4	0.4556	0.6879	0.7076	1					
5	0.5037	0.7359	0.7266	0.5847	1				
6	0.5616	0.6624	0.6968	0.6395	0.5484	1			
7	0.5439	0.7400	0.7305	0.6715	0.5839	0.6632	1		
8	0.5012	0.6952	0.6842	0.5588	0.5811	0.5476	0.6221	1	
9	0.3317	0.4221	0.4419	0.4263	0.4520	0.3330	0.3697	0.3868	1
Output	-0.7043	-0.8026	-0.8135	-0.6780	-0.6634	-0.7771	-0.7417	-0.6953	-0.4275
Ordering	5	2	1	7	8	3	4	6	9

Table 4.5: Correlation Index and Feature Ordering (Cancer)

	$\sum_{j=1}^{n} \lvert Correlation(Input_i, Input_j) \rvert$	$\lvert Correlation(Input_i, Output) \rvert$	Feature	Correlation Index
1	4.8797	0.7043	1	0.016037
2	5.4296	0.7771	6	0.015903
3	5.2723	0.6953	8	0.014653
4	5.6665	0.7417	7	0.014544
5	6.3587	0.8135	3	0.014215
6	6.2828	0.8026	2	0.014194
7	5.4099	0.6780	4	0.013925
8	5.3797	0.6634	5	0.013702
9	3.5910	0.4275	9	0.013228

Table 4.6: Classification Results of Correlation-based Feature Ordering(Cancer)

	Approach	Feature Ordering	Classification Error (%)		
			ITID (ILIA1)	ITID (ILIA2)	AVG
1	Input-Output Correlation-based	3-2-6-7-1-8-4-5-9	1.69541	1.72414	1.70977
2	Integrated Correlation-based	1-6-8-7-3-2-4-5-9	1.83908	2.01150	1.92529
3	Conventional Method	No Feature Ordering	1.86782		

3. Glass

Glass has six output attributes. It is carried out by both simple input-output correlation-based feature ordering and integrated correlation feature ordering. Aside from this, both of these two correlations are combined with the weights derived from different pattern numbers belonging to different classes. Table 4.7 demonstrates three different kinds of feature orderings, where

Ordering 1 is obtained by directly merging all six output attributes into one single output; Ordering 2 is derived by the average number of all input-output correlation; in Ordering 3, all outputs are improved by weights, which are shown in Table 4.8. Table 4.9, 4.10, 4.11 show that the Correlation Index derived by three different ways: average output, weighted output and output integration, respectively. Final classification results are shown in Table 4.12, where ITID (ILIA2) with correlation-base feature ordering can obtain a better performance compared with conventional methods. However, the classification error rates of Output Integration using ITID (ILIA1) are not satisfactory.

Table 4.7: Correlations of Features and Outputs (Glass)

Feature	1	2	3	4	5	6	7	8	9
1	1								
2	-0.2469	1							
3	-0.0324	-0.2144	1						
4	-0.4179	0.0719	-0.4819	1					
5	-0.4901	0.0145	-0.1856	-0.1192	1				
6	-0.3101	-0.2677	-0.1335	0.4204	-0.2635	1			
7	0.8317	-0.2844	-0.4430	-0.2237	-0.1808	-0.2912	1		
8	-0.2086	0.3946	-0.3821	0.5488	-0.0559	-0.0681	-0.1896	1	
9	0.2510	-0.2566	0.0012	-0.0055	-0.0716	-0.0334	0.1948	-0.0519	1
Output	-0.0276	-0.0438	-0.1712	0.1074	-0.0903	0.2061	0.1261	-0.2405	0.0155
Ordering 1	8	7	3	5	6	2	4	1	9
ABS[3](Output 1)	0.1687	0.0682	0.4565	0.4381	0.0893	0.0984	0.0865	0.2105	0.0526
ABS(Output 2)	0.0879	0.2481	0.1757	0.0945	0.0320	0.0283	0.0578	0.2082	0.2132
ABS(Output 3)	0.0424	0.0350	0.1563	0.0571	0.1132	0.0180	0.0403	0.0758	0.1014
ABS(Output 4)	0.0123	0.1775	0.4146	0.4218	0.1612	0.4374	0.1648	0.1010	0.0599
ABS(Output 5)	0.1349	0.3919	0.1833	0.0311	0.1987	0.1532	0.0190	0.0758	0.1395
ABS(Output 6)	0.2536	0.4018	0.5503	0.4746	0.2854	0.1104	0.0759	0.6673	0.1295
Output AVG	0.1166	0.2204	0.3228	0.2529	0.1466	0.1409	0.0740	0.2231	0.1160
Ordering 2	7	4	1	2	5	6	9	3	8
Output Weight	0.02074	0.03204	0.05543	0.04620	0.01742	0.01813	0.01286	0.03838	0.02073
Ordering 3	5	4	1	2	8	7	9	3	6

[3] ABS stands for Absolute Value.

Table 4.8: Output Weight (Glass)

Output	Number	Total Number of Training Patterns	Weight
Output 1	36	107	0.336449
Output 2	37	107	0.345794
Output 3	6	107	0.056075
Output 4	11	107	0.102804
Output 5	6	107	0.056075
Output 6	11	107	0.102804

Table 4.9: Correlation Index and Feature Ordering (Output Average Weight, Glass)

	$\sum_{j=1}^{n}\lvert Correlation(Input_i, Input_j)\rvert$	$\lvert Correlation(Input_i, Output)\rvert$	Feature	Correlation Index
1	2.0453	0.3228	3	0.017536
2	0.8815	0.1160	9	0.014622
3	1.7948	0.2204	2	0.013644
4	2.3967	0.2529	4	0.011724
5	2.1401	0.2231	8	0.011583
6	1.4715	0.1466	5	0.011070
7	1.9940	0.1409	6	0.007851
8	2.8163	0.1166	1	0.004600
9	2.7653	0.0740	7	0.002973

Table 4.10: Correlation Index and Feature Ordering (Weighted Output, Glass)

	$\sum_{j=1}^{n}\lvert Correlation(Input_i, Input_j)\rvert$	$\lvert Correlation(Input_i, Output)\rvert$	Feature	Correlation Index
1	2.0453	0.05543	3	0.003011
2	0.8815	0.02073	9	0.002613
3	2.3967	0.04620	4	0.002142
4	2.1401	0.03838	8	0.001993
5	1.7948	0.03204	2	0.001984
6	1.4715	0.01742	5	0.001315
7	1.9940	0.01813	6	0.001010
8	2.8163	0.02074	1	0.000818
9	2.7653	0.01286	7	0.000517

Table 4.11: Correlation Index and Feature Ordering (Output Integration, Glass)

	$\sum_{j=1}^{n}\lvert Correlation(Input_i, Input_j)\rvert$	$\lvert Correlation(Input_i, Output)\rvert$	Feature	Correlation Index
1	2.1401	0.2405	8	0.012486
2	1.9940	0.2061	6	0.011484
3	2.0453	0.1712	3	0.009300
4	1.4715	0.0903	5	0.006818
5	2.7653	0.1261	7	0.005067
6	2.3967	0.1074	4	0.004979
7	1.7948	0.0438	2	0.002712
8	0.8815	0.0155	9	0.001954
9	2.8163	0.0276	1	0.001089

Table 4.12: Classification Results of Correlation-based Feature Ordering (Glass)

	Approach	Feature Ordering	Classification Error (%)		
			ITID (ILIA1)	ITID (ILIA2)	AVG
1	Input-Output Correlation-based Output AVG	3-4-8-2-5-6-1-9-7	40.37733	35.56605	37.97169
2	Input-Output Correlation-based Output Weight	3-4-8-2-1-9-6-5-7	40.28300	35.66040	37.97170
3	Input-Output Correlation-based Output Integration	8-6-3-7-4-5-2-1-9	54.15097	38.20755	46.17926
4	Integrated Correlation-based Output AVG	3-9-2-4-8-5-6-1-7	34.24530	32.26417	33.25474
5	Integrated Correlation-based Output Weight	3-9-4-8-2-5-6-1-7	41.03771	36.60378	38.82075
6	Integrated Correlation-based Output Integration	8-6-3-5-7-4-2-9-1	53.39625	38.20755	45.80190
7	Conventional Method	No Feature Ordering	41.22641		

4. Thyroid

Thyroid has three output attributes. Similar to Glass, it is carried out by both simple input-output correlation-based feature ordering and integrated correlation feature ordering. Moreover, both of these two correlations are combined with weights derived from different pattern numbers belonging to different classes. Table 4.13 shows three different kinds of feature orderings, where Ordering 1 is obtained by consolidating all the three outputs into one output; Ordering 2 is derived by the average number of all input-output correlation; in Ordering 3, all outputs were improved by weights, which are shown in Table 4.14. Tables 4.15, 4.16, 4.17 show Correlation

Index derived by three different ways: average weight output, weighted output and integration, respectively. The final classification results are shown in Table 4.18. According to the results, it is obvious that feature ordering based on correlation can always obtain a better results with ITID (ILIA2) than conventional methods. However, the classification error rates of those using ITID (ILIA1) are not very good.

Table 4.13: Correlations of Features and Outputs (Thyroid)

Feature	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	1																				
2	0.0015	1																			
3	0.0206	-0.1012	1																		
4	-0.0173	0.0346	0.0036	1																	
5	-0.0679	-0.0281	0.0017	-0.0118	1																
6	0.0880	0.0018	-0.0402	0.0162	-0.0211	1															
7	-0.1208	-0.0798	0.0115	0.0492	0.0773	-0.0244	1														
8	-0.0284	-0.0357	0.0291	0.0084	-0.0123	0.0012	-0.0142	1													
9	0.0582	-0.0167	0.0652	-0.0139	0.0092	-0.0248	-0.0149	0.0052	1												
10	0.0466	-0.0441	0.0970	-0.0289	-0.0158	0.0258	-0.0215	-0.0202	0.0441	1											
11	-0.0409	-0.0642	-0.0244	-0.0077	0.1305	-0.0332	0.1250	0.0203	0.0642	0.0214	1										
12	-0.0320	-0.0102	-0.0011	-0.0077	-0.0072	-0.0139	-0.0083	-0.0081	-0.0085	-0.0009	0.0164	1									
13	-0.0489	0.0060	-0.0157	-0.0103	-0.0096	-0.0184	0.0145	-0.0108	-0.0113	-0.0236	-0.0233	-0.0063	1								
14	-0.0266	-0.0715	-0.0333	-0.0022	-0.0169	0.0134	0.1288	-0.0036	-0.0199	-0.0340	0.0553	-0.0111	0.0047	1							
15	-0.0269	0.0253	-0.0062	0.1482	-0.0017	-0.0034	-0.0020	-0.0020	-0.0021	-0.0043	-0.0042	-0.0011	-0.0015	-0.0027	1						
16	-0.1107	0.0916	-0.0730	-0.0255	-0.0237	-0.0389	-0.0167	-0.0266	-0.0280	-0.0155	-0.0577	0.0408	-0.0066	-0.0201	-0.0038	1					
17	-0.0544	-0.0327	0.0143	-0.0143	-0.0096	-0.0215	-0.0198	0.0293	-0.0011	0.0294	-0.0097	-0.0060	-0.0131	-0.0148	-0.0032	-0.0278	1				
18	-0.2364	-0.0641	0.0124	0.0034	0.0798	-0.0741	0.1862	-0.0307	-0.0015	-0.0500	0.1680	0.0107	0.0155	0.0948	-0.0162	0.0289	-0.1603	1			
19	-0.0382	-0.1647	0.2120	-0.0030	0.0314	-0.0355	0.1764	-0.0341	-0.0180	-0.0113	0.1324	-0.0108	-0.0175	0.0521	-0.0262	0.0029	-0.2623	0.5152	1		
20	-0.1718	-0.2154	0.0508	0.0041	0.0556	-0.0414	0.3379	0.0330	0.0071	0.0159	0.0839	0.0230	0.0456	0.0898	0.0068	-0.0202	0.0734	0.4013	0.4213	1	
21	0.0619	-0.0471	0.1816	-0.0034	-0.0059	-0.0178	-0.0131	-0.0400	-0.0200	-0.0219	0.0959	-0.0246	-0.0415	0.0037	-0.0293	0.0120	-0.2875	0.3174	0.7705	-0.1868	1
Output	0.0111	-0.0478	-0.0860	-0.0091	-0.0170	0.0075	-0.0332	-0.0218	0.0060	0.0827	-0.0065	-0.0012	-0.0251	0.0098	-0.0046	-0.0285	0.2938	-0.1573	-0.2285	0.0151	-0.2449
Ordering 1	14	7	5	16	12	17	8	11	19	6	18	21	10	15	20	9	1	4	3	13	2
ABS(Output1)	0.0148	0.0253	0.0117	0.0181	0.0169	0.0324	0.0195	0.0116	0.0092	0.0394	0.0188	0.0111	0.0148	0.0034	0.0027	0.0365	0.6071	0.2272	0.3456	0.0396	0.3685

ABS(Output2)	0.0170	0.0411	0.0861	0.0030	0.0118	0.0195	0.0279	0.0270	0.0030	0.0727	0.0001	0.0027	0.0211	0.0115	0.0038	0.0168	0.0904	0.0836	0.1159	0.0017	0.1247
ABS(Output3)	0.0052	0.0490	0.0781	0.0132	0.0198	0.0031	0.0346	0.0154	0.0080	0.0835	0.0112	0.0044	0.0262	0.0075	0.0048	0.0356	0.4353	0.2040	0.3010	0.0249	0.3220
Output AVG	0.0123	0.0385	0.0586	0.0115	0.0161	0.0183	0.0273	0.0180	0.0067	0.0652	0.0100	0.0060	0.0207	0.0075	0.0038	0.0296	0.3776	0.1716	0.2541	0.0220	0.2718
Ordering 2	15	7	6	16	14	12	9	13	19	5	17	20	11	18	21	8	1	4	3	10	2
Output Weight	0.00202	0.01601	0.02562	0.00428	0.00643	0.00157	0.01130	0.00531	0.00258	0.02728	0.00361	0.00148	0.00855	0.00254	0.00156	0.01155	0.14078	0.06618	0.09761	0.00803	0.10443
Ordering 3	18	7	6	14	12	19	9	13	16	5	15	21	10	17	20	8	1	4	3	11	2

Table 4.14: Output Weight (Thyroid)

Output	Number	Total Number of Training Patterns	Weight
Output 1	91	3600	0.025278
Output 2	181	3600	0.050278
Output 3	3328	3600	0.924444

Table 4.15: Correlation Index and Feature Ordering (Output Average Weight, Thyroid)

	$\sum_{j=1}^{n} \lvert Correlation(Input_i, Input_j) \rvert$	$\lvert Correlation(Input_i, Output) \rvert$	Feature	Correlation Index
1	1.3783	0.3776	17	0.091320
2	2.4268	0.2718	21	0.037333
3	0.6549	0.0652	10	0.033186
4	3.1643	0.2541	19	0.026767
5	2.6242	0.1716	18	0.021797
6	0.3698	0.0207	13	0.018659
7	1.0809	0.0586	3	0.018071
8	0.4150	0.0180	8	0.014458
9	0.6995	0.0296	16	0.014105
10	0.5625	0.0183	6	0.010844
11	1.1841	0.0385	2	0.010838
12	0.4228	0.0115	4	0.009067
13	0.6341	0.0161	5	0.008463
14	0.2499	0.0060	12	0.008003
15	1.4755	0.0273	7	0.006167
16	0.4399	0.0067	9	0.005077
17	0.3217	0.0038	15	0.003937
18	0.7091	0.0075	14	0.003526
19	2.3002	0.0220	20	0.003188
20	1.3091	0.0123	1	0.003132
21	1.1851	0.0100	11	0.002813

Table 4.16: Correlation Index and Feature Ordering (Weighted Output, Thyroid)

	$\sum_{j=1}^{n} \lvert Correlation(Input_i, Input_j) \rvert$	$\lvert Correlation(Input_i, Output) \rvert$	Feature	Correlation Index
1	1.3783	0.14078	17	0.034047
2	2.4268	0.10443	21	0.014344
3	0.6549	0.02728	10	0.013885
4	3.1643	0.09761	19	0.010282
5	2.6242	0.06618	18	0.008406
6	1.0809	0.02562	3	0.007901
7	0.3698	0.00855	13	0.007707
8	0.6995	0.01155	16	0.005504
9	1.1841	0.01601	2	0.004507
10	0.4150	0.00531	8	0.004265
11	0.6341	0.00643	5	0.003380
12	0.4228	0.00428	4	0.003374
13	1.4755	0.01130	7	0.002553
14	0.2499	0.00148	12	0.001974
15	0.4399	0.00258	9	0.001955
16	0.3217	0.00156	15	0.001616
17	0.7091	0.00254	14	0.001194
18	2.3002	0.00803	20	0.001164
19	1.1851	0.00361	11	0.001015
20	0.5625	0.00157	6	0.000930
21	1.3091	0.00202	1	0.000514

Table 4.17: Correlation Index and Feature Ordering (Output Integration, Thyroid)

	$\sum_{j=1}^{n} \lvert Correlation(Input_i, Input_j) \rvert$	$\lvert Correlation(Input_i, Output) \rvert$	Feature	Correlation Index
1	1.3783	0.2938	17	0.010151
2	0.6549	0.0827	10	0.006013
3	2.4268	0.2449	21	0.004805
4	1.0809	0.0860	3	0.003789
5	3.1643	0.2285	19	0.003439
6	0.3698	0.0251	13	0.003232
7	2.6242	0.1573	18	0.002854
8	0.4150	0.0218	8	0.002501
9	0.6995	0.0285	16	0.001940
10	1.1841	0.0478	2	0.001922
11	0.6341	0.0170	5	0.001277
12	1.4755	0.0332	7	0.001071
13	0.4228	0.0091	4	0.001025
14	0.3217	0.0046	15	0.000681
15	0.7091	0.0098	14	0.000658
16	0.4399	0.0060	9	0.000649
17	0.5625	0.0075	6	0.000635
18	1.3091	0.0111	1	0.000404
19	2.3002	0.0151	20	0.000313
20	1.1851	0.0065	11	0.000261
21	0.2499	0.0012	12	0.000229

Table 4.18: Classification Results of Correlation-based Feature Ordering (Thyroid)

	Approach	Feature Ordering	Classification Error (%)		
			ITID (ILIA1)	ITID (ILIA2)	AVG
1	Input-Output Correlation-based Output AVG	17-21-19-18-10-3-2-16-7-20-13-6-8-5-1-4-11-14-9-12-15	2.50000	1.68611	2.09306
2	Input-Output Correlation-based Output Weight	17-21-19-18-10-3-2-16-7-13-20-5-8-4-11-9-14-1-6-15-12	2.50833	1.68889	2.09861
3	Input-Output Correlation-based Output Integration	17-21-19-18-3-10-2-7-16-13-8-5-20-1-14-4-6-11-9-15-12	2.47778	1.74445	2.11111
4	Integrated Correlation-based Output AVG	17-21-10-19-18-13-3-8-16-6-2-4-5-12-7-9-15-14-20-1-11	2.20000	1.85833	2.02917
5	Integrated Correlation-based Output Weight	17-21-10-19-18-3-13-16-2-8-5-4-7-12-9-15-14-20-11-6-1	2.51667	1.72222	2.11944
6	Integrated Correlation-based Output Integration	17-10-21-3-19-13-18-8-16-2-5-7-4-15-14-9-6-1-20-11-12	2.28056	1.57500	1.92778
7	Conventional Method	No Feature Ordering	1.86389		

4.6 Summary

This chapter carried out the computing of feature ordering based on statistical correlations. Correlation has already been confirmed as a useful metric for feature discriminative ability measure in feature selection. In this study, it is employed for feature ordering. Two kinds of correlation metrics are employed for the feature ordering measurement: simple input-output correlation and integrated correlation. Specially, for multiple category classification problems, different classes usually have different numbers of patterns, the number of patterns belonging to different classes is also an element which may impact on the final results. According to the experimental results, statistical correlation is applicable for feature ordering. Most of the results derived by ITID (ILIA2) with feature ordering obtained by this correlation-base approaches are

better and more acceptable than conventional batch training method. Moreover, the process of correlation-based feature ordering is much faster than contribution-based feature ordering, because it is a filter feature ordering approach. However, some results of IAL are still worse than conventional method, which indicates that results derived by feature ordering based on correlation is unstable. Sometimes, it cannot produce lower classification results than conventional approaches. Generally, the correlation-based feature ordering is feasible for IAL but the results derived by such an ordering are not always stable. In the next chapter, some other feature ordering approaches will be studied for IAL.

Chapter 5

Feature Ordering based on Mutual Information

5.1 Mutual Information

Mutual Information (MI) is a measure of the independence of two variables in probability theory and information theory. Thus it can also be employed for the measurement of correlations between two variables. Formally, mutual information of two discrete variables *X* and *Y* can be calculated by

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \, log \left[\frac{p(x,y)}{p(x)p(y)} \right] \tag{5.1}$$

where $p(x,y)$ is the joint probability of X and Y, and $p(x)$ and $p(y)$ are the marginal probability of *X* and *Y* , respectively.

Intuitively, MI measures how much information that *X* and *Y* are sharing together. It is a metric for calculating how much knowledge one of these variables reduces uncertainty about the other. For example, if *X* and *Y* are independent, then *X* does not give any information about *Y* and vice versa. Therefore, their mutual information is zero. At the other extreme, if *X* and *Y* are identical, then all the knowledge conveyed by *X* is the same with *Y*. This simply means that knowing *X* is knowing *Y*, and vice versa.

5.2 Minimal-redundancy-maximal-relevance Criterion

In previous studies, Mutual Information has been found to be a useful measure for redundancy and relevance between input features and output attributes in pattern recognition and statistical machine learning. It has been confirmed that it is applicable for feature selection with a criterion called **Minimal-redundancy-maximal-relevance criterion (mRMR)** by Peng et al. [61].

General speaking, mRMR is a method for first-order incremental feature selection. In mRMR criterion, features which have both minimum redundancy for input features and maximum relevancy for output classes should be selected. Therefore, this method is based on two important metrics. One is mutual information between an output and each input, which is used to measure relevancy, and the other is the mutual information between two inputs, which is used to calculate redundancy between these inputs.

More specifically, let S denotes the subset of selected features, and Ω is a pool of all input features, the minimum redundancy can be computed by

$$\min_{S\subset\Omega} \frac{1}{|S|^2} \sum_{i,j\in S} I(f_i; f_j) \tag{5.2}$$

where $I(f_i, f_j)$ is mutual information between f_i and f_j, and $|S|$ is the number of input feature of S. On the other hand, mutual information $I(c, f_i)$ is usually employed to calculate discrimination ability from feature f_i to class $c = \{c_1, \dots, c_k\}$. Therefore, the maximum relevancy can be calculated by

$$\max_{S\subset\Omega} \frac{1}{|S|} \sum_{i\in S} I(c; f_i) \tag{5.3}$$

Combining Eq.(5.2) with Eq.(5.3), mRMR feature selection criterion can be obtained as below, either in quotient form:

$$\max_{S\subset\Omega} \left\{ \sum_{i\in S} I(c; f_i) \Big/ \left[\frac{1}{|S|} \sum_{i,j\in S} I(f_i; f_j) \right] \right\} \tag{5.4}$$

or in a different form:

$$\max_{S\subset\Omega} \left\{ \sum_{i\in S} I(c; f_i) - \left[\frac{1}{|S|} \sum_{i,j\in S} I(f_i; f_j) \right] \right\} \tag{5.5}$$

In the solutions of mRMR, features are incrementally added into the selected feature subset. According to such a process, the sequence of the incremental addition can be regarded as an order of discrimination ability of features. Thus, feature ordering can also be calculated by Eq.(5.4) or Eq.(5.5), assuming all features have been put into the selected subset by mRMR.

5.3 Feature Ordering based on mRMR

Based on mRMR's properties, it is manifested that such a criterion is applicable in IAL feature ordering. Feature ordering is unique to data preparation work of IAL. Compared with conventional approaches where input features are trained in one batch, features will be gradually imported into pattern recognition one after another in IAL. In this process, the method to derive an order for training is very important. Hence, feature ordering is seldom used in conventional pattern recognition techniques, which is indispensable in IAL. Moreover, the computing procedure of feature ordering is different from that of feature selection methods, where feature selection discards some features from the original feature set, while feature ordering merely puts down all features in a given order which may be different from the original sequence. Our previous studies have illustrated the feasibility of mRMR feature ordering approach [62]. Based on it, this mutual information feature ordering approach is extended to high dimensional classification problems in this section.

Due to the fact that the calculation of feature ordering in the previous studies is based on wrapper contribution-based feature ordering approaches, which are time-consuming compared with filters, using filter methods is able to bring benefits for feature ordering of IAL in the time-saving aspect. Apart from saving time in preprocessing, there are some other advantages in the calculation of feature ordering using filters. For example, feature reduction and feature ordering can be applied simultaneously, because the former severely relies on discrimination ability while the calculation of discrimination ability is the key in the latter.

Although these two mRMR methods are different, both are applicable in the calculation of input feature ordering, and each ordering can be employed in training. In ordered IAL, feature ordering can be calculated by mRMR. In the process, the discrimination ability of each feature is

calculated on the basis of training dataset, and the ordering ranking results of this mRMR-based approach are sorted in a descending order according to mRMR's properties. This is similar to the contribution-based discrimination ability which is also placed in a descending order. Previous studies have confirmed that the descending order can produce better results than the other orders such as random and ascending ordering [2]. In the next phase, the formal machine learning process starts. Patterns are randomly divided into three different datasets: training, validation and testing [44], and patterns in both validation and testing set will be formatted with the same ordering in training set. The process in this learning step is based on these datasets, and feature ordering for importing features is a foundation for pattern recognition in each round.

5.4 Experiments

The proposed ordered IAL method using mRMR and ITID were tested on five classification benchmarks which are Diabetes, Cancer, Glass, Thyroid and Semeion cases. In these experiments, all the patterns were randomly divided into three groups: training set (50%), validation set (25%) and testing set (25%). The training data were firstly used to rank feature ordering based on mRMR in the first place as a preprocessing task while ITID was employed for classification according to this feature ordering in the following step.

In the experiments, both of mRMR methods are applicable for feature ordering, thus two streams of experiments based on mRMR are implemented as well. Tables 5.1~5.5 and Figures 5.1~5.5 present the details of experiments using different datasets. According to these tables, it is obvious that ITID (ILIA2) with feature ordering derived by both mRMR approaches, no matter by difference or quotient, can exhibit better performance than conventional one-batch training approach in many cases. Most of these experimental results have lower error rates than those derived by conventional approach. However, although ITID (ILIA1) also obtained better classification results in Diabetes, Glass and Thyroid, it failed in Cancer and Semeion. Therefore, it seems that ILIA2 is more stable than ILIA1, which has already been mentioned in the previous research on ILIA [21]. Generally, as a well-known method of feature selection, mRMR is also available for feature ordering. It can be used with IAL approaches and reduce classification error

rates by ITID (ILIA2) algorithm.

1. Diabetes

Table 5.1: Experimental Results based on mRMR Feature Ordering (Diabetes)

	Approach	Feature Ordering	Classification Error (%)		
			ITID (ILIA1)	ITID (ILIA2)	AVG
1	mRMR-Difference	2-6-1-7-3-8-4-5	22.86459	23.56770	23.21615
2	mRMR-Quotient	2-6-1-7-3-8-5-4	22.96876	23.82813	23.39845
3	Conventional Method	No Feature Ordering	23.93229		

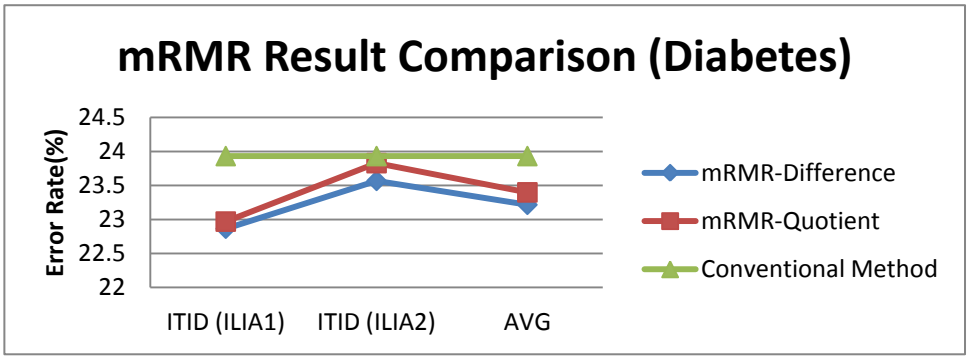


Figure 5.1: Comparison of Results based on mRMR Feature Ordering (Diabetes)

2. Cancer

Table 5.2: Experimental Results based on mRMR Feature Ordering (Cancer)

	Approach	Feature Ordering	Classification Error (%)		
			ITID (ILIA1)	ITID (ILIA2)	AVG
1	mRMR-Difference	2-6-1-7-3-8-5-4-9	2.29885	1.58046	1.93966
2	mRMR-Quotient	2-6-1-7-8-3-5-4-9	2.29885	1.81035	2.05460
3	Conventional Method	No Feature Ordering	1.86782		

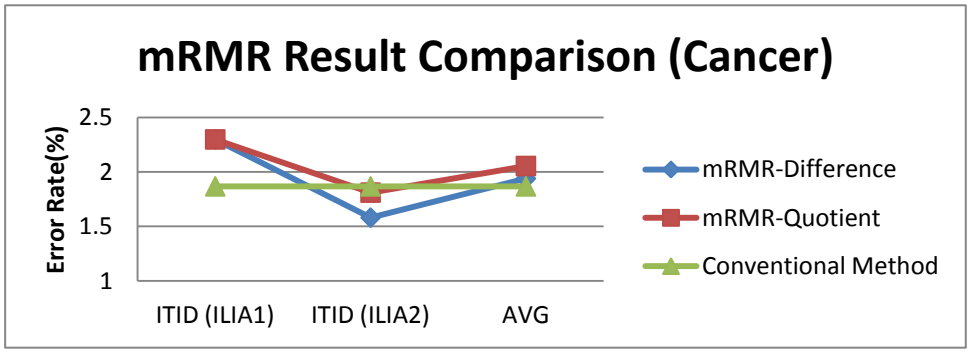


Figure 5.2: Comparison of Results based on mRMR Feature Ordering (Cancer)

3. Glass

Table 5.3: Experimental Results based on mRMR Feature Ordering (Glass)

	Approach	Feature Ordering	Classification Error (%)		
			ITID (ILIA1)	ITID (ILIA2)	AVG
1	mRMR-Difference	3-2-4-5-7-9-8-6-1	39.05663	35.09436	37.07550
2	mRMR-Quotient	3-5-2-8-9-4-7-6-1	35.28304	31.50946	33.39625
3	Conventional Method	No Feature Ordering	41.22641		

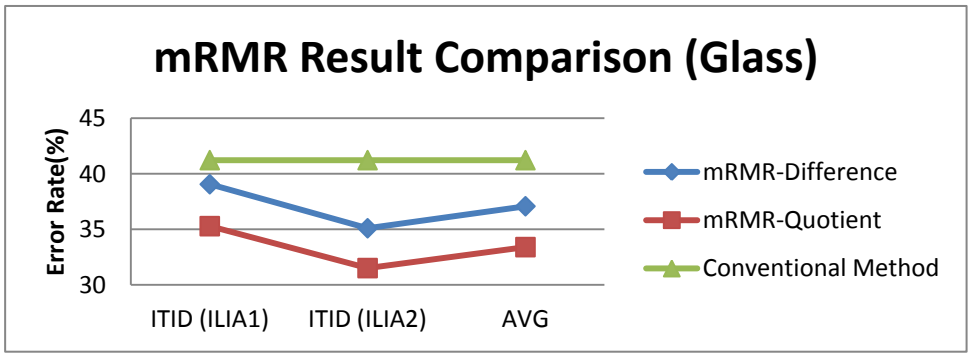


Figure 5.3: Comparison of Results based on mRMR Feature Ordering (Glass)

4. Thyroid

Table 5.4: Experimental Results based on mRMR Feature Ordering (Thyroid)

	Approach	Feature Ordering	Classification Error (%)		
			ITID (ILIA1)	ITID (ILIA2)	AVG
1	mRMR-Difference	3-7-17-10-6-8-13-16-4-5-12-21 -18-19-2-20-15-9-14-11-1	1.61944	1.29722	1.45833
2	mRMR-Quotient	3-10-16-7-6-17-2-8-13-5-1-4-11 -12-14-9-21-15-18-19-20	1.62500	1.42222	1.52361
3	Conventional Method	No Feature Ordering	1.86389		

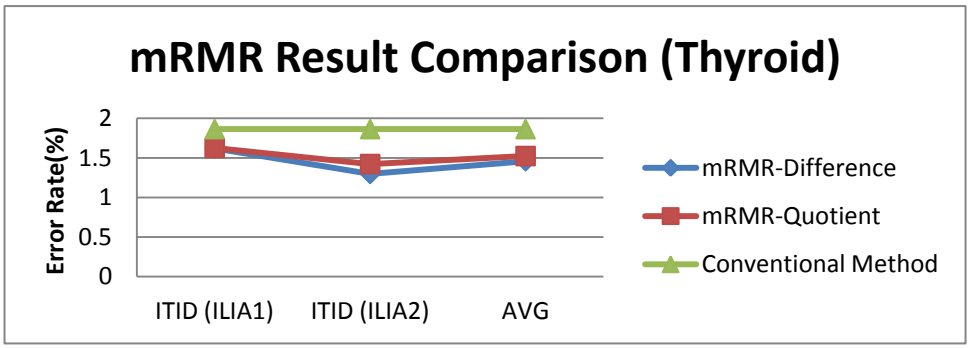


Figure 5.4: Comparison of Results based on mRMR Feature Ordering (Thyroid)

5. Semeion

Table 5.5: Experimental Results based on mRMR Feature Ordering (Semeion)

	Approach	Feature Ordering	Classification Error (%)		
			ITID (ILIA1)	ITID (ILIA2)	AVG
1	mRMR-Difference	162-79-82-146-178-111-8-130-194-63-231-98-161-95-62-145-163-9-112-177-66-229-47-191-128-77-11-129-179-96-230-105-93-114-193-127-10-232-83-7-76-195-228-99-78-143-147-113-234-64-3-50-174-84-152-80-103-233-92-136-81-210-175-46-4-67-245-91-61-108-51-104-159-97-121-246-150-1-167-75-6-135-107-12-207-100-65-119-109-5-151-189-106-18-48-227-153-164-2-94-188-120-68-144-168-16-102-247-90-166-192-101-235-149-211-137-17-238-60-115-183-89-180-190-158-45-165-35-122-131-182-31-237-36-123-59-134-37-173-209-138-69-226-19-13-124-118-23-157-236-244-52-187-208-154-184-85-248-22-74-148-155-206-169-205-139-255-181-222-212-34-254-172-15-110-21-53-142-204-196-30-160-88-239-249-140-20-58-49-141-32-253-33-176-170-14-156-125-221-240-171-199-223-38-24-185-225-220-126-116-186-198-54-73-203-197-86-70-87-256-117-44-243-57-25-133-55-252-241-224-71-213-41-250-56-219-242-39-27-251-132-29-26-200-218-43-40-216-72-217-42-28-215-202-214-201	17.86432	12.88945	15.37688
2	mRMR-Quotient	162-63-82-233-238-135-228-45-103-8-143-195-100-130-1-152-11-77-95-163-174-188-50-194	17.72613	13.34172	15.53392

		-231-105-68-3-146-183-16-79-245-108-191-178-128-62-165-98-155-235-75-122-229-111-84-9-51-147-101-179-47-157-119-18-145-255-150-12-66-177-168-76-232-112-93-23-136-5-99-246-161-64-129-237-46-210-10-159-189-230-167-164-96-78-102-234-89-193-83-2-61-81-7-127-180-175-121-109-153-114-226-80-91-17-187-149-35-182-211-144-85-158-253-107-247-67-104-139-4-151-48-131-65-192-113-94-205-227-240-137-124-184-60-37-173-92-148-196-13-118-69-166-207-115-106-244-97-120-141-254-190-154-212-22-236-59-6-134-90-123-172-52-169-138-33-248-209-36-204-19-74-208-181-31-110-24-140-88-239-220-142-156-170-206-21-58-249-256-49-199-53-160-30-116-34-15-186-125-222-86-171-198-225-185-20-197-221-32-38-87-126-117-176-73-241-203-14-25-252-54-223-71-213-44-243-133-57-70-27-224-55-219-29-250-41-242-132-56-216-26-251-218-39-28-200-43-217-72-215-40-214-42-202-201			
3	Conventional Method	No Feature Ordering	13.32915		

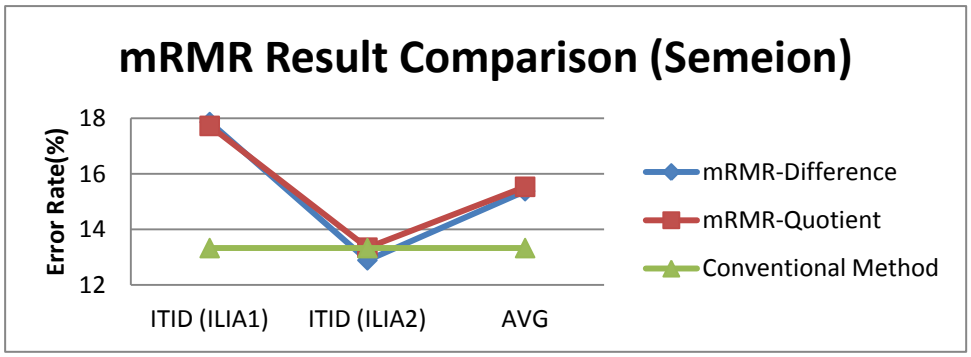


Figure 5.5: Comparison of Results based on mRMR Feature Ordering (Semeion)

5.5 Summary

In this chapter, mRMR criterion was employed to rank feature discrimination ability and produce feature orderings for IAL. mRMR was previously used as a feature selection approach. Due to the fact that it can rank features with minimal redundancy and maximal relevance, feature's

discrimination ability also can be calculated by this criterion very well. Most of the experimental results indicated that IAL with feature ordering derived from mRMR can exhibit acceptable performance when ITID (ILIA2) was employed. Moreover, in this chapter, feature ordering with ITID for IAL was also successfully applied in a high dimensional problem, which showed the feasibility and efficiency of mRMR in the field of feature ordering. Nonetheless, in ITID (ILIA1), mRMR cannot always obtain lower error rates than conventional method, especially in Cancer and Semeion datasets. Furthermore, in Semeion, even ITID (ILIA2) was employed, result of mRMR-Quotient still did not overcome that of conventional method. Such a phenomenon denotes that mRMR is not very stable for IAL feature ordering. Therefore, it is still necessary to seek an optimum feature ordering approach for IAL. In Chapter 6, feature ordering will be studied based on linear discriminant.

Chapter 6

Feature Ordering based on Linear Discriminant

6.1 Fisher's Linear Discriminant

Fisher's Linear Discriminant (FLD) can be regarded as a linear statistical classifier [63]. It provides simple ways to estimate the accuracy of classification problems. It firstly assumes that the datasets used in FLD are Gaussian conditional density models, where the data have normal distributed classes or equal class covariance. The Fisher criterion aims to search a direction where the distance between different classes is the farthest and the distance of each pattern within every class is the closest. Thus, in this direction, the ratio of distance between-classes and within-classes is the largest compared with other directions. Such a direction often leads to the simplest classification. Mathematically, FLD in two-category classification is

$$J(w) = \frac{(\tilde{\mu}_2 - \tilde{\mu}_1)^2}{s_1^2 + s_2^2} \tag{6.1}$$

where $\tilde{\mu}_1$ and $\tilde{\mu}_2$ are two means of projected classes, and s_1 and s_2 are within-class variances. The objective of FLD is to search the matrix w for maximum $J(w)$. Larger the $J(w)$, easier the classification. FLD with such an objective can be treated as traditional linear discriminant. It has a capacity to create new features to classify original datasets. However, such a process cannot calculate the discrimination ability for each feature. An approach is to make the directions stable, namely make the current existing feature's direction as $J(w)$'s direction. Thus feature's discrimination ability can be calculated by Eq.(6.1), and feature ordering derived by FLD can be obtained with the descending order of FLD score by $J(w)$.

However, due to the fact that $J(w)$ in Eq.(6.1) is impacted by two classes, it will be difficult to calculate $J(w)$ for patterns belonging to three or more classes at the same time. Therefore, Eq.(6.1) should be modified to address this issue; otherwise it will lose its applicability in reality.

6.2 Multivariate Fisher's Linear Discriminant

Multivariate Fisher's Linear Discriminant (MFLD) is a variant of FLD [64]. It can cope with multivariate output classification problems. When MFLD is carried out, Eq.(6.1) must be extended from two-dimension to multiple dimensions. A feasible way for such an extension is to convert multivariate output classification problems into some univariate output classification problems, where the outputs of each univariate output classification problem only have two different types: "belong-to" and "not-belong-to". After such a conversion, an x-category classification problem will become x 2-category classification problem. Each of these 2-category classification problems has a property in the output: "one-against-all". These "one-against-all" problems then will be integrated into one in the solution of the whole problem.

Fisher Score (FS) [65] is one representative extended from FLD to MFLD. This metric individually inspect FLD of each feature with the original direction. All features are calculated one by one in "one-against-all" style at the first place, and then, they are integrated by the sum of each individual result of each class. Moreover, FS also combined with feature weighting. Here is the formula of the FS of feature i:

$$FS(f_i) = \frac{\sum_{j=1}^{n} \omega_j \left(\tilde{\mu}_{i,c\in j} - \tilde{\mu}_i\right)^2}{\sum_{j=1}^{n} \omega_j s_{i,c\in j}^2} \tag{6.2}$$

where ω_j is the weight of j-th class. Obviously, feature ordering derived by Eq.(6.2) must be sorted with descending order. FS individually valuates features, therefore it cannot eliminate feature redundancy [65].

6.3 Single Discriminability

In IAL, each feature's discrimination ability can be estimated in this feature's one-dimensional space. Features can be ordered by the ranking value of the feature discrimination ability. For two-class classification problems (c_2), based on Eq.(6.1), the discrimination ability of feature f_i can be given by

$$D(f_i) = \frac{(\mu_2 - \mu_1)^2}{s_1^2 + s_2^2} \tag{6.3}$$

where μ_1 and μ_2 are the means of two classes, and s_1 and s_2 are within-class variances.

However, Eq.(6.3) is too simple to cope with multi-category classifications, because the between-class scatter is difficult to describe merely by distance between patterns. Here, the difference between the centres of these multiple classes should be replaced by standard deviations of centres and standard deviations of patterns, so that the influence brought by classes whose mean is not the smallest or the largest of all the means of classes can be measured.

Definition 6.1: ***Single Discriminability*** **(*SD*)** is a ratio between a feature by the standard deviation of all class centres and the sum of standard deviations of all patterns in each class.

SD for both two-category and *n*-category classification problems can be integrated as

$$SD(f_i) = \frac{std\left[\left(\mu_{f_i,j}\right)_{j=1}^{j=n}\right]}{\sum_{j=1}^{j=n} std\,(f_i)_j} \tag{6.4}$$

where *n* is the total number of classes, and *std* denotes the standard deviations, one for all patterns belonging to c_j in feature *i*, and the other for the vector consisting of the means of all classes in feature *i*. Let **x** be the vector for standard deviation calculation, the standard deviation of **x** is:

$$std(\mathbf{x}) = \sqrt{\frac{\sum_{k=1}^{k=r}(x_k - \mu)^2}{r-1}} \tag{6.5}$$

where the vector $\mathrm{x} = \{\mathrm{x_k}\}_{\mathrm{k=1}}^{\mathrm{k=r}}$, x_k is the value of k^{th} pattern, and *r* is the total number of patterns. Obviously, in Eq.(6.5), the part of $(x_k - \mu)$ is a distance between k^{th} pattern and its mean. Thus, let *dist* replace this part, then Eq.(6.5) can be re-written as:

$$std(\mathbf{x}) = \sqrt{\frac{\sum_{k=1}^{k=r} dist_{x_k,\mu}{}^2}{r-1}} \tag{6.6}$$

where $\mathrm{dist}_{x_k,\mu}$ denotes the distance of k^{th} pattern in **x** and its mean μ .

Obviously, according to Eq.(6.6), the essence of SD indicates two kinds of distance, one is the distance between classes, and the other is the distance within each class. These are similar to FLD, where the further the distance between different classes and the nearer the distance between each pattern and its class centre, the easier these classes can be distinguished. Here, easier means the probability of correct prediction in pattern recognition is higher. For example, Figure 6.1 shows a normalized dataset which has two classes. The class centres are *a* and *b*, and *x* is one of its features. According to *a* and *b*, the feature space of *x* can be divided into three parts: [0, *a*], (*a*, *b*], and (*b*, 1]. Taking a random number produced by a classifier as a segmentation point, the probability of a random number in [0, *a*] is $P_1=a/1=a$; that in (*a*, *b*] is $P_2=(b-a)/1=b-a$; and that in (*b*, 1] is $P_3=(1-b)/1=1-b$. If we want to make the classification easier, we must enhance P_2 and reduce P_1 and P_3. Therefore, for P_1, *a* should be reduced; for P_2, *b* should be increased and *a* should be reduced; and for P_3, *b* should be increased. As a result of reducing *a* and increasing *b*, the distance between *a* and *b* will be larger.

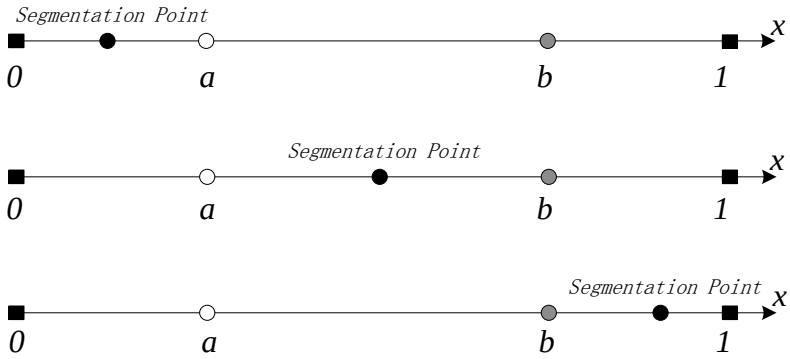


Figure 6.1: Segmentations on *x*.

This is similar to FLD, where the greater the standard deviation of *a* and *b*, the easier the classification. In the example shown in Figure 6.1, if μ is the mean of *a* and *b*, the standard deviation of *a* and *b* is

$$std = \sqrt{\frac{(a-\mu)^2 + (b-\mu)^2}{2}} \tag{6.7}$$

Substituting for $\mu = \frac{a+b}{2}$ and simplifying:

$$std = \sqrt{\frac{\left(a - \frac{a+b}{2}\right)^2 + \left(b - \frac{a+b}{2}\right)^2}{2}}$$

$$= \sqrt{\frac{\left(\frac{a}{2} - \frac{b}{2}\right)^2 + \left(\frac{b}{2} - \frac{a}{2}\right)^2}{2}}$$

$$= \frac{|a-b|}{2}.$$

Since $b \geq a$, we get

$$std = \frac{b-a}{2} \tag{6.8}$$

Therefore, according to Eq.(6.8), if the distance between *a* and *b* is greater, the standard deviation of *a* and *b* will also be greater. Namely, greater distance indicates easier classification, and greater standard deviation will also imply easier classification.

If there are three or more classes in one feature space, Eq.(6.5) also works very well. Assuming that there are two pattern sets, one is $\boldsymbol{a}=\{a_1, a_2, \ldots, a_i\}$, and the other is $\boldsymbol{b}=\{b_1, b_2, \ldots, b_j\}$, $a \cap b = \emptyset$, then relations of the mean and standard deviation are:

$$\mu_{\boldsymbol{a}\cup\boldsymbol{b}} = \frac{1}{i+j}(i \cdot \mu_{\boldsymbol{a}} + j \cdot \mu_{\boldsymbol{b}}) \tag{6.9}$$

$$std_{\boldsymbol{a}\cup\boldsymbol{b}} = \sqrt{\frac{1}{i+j-1}[(i-1) \cdot std_{\boldsymbol{a}}^2 + i \cdot \mu_{\boldsymbol{a}}^2 + (j-1) \cdot std_{\boldsymbol{b}}^2 + j \cdot \mu_{\boldsymbol{b}}^2 - (i+j)\mu_{\boldsymbol{a}\cup\boldsymbol{b}}^2]}\ . \tag{6.10}$$

According to Eq.(6.9) and Eq.(6.10), when *n*=3, if the two pattern sets are $\{a_1, a_2\}$, and $\{a_3\}$, then

$$\mu_{\{a_1,a_2\}\cup\{a_3\}} = \frac{1}{3}(2\mu_{\{a_1,a_2\}} + \mu_{\{a_3\}}) \tag{6.11}$$

$$std_{\{a_1,a_2\}\cup\{a_3\}} = \sqrt{\frac{1}{2}\left[std_{\{a_1,a_2\}}^2 + 2\mu_{\{a_1,a_2\}}^2 + 0 \cdot std_{\{a_3\}}^2 + \mu_{\{a_3\}}^2 - 3\mu_{\{a_1,a_2\}\cup\{a_3\}}^2\right]}$$

$$= \sqrt{\frac{1}{2}\left[std_{\{a_1,a_2\}}^2 + 2\mu_{\{a_1,a_2\}}^2 + \mu_{\{a_3\}}^2 - 3\left(\frac{1}{3}\left(2\mu_{\{a_1,a_2\}} + \mu_{\{a_3\}}\right)\right)^2\right]}$$

$$= \sqrt{\frac{1}{2}\left[std_{\{a_1,a_2\}}^2 + 2\mu_{\{a_1,a_2\}}^2 + \mu_{\{a_3\}}^2 - \frac{1}{3}\left(4\mu_{\{a_1,a_2\}}^2 + 4\mu_{\{a_1,a_2\}}\mu_{\{a_3\}} + \mu_{\{a_3\}}^2\right)\right]}$$

$$= \sqrt{\frac{1}{2}std_{\{a_1,a_2\}}^2 + \frac{1}{3}\left(\mu_{\{a_1,a_2\}} - \mu_{\{a_3\}}\right)^2}. \tag{6.12}$$

Obviously, if $std_{\{a_1,a_2\}}$ and $\mu_{\{a_1,a_2\}} - \mu_{\{a_3\}}$ both increase, then $std_{\{a_1,a_2\}\cup\{a_3\}}$ will increase correspondingly. The key elements here are the distance between a_1 and a_2, and the distance between the centres of $\{a_1, a_2\}$, and a_3. The further the distance, the lower the classification error rate is.

Similarly, when $n=4$, if the two pattern sets are $\{a_1, a_2, a_3\}$, and $\{a_4\}$, then

$$\mu_{\{a_1,a_2,a_3\}\cup\{a_4\}} = \frac{1}{4}(3\mu_{\{a_1,a_2,a_3\}} + \mu_{\{a_4\}}) \tag{6.13}$$

$$std_{\{a_1,a_2,a_3\}\cup\{a_4\}}$$

$$= \sqrt{\frac{1}{3}\left[2std^2_{\{a_1,a_2,a_3\}} + 3\mu^2_{\{a_1,a_2,a_3\}} + 0\cdot std^2_{\{a_4\}} + \mu^2_{\{a_4\}} - 4\mu^2_{\{a_1,a_2,a_3\}\cup\{a_4\}}\right]}$$

$$= \sqrt{\frac{1}{3}\left[2std^2_{\{a_1,a_2,a_3\}} + 3\mu^2_{\{a_1,a_2,a_3\}} + \mu^2_{\{a_4\}} - 4\left(\frac{1}{4}(3\mu_{\{a_1,a_2,a_3\}} + \mu_{\{a_4\}})\right)^2\right]}$$

$$= \sqrt{\frac{1}{3}\left[2std^2_{\{a_1,a_2,a_3\}} + 3\mu^2_{\{a_1,a_2,a_3\}} + \mu^2_{\{a_4\}} - \frac{1}{4}\left(9\mu^2_{\{a_1,a_2,a_3\}} + 6\mu_{\{a_1,a_2,a_3\}}\mu_{\{a_4\}} + \mu^2_{\{a_4\}}\right)\right]}$$

$$= \sqrt{\frac{2}{3}std^2_{\{a_1,a_2,a_3\}} + \frac{1}{4}\left(\mu_{\{a_1,a_2,a_3\}} - \mu_{\{a_4\}}\right)^2} \tag{6.14}$$

Similar to the situation of n=3, the further the distance between a_1, a_2, a_3, and a_4, the better the pattern recognition performance.

We assume that there is an n-category classification problem, then it needs n-1 segmentation points. When $n=k$,

$$std_{\{a_1,a_2,\ldots,a_{k-1}\}\cup\{a_k\}} = \sqrt{\frac{k-2}{k-1}std^2_{\{a_1,a_2,\ldots,a_{k-1}\}} + \frac{1}{k}\left(\mu_{\{a_1,a_2,\ldots,a_{k-1}\}} - \mu_{\{a_k\}}\right)^2} \tag{6.15}$$

Eq.(6.15) shows that the standard deviation depends on the distances between patterns. Then when $n=k+1$, the two pattern sets are $\{a_1, a_2, \ldots, a_k\}$, and $\{a_{k+1}\}$.

$$\mu_{\{a_1,a_2,\ldots,a_k\}\cup\{a_{k+1}\}} = \frac{1}{k+1}(k\mu_{\{a_1,a_2,\ldots,a_k\}} + \mu_{\{a_{k+1}\}}) \tag{6.16}$$

$$std_{\{a_1,a_2,\ldots,a_k\}\cup\{a_{k+1}\}}$$

$$= \sqrt{\frac{1}{k}\left[(k-1)std^2_{\{a_1,a_2,\ldots,a_k\}} + k\mu^2_{\{a_1,a_2,\ldots,a_k\}} + 0\cdot std^2_{\{a_{k+1}\}} + \mu^2_{\{a_{k+1}\}} - (k+1)\mu^2_{\{a_1,a_2,\ldots,a_k\}\cup\{a_{k+1}\}}\right]}$$

$$= \sqrt{\frac{1}{k}\left[(k-1)std^2_{\{a_1,a_2,\ldots,a_k\}} + k\mu^2_{\{a_1,a_2,\ldots,a_k\}} + \mu^2_{\{a_{k+1}\}} - (k+1)\left(\frac{1}{k+1}(k\mu_{\{a_1,a_2,\ldots,a_k\}} + \mu_{\{a_{k+1}\}})\right)^2\right]}$$

$$= \sqrt{\frac{\mathrm{k}-1}{\mathrm{k}} std^2_{\{a_1,a_2,\ldots,a_k\}} + \frac{1}{k+1}\left(\mu_{\{a_1,a_2,\ldots,a_k\}} - \mu_{\{a_k\}}\right)^2} \quad (6.17)$$

Therefore, when n=k+1, the $\mathrm{std}_{\{a_1,a_2,\ldots,a_k\}\cup\{a_{k+1}\}}$ also depends on the distance between patterns. More specifically, $\mathrm{std}_{\{a_1,a_2,\ldots,a_k\}\cup\{a_{k+1}\}}$ is decided by $std_{\{a_1,a_2,\ldots,a_k\}}$ and the distance between centres of $\{a_1, a_2, \ldots, a_k\}$, and $\{a_k\}$. However, $std_{\{a_1,a_2,\ldots,a_k\}}$ and $\mu_{\{a_1,a_2,\ldots,a_k\}}$ depend on the value of $std_{\{a_1,a_2,\ldots,a_{k-1}\}}$ and $\mu_{\{a_1,a_2,\ldots,a_{k-1}\}}$. Eventually, they depend on the distance between every two samples.

Therefore, based on these properties of standard deviation and mean, the calculation of SD which is presented in Eq. (6.4) is obviously applicable in IAL feature discrimination ability computing. During the process, standard deviations between multiple classes and within classes can be calculated. Generally, the greater the "between" standard deviation means the greater the total distance, then the lower the probability of errors are. Absolutely, in the mean while, the "within" standard deviation which is influenced by the pattern distribution of each class, can reflect the tightness of each class centre.

Generally speaking, to effectively distinguish patterns from each other, it is necessary to ensure that the total distance between pattern centres should be the greatest. Therefore, SD, which is inspired from FLD can deduce feature discrimination ability well. Thus, SD is suitable to address the problems in multi-category classification. However, similar to FS, SD also computes features one by one, therefore, it also cannot handle feature redundancy during the feature ordering calculations.

6.4 Accumulative Discriminability

Although previous research shows that filter approaches are more effective and simpler for feature ordering, they still have several shortcomings, which may bring bias into the final result. This problem is that, in almost all previous studies on feature ordering, features are sorted into a certain order by their ranking of individual discrimination ability or contribution, while the combined accumulative effects derived from two or more continuously imported features are ignored. This is inappropriate. Here a new metric for filter feature ordering with accumulative

calculations on feature discrimination ability is inspired from FLD and SD.

With an increasing number of new features in IAL, the number of dimensions in feature space is also growing. Growing feature space has been regarded as one of the unique characteristics of IAL. In such a space, the standard deviation, which is the core of evolving linear discriminant, should be generalized from that in one dimension. More specifically, the standard deviation in one dimensional space is based on the distance between each pattern and their mean of the same class. This distance should be extended to a higher dimensional space, when the feature space is growing. If ‖D‖ is the Euclidean norm of *d*-dimensional feature space, Eq.(6.6) can be given in a high-dimensional style by:

$$std(\mathbf{x}) = \sqrt{\frac{\sum_{k=1}^{k=r} \left\| D_{x_k,\tilde{\mu}} \right\|^2}{r-1}} \tag{6.18}$$

where $\tilde{\mu}$ is the centroid of **x**, and

$$\left\| D_{x_k,\tilde{\mu}} \right\| = \sqrt{\sum_{i=1}^{d} \left(x_{k,i} - \mu_i\right)^2} \tag{6.19}$$

Here *d* is the total number of features imported so far. Therefore, to calculate the standard deviation of *r* patterns in two dimensions, Eq.(6.18) can be written as

$$std(\mathbf{x}) = \sqrt{\frac{\sum_{k=1}^{k=r} \left[\left(x_{k,1} - \mu_1\right)^2 + \left(x_{k,2} - \mu_2\right)^2 \right]}{r-1}}, \mathbf{x} = \left\{x_{k,d}\right\}_{k=1,d=1}^{k=r,d=2} \in \mathbb{R}_{feature}^{r\times 2} \tag{6.20}$$

and for a tri-dimensional space, the equation is

$$std(\mathbf{x}) = \sqrt{\frac{\sum_{k=1}^{k=r} \left[\left(x_{k,1} - \mu_1\right)^2 + \left(x_{k,2} - \mu_2\right)^2 + \left(x_{k,3} - \mu_3\right)^2 \right]}{r-1}}, \quad \mathbf{x} = \left\{x_{k,d}\right\}_{k=1,d=1}^{k=r,d=3} \in \mathbb{R}_{feature}^{r\times 3} \tag{6.21}$$

Accordingly, multidimensional standard deviation of *r* patterns in an *m*-dimensional space is

$$std(\mathbf{x}) = \sqrt{\frac{\sum_{k=1}^{k=r} \sum_{i=1}^{i=m} \left(x_{k,i} - \mu_i\right)^2}{r-1}}, \mathbf{x} = \left\{x_{k,d}\right\}_{k=1,d=1}^{k=r,d=m} \in \mathbb{R}_{feature}^{r\times m} \tag{6.22}$$

Based on Eq.(6.22), when some new features are incrementally introduced into the system, formula in Eq.(6.4), the standard deviation-based linear discriminant of IAL in one feature dimension should be upgraded to fit in this gradually increasing dimensional space, because Eq.(6.4) pays little attention to gradually importing new features.

Definition 6.2: ***Accumulative Discriminability*** **(*AD*)** is the ratio in *d*-feature space between the multidimensional standard deviation of all class centres and the sum of all multidimensional standard deviations of all patterns in each class.

If $\{f_1, f_2, \dots, f_m\}$ is the pool of input features, $\mathrm{f} = \{\mathrm{f}_{\mathrm{k,d}}\}_{\mathrm{k=1,d=1}}^{\mathrm{k=r,d=m}} \in \mathrm{R}_{\mathrm{feature}}^{\mathrm{r\times m}}$, when the d^{th} ($1 \le d \le m$) feature is imported, AD is

$$AD(f_1, f_2, \dots, f_d) = \frac{std\left[\left(\tilde{\boldsymbol{\mu}}_j\right)_{j=1}^{j=n}\right]}{\sum_{j=1}^{j=n} std\left[(f_i)_{i=1}^{i=d}\right]_j}, (1 \le d \le m) \tag{6.23}$$

where $\tilde{\mu}_j$ is the centroid of vector $(f_1, f_2, \dots, f_d)$ with patterns belonging to *j*. Therefore, the results of Eq.(6.23) are calculated on the run when new features are gradually imported into training. To obtain better classification results, it is necessary to ensure the result of Eq.(6.23) is the maximum in every step of feature importing. Obviously, different from SD, AD has the capacity to show the redundancy between features. When a new feature *d* is imported, the difference between $AD(f_1, f_2, \dots, f_d)$ and $AD(f_1, f_2, \dots, f_{d-1})$ indicates this kind of redundancy. Intuitively, the less change exists between ADs, the more redundancy between features.

6.5 Maximum Mean Discriminative Criterion

To obtain the best accurate classification result in IAL, it is necessary to ensure that the datasets have the greatest discrimination ability in every step when a new feature is imported into the predictive system and the feature dimension is increased from *d* to *d+1*. Therefore, the ratio in Eq.(6.23) should be the largest all the time, as only in this way can it guarantee that different classes can be separated in the easiest way. Therefore, the criterion for optimum classification results, as well as the greatest discrimination ability, is to produce an optimum feature ordering which contains the greatest discrimination ability in each round of feature importing. Obviously, after all features are imported, the optimum feature ordering will have the largest sum or mean of feature discrimination ability calculated in each step of the process. Hence such a criterion for obtaining the optimum feature ordering can be given with maximum discrimination ability mean by

$$\max \frac{1}{d}\sum_{d=1}^{d} AD(\mathbf{f}_{1:d}), (1 \le d \le m) \tag{6.24}$$

where $\mathbf{f}_{1:d}$ is the feature subset of $\{f_1, f_2, \dots, f_m\}$ during the feature importing process.

Usually, features with greater SD calculated by Eq.(6.4) may not always have greater AD, because Eq.(6.23) has an additional value produced by the Euclidean distance in high dimensional space. Such a value is disproportionate with the value in Eq.(6.4). Thus, features which have greater SD may also have weaker AD in IAL feature importing. Therefore, for IAL classification, Eq.(6.24) will likely produce more accurate results than Eq.(6.4).

In addition, it is obvious that if all patterns and all features are imported into this computation, the final AD in the highest dimensionality will be the same, no matter the process of importing feature ordering. However, because the process of importing random feature ordering is different in each round, the AD calculated in each step of every round is also different. Therefore, for an input matrix, although the values of the final AD with different feature ordering are equal, the means derived from AD obtained in each step is different. The mean with a greater value indicates that the corresponding feature ordering has greater discrimination ability than the others. Hence, the **Maximum Mean Discriminative Criterion (MMDC)** has the capacity to select optimum feature ordering.

Therefore, before MMDC is employed, a pool filled with different feature ordering should be firstly prepared. This feature ordering can be randomly initialized. The greater the size of the pool and the more means we compare, the higher probability the system has for obtaining optimum feature ordering. However, it will be difficult to compare all combinations of different feature ordering because of the dimensional curses. A feasible way for this is to compute it using heuristic or evolutional approaches, based on the initialized random feature ordering pool.

6.6 GA-Based Optimum Feature Ordering

Final classification results can be estimated by discrimination ability. For SD and other methods with stable metrics, feature ordering can be directly obtained by individual ranking. However, AD is more complex because MMDC is set upon the foundation of comparison with different AD

means, which makes it compulsory to carry out the computation many times. Before AD is computed, a random feature ordering should already exist. Fortunately, the computation of AD is not so complex. When some heuristic or evolutional approaches are employed, the algorithm for AD is fairly efficient.

Apart from random feature ordering generation which is applicable but not as efficient, a more rational approach is to employ intelligent machine learning algorithms, such as GA or NN. For example, an evolutionary algorithm can be employed to obtain the maximum mean of features' discrimination ability for optimal feature ordering according to AD and MMDC. Here the algorithm used to obtain optimum feature ordering, based on AD, was GA. GA has a range of genetic operators such as crossover and mutation. As every feature should appear at least once in the ordering, only crossover is needed in AD feature ordering computation. In addition, the fitness function is the maximum mean calculated in the comparison, according to MMDC with Eq.(6.24).

The **GA for Optimum Feature Ordering (GAOFO)** based on MMDC and AD is not very complicated. It can be carried out as follows:

Step 1: The algorithm randomly produce a set of seeds in different feature orderings, and each seeds evolutes in parallel.

Step 2: More than two places in the ordering of each seed are exchanged to generate a new ordering. Such an exchange is similar to crossover in GA;

Step 3: According to MMDC with Eq.(6.24), if the seed gets the greatest mean of AD in its evolutional history, it will be recorded;

Step 4: To repeat Step 2 and 3 with a number of epochs of evolution. During the process, the recorded feature ordering of each seed will be compared with one another, and the seed with the greatest mean will be selected as potential global optimization. Naturally, the number of potential global optimum feature orderings equals to the number of seeds produced in Step 1.

Step5: If all or most of the potential global optimum feature orderings are the same, the ordering can be concluded as the real global optimum feature ordering, otherwise, they should be repeated in Step 4 with a longer epochs.

Because of the large feature number and limitations of the evolutionary generation number, it is difficult to obtain the global optimum solution. In this case, the feature ordering will be close to the real optimum solution. Sometimes, due to the time limitation, it may be impossible to wait for the global optimum feature ordering, then an approaching optimum feature ordering can be obtained by this method.

Obviously, the ordering transformed data based on the global optimal feature ordering can be directly employed in training, validation and testing. The speed of producing such a transformed dataset depends on the feature dimensional numbers, the number of evolving generations and the number of random seeds.

Compared with other approaches to obtain optimum feature ordering such as wrappers, filter approaches are able to save more time for data preparation. For high-dimensional classification problems, if the GA with MMDC for AD optimum feature ordering is employed, it only spends a very short time in computation. However, for the same problem with the consideration of accumulative influence in dynamic increasing feature space, if using wrapper contribution-based approaches to calculate feature ordering, it will be a far more time consuming task.

6.7 Max. AD Mean Feature Ordering based on SD

The GA-based optimum feature ordering derived by MMDC and AD is able to seek the optimum feature ordering for pattern classification problems. The essence of GA-based feature ordering approach is to search the optimum result within a feature space. Searching in a very large space of possible hypotheses to determine one that best fits the observed data is one of the main tasks of machine learning [66]. During the evolutionary process like GA, the number of epochs of searching is relevant to the number of input features, thus it is difficult to determine how many iteration rounds should be set for the evolution. Although such an approach can provide us an approaching optimum feature ordering for some situation, we still wonder whether there are some other approaches for optimum feature ordering based on MMDC.

Fortunately, another heuristic feature ordering approach called **Maximum AD Mean**

Feature Ordering (MAMFO) based on SD is developed. This approach is not only directly based on MMDC and AD, but also on the basis of SD. The pseudo-code of this SD-based feature ordering approach is shown in Figure 6.2. In this algorithm, SD of each input feature should be computed in the first place. Then, the feature with the greatest SD will be selected as the first feature into the ordering process. From the second feature to the last one, AD and MMDC are employed. Each of the remaining features will be solely imported into the calculation of AD mean one by one. It is combined with all previously selected features. They are calculated together. If the newly combined feature subset has the maximum mean, the new imported feature will be selected as the second feature because of the greatest discrimination ability. For the subsequent features, calculation of AD mean is done in the same manner. Finally, a new feature ordering will be obtained. Obviously, the performance of such a feature ordering approach will be the same as GA-based global optimum feature ordering.

Maximum AD Mean Feature Ordering based on SD

```
1.  Initialize Ordering[]: feature ordering, m: feature number, n: class number ;
2.  Get SD from each input feature;
3.  // the first feature
4.  For i=1:m
5.      if f=argmax(SD(f_i)) then Ordering[1]=f;
6.  End
7.  //the rest features
8.  For i=2:m
9.      For j=1:m
10.         if f_j has been selected, then continue
11.         else if f_j=argmax(AD(Ordering[i-1],f_j)) then Ordering[i]=f_j;
12.     End;
13. End;
14. Print Ordering[];
```

Figure 6.2: The Pseudo-code of Maximum AD Mean Feature Ordering based on SD

6.8 Experiments

Experiments on feature ordering derived by FS, SD, and AD are implemented in this section. Seeing from the time cost shown in Table 6.1, GA-based feature ordering approach is much more

time-consuming than the SD-based feature ordering. These results are derived on PC with the CPU Intel Core i7-2640 @ 2.8GHz and the size of memory is 8GB.

Table 6.1: Time Cost Comparison between GA and SD -based Feature Ordering

	GA AD Feature Ordering	Max. AD Mean Feature Ordering based on SD
Diabetes	More than 10 min (10 seeds)	0.7100 s
Cancer	More than 10 min (10 seeds)	1.0700 s
Glass	More than 10 min (10 seeds)	0.4500 s
Thyroid	More than 2 days (10 seeds)	165.9300 s
Semeion	Too long	59785.0000 s

In the following subsections, the performance exhibited in UCI machine learning repository is showed. According to these results, it is manifested that AD is much more stable than SD and Fisher Score. Except Semeion, all the results derived by AD, no matter it is from ITID (ILIA1) or from ITID (ILIA2), exhibit a better performance than conventional one-batch training method. In Semeion, although AD with ITID (ILIA1) cannot produce better results than conventional method, AD with ITID (ILIA2) can also exhibit an acceptable performance. Comparing with AD, the performances of SD and FS are weaker. For example, in Semeion, their performances are similar to AD. Furthermore, in Thyroid, both of these two approaches got higher classification error rates than conventional method based on ITID (ILIA1). Figures 6.4, 6.5, 6.6, and 6.7 demonstrate the GA evolution of Diabetes, Cancer, Glass and Thyroid. It is obvious that all the results of feature ordering converge into one. Moreover, all of these feature orderings begin with the feature with the largest SD. In addition, both GAOFO and SD-based MAMFO produced the same feature ordering. Such a phenomenon not only confirmed that it is applicable to employ GAOFO and MAMFO for the optimum feature ordering seeking, but also showed the uniqueness of the optimum feature ordering according to MMDC.

1. Diabetes

Diabetes is a univariate output classification problem. Table 6.2 shows the value of SD, AD and FS of the features in Diabetes dataset. According to the values, SD and FS obtains the same feature ordering, while the feature ordering derived by AD is different from the former two. Obviously, the final classification results derived by SD and FS are identical, while that of AD

has a slight difference. In Table 6.3, all the classification error rates obtained by the feature ordering derived by SD, AD and FS are much lower than those derived by one-batch-training conventional methods. Moreover, AD achieves the lowest classification error in ITID (ILIA1) approach, and the other two IAL methods obtain the lowest error rate in ITID (ILIA2) approach. However, in average, AD obtains the lowest classification error rate.

Table 6.2: Discriminabilities and Fisher Scores of Diabetes

Ordering	Single Discriminability		Accumulative Discriminability		Fisher Score	
	Discriminability	Feature Index	Discriminability	Feature Index	Score	Feature Index
1	0.3694	2	0.3694	2	0.4449	2
2	0.2630	6	0.3317	6	0.2240	6
3	0.1844	8	0.2751	7	0.1279	8
4	0.1567	7	0.2468	8	0.0679	7
5	0.1437	1	0.2268	5	0.0634	1
6	0.1047	4	0.2093	4	0.0342	4
7	0.0960	5	0.1953	1	0.0286	5
8	0.0509	3	0.1828	3	0.0084	3

Table 6.3: Results derived by Linear Discriminant Feature Ordering(Diabetes)

	Approach	Feature Ordering	Classification Error (%)		
			ITID (ILIA1)	ITID (ILIA2)	AVG
1	SD	2-6-8-7-1-4-5-3	21.84896	22.39583	22.12240
2	AD	2-6-7-8-5-4-1-3	21.61458	22.60416	22.10937
3	FS	2-6-8-7-1-4-5-3	21.84896	22.39583	22.12240
4	Conventional Method	No Feature Ordering	23.93229		

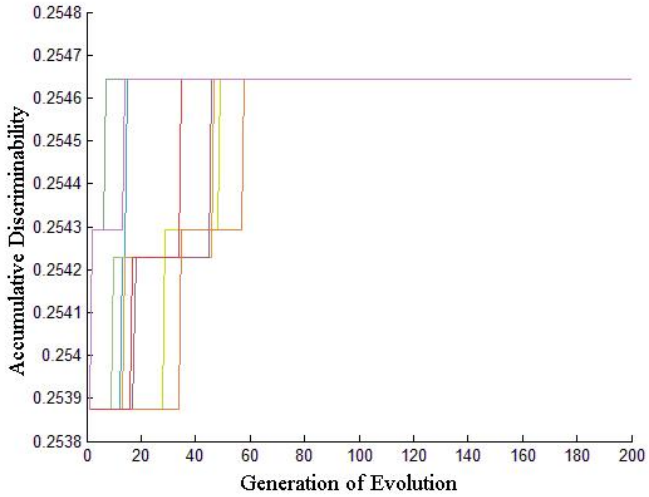


Figure 6.3: GA Evolution for optimum feature ordering (Diabetes)

2. Cancer

Cancer is a univariate output classification problem. Table 6.4 shows the value of SD, AD and FS of the features in Cancer dataset. According to the values, Single SD and FS obtains the same feature ordering again, while the feature ordering derived by AD is different from the former two. Obviously, the final classification results derived by SD and FS are identical, while that of AD is different. In Table 6.5, all the classification errors obtained by the feature ordering derived by SD, AD and FS are much lower than those derived by one-batch-training conventional methods. Moreover, in Cancer, AD achieves the lowest classification error both in ITID (ILIA1) approach and in ITID (ILIA2) approach. It is manifested that AD obtains the lowest classification error rate in average.

Table 6.4: Discriminabilities and Fisher Scores of Cancer

Ordering	Single Discriminability		Accumulative Discriminability		Fisher Score	
	Discriminability	Feature Index	Discriminability	Feature Index	Score	Feature Index
1	0.9888	3	0.9888	3	1.95676	3
2	0.9566	2	0.9713	2	1.81004	2
3	0.8725	6	0.9283	6	1.52486	6
4	0.7793	7	0.9025	7	1.22296	7
5	0.7039	1	0.8617	5	0.98401	1
6	0.6895	8	0.8314	1	0.93583	8
7	0.6604	4	0.8064	8	0.85083	4
8	0.6260	5	0.7869	4	0.78583	5
9	0.3534	9	0.7605	9	0.22363	9

Table 6.5: Results derived by Linear Discriminant Feature Ordering(Cancer)

	Approach	Feature Ordering	Classification Error (%)		
			ITID (ILIA1)	ITID (ILIA2)	AVG
1	SD	3-2-6-7-1-8-4-5-9	1.69541	1.72414	1.70977
2	AD	3-2-6-7-5-1-8-4-9	1.55173	1.60920	1.58046
3	FS	3-2-6-7-1-8-4-5-9	1.69541	1.72414	1.70977
4	Conventional Method	No Feature Ordering	1.86782		

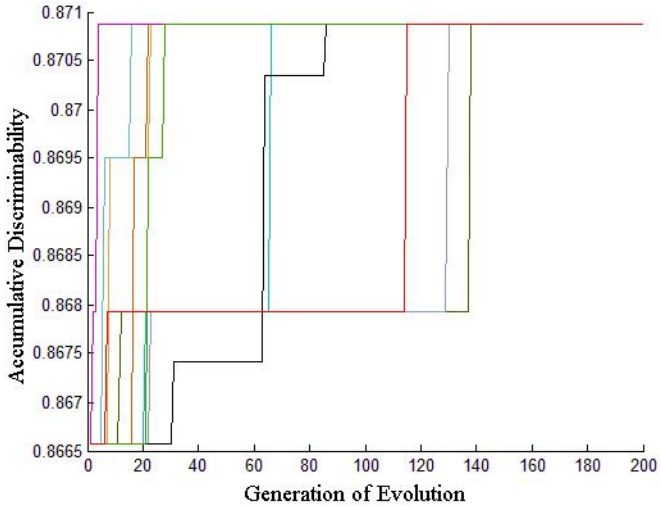


Figure 6.4: GA Evolution for optimum feature ordering (Cancer)

3. Glass

Glass is a multivariate output classification problem. Table 6.6 shows the value of SD, AD and FS of the features in Glass dataset. Three different feature orderings are obtained respectively. Table 6.7 demonstrates all the classification error rates obtained by the feature ordering derived by SD, AD and FS. Obviously, although all of them are much lower than the results derived by one-batch-training conventional methods, the performance of FS is not as good as two Discriminability approaches. Moreover, AD achieves the lowest classification error in ITID (ILIA1) approach, while SD obtains the lowest error rate in ITID (ILIA2) approach. However, in average, AD obtains the lowest classification error rate again.

Table 6.6: Discriminabilities and Fisher Scores of Glass

Ordering	Single Discriminability		Accumulative Discriminability		Fisher Score	
	Discriminability	Feature Index	Discriminability	Feature Index	Score	Feature Index
1	0.3226	3	0.3226	3	1.7710	3
2	0.2605	8	0.2896	8	0.9954	4
3	0.1716	4	0.2523	2	0.9130	8
4	0.1566	2	0.2331	4	0.5688	2
5	0.1514	6	0.2156	6	0.2690	6
6	0.0976	5	0.2018	7	0.1789	5
7	0.0802	9	0.1904	1	0.1119	1
8	0.0764	1	0.1773	5	0.0845	9
9	0.0542	7	0.1661	9	0.0401	7

Table 6.7: Results derived by Linear Discriminant Feature Ordering(Glass)

	Approach	Feature Ordering	Classification Error (%)		
			ITID (ILIA1)	ITID (ILIA2)	AVG
1	SD	3-8-4-2-6-5-9-1-7	34.81133	28.96228	31.88681
2	AD	3-8-2-4-6-7-1-5-9	34.33964	29.24530	31.79247
3	FS	3-4-8-2-6-5-1-9-7	39.62261	35.56606	37.59434
4	Conventional Method	No Feature Ordering	41.22641		

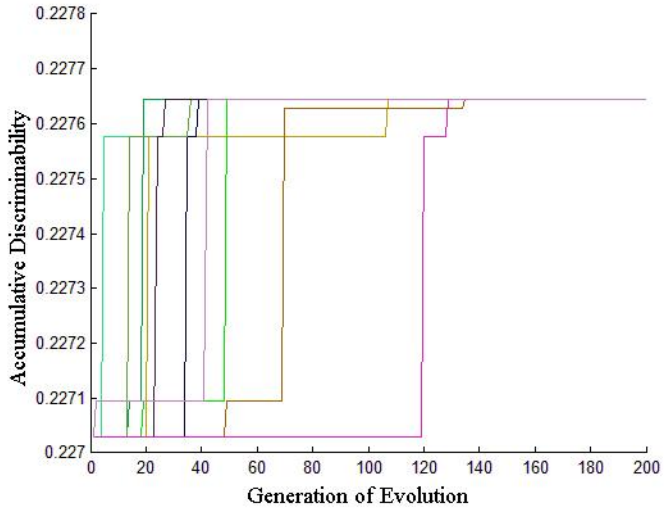


Figure 6.5: GA Evolution for optimum feature ordering (Glass)

4. Thyroid

Thyroid is a multivariate output classification problem. Table 6.8 shows the value of SD, AD and FS of the features in Thyroid dataset. Three different feature orderings are obtained respectively.

Table 6.9 demonstrates all the classification error rates obtained by the feature ordering derived by SD, AD and FS. Nevertheless, not all of these results can beat the results derived by one-batch-training conventional methods. The performance of SD and FS is not as good as AD approach, where the ITID (ILIA1) classification error rates by SD and FS are quite higher than conventional batch training. Only AD can achieve a lower classification error in both ITID (ILIA1) and ITID (ILIA2) approaches than the conventional batch training approach. Obviously, in average, AD obtains the lowest classification error rate.

Table 6.8: Discriminabilities and Fisher Scores of Thyroid

Ordering	Single Discriminability		Accumulative Discriminability		Fisher Score	
	Discriminability	Feature Index	Discriminability	Feature Index	Score	Feature Index
1	0.5890	21	0.5890	21	0.61639	17
2	0.5272	19	0.5724	18	0.18344	21
3	0.3816	17	0.5493	19	0.15739	19
4	0.2883	18	0.5334	15	0.06392	18
5	0.1067	3	0.4753	20	0.00769	3
6	0.0727	7	0.3676	17	0.00711	10
7	0.0672	6	0.2911	13	0.00242	2
8	0.0648	16	0.2416	7	0.00167	16
9	0.0551	13	0.2034	12	0.00158	20
10	0.0487	20	0.1771	5	0.00139	6
11	0.0478	10	0.1532	4	0.00120	7
12	0.0420	8	0.1297	8	0.00084	8
13	0.0373	2	0.1111	3	0.00069	13
14	0.0321	4	0.1012	9	0.00049	1
15	0.0316	5	0.0933	16	0.00044	5
16	0.0278	1	0.0874	6	0.00035	11
17	0.0247	11	0.0815	14	0.00034	4
18	0.0209	12	0.0759	1	0.00014	14
19	0.0120	14	0.0699	11	0.00013	12
20	0.0100	15	0.0649	10	0.00010	9
21	0.0095	9	0.0573	2	0.00002	15

Table 6.9: Results derived by Linear Discriminant Feature Ordering(Thyroid)

	Approach	Feature Ordering	Classification Error (%)		
			ITID (ILIA1)	ITID (ILIA2)	AVG
1	SD	21-19-17-18-3-7-6-16-13-20-10 -8-2-4-5-1-11-12-14-15-9	1.92778	1.52500	1.72639
2	AD	21-18-19-15-20-17-13-7-12-5-4 -8-3-9-16-6-14-1-11-10-2	1.52500	1.21667	1.37083
3	FS	17-21-19-18-3-10-2-16-20-6-7- 8-13-1-5-11-4-14-12-9-15	2.52500	1.77222	2.14861
4	Conventional Method	No Feature Ordering	1.86389		

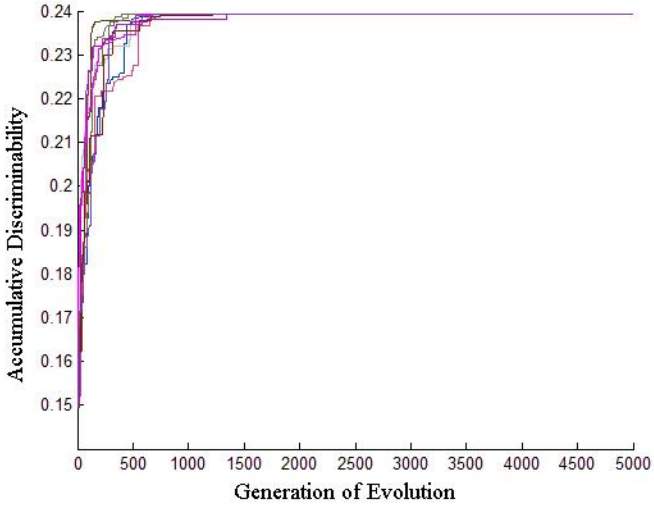


Figure 6.6: GA Evolution for optimum feature ordering (Thyroid)

5. Semeion

Semeion is a multivariate output classification problem. Table 6.10 shows the value of SD, AD and FS of the features in Semeion dataset. Three different feature orderings are obtained respectively. Table 6.11 demonstrates all the classification error rates obtained by the feature ordering derived by SD, AD and FS. Nevertheless, all the results of ITID (ILIA1) cannot obtain better results than one-batch-training conventional methods. The classification error rates derived from SD, AD and FS are quite higher than conventional batch training. However, all of the classification results derived by ITID (ILIA2) with SD, AD and FS are better than that of conventional methods. Moreover, ITID (ILIA2) with the preprocessing using FS produces the lowest error rate, and ITID (ILIA2) with AD gets the lowest error rate in average.

Table 6.10: Discriminabilities and Fisher Scores of Semeion

Ordering	Single Discriminability		Accumulative Discriminability		Fisher Score	
	Discriminability	Feature Index	Discriminability	Feature Index	Score	Feature Index
1	0.14062	112	0.14062	112	1.27515	162
2	0.13314	162	0.12917	96	1.12783	112
3	0.12853	96	0.12235	162	1.07967	146
4	0.12233	128	0.11749	178	1.01445	178
5	0.11448	146	0.11507	146	0.98727	96
6	0.11360	178	0.11238	128	0.82843	111
7	0.10574	111	0.11020	111	0.81290	145
8	0.10272	79	0.10800	95	0.80294	130
9	0.10262	95	0.10594	79	0.78989	161
10	0.10242	161	0.10415	161	0.77289	128
11	0.09879	145	0.10286	145	0.73106	79
12	0.09655	1	0.10169	177	0.72687	177
13	0.09542	130	0.10049	130	0.71563	95
14	0.09494	177	0.09938	256	0.69904	194
15	0.09316	80	0.09831	80	0.63922	127
16	0.08930	194	0.09729	127	0.59078	82
17	0.08792	63	0.09640	194	0.58035	63
18	0.08767	127	0.09531	63	0.56589	129
19	0.08611	82	0.09429	1	0.56325	98
20	0.08272	98	0.09323	82	0.52041	9
21	0.08255	129	0.09225	129	0.50587	47
22	0.08002	113	0.09136	98	0.50146	66
23	0.07869	163	0.09050	66	0.48465	114
24	0.07843	66	0.08967	113	0.48198	113
25	0.07809	114	0.08890	163	0.47737	80
26	0.07717	47	0.08819	9	0.47721	8
27	0.07681	9	0.08750	47	0.47670	163
28	0.07572	64	0.08683	114	0.47210	193
29	0.07456	62	0.08619	193	0.44787	10
30	0.07421	93	0.08560	64	0.43936	230
31	0.07402	193	0.08503	8	0.43904	229
32	0.07384	8	0.08448	81	0.43886	231
33	0.07204	77	0.08393	179	0.42817	93
34	0.07193	81	0.08339	93	0.42034	11
35	0.07189	179	0.08288	97	0.41921	179
36	0.07144	78	0.08239	65	0.40611	232
37	0.07142	10	0.08193	144	0.40089	77
38	0.07044	231	0.08149	10	0.39915	143
39	0.07033	2	0.08106	231	0.39882	62

40	0.07009	230	0.08065	230	0.39817	195
41	0.06972	229	0.08026	229	0.39127	64
42	0.06949	11	0.07988	2	0.38924	1
43	0.06916	97	0.07952	11	0.38761	228
44	0.06907	143	0.07916	77	0.38533	81
45	0.06889	3	0.07882	195	0.38246	7
46	0.06842	17	0.07850	62	0.37839	147
47	0.06794	83	0.07819	143	0.37653	83
48	0.06770	195	0.07788	3	0.36746	97
49	0.06712	232	0.07758	232	0.36532	233
50	0.06710	65	0.07729	17	0.36516	78
51	0.06693	144	0.07700	147	0.36508	99
52	0.06680	147	0.07672	78	0.36048	210
53	0.06667	99	0.07644	83	0.35478	3
54	0.06624	50	0.07616	7	0.35467	105
55	0.06587	7	0.07590	228	0.34505	65
56	0.06551	228	0.07563	50	0.34257	191
57	0.06526	105	0.07538	99	0.34118	92
58	0.06461	76	0.07512	233	0.33905	234
59	0.06442	92	0.07485	255	0.33805	50
60	0.06414	191	0.07459	92	0.32984	67
61	0.06355	233	0.07434	210	0.32470	76
62	0.06350	67	0.07410	4	0.32281	4
63	0.06311	210	0.07386	105	0.32226	175
64	0.06263	4	0.07363	67	0.31890	103
65	0.06159	84	0.07341	191	0.31829	144
66	0.06148	234	0.07317	76	0.31385	2
67	0.06090	152	0.07295	48	0.31090	46
68	0.06088	103	0.07273	234	0.30897	152
69	0.06065	175	0.07250	51	0.30824	108
70	0.06058	46	0.07227	109	0.30624	174
71	0.06046	51	0.07206	152	0.30616	159
72	0.06038	108	0.07184	175	0.30525	84
73	0.06008	48	0.07164	84	0.29662	109
74	0.05977	91	0.07143	103	0.29637	51
75	0.05958	159	0.07123	108	0.29349	91
76	0.05948	174	0.07104	46	0.29326	107
77	0.05904	109	0.07085	240	0.29308	48
78	0.05898	94	0.07066	192	0.28807	136
79	0.05896	61	0.07046	94	0.28638	6
80	0.05885	18	0.07027	159	0.28621	12
81	0.05850	107	0.07008	91	0.28356	104
82	0.05839	192	0.06990	174	0.27857	246

83	0.05831	136	0.06971	18	0.27583	192
84	0.05823	167	0.06953	107	0.27179	135
85	0.05800	104	0.06934	254	0.27156	5
86	0.05717	75	0.06915	167	0.26772	121
87	0.05706	6	0.06897	151	0.26715	17
88	0.05680	151	0.06879	136	0.26658	207
89	0.05646	5	0.06862	104	0.26451	94
90	0.05645	245	0.06845	16	0.26441	106
91	0.05630	12	0.06828	6	0.26429	151
92	0.05621	135	0.06811	12	0.26222	227
93	0.05614	246	0.06795	75	0.26178	16
94	0.05612	207	0.06779	5	0.26095	75
95	0.05609	121	0.06763	135	0.25922	245
96	0.05578	150	0.06747	188	0.25620	102
97	0.05567	106	0.06731	150	0.25329	100
98	0.05557	168	0.06716	207	0.25317	188
99	0.05548	16	0.06701	246	0.25307	120
100	0.05482	188	0.06686	183	0.25266	211
101	0.05464	153	0.06671	168	0.25227	61
102	0.05451	166	0.06657	166	0.25203	189
103	0.05437	120	0.06642	106	0.25079	150
104	0.05432	100	0.06628	61	0.25072	119
105	0.05425	119	0.06614	121	0.25071	247
106	0.05415	68	0.06600	68	0.24699	167
107	0.05410	183	0.06586	102	0.24417	18
108	0.05401	189	0.06572	149	0.24169	153
109	0.05395	227	0.06559	182	0.24035	68
110	0.05368	102	0.06545	100	0.23814	31
111	0.05362	90	0.06532	189	0.23700	90
112	0.05338	164	0.06519	245	0.23632	115
113	0.05307	149	0.06506	227	0.23377	235
114	0.05286	247	0.06493	119	0.23369	166
115	0.05276	211	0.06481	120	0.23333	149
116	0.05266	255	0.06469	153	0.23294	101
117	0.05222	115	0.06457	208	0.23092	168
118	0.05204	182	0.06445	164	0.22984	131
119	0.05185	60	0.06433	90	0.22245	137
120	0.05166	31	0.06421	211	0.22245	209
121	0.05163	101	0.06410	49	0.22244	164
122	0.05145	256	0.06398	247	0.21918	190
123	0.05129	235	0.06386	115	0.21850	89
124	0.05108	137	0.06374	184	0.21351	183
125	0.05087	89	0.06362	59	0.20788	165

126	0.05080	165	0.06351	31	0.20635	158
127	0.05035	190	0.06339	101	0.20580	60
128	0.05033	131	0.06328	165	0.20266	36
129	0.05032	209	0.06316	209	0.20255	182
130	0.05021	59	0.06305	110	0.20218	45
131	0.04996	208	0.06294	131	0.20188	59
132	0.04981	180	0.06283	235	0.20159	13
133	0.04960	35	0.06272	89	0.20012	122
134	0.04888	36	0.06261	60	0.19881	134
135	0.04875	158	0.06250	33	0.19786	180
136	0.04850	45	0.06239	180	0.19734	35
137	0.04839	122	0.06228	137	0.19553	37
138	0.04799	134	0.06217	190	0.19237	208
139	0.04786	184	0.06206	35	0.18852	226
140	0.04783	254	0.06195	36	0.18814	69
141	0.04767	37	0.06183	187	0.18587	110
142	0.04752	238	0.06172	241	0.18468	248
143	0.04711	110	0.06161	32	0.18420	118
144	0.04709	13	0.06150	45	0.18264	187
145	0.04705	69	0.06139	69	0.18189	124
146	0.04699	19	0.06129	37	0.18149	52
147	0.04677	52	0.06118	52	0.18119	184
148	0.04673	123	0.06107	74	0.18102	238
149	0.04668	187	0.06097	253	0.18087	173
150	0.04630	226	0.06086	134	0.18060	15
151	0.04625	118	0.06076	158	0.17884	123
152	0.04618	124	0.06065	181	0.17860	237
153	0.04605	240	0.06055	122	0.17762	19
154	0.04602	74	0.06044	148	0.17731	74
155	0.04597	173	0.06034	13	0.17485	236
156	0.04585	49	0.06024	169	0.17437	23
157	0.04582	237	0.06014	160	0.17374	85
158	0.04525	148	0.06004	19	0.17277	22
159	0.04521	248	0.05994	238	0.17240	148
160	0.04518	169	0.05984	124	0.17235	212
161	0.04515	138	0.05974	58	0.17059	32
162	0.04481	236	0.05964	118	0.16763	138
163	0.04476	85	0.05954	199	0.16688	154
164	0.04474	15	0.05944	176	0.16676	30
165	0.04465	32	0.05934	85	0.16615	206
166	0.04464	181	0.05925	34	0.16235	21
167	0.04420	34	0.05915	226	0.16092	254
168	0.04413	154	0.05905	185	0.16077	181

169	0.04412	206	0.05895	123	0.16068	204
170	0.04411	23	0.05886	173	0.15975	14
171	0.04410	22	0.05876	53	0.15928	222
172	0.04360	212	0.05867	15	0.15892	157
173	0.04356	160	0.05857	248	0.15889	169
174	0.04326	53	0.05848	125	0.15871	142
175	0.04321	222	0.05838	237	0.15694	34
176	0.04315	244	0.05829	154	0.15632	176
177	0.04313	176	0.05820	196	0.15593	155
178	0.04306	157	0.05810	212	0.15591	172
179	0.04292	204	0.05801	224	0.15588	53
180	0.04289	30	0.05792	236	0.15583	249
181	0.04279	21	0.05783	22	0.15555	49
182	0.04270	142	0.05774	204	0.15545	205
183	0.04264	205	0.05765	186	0.15528	125
184	0.04258	155	0.05756	239	0.15462	255
185	0.04257	172	0.05747	23	0.14986	196
186	0.04215	58	0.05738	170	0.14967	244
187	0.04192	14	0.05729	138	0.14683	160
188	0.04190	196	0.05720	198	0.14549	88
189	0.04190	125	0.05712	206	0.14544	20
190	0.04174	33	0.05703	172	0.14325	171
191	0.04151	249	0.05694	126	0.14293	141
192	0.04114	139	0.05686	88	0.14182	58
193	0.04111	88	0.05677	155	0.13917	139
194	0.04092	253	0.05669	171	0.13718	170
195	0.04087	20	0.05661	30	0.13710	240
196	0.04034	239	0.05652	142	0.13605	140
197	0.04033	171	0.05644	21	0.13546	185
198	0.04023	141	0.05636	244	0.13389	199
199	0.04011	170	0.05627	222	0.12903	126
200	0.04009	199	0.05619	157	0.12646	38
201	0.03991	185	0.05611	205	0.12616	253
202	0.03938	140	0.05602	14	0.12613	156
203	0.03882	223	0.05594	20	0.12584	223
204	0.03879	126	0.05586	249	0.12385	239
205	0.03806	38	0.05578	225	0.12327	225
206	0.03789	225	0.05569	54	0.12138	186
207	0.03788	156	0.05561	55	0.12099	24
208	0.03769	186	0.05553	70	0.11941	70
209	0.03741	221	0.05544	223	0.11921	221
210	0.03681	54	0.05536	56	0.11827	54
211	0.03672	198	0.05528	141	0.11772	203

212	0.03660	24	0.05520	73	0.11423	33
213	0.03643	203	0.05511	38	0.11330	198
214	0.03637	70	0.05503	139	0.10999	116
215	0.03537	73	0.05495	140	0.10634	55
216	0.03520	116	0.05486	203	0.10622	73
217	0.03459	86	0.05478	57	0.10578	86
218	0.03458	55	0.05470	242	0.10326	44
219	0.03452	220	0.05461	197	0.10268	87
220	0.03415	44	0.05453	200	0.10161	220
221	0.03413	87	0.05445	156	0.10132	41
222	0.03369	197	0.05436	71	0.10063	117
223	0.03353	117	0.05428	86	0.09512	197
224	0.03344	41	0.05420	252	0.09400	133
225	0.03243	133	0.05411	41	0.09350	71
226	0.03240	71	0.05403	116	0.09214	25
227	0.03223	57	0.05394	87	0.09208	256
228	0.03209	56	0.05386	44	0.09059	56
229	0.03189	224	0.05377	24	0.08707	57
230	0.03175	25	0.05369	221	0.08665	250
231	0.03100	250	0.05360	243	0.08352	213
232	0.03063	243	0.05351	117	0.08121	224
233	0.03055	241	0.05342	133	0.08053	243
234	0.03042	213	0.05332	220	0.07607	252
235	0.03010	252	0.05323	40	0.07529	39
236	0.02900	39	0.05314	251	0.07490	40
237	0.02871	40	0.05305	202	0.07431	132
238	0.02853	132	0.05296	42	0.07090	251
239	0.02841	251	0.05286	72	0.06923	219
240	0.02823	242	0.05277	250	0.06862	26
241	0.02794	200	0.05268	43	0.06815	200
242	0.02790	219	0.05259	213	0.06744	242
243	0.02730	26	0.05250	201	0.06704	27
244	0.02728	43	0.05241	39	0.06605	43
245	0.02718	42	0.05232	25	0.06566	42
246	0.02698	27	0.05222	132	0.06546	241
247	0.02681	218	0.05213	216	0.06530	218
248	0.02627	29	0.05203	217	0.06299	29
249	0.02615	217	0.05193	218	0.06211	217
250	0.02543	216	0.05184	215	0.05838	216
251	0.02481	72	0.05174	219	0.05393	72
252	0.02412	202	0.05164	214	0.05167	202
253	0.02357	28	0.05154	26	0.05097	28
254	0.02280	201	0.05143	27	0.04627	201

255	0.02272	215	0.05133	29	0.04605	215
256	0.02221	214	0.05122	28	0.04447	214

Table 6.11: Results derived by Linear Discriminant Feature Ordering(Semeion)

	Approach	**Feature Ordering**	**Classification Error (%)**		
			ITID (ILIA1)	**ITID (ILIA2)**	**AVG**
1	SD	112-162-96-128-146-178-111-79-95-161-145-1-130-177-80-194-63-127-82-98-129-113-163-66-114-47-9-64-62-93-193-8-77-81-179-78-10-231-2-230-229-11-97-143-3-17-83-195-232-65-144-147-99-50-7-228-105-76-92-191-233-67-210-4-84-234-152-103-175-46-51-108-48-91-159-174-109-94-61-18-107-192-136-167-104-75-6-151-5-245-12-135-246-207-121-150-106-168-16-188-153-166-120-100-119-68-183-189-227-102-90-164-149-247-211-255-115-182-60-31-101-256-235-137-89-165-190-131-209-59-208-180-35-36-158-45-122-134-184-254-37-238-110-13-69-19-52-123-187-226-118-124-240-74-173-49-237-148-248-169-138-236-85-15-32-181-34-154-206-23-22-212-160-53-222-244-176-157-204-30-21-142-205-155-172-58-14-196-125-33-249-139-88-253-20-239-171-141-170-199-185-140-223-126-38-225-156-186-221-54-198-24-203-70-73-116-86-55-220-44-87-197-117-41-133-71-57-56-224-25-250-243-241-213-252-39-40-132-251-242-200-219-26-43-42-27-218-29-217-216-72-202-28-201-215-214	18.85678	12.96483	15.91081
2	AD	112-96-162-178-146-128-111-95-79-161-145-177-130-256-80-127-194-63-1-82-129-98-66-113-163-9-47-114-193-64-8-81-179-93-97-65-144-10-231-230-229-2-11-77-195-62-143-3-232-17-147-78-83-7-228-50-99-233-255-92-210-4-105-67-191-76-48-234-51-109-152-175-84-103-108-46-240-192-94-159-91-174-18-107-254-167-151-136-104-16-6-12-75-5-135-188-150-207-246-183-168-166-106-61-121-68-102-149-182-100-189-245-227-119-120-153-208-164-90-211-49-247-115-184-59-31-101-165-209-110-131-235-89-60-	18.02764	12.83922	15.43343

		33-180-137-190-35-36-187-241-32-45-69-37-52-74-253-134-158-181-122-148-13-169-160-19-238-124-58-118-199-176-85-34-226-185-123-173-53-15-248-125-237-154-196-212-224-236-22-204-186-239-23-170-138-198-206-172-126-88-155-171-30-142-21-244-222-157-205-14-20-249-225-54-55-70-223-56-141-73-38-139-140-203-57-242-197-200-156-71-86-252-41-116-87-44-24-221-243-117-133-220-40-251-202-42-72-250-43-213-201-39-25-132-216-217-218-215-219-214-26-27-29-28			
3	FS	162-112-146-178-96-111-145-130-161-128-79-177-95-194-127-82-63-129-98-9-47-66-114-113-80-8-163-193-10-230-229-231-93-11-179-232-77-143-62-195-64-1-228-81-7-147-83-97-233-78-99-210-3-105-65-191-92-234-50-67-76-4-175-103-144-2-46-152-108-174-159-84-109-51-91-107-48-136-6-12-104-246-192-135-5-121-17-207-94-106-151-227-16-75-245-102-100-188-120-211-61-189-150-119-247-167-18-153-68-31-90-115-235-166-149-101-168-131-137-209-164-190-89-183-165-158-60-36-182-45-59-13-122-134-180-35-37-208-226-69-110-248-118-187-124-52-184-238-173-15-123-237-19-74-236-23-85-22-148-212-32-138-154-30-206-21-254-181-204-14-222-157-169-142-34-176-155-172-53-249-49-205-125-255-196-244-160-88-20-171-141-58-139-170-240-140-185-199-126-38-253-156-223-239-225-186-24-70-221-54-203-33-198-116-55-73-86-44-87-220-41-117-197-133-71-25-256-56-57-250-213-224-243-252-39-40-132-251-219-26-200-242-27-43-42-241-218-29-217-216-72-202-28-201-215-214	19.58543	12.73869	16.16206
4	Conventional Method	No Feature Ordering	13.32915		

6.9 Summary

This chapter presented several feature ordering approaches based on linear discriminant and

obtained the following achievements:

1. FS, an MFLD metric based on FLD, was employed for feature ordering. It can cope with not only univariate but also multivariate classification problems. Most of the results derived by FS-based feature ordering with ITID (ILIA2) can overcome those derived by conventional batch training method.
2. Similar to the principle of FLD, SD, a novel linear discriminant metric, was proposed in this study. Different from FS that is based on the differences between patterns, SD is based on standard deviations among patterns. The basic element of standard deviation is difference, hence SD has the same function of FLD. Moreover, because SD is derived from standard deviation, it can also cope with classification problems with multiple categories. Experimental results indicated that SD feature ordering can produce much lower error rates using ITID (ILIA2) in IAL.
3. Based on SD, AD was developed to make the solution always get the maximum discrimination ability in IAL's gradually growing feature space. A corresponding criterion called MMDC was presented at the same time. Moreover, different from SD, AD has the capacity to show the redundancy between features.
4. According to MMDC, two kinds of feature ordering approaches were developed. The GA-based approach can obtain an optimum feature ordering which often produce a lower error rate than some other approaches including the conventional batch training approach. A more interesting thing which is necessary to be mentioned is that the GA-based approach often converge to one feature ordering during the evolutionary process. Moreover, the first feature in the ordering derived from GA-based approach is always the first feature in SD-based feature ordering. As a result of that MAMFO based on SD was presented. Experiments confirmed that GAOFO and SD-based MAMFO approaches can obtain the same feature ordering. They produced AD-based results which were the optimum feature orderings. Moreover, they often get the lowest error rates, especially in ITID (ILIA2). However, compare with these two kinds of feature orderings, SD-based MAMFO is more time-saving.
5. Based on the study of AD, feature ordering calculated with the consideration about growing feature space is more validated to produce stable results with lower error rates.

Therefore, it is necessary to compute feature orderings with the influence from the increasing number of feature dimensions.

Generally, according to the research of this chapter, feature ordering, which was previously treated as a by-product of IAL, now should be formally regarded as an independent and common preprocessing stage in IAL. However, in the meanwhile, it is also unique to those conventional approaches which trains features in one batch.

In addition, according to the experimental results, it seems that the algorithms are sensitive to different datasets. The reasons of that are complex, which generally come from three aspects, firstly, the data distribution, secondly, the feature ordering methods, and the third is the neural network structure. Data distribution, preprocessing methods and predictive algorithms are three important elements which may influence the final results. What is the relations among these elements and classification performance will be an issue to be researched in the future.

Chapter 7

Experimental Analysis of Feature Ordering

7.1 Overview

This chapter aims to compare the experimental results derived by the approaches presented in previous chapters. Approaches about feature ordering and corresponding techniques are analyzed and compared in the meanwhile. The significance of different feature ordering metrics and approaches are also shown in this chapter, and moreover, the relationship between some good feature ordering metrics and the final pattern classification results are illustrated by tables and figures with statistical demonstrations. More specifically, in Section 7.2, experimental results derived from different feature ordering approaches based on instance correlation, mutual information and linear discriminant are individually compared with each other by datasets; the feature ordering measure with the best performance is selected to analyze the relations with final classification error rates in Section 7.3; in the last section of this chapter, the best results obtained in this study and the best results derived by the best well-performed approach are compared with some recent state of the art experimental results from other neural network researchers.

7.2 Comparisons with different feature ordering

7.2.1 Diabetes

The results derived by ITID with different feature ordering approaches have been shown in Table 7.1. Figure 7.1 shows the comparison of Diabetes results. The Integrated Correlation-based feature ordering produces the lowest classification error rate in ITID (ILIA1), and the Contribution-based feature ordering obtains the best result in ITID (ILIA2). Likewise, Integrated Correlation-based feature ordering also gets the lowest error rate in average. Besides that, AD exhibits the second best performance, and Input-Output Correlation-based feature ordering, SD and FS get the third ranking to their identical feature ordering. In addition, no feature ordering algorithms with IAL produces higher error rates than one-batch-training conventional method.

Table 7.1: Classification Result Comparison (Diabetes)

	Approach	Feature Ordering	Classification Error (%)		
			ITID (ILIA1)	ITID (ILIA2)	AVG
1	Contribution-based	2-8-1-5-7-4-3-6	22.31772	**22.03125**	22.17448
2	Input-Output Correlation-based	2-6-8-7-1-4-5-3	21.84896	22.39583	22.12240
3	Integrated Correlation-based	2-7-6-8-1-4-5-3	**21.32812**	22.47396	**21.90104**
4	mRMR-Difference	2-6-1-7-3-8-4-5	22.86459	23.56770	23.21615
5	mRMR-Quotient	2-6-1-7-3-8-5-4	22.96876	23.82813	23.39845
6	SD	2-6-8-7-1-4-5-3	21.84896	22.39583	22.12240
7	AD	2-6-7-8-5-4-1-3	21.61458	22.60416	22.10937
8	FS	2-6-8-7-1-4-5-3	21.84896	22.39583	22.12240
9	Original Ordering	1-2-3-4-5-6-7-8	22.86458	23.80209	23.33334
10	Conventional Method	No Feature Ordering	23.93229		

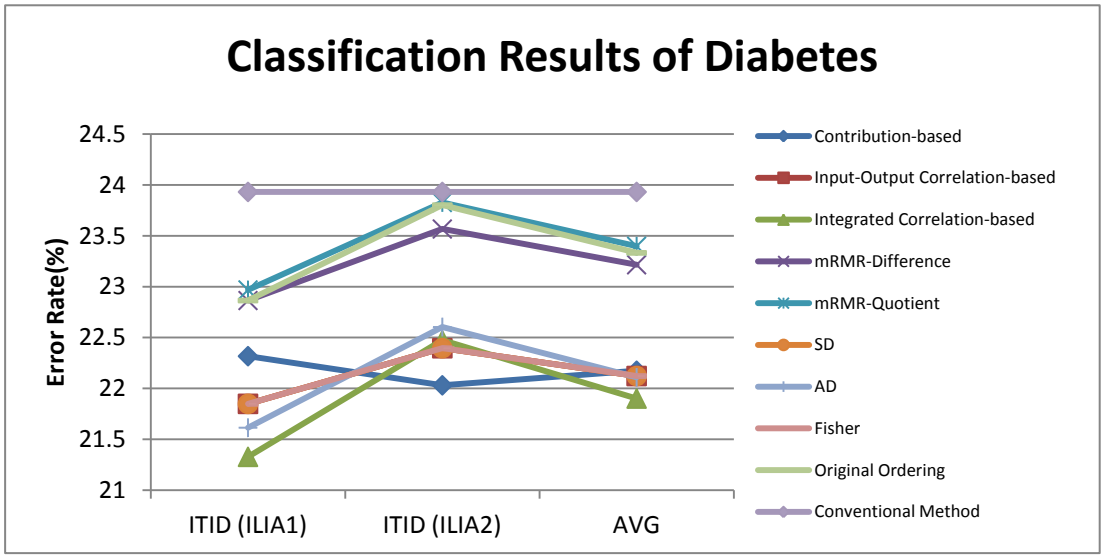


Figure 7.1: Classification Results of Diabetes

7.2.2 Cancer

Table 7.2 compares the results derived by ITID with different feature ordering approaches, and Figure 7.2 shows the comparison of these Cancer results. Feature ordering derived by AD produces the lowest classification error rate in ITID (ILIA1), and the second lowest error rate in ITID (ILIA2). The mRMR-Difference feature ordering obtains the best result in ITID (ILIA2). However, in the aspect of ITID average, AD exhibits the best performance with lowest error rate. Input-Output Correlation-based feature ordering, SD and FS get the second ranking as their feature orderings are identical. Integrated Correlation-based feature ordering gets the third ranking, while the mRMR-Difference, the approach which obtains the best result in ITID (ILIA2) only in the fourth place. Moreover, another mRMR approach, mRMR-Quotient exhibits worse performance than conventional batch training methods. This indicates that mRMR approaches are unstable approach, and feature ordering obtained by these approach may not obtain good results. In addition, in this experiment, only Input-Output Correlation-based feature ordering, SD, AD, and FS feature ordering can beat conventional approach by both ITID (ILIA1) and ITID (ILIA2). This denotes that these four approaches are more stable than others. Another fact which is worth mentioning is that neither the feature ordering derived by Contribution-based nor Original Feature Ordering can conquer the conventional batch training approach. It obviously shows that

IAL itself cannot always beat batch training conventional method, proper feature ordering is more important for obtaining better result with lower error rate.

Table 7.2: Classification Result Comparison (Cancer)

	Approach	Feature Ordering	Classification Error (%)		
			ITID (ILIA1)	ITID (ILIA2)	AVG
1	Contribution-based	2-3-5-8-6-7-4-1-9	2.50000	1.92529	2.21264
2	Input-Output Correlation-based	3-2-6-7-1-8-4-5-9	1.69541	1.72414	1.70977
3	Integrated Correlation-based	1-6-8-7-3-2-4-5-9	1.83908	2.01150	1.92529
4	mRMR-Difference	2-6-1-7-3-8-5-4-9	2.29885	**1.58046**	1.93966
5	mRMR-Quotient	2-6-1-7-8-3-5-4-9	2.29885	1.81035	2.05460
6	SD	3-2-6-7-1-8-4-5-9	1.69541	1.72414	1.70977
7	AD	3-2-6-7-5-1-8-4-9	**1.55173**	1.60920	**1.58046**
8	FS	3-2-6-7-1-8-4-5-9	1.69541	1.72414	1.70977
9	Original Ordering	1-2-3-4-5-6-7-8-9	2.90230	2.18391	2.54310
10	Conventional Method	No Feature Ordering	1.86782		

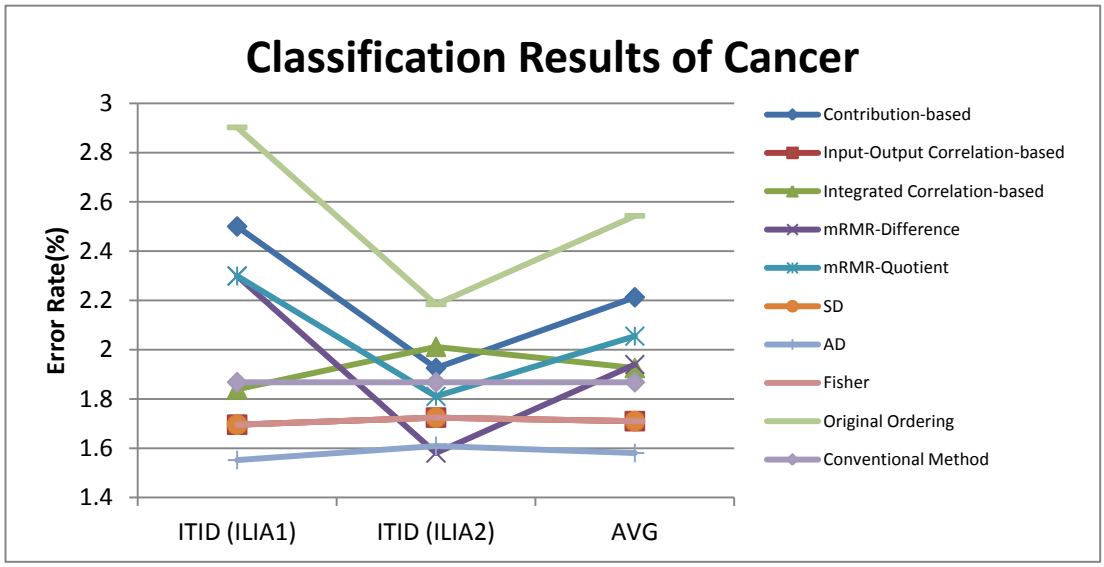


Figure 7.2: Classification Results of Cancer

7.2.3 Glass

Table 7.3 and Figure 7.3 illustrate the classification error rates derived by different feature ordering with IAL approaches. In both of ITID (ILIA1) and ITID (ILIA2), AD obtains the second best results. The Integrated Correlation-based Output AVG feature ordering and SD feature

ordering obtain the lowest error rate in ITID (ILIA1) and ITID (ILIA2), respectively. However, AD is most stable feature ordering. According to the average value, AD is the lowest. Moreover, in Glass, not all results derived by IAL approaches can exhibit better performance than Conventional Method. Some Correlation-based feature orderings and Original Ordering obtain high error rates in ITID (ILIA1).

Table 7.3: Classification Result Comparison (Glass)

	Approach	Feature Ordering	Classification Error (%)		
			ITID (ILIA1)	ITID (ILIA2)	AVG
1	Contribution-based	4-2-8-3-6-9-1-7-5	36.41510	33.11322	34.76416
2	Input-Output Correlation-based Output AVG	3-4-8-2-5-6-1-9-7	40.37733	35.56605	37.97169
3	Input-Output Correlation-based Output Weight	3-4-8-2-1-9-6-5-7	40.28300	35.66040	37.97170
4	Input-Output Correlation-based Output Integration	8-6-3-7-4-5-2-1-9	54.15097	38.20755	46.17926
5	Integrated Correlation-based Output AVG	3-9-2-4-8-5-6-1-7	**34.24530**	32.26417	33.25474
6	Integrated Correlation-based Output Weight	3-9-4-8-2-5-6-1-7	41.03771	36.60378	38.82075
7	Integrated Correlation-based Output Integration	8-6-3-5-7-4-2-9-1	53.39625	38.20755	45.80190
8	mRMR-Difference	3-2-4-5-7-9-8-6-1	39.05663	35.09436	37.07550
9	mRMR-Quotient	3-5-2-8-9-4-7-6-1	35.28304	31.50946	33.39625
10	SD	3-8-4-2-6-5-9-1-7	34.81133	**28.96228**	31.88681
11	AD	3-8-2-4-6-7-1-5-9	34.33964	29.24530	**31.79247**
12	FS	3-4-8-2-6-5-1-9-7	39.62261	35.56606	37.59434
13	Original Ordering	1-2-3-4-5-6-7-8-9	45.18870	36.03775	40.61323
14	Conventional Method	No Feature Ordering	41.22641		

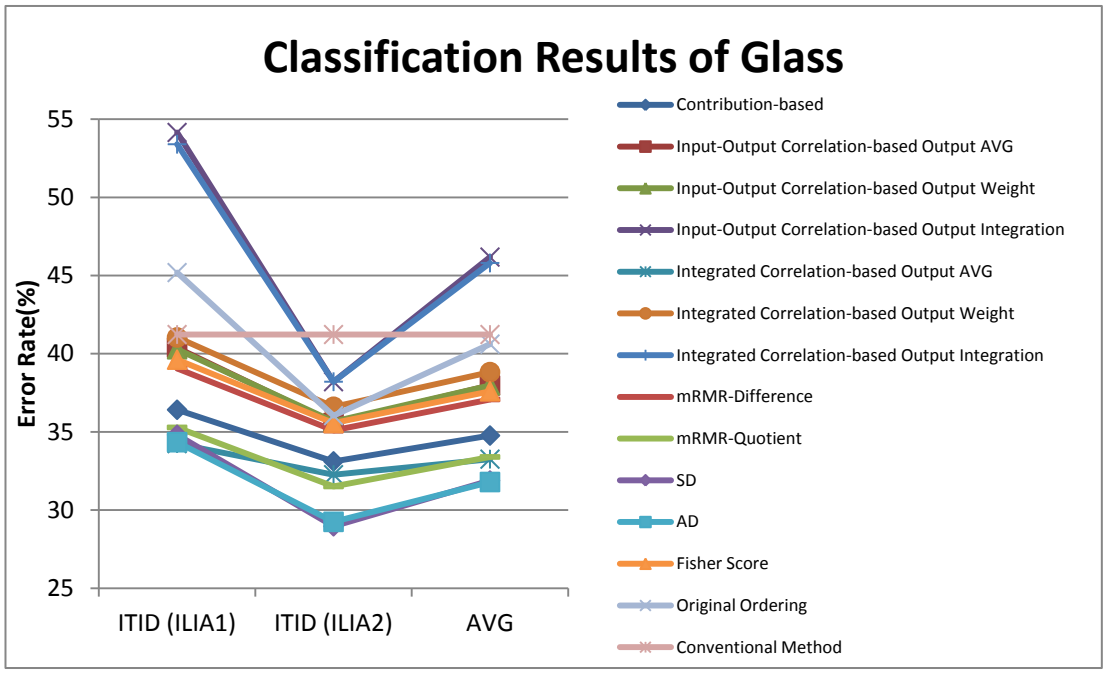


Figure 7.3: Classification Results of Glass

7.2.4 Thyroid

Results comparison of Thyroid is presented in Table 7.4 and Figure 7.4. Obviously, AD exhibits the best performance, which obtains the lowest classification error rate in both ITID (ILIA1) and ITID (ILIA2) comparing with all the other feature ordering approaches. It is manifest that the lowest average error rate also achieved by AD. However, besides AD, only two mRMR approaches can defeat the conventional one-batch-training method. The mRMR-Difference and mRMR-Quotient obtain the second and the third best result, respectively, while all the other feature ordering approaches failed with ITID (ILIA1) in prediction. This indicates that all the other feature ordering approaches is unstable and may negatively impact the influence of IAL. Anyway, all the results derived by corresponding feature selection and ITID (ILIA2) is acceptable.

Table 7.4: Classification Result Comparison (Thyroid)

	Approach	Feature Ordering	Classification Error (%)		
			ITID (ILIA1)	ITID (ILIA2)	AVG
1	Contribution-based	17-21-19-18-1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-20	2.50556	1.72222	2.11389
2	Input-Output Correlation-based Output AVG	17-21-19-18-10-3-2-16-7-20-13-6-8-5-1-4-11-14-9-12-15	2.50000	1.68611	2.09306
3	Input-Output Correlation-based Output Weight	17-21-19-18-10-3-2-16-7-13-20-5-8-4-11-9-14-1-6-15-12	2.50833	1.68889	2.09861
4	Input-Output Correlation-based Output Integration	17-21-19-18-3-10-2-7-16-13-8-5-20-1-14-4-6-11-9-15-12	2.47778	1.74445	2.11111
5	Integrated Correlation-based Output AVG	17-21-10-19-18-13-3-8-16-6-2-4-5-12-7-9-15-14-20-1-11	2.20000	1.85833	2.02917
6	Integrated Correlation-based Output Weight	17-21-10-19-18-3-13-16-2-8-5-4-7-12-9-15-14-20-11-6-1	2.51667	1.72222	2.11944
7	Integrated Correlation-based Output Integration	17-10-21-3-19-13-18-8-16-2-5-7-4-15-14-9-6-1-20-11-12	2.28056	1.57500	1.92778
8	mRMR-Difference	3-7-17-10-6-8-13-16-4-5-12-21-18-19-2-20-15-9-14-11-1	1.61944	1.29722	1.45833
9	mRMR-Quotient	3-10-16-7-6-17-2-8-13-5-1-4-11-12-14-9-21-15-18-19-20	1.62500	1.42222	1.52361
10	SD	21-19-17-18-3-7-6-16-13-20-10-8-2-4-5-1-11-12-14-15-9	1.92778	1.52500	1.72639
11	AD	21-18-19-15-20-17-13-7-12-5-4-8-3-9-16-6-14-1-11-10-2	**1.52500**	**1.21667**	**1.37083**
12	FS	17-21-19-18-3-10-2-16-20-6-7-8-13-1-5-11-4-14-12-9-15	2.52500	1.77222	2.14861
13	Original Ordering	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21	2.05000	1.59167	1.82083
14	Conventional Method	No Feature Ordering	1.86389		

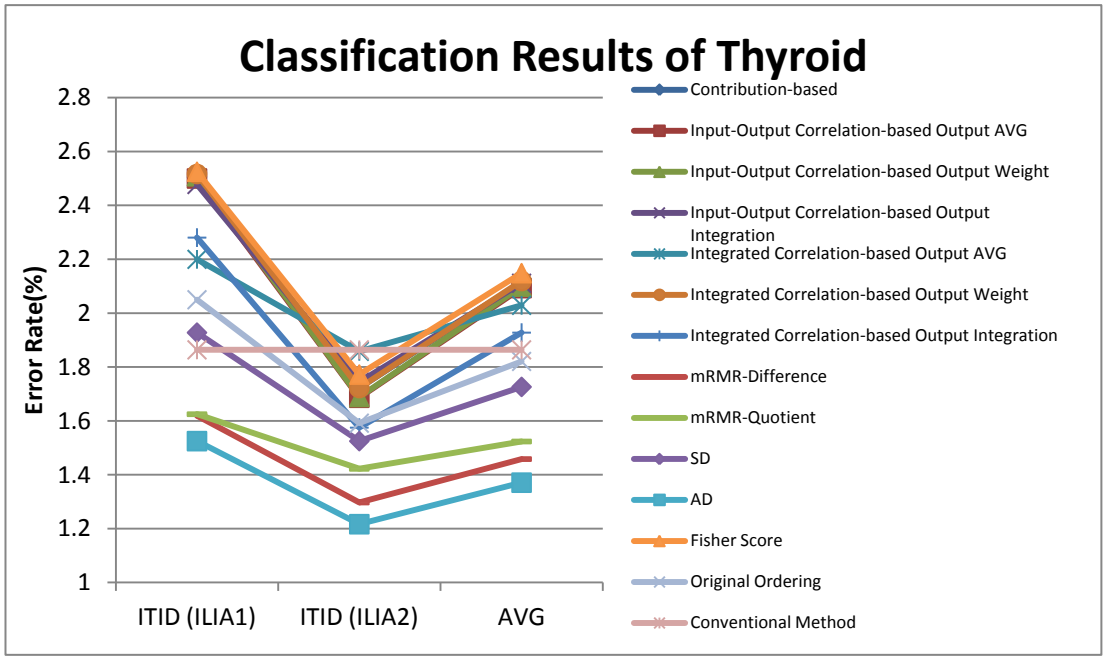


Figure 7.4: Classification Results of Thyroid

7.2.5 Semeion

Results comparison of Semeion is presented in Table 7.5 and Figure 7.5. Obviously, no ITID (ILIA1) results are better than that derived by conventional batch training method. However, in another aspect of ITID (ILIA2), only the error rate of feature ordering based on mRMR-Quotient is a little greater than conventional method, all the other results are better than the batch training approach. Moreover, in ITID (ILIA2), FS gets the lowest error rate and AD exhibits the second best performance. Anyway, all the results derived by corresponding feature selection and ITID (ILIA2) is acceptable.

Table 7.5: Classification Result Comparison (Semeion)

	Approach	Feature Ordering	Classification Error (%)		
			ITID (ILIA1)	ITID (ILIA2)	AVG
1	Contribution-based	162-178-146-145-95-96-194-8-79-112-7-111-129-230-161-143-191-9-113-127-114-80-82-229-231-130-6-63-179-109-207-98-177-193-128-22-37-52-67-106-110-121-157-10-36-78-108-122-228-195-47-211-4-97-107-236-104-	18.25378	12.95226	15.60302

		173-221-222-103-163-5-159-136-94-119-137-152-192-206-144-210-105-235-232-83-62-92-141-153-180-247-237-171-93-189-12-246-11-14-46-115-118-151-154-156-172-238-248-15-77-61-16-102-13-66-138-158-212-50-175-135-205-123-167-164-234-20-35-89-120-134-176-223-227-3-147-181-174-31-45-68-142-188-131-48-21-169-170-203-51-29-30-34-38-165-208-168-60-23-245-124-53-44-88-140-233-239-204-2-64-125-196-139-81-24-99-100-116-133-150-160-187-74-226-65-55-85-59-148-126-166-87-183-69-249-91-190-209-90-76-220-70-185-26-40-84-186-218-219-182-75-244-19-184-58-73-117-201-217-1-202-155-54-43-49-225-243-240-253-18-25-32-42-86-199-39-198-213-17-71-132-56-216-250-215-254-33-214-41-57-101-27-200-224-197-251-149-252-242-28-241-255-72-256			
2	mRMR-Difference	162-79-82-146-178-111-8-130-194-63-231-98-161-95-62-145-163-9-112-177-66-229-47-191-128-77-11-129-179-96-230-105-93-114-193-127-10-232-83-7-76-195-228-99-78-143-147-113-234-64-3-50-174-84-152-80-103-233-92-136-81-210-175-46-4-67-245-91-61-108-51-104-159-97-121-246-150-1-167-75-6-135-107-12-207-100-65-119-109-5-151-189-106-18-48-227-153-164-2-94-188-120-68-144-168-16-102-247-90-166-192-101-235-149-211-137-17-238-60-115-183-89-180-190-158-45-165-35-122-131-182-31-237-36-123-59-134-37-173-209-138-69-226-19-13-124-118-23-157-236-244-52-187-208-154-184-85-248-22-74-148-155-206-169-205-139-255-181-222-212-34-254-172-15-110-21-53-142-204-196-30-160-88-239-249-140-20-58-49-141-32-253-33-176-170-14-156-125-221-240-171-199-223-38-24-185-225-220-126-116-186-198-54-73-203-197-86-70-87-256-117-44-243-57-25-133-55-252-241-224-71-213-41-250-56-219-242-39-27-251-132-29-26-200-218-43-40-216-72-217-42-28-215-202-214-201	17.86432	12.88945	**15.37688**
3	mRMR-Quotient	162-63-82-233-238-135-228-45-103-8-143-195-100-130-1-152-11-77-95-163-174-188-50-194-231-105-68-3-146-183-16-79-245-108-191-	**17.72613**	13.34172	15.53392

		178-128-62-165-98-155-235-75-122-229-111-84-9-51-147-101-179-47-157-119-18-145-255-150-12-66-177-168-76-232-112-93-23-136-5-99-246-161-64-129-237-46-210-10-159-189-230-167-164-96-78-102-234-89-193-83-2-61-81-7-127-180-175-121-109-153-114-226-80-91-17-187-149-35-182-211-144-85-158-253-107-247-67-104-139-4-151-48-131-65-192-113-94-205-227-240-137-124-184-60-37-173-92-148-196-13-118-69-166-207-115-106-244-97-120-141-254-190-154-212-22-236-59-6-134-90-123-172-52-169-138-33-248-209-36-204-19-74-208-181-31-110-24-140-88-239-220-142-156-170-206-21-58-249-256-49-199-53-160-30-116-34-15-186-125-222-86-171-198-225-185-20-197-221-32-38-87-126-117-176-73-241-203-14-25-252-54-223-71-213-44-243-133-57-70-27-224-55-219-29-250-41-242-132-56-216-26-251-218-39-28-200-43-217-72-215-40-214-42-202-201			
4	SD	112-162-96-128-146-178-111-79-95-161-145-1-130-177-80-194-63-127-82-98-129-113-163-66-114-47-9-64-62-93-193-8-77-81-179-78-10-231-2-230-229-11-97-143-3-17-83-195-232-65-144-147-99-50-7-228-105-76-92-191-233-67-210-4-84-234-152-103-175-46-51-108-48-91-159-174-109-94-61-18-107-192-136-167-104-75-6-151-5-245-12-135-246-207-121-150-106-168-16-188-153-166-120-100-119-68-183-189-227-102-90-164-149-247-211-255-115-182-60-31-101-256-235-137-89-165-190-131-209-59-208-180-35-36-158-45-122-134-184-254-37-238-110-13-69-19-52-123-187-226-118-124-240-74-173-49-237-148-248-169-138-236-85-15-32-181-34-154-206-23-22-212-160-53-222-244-176-157-204-30-21-142-205-155-172-58-14-196-125-33-249-139-88-253-20-239-171-141-170-199-185-140-223-126-38-225-156-186-221-54-198-24-203-70-73-116-86-55-220-44-87-197-117-41-133-71-57-56-224-25-250-243-241-213-252-39-40-132-251-242-200-219-26-43-42-27-218-29-217-216-72-202-28-201-215-214	18.85678	12.96483	15.91081

5	AD	112-96-162-178-146-128-111-95-79-161-145-177-130-256-80-127-194-63-1-82-129-98-66-113-163-9-47-114-193-64-8-81-179-93-97-65-144-10-231-230-229-2-11-77-195-62-143-3-232-17-147-78-83-7-228-50-99-233-255-92-210-4-105-67-191-76-48-234-51-109-152-175-84-103-108-46-240-192-94-159-91-174-18-107-254-167-151-136-104-16-6-12-75-5-135-188-150-207-246-183-168-166-106-61-121-68-102-149-182-100-189-245-227-119-120-153-208-164-90-211-49-247-115-184-59-31-101-165-209-110-131-235-89-60-33-180-137-190-35-36-187-241-32-45-69-37-52-74-253-134-158-181-122-148-13-169-160-19-238-124-58-118-199-176-85-34-226-185-123-173-53-15-248-125-237-154-196-212-224-236-22-204-186-239-23-170-138-198-206-172-126-88-155-171-30-142-21-244-222-157-205-14-20-249-225-54-55-70-223-56-141-73-38-139-140-203-57-242-197-200-156-71-86-252-41-116-87-44-24-221-243-117-133-220-40-251-202-42-72-250-43-213-201-39-25-132-216-217-218-215-219-214-26-27-29-28	18.02764	12.83922	15.43343
6	FS	162-112-146-178-96-111-145-130-161-128-79-177-95-194-127-82-63-129-98-9-47-66-114-113-80-8-163-193-10-230-229-231-93-11-179-232-77-143-62-195-64-1-228-81-7-147-83-97-233-78-99-210-3-105-65-191-92-234-50-67-76-4-175-103-144-2-46-152-108-174-159-84-109-51-91-107-48-136-6-12-104-246-192-135-5-121-17-207-94-106-151-227-16-75-245-102-100-188-120-211-61-189-150-119-247-167-18-153-68-31-90-115-235-166-149-101-168-131-137-209-164-190-89-183-165-158-60-36-182-45-59-13-122-134-180-35-37-208-226-69-110-248-118-187-124-52-184-238-173-15-123-237-19-74-236-23-85-22-148-212-32-138-154-30-206-21-254-181-204-14-222-157-169-142-34-176-155-172-53-249-49-205-125-255-196-244-160-88-20-171-141-58-139-170-240-140-185-199-126-38-253-156-223-239-225-186-24-70-221-54-203-33-198-116-55-73-86-44-87-220-41-117-197-133-71-25-256-56-57-250-213-224-243-252-39-40-132-251-219-26-200-242-27-	19.58543	**12.73869**	16.16206

		43-42-241-218-29-217-216-72-202-28-201-215-214			
7	Original Ordering	1-2-3-...-256	24.84925	13.00251	18.92588
8	Conventional Method	No Feature Ordering	13.32915		

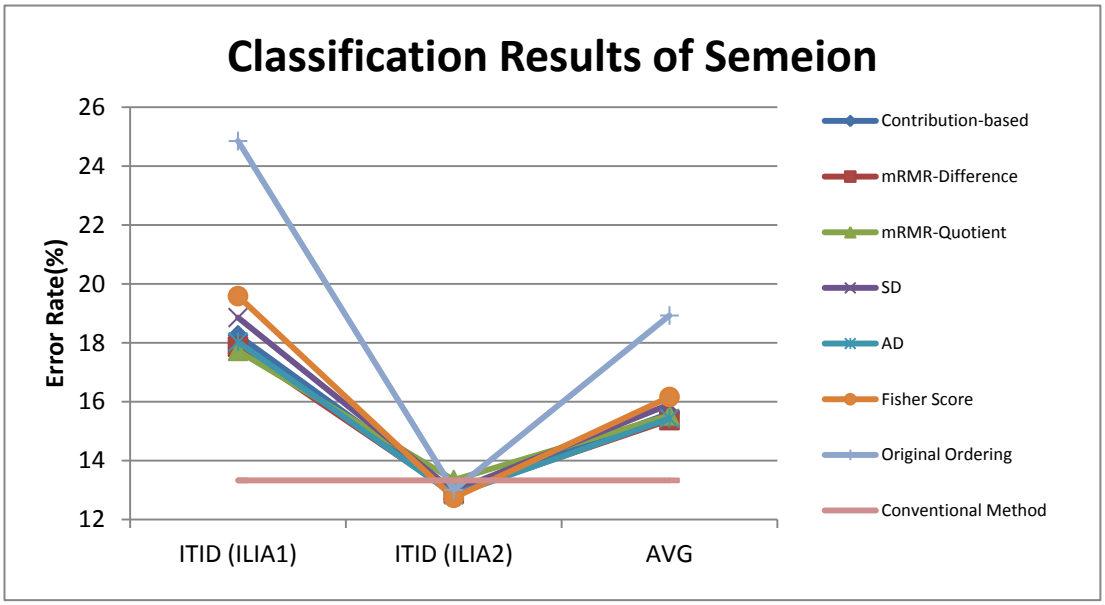


Figure 7.5: Classification Results of Semeion

7.2.6 Discussions

In this thesis, research is about the preprocessing of IAL, especially the influence from different feature orderings. Nine different feature ordering algorithms are carried out in this study. According to the results, it is manifested that most of these feature ordering algorithms cannot always exhibit better performance than conventional method which trains, validates and tests all features in one batch. In Table 7.6, approaches which cannot exhibit better performance than conventional batch training method will be shaded in blue. Those obtain lower classification results will be kept in white. Those cells in grey indicate that there are no experiments done using such an approach. Except approaches 2 and 3, the performance of all the other approaches are ranked according to their error rates. Moreover, a total score of this ranking is also given in the Table 7.6. They are also sorted from low to high at the last column of the table. According to the

performance ranking, AD is in the first place, following by SD. Two mRMR approaches are sorted in the third and fourth places. Because Correlation-based approaches have not exhibited better performance in Thyroid, it is undoubted that AD is the best feature ordering approach. Although sometimes "no matter what algorithm you use, there is at least one target function for which random guessing is a better algorithm" [49], it is still necessary to state that the performance AD is not only very good, but also very stable.

According to Table 7.6, both ITID (ILIA1) and ITID (ILIA2) may produce higher error rates than conventional results. Moreover, it seems no feature ordering approach can always exhibit better performance than conventional method in classification problems, no matter what kinds of prediction methods is applied. Taking each result cell in Table 7.6 as an experimental unit, there are 65 experiments in total. Statistically, 25 of the total have higher error rates than conventional method. Three of them have no impact on the average performance. They are Original Ordering in Glass and Thyroid, and SD in Thyroid. All of them are in ITID (ILIA1). However, average values are seriously influenced in the other experiments. More specifically, three of these bad performance are derived from both ITID (ILIA1) and ITID (ILIA2), which are mRMR-Quotient feature ordering in Semeion, and Contribution-based and Original Feature Ordering in Cancer. Only one is directly influenced by ITID (ILIA2), which is Integrated Correlation-based feature ordering in Cancer, and rest eighteen experiments obtain higher error rates merely because of ITID (ILIA1). According to the proportion of these unsatisfactory performance, where most of the high error rates are produced by ITID (ILIA1), it confirms again that ILIA2 is better than other ILIA algorithms, which has been illustrated in [21]. Table 7.7 shows the reduction rate of error rate using different IAL feature ordering for benchmark datasets.

Table 7.6: Classification Performance Ranking

	Approaches		Diabetes			Cancer			Glass			Thyroid			Semeion[4]			Total Score	Rank
			ITID (ILIA1)	ITID (ILIA2)	AVG	ITID (ILIA1)	ITID (ILIA2)	AVG	ITID (ILIA1)	ITID (ILIA2)	AVG	ITID (ILIA1)	ITID (ILIA2)	AVG	ITID (ILIA1)	ITID (ILIA2)	AVG		
1	Contribution-based		4	1	4	6	6	6	4	4	4	6	6	6	4	4	4	69	6
2	Input-Output Correlation -based	AVG													-	-	-		
		Weight													-	-	-		
		Integration													-	-	-		
3	Integrated Correlation -based	AVG													-	-	-		
		Weight													-	-	-		
		Integration													-	-	-		
4	mRMR-Difference		6	5	5	4	1	4	5	5	5	2	2	2	2	3	1	52	3
5	mRMR-Quotient		7	7	7	4	5	5	3	3	3	3	3	3	1	7	3	64	4
6	SD		2	2	2	2	3	2	2	1	2	4	4	4	5	5	5	45	2
7	AD		1	4	1	1	2	1	1	2	1	1	1	1	3	2	2	24	1
8	Fisher Score		2	2	2	2	3	2	6	6	6	7	7	7	6	1	6	65	5
9	Original Ordering		5	6	6	7	7	7	7	7	7	5	5	5	7	6	7	94	7

[4] The correlation-based feature ordering on Semeion dataset was not presented in this thesis, because the feature number of Semeion is too large to print on the paper.

Table 7.7: Classification Error Rate Reduction Compared with Conventional Method

	Approaches		Diabetes			Cancer			Glass			Thyroid			Semeion		
			ITID (ILIA1)	ITID (ILIA2)	AVG	ITID (ILIA1)	ITID (ILIA2)	AVG	ITID (ILIA1)	ITID (ILIA2)	AVG	ITID (ILIA1)	ITID (ILIA2)	AVG	ITID (ILIA1)	ITID (ILIA2)	AVG
1	Contribution-based		-6.75%	-7.94%	-7.34%	33.85%	3.08%	18.46%	-11.67%	-19.68%	-15.68%	34.43%	-7.60%	13.41%	36.95%	-2.83%	17.06%
2	Input-Output Correlation -based	AVG	-8.71%	-6.42%	-7.56%	-9.23%	-7.69%	-8.46%	-2.06%	-13.73%	-7.89%	34.13%	-9.54%	12.30%			
		Weight							-2.29%	-13.50%	-7.89%	34.58%	-9.39%	12.59%			
		Integration							31.35%	-7.32%	12.01%	32.94%	-6.41%	13.26%			
3	Integrated Correlation -based	AVG	-10.88%	-6.09%	-8.49%	-1.54%	7.69%	3.08%	-16.93%	-21.74%	-19.34%	18.03%	-0.30%	8.87%			
		Weight							-0.46%	-11.21%	-5.84%	35.02%	-7.60%	13.71%			
		Integration							29.52%	-7.32%	11.10%	22.35%	-15.50%	3.43%			
4	mRMR-Difference		-4.46%	-1.52%	-2.99%	23.08%	-15.38%	3.85%	-5.26%	-14.87%	-10.07%	-13.12%	-30.40%	-21.76%	34.02%	-3.30%	15.36%
5	mRMR-Quotient		-4.03%	-0.44%	-2.23%	23.08%	-3.08%	10.00%	-14.42%	-23.57%	-18.99%	-12.82%	-23.70%	-18.26%	32.99%	0.09%	16.54%
6	SD		-8.71%	-6.42%	-7.56%	-9.23%	-7.69%	-8.46%	-15.56%	-29.75%	-22.65%	3.43%	-18.18%	-7.38%	41.47%	-2.73%	19.37%
7	AD		-9.68%	-5.55%	-7.62%	-16.92%	-13.85%	-15.38%	-16.70%	-29.06%	-22.88%	-18.18%	-34.72%	-26.45%	35.25%	-3.68%	15.79%
8	Fisher Score		-8.71%	-6.42%	-7.56%	-9.23%	-7.69%	-8.46%	-3.89%	-13.73%	-8.81%	35.47%	-4.92%	15.28%	46.94%	-4.43%	21.25%
9	Original Ordering		-4.46%	-0.54%	-2.50%	55.38%	16.92%	36.15%	9.61%	-12.59%	-1.49%	9.99%	-14.60%	-2.31%	86.43%	-2.45%	41.99%

7.3 Relation between AD and Classification Error

According to the results comparison and discussion presented in the last subsection, it seems that the process where AD is employed in preprocessing and ILIA2 is used with ITID in prediction based on neural networks is a more stable and reliable way to obtain better results. Thus, it is necessary to find out the answers for the questions: why AD is so powerful? why the feature ordering produced by AD can cope well with classification problems?

AD is a metric for feature's discrimination ability measurement in IAL. Because IAL has its own property on gradual sequential feature training, it is necessary to guarantee the imported feature subset always has the greatest feature discrimination ability during the pattern recognition process. Thus, the MMDC is very important. As a result of this, the AD values of each step of the feature ordering directly derived from AD should be greater than those derived from other feature ordering approaches. As such, for the same dataset, the AD mean of the feature ordering derived from AD will be larger than those AD means of the feature orderings calculated by other approaches. Tables 7.8, 7.9, 7.10, 7.11, and 7.12 show the AD means of all the feature orderings employed in the experiments of this study. It is obvious that AD means of feature orderings directly derived by AD is the greatest in each dataset. AD means in other feature ordering is smaller. In another aspect, experiments in this study have confirmed that feature orderings which are obtained by AD can often produce results with accuracy, stability and acceptability. Consequently, the questions change to whether higher AD mean is a good prediction for accurate and steady results.

Checking the AD means shown in Tables 7.8 ~ 7.12 with classification results presented in previous sections, the correlations between AD means and classification error rates are more or less the negative. Table 7.13 shows the correlations between classification error rates derived by ITID (ILIA2) and AD means. The correlations is statistically calculated according to every feature ordering approach presented in this study. Moreover, Figures 7.6, 7.7, 7.8, 7.9 and 7.10 illustrate the negative correlation between classification error rates derived by ITID (ILIA2) and AD means. The black lines in these diagrams show the regression directions and trends. According to these black lines, it is obvious that the higher the AD value, the lower is error rate.

Table 7.8: AD Mean derived from different approaches (Diabetes)

	ITID-AD		ITID-SD		Fisher Score		mRMR-Difference		mRMR-Quotient		Contribution-based		Input-Output Correlation-based		Integrated Correlation-based		Original Ordering	
	Order	AD	Order	AD	Order	AD	Order	AD	Order	AD	Order	AD	Order	AD	Order	AD	Order	AD
1	2	0.3694	2	0.3694	2	0.3694	2	0.3694	2	0.3694	2	0.3694	2	0.3694	2	0.3694	1	0.1437
2	6	0.3317	6	0.3317	6	0.3317	6	0.3317	6	0.3317	6	0.3317	6	0.3317	7	0.2788	2	0.2414
3	7	0.2751	8	0.2718	8	0.2718	1	0.2453	1	0.2453	1	0.2453	8	0.2718	6	0.2751	3	0.2043
4	8	0.2468	7	0.2468	7	0.2468	7	0.2281	7	0.2281	7	0.2281	7	0.2468	8	0.2468	4	0.1873
5	5	0.2268	1	0.2183	1	0.2183	3	0.2037	3	0.2037	3	0.2037	1	0.2183	1	0.2183	5	0.1770
6	4	0.2093	4	0.2044	4	0.2044	8	0.1999	8	0.1999	8	0.1999	4	0.2044	4	0.2044	6	0.1863
7	1	0.1953	5	0.1953	5	0.1953	4	0.1898	5	0.1912	5	0.1912	5	0.1953	5	0.1953	7	0.1826
8	3	0.1828	3	0.1828	3	0.1828	5	0.1828	4	0.1828	4	0.1828	3	0.1828	3	0.1828	8	0.1828
	AVG	0.2546	AVG	0.2526	AVG	0.2526	AVG	0.2438	AVG	0.2440	AVG	0.2440	AVG	0.2526	AVG	0.2464	AVG	0.1882

Table 7.9: AD Mean derived from different approaches (Cancer)

	ITID-AD		ITID-SD		Fisher Score		mRMR-Difference		mRMR-Quotient		Contribution-based		Input-Output Correlation-based		Integrated Correlation-based		Original Ordering	
	Order	AD	Order	AD	Order	AD	Order	AD	Order	AD	Order	AD	Order	AD	Order	AD	Order	AD
1	3	0.9888	3	0.9888	3	0.9888	2	0.9566	2	0.9566	2	0.9566	3	0.9888	1	0.7039	1	0.7039
2	2	0.9713	2	0.9713	2	0.9713	6	0.9065	6	0.9065	3	0.9713	2	0.9713	6	0.7994	2	0.8232
3	6	0.9283	6	0.9283	6	0.9283	1	0.8429	1	0.8429	5	0.8780	6	0.9283	8	0.7615	3	0.8722
4	7	0.9025	7	0.9025	7	0.9025	7	0.8325	7	0.8325	8	0.8188	7	0.9025	7	0.7642	4	0.8151
5	5	0.8617	1	0.8595	1	0.8595	3	0.8595	8	0.7997	6	0.8334	1	0.8595	3	0.8019	5	0.7873
6	1	0.8314	8	0.8263	8	0.8263	8	0.8263	3	0.8263	7	0.8267	8	0.8263	2	0.8263	6	0.8073
7	8	0.8064	4	0.8021	4	0.8021	5	0.8064	5	0.8064	4	0.8017	4	0.8021	4	0.8021	7	0.8044
8	4	0.7869	5	0.7869	5	0.7869	4	0.7869	4	0.7869	1	0.7869	5	0.7869	5	0.7869	8	0.7869
9	9	0.7605	9	0.7605	9	0.7605	9	0.7605	9	0.7605	9	0.7605	9	0.7605	9	0.7605	9	0.7605
	AVG	0.8709	AVG	0.8696	AVG	0.8696	AVG	0.8420	AVG	0.8354	AVG	0.8482	AVG	0.8696	AVG	0.7785	AVG	0.7956

Table 7.10: AD Mean derived from different approaches (Glass)

	ITID-AD		ITID-SD		Fisher Score		mRMR-Difference		mRMR-Quotient		Contribution-based		Original Ordering	
	Order	AD	Order	AD	Order	AD	Order	AD	Order	AD	Order	AD	Order	AD
1	3	0.3226	3	0.3226	3	0.3226	3	0.3226	3	0.3226	4	0.1716	1	0.0764
2	8	0.2896	8	0.2896	4	0.2652	2	0.2692	5	0.2380	2	0.1625	2	0.1200
3	2	0.2523	4	0.2522	8	0.2522	4	0.2415	2	0.2208	8	0.1669	3	0.2378
4	4	0.2331	2	0.2331	2	0.2331	5	0.2102	8	0.2144	3	0.2331	4	0.2211
5	6	0.2156	6	0.2156	6	0.2156	7	0.1967	9	0.1899	6	0.2156	5	0.1974
6	7	0.2018	5	0.1954	5	0.1954	9	0.1788	4	0.1866	9	0.1936	6	0.1865
7	1	0.1904	9	0.1796	1	0.1856	8	0.1780	7	0.1780	1	0.1839	7	0.1778
8	5	0.1773	1	0.1725	9	0.1725	6	0.1723	6	0.1723	7	0.1758	8	0.1773
9	9	0.1661	7	0.1661	7	0.1661	1	0.1661	1	0.1661	5	0.1661	9	0.1661
	AVG	0.2276	AVG	0.2252	AVG	0.2231	AVG	0.2150	AVG	0.2099	AVG	0.1855	AVG	0.1734

(Continued on the next page)

(Continued from the previous page)

	Input-Output Correlation-based Output AVG		**Input-Output Correlation-based Output Weight**		**Input-Output Correlation-based Output Integration**		**Integrated Correlation-based Output AVG**		**Integrated Correlation-based Output Weight**		**Integrated Correlation-based Output Integration**	
	Order	AD	Order	AD	Order	AD	Order	AD	Order	AD	Order	AD
1	3	0.3226	3	0.3226	8	0.2605	3	0.3226	3	0.3226	8	0.2605
2	4	0.2652	4	0.2652	6	0.1858	9	0.2386	9	0.2386	6	0.1858
3	8	0.2522	8	0.2522	3	0.2505	2	0.2175	4	0.2193	3	0.2505
4	2	0.2331	2	0.2331	7	0.2254	4	0.2071	8	0.2141	5	0.2101
5	5	0.2064	1	0.2151	4	0.2120	8	0.2037	2	0.2037	7	0.1958
6	6	0.1954	9	0.1923	5	0.1910	5	0.1866	5	0.1866	4	0.1910
7	1	0.1856	6	0.1839	2	0.1855	6	0.1796	6	0.1796	2	0.1855
8	9	0.1725	5	0.1725	1	0.1773	1	0.1725	1	0.1725	9	0.1723
9	7	0.1661	7	0.1661	9	0.1661	7	0.1661	7	0.1661	1	0.1661
	AVG	0.2221	AVG	0.2226	AVG	0.2060	AVG	0.2105	AVG	0.2115	AVG	0.2020

Table 7.11: AD Mean derived from different approaches (Thyroid)

	ITID-AD		ITID-SD		Fisher Score		mRMR-Difference		mRMR-Quotient		Contribution-based		Original Ordering	
	Order	AD	Order	AD	Order	AD	Order	AD	Order	AD	Order	AD	Order	AD
1	21	0.5890	21	0.5890	17	0.3816	3	0.1067	3	0.1067	18	0.2883	1	0.0278
2	18	0.5724	19	0.5564	21	0.3817	7	0.1040	10	0.0672	17	0.3682	2	0.0359
3	19	0.5493	17	0.3921	19	0.3921	17	0.1225	16	0.0652	19	0.3661	3	0.0527
4	15	0.5334	18	0.3906	18	0.3906	10	0.0774	7	0.0647	20	0.3439	4	0.0522
5	20	0.4753	3	0.1475	3	0.1475	6	0.0741	6	0.0634	11	0.0951	5	0.0518
6	17	0.3676	7	0.1436	10	0.0913	8	0.0725	17	0.0719	21	0.1114	6	0.0520
7	13	0.2911	6	0.1106	2	0.0679	13	0.0721	2	0.0588	15	0.1113	7	0.0519
8	7	0.2416	16	0.1015	16	0.0669	16	0.0702	8	0.0582	10	0.0777	8	0.0514
9	12	0.2034	13	0.1008	20	0.0669	4	0.0693	13	0.0581	3	0.0807	9	0.0502
10	5	0.1771	20	0.1007	6	0.0661	5	0.0687	5	0.0578	8	0.0790	10	0.0495
11	4	0.1532	10	0.0819	7	0.0659	12	0.0682	1	0.0561	13	0.0786	11	0.0476
12	8	0.1297	8	0.0802	8	0.0652	21	0.0734	4	0.0557	7	0.0780	12	0.0475
13	3	0.1111	2	0.0650	13	0.0650	18	0.0734	11	0.0536	1	0.0733	13	0.0474
14	9	0.1012	4	0.0646	1	0.0630	19	0.0779	12	0.0535	2	0.0611	14	0.0464
15	16	0.0933	5	0.0642	5	0.0627	2	0.0640	14	0.0523	12	0.0609	15	0.0463
16	6	0.0874	1	0.0622	11	0.0601	20	0.0640	9	0.0515	16	0.0605	16	0.0467
17	14	0.0815	11	0.0598	4	0.0598	15	0.0640	21	0.0546	6	0.0602	17	0.0515
18	1	0.0759	12	0.0596	14	0.0584	9	0.0628	15	0.0546	5	0.0599	18	0.0515
19	11	0.0699	14	0.0582	12	0.0582	14	0.0612	18	0.0546	4	0.0596	19	0.0544
20	10	0.0649	15	0.0582	9	0.0573	11	0.0588	19	0.0573	14	0.0582	20	0.0544
21	2	0.0573	9	0.0573	15	0.0573	1	0.0573	20	0.0573	9	0.0573	21	0.0573
	AVG	0.2393	AVG	0.1592	AVG	0.1298	AVG	0.0744	AVG	0.0606	AVG	0.1252	AVG	0.0489

	Input-Output Correlation -based Output AVG		**Input-Output Correlation -based Output Weight**		**Input-Output Correlation -based Output Integration**		**Integrated Correlation-based Output AVG**		**Integrated Correlation-based Output Weight**		**Integrated Correlation-based Output Integration**	
	Order	AD	Order	AD	Order	AD	Order	AD	Order	AD	Order	AD
1	17	0.3816	17	0.3816	17	0.3816	17	0.3816	17	0.3816	17	0.3816
2	21	0.3817	21	0.3817	21	0.3817	21	0.3817	21	0.3817	10	0.0717
3	19	0.3921	19	0.3921	19	0.3921	10	0.0843	10	0.0843	21	0.0843
4	18	0.3906	18	0.3906	18	0.3906	19	0.0943	19	0.0943	3	0.0852
5	10	0.0945	10	0.0945	3	0.1475	18	0.0945	18	0.0945	19	0.0912
6	3	0.0913	3	0.0913	10	0.0913	13	0.0928	3	0.0913	13	0.0905
7	2	0.0679	2	0.0679	2	0.0679	3	0.0906	13	0.0906	18	0.0906
8	16	0.0669	16	0.0669	7	0.0676	8	0.0882	16	0.0861	8	0.0882
9	7	0.0667	7	0.0667	16	0.0667	16	0.0842	2	0.0667	16	0.0842
10	20	0.0667	13	0.0665	13	0.0665	6	0.0808	8	0.0660	2	0.0660
11	13	0.0665	20	0.0665	8	0.0658	2	0.0653	5	0.0656	5	0.0656
12	6	0.0657	5	0.0661	5	0.0654	4	0.0648	4	0.0651	7	0.0654
13	8	0.0650	8	0.0654	20	0.0654	5	0.0644	7	0.0649	4	0.0649
14	5	0.0647	4	0.0649	1	0.0632	12	0.0642	12	0.0647	15	0.0649
15	1	0.0627	11	0.0619	14	0.0615	7	0.0640	9	0.0634	14	0.0630
16	4	0.0622	9	0.0607	4	0.0610	9	0.0628	15	0.0633	9	0.0618
17	11	0.0598	14	0.0592	6	0.0607	15	0.0628	14	0.0616	6	0.0614
18	14	0.0584	1	0.0577	11	0.0584	14	0.0612	20	0.0616	1	0.0597
19	9	0.0575	6	0.0575	9	0.0575	20	0.0612	11	0.0590	20	0.0597
20	12	0.0574	15	0.0575	15	0.0575	1	0.0595	6	0.0588	11	0.0575
21	15	0.0573	12	0.0573	12	0.0573	11	0.0573	1	0.0573	12	0.0573
	AVG	0.1275	AVG	0.1274	AVG	0.1299	AVG	0.1029	AVG	0.1011	AVG	0.0864

Table 7.12: AD Mean derived from different approaches (Semeion)

	ITID-AD		ITID-SD		Fisher Score		mRMR-Difference		mRMR-Quotient		Contribution-based		Original Ordering	
	Order	AD	Order	AD	Order	AD	Order	AD	Order	AD	Order	AD	Order	AD
1	112	0.1406	112	0.1406	162	0.1331	162	0.1331	162	0.1331	162	0.1331	1	0.0965
2	96	0.1292	162	0.1242	112	0.1242	79	0.1107	63	0.1035	178	0.1203	2	0.0767
3	162	0.1223	96	0.1223	146	0.1180	82	0.0975	82	0.0933	146	0.1171	3	0.0713
4	178	0.1175	128	0.1156	178	0.1158	146	0.1006	233	0.0846	145	0.1117	4	0.0667
5	146	0.1151	146	0.1130	96	0.1151	178	0.1024	238	0.0767	95	0.1073	5	0.0629
6	128	0.1124	178	0.1124	111	0.1120	111	0.1017	135	0.0729	96	0.1074	6	0.0610
7	111	0.1102	111	0.1102	145	0.1087	8	0.0963	228	0.0716	194	0.1033	7	0.0617
8	95	0.1080	79	0.1073	130	0.1056	130	0.0958	45	0.0686	8	0.0981	8	0.0633
9	79	0.1059	95	0.1059	161	0.1044	194	0.0948	103	0.0676	79	0.0972	9	0.0646
10	161	0.1041	161	0.1041	128	0.1036	63	0.0933	8	0.0679	112	0.0981	10	0.0650
11	145	0.1029	145	0.1029	79	0.1022	231	0.0908	143	0.0677	7	0.0939	11	0.0652
12	177	0.1017	1	0.1006	177	0.1011	98	0.0896	195	0.0676	111	0.0941	12	0.0642
13	130	0.1005	130	0.0996	95	0.1005	161	0.0900	100	0.0664	129	0.0927	13	0.0626
14	256	0.0994	177	0.0988	194	0.0992	95	0.0901	130	0.0683	230	0.0905	14	0.0608
15	80	0.0983	80	0.0978	127	0.0982	62	0.0884	1	0.0683	161	0.0908	15	0.0595
16	127	0.0973	194	0.0968	82	0.0967	145	0.0887	152	0.0677	143	0.0889	16	0.0591
17	194	0.0964	63	0.0957	63	0.0956	163	0.0878	11	0.0677	191	0.0869	17	0.0590
18	63	0.0953	127	0.0951	129	0.0944	9	0.0870	77	0.0676	9	0.0862	18	0.0587
19	1	0.0943	82	0.0939	98	0.0934	112	0.0877	95	0.0685	113	0.0855	19	0.0580
20	82	0.0932	98	0.0928	9	0.0922	177	0.0879	163	0.0687	127	0.0855	20	0.0571
21	129	0.0923	129	0.0920	47	0.0911	66	0.0872	174	0.0681	114	0.0848	21	0.0563

22	98	0.0914	113	0.0910	66	0.0903	229	0.0862	188	0.0675	80	0.0845	22	0.0557
23	66	0.0905	163	0.0901	114	0.0894	47	0.0856	50	0.0672	82	0.0844	23	0.0551
24	113	0.0897	66	0.0894	113	0.0887	191	0.0845	194	0.0681	229	0.0835	24	0.0542
25	163	0.0889	114	0.0886	80	0.0884	128	0.0846	231	0.0681	231	0.0828	25	0.0533
26	9	0.0882	47	0.0879	8	0.0876	77	0.0839	105	0.0679	130	0.0833	26	0.0522
27	47	0.0875	9	0.0873	163	0.0870	11	0.0832	68	0.0673	6	0.0820	27	0.0513
28	114	0.0868	64	0.0866	193	0.0864	129	0.0830	3	0.0672	63	0.0819	28	0.0503
29	193	0.0862	62	0.0857	10	0.0857	179	0.0825	146	0.0686	179	0.0814	29	0.0494
30	64	0.0856	93	0.0851	230	0.0849	96	0.0829	183	0.0681	109	0.0806	30	0.0492
31	8	0.0850	193	0.0846	229	0.0843	230	0.0824	16	0.0676	207	0.0796	31	0.0492
32	81	0.0845	8	0.0841	231	0.0837	105	0.0816	79	0.0683	98	0.0796	32	0.0491
33	179	0.0839	77	0.0835	93	0.0832	93	0.0812	245	0.0678	177	0.0799	33	0.0489
34	93	0.0834	81	0.0830	11	0.0826	114	0.0809	108	0.0675	193	0.0797	34	0.0487
35	97	0.0829	179	0.0825	179	0.0822	193	0.0806	191	0.0673	128	0.0799	35	0.0487
36	65	0.0824	78	0.0820	232	0.0817	127	0.0807	178	0.0684	22	0.0787	36	0.0486
37	144	0.0819	10	0.0815	77	0.0812	10	0.0804	128	0.0687	37	0.0777	37	0.0486
38	10	0.0815	231	0.0811	143	0.0808	232	0.0799	62	0.0686	52	0.0769	38	0.0483
39	231	0.0811	2	0.0807	62	0.0804	83	0.0795	165	0.0681	67	0.0765	39	0.0478
40	230	0.0806	230	0.0803	195	0.0800	7	0.0790	98	0.0684	106	0.0758	40	0.0474
41	229	0.0803	229	0.0800	64	0.0797	76	0.0785	155	0.0677	110	0.0752	41	0.0471
42	2	0.0799	11	0.0796	1	0.0795	195	0.0782	235	0.0672	121	0.0746	42	0.0467
43	11	0.0795	97	0.0793	228	0.0791	228	0.0779	75	0.0669	157	0.0737	43	0.0463
44	77	0.0792	143	0.0789	81	0.0788	99	0.0775	122	0.0664	10	0.0735	44	0.0460
45	195	0.0788	3	0.0786	7	0.0785	78	0.0772	229	0.0665	36	0.0730	45	0.0461
46	62	0.0785	17	0.0783	147	0.0781	143	0.0769	111	0.0670	78	0.0727	46	0.0464

47	143	0.0782	83	0.0779	83	0.0778	147	0.0766	84	0.0668	108	0.0724	47	0.0470
48	3	0.0779	195	0.0777	97	0.0776	113	0.0766	9	0.0669	122	0.0718	48	0.0472
49	232	0.0776	232	0.0774	233	0.0772	234	0.0762	51	0.0667	228	0.0716	49	0.0472
50	17	0.0773	65	0.0771	78	0.0769	64	0.0760	147	0.0667	195	0.0715	50	0.0474
51	147	0.0770	144	0.0769	99	0.0766	3	0.0758	101	0.0664	47	0.0715	51	0.0476
52	78	0.0767	147	0.0766	210	0.0763	50	0.0756	179	0.0664	211	0.0711	52	0.0476
53	83	0.0764	99	0.0763	3	0.0761	174	0.0752	47	0.0666	4	0.0709	53	0.0475
54	7	0.0762	50	0.0760	105	0.0757	84	0.0748	157	0.0660	97	0.0708	54	0.0473
55	228	0.0759	7	0.0758	65	0.0756	152	0.0745	119	0.0658	107	0.0705	55	0.0471
56	50	0.0756	228	0.0755	191	0.0753	80	0.0745	18	0.0656	236	0.0699	56	0.0469
57	99	0.0754	105	0.0752	92	0.0750	103	0.0742	145	0.0661	104	0.0697	57	0.0467
58	233	0.0751	76	0.0749	234	0.0747	233	0.0740	255	0.0659	173	0.0692	58	0.0466
59	255	0.0748	92	0.0747	50	0.0745	92	0.0738	150	0.0657	221	0.0686	59	0.0466
60	92	0.0746	191	0.0744	67	0.0742	136	0.0734	12	0.0655	222	0.0681	60	0.0466
61	210	0.0743	233	0.0742	76	0.0740	81	0.0733	66	0.0656	103	0.0679	61	0.0467
62	4	0.0741	67	0.0740	4	0.0738	210	0.0731	177	0.0660	163	0.0680	62	0.0471
63	105	0.0739	210	0.0737	175	0.0735	175	0.0729	168	0.0658	5	0.0678	63	0.0476
64	67	0.0736	4	0.0735	103	0.0732	46	0.0726	76	0.0657	159	0.0676	64	0.0479
65	191	0.0734	84	0.0733	144	0.0731	4	0.0724	232	0.0657	136	0.0674	65	0.0480
66	76	0.0732	234	0.0730	2	0.0730	67	0.0722	112	0.0661	94	0.0672	66	0.0485
67	48	0.0729	152	0.0728	46	0.0727	245	0.0719	93	0.0661	119	0.0670	67	0.0487
68	234	0.0727	103	0.0726	152	0.0725	91	0.0717	23	0.0657	137	0.0667	68	0.0487
69	51	0.0725	175	0.0723	108	0.0722	61	0.0714	136	0.0656	152	0.0666	69	0.0486
70	109	0.0723	46	0.0721	174	0.0720	108	0.0712	5	0.0654	192	0.0664	70	0.0485
71	152	0.0721	51	0.0719	159	0.0718	51	0.0710	99	0.0654	206	0.0661	71	0.0483

72	175	0.0718	108	0.0717	84	0.0716	104	0.0708	246	0.0652	144	0.0660	72	0.0480
73	84	0.0716	48	0.0715	109	0.0714	159	0.0706	161	0.0656	210	0.0659	73	0.0479
74	103	0.0714	91	0.0713	51	0.0712	97	0.0705	64	0.0656	105	0.0659	74	0.0478
75	108	0.0712	159	0.0711	91	0.0710	121	0.0702	129	0.0657	235	0.0657	75	0.0479
76	46	0.0710	174	0.0709	107	0.0708	246	0.0700	237	0.0654	232	0.0657	76	0.0481
77	240	0.0708	109	0.0707	48	0.0706	150	0.0698	46	0.0653	83	0.0657	77	0.0483
78	192	0.0707	94	0.0705	136	0.0704	1	0.0697	210	0.0653	62	0.0657	78	0.0485
79	94	0.0705	61	0.0703	6	0.0702	167	0.0695	10	0.0653	92	0.0656	79	0.0491
80	159	0.0703	18	0.0701	12	0.0700	75	0.0694	159	0.0652	141	0.0653	80	0.0493
81	91	0.0701	107	0.0699	104	0.0697	6	0.0692	189	0.0651	153	0.0651	81	0.0494
82	174	0.0699	192	0.0697	246	0.0695	135	0.0690	230	0.0651	180	0.0649	82	0.0498
83	18	0.0697	136	0.0695	192	0.0694	107	0.0688	167	0.0650	247	0.0647	83	0.0500
84	107	0.0695	167	0.0693	135	0.0692	12	0.0686	164	0.0648	237	0.0644	84	0.0501
85	254	0.0693	104	0.0691	5	0.0690	207	0.0684	96	0.0650	171	0.0642	85	0.0500
86	167	0.0692	75	0.0690	121	0.0688	100	0.0682	78	0.0650	93	0.0642	86	0.0499
87	151	0.0690	6	0.0688	17	0.0687	65	0.0682	102	0.0649	189	0.0641	87	0.0497
88	136	0.0688	151	0.0686	207	0.0685	119	0.0680	234	0.0648	12	0.0640	88	0.0496
89	104	0.0686	5	0.0684	94	0.0684	109	0.0679	89	0.0646	246	0.0639	89	0.0496
90	16	0.0684	245	0.0682	106	0.0682	5	0.0677	193	0.0647	11	0.0639	90	0.0496
91	6	0.0683	12	0.0681	151	0.0680	151	0.0676	83	0.0647	14	0.0636	91	0.0497
92	12	0.0681	135	0.0679	227	0.0678	189	0.0674	2	0.0647	46	0.0636	92	0.0498
93	75	0.0680	246	0.0677	16	0.0677	106	0.0672	61	0.0645	115	0.0634	93	0.0500
94	5	0.0678	207	0.0676	75	0.0675	18	0.0671	81	0.0646	118	0.0632	94	0.0501
95	135	0.0676	121	0.0674	245	0.0674	48	0.0670	7	0.0646	151	0.0631	95	0.0504
96	188	0.0675	150	0.0673	102	0.0672	227	0.0668	127	0.0647	154	0.0629	96	0.0507

97	150	0.0673	106	0.0671	100	0.0670	153	0.0667	180	0.0645	156	0.0627	97	0.0508
98	207	0.0672	168	0.0670	188	0.0669	164	0.0665	175	0.0645	172	0.0624	98	0.0511
99	246	0.0670	16	0.0668	120	0.0667	2	0.0665	121	0.0644	238	0.0623	99	0.0513
100	183	0.0669	188	0.0667	211	0.0666	94	0.0664	109	0.0643	248	0.0621	100	0.0513
101	168	0.0667	153	0.0665	61	0.0664	188	0.0662	153	0.0642	15	0.0619	101	0.0513
102	166	0.0666	166	0.0664	189	0.0663	120	0.0661	114	0.0642	77	0.0619	102	0.0513
103	106	0.0664	120	0.0662	150	0.0661	68	0.0659	226	0.0640	61	0.0618	103	0.0514
104	61	0.0663	100	0.0661	119	0.0660	144	0.0659	80	0.0641	16	0.0618	104	0.0514
105	121	0.0661	119	0.0659	247	0.0658	168	0.0658	91	0.0640	102	0.0617	105	0.0516
106	68	0.0660	68	0.0658	167	0.0657	16	0.0657	17	0.0640	13	0.0615	106	0.0516
107	102	0.0659	183	0.0657	18	0.0656	102	0.0655	187	0.0638	66	0.0616	107	0.0516
108	149	0.0657	189	0.0655	153	0.0655	247	0.0654	149	0.0637	138	0.0615	108	0.0517
109	182	0.0656	227	0.0654	68	0.0654	90	0.0653	35	0.0635	158	0.0613	109	0.0518
110	100	0.0655	102	0.0653	31	0.0652	166	0.0651	182	0.0634	212	0.0612	110	0.0517
111	189	0.0653	90	0.0652	90	0.0651	192	0.0651	211	0.0633	50	0.0612	111	0.0520
112	245	0.0652	164	0.0650	115	0.0650	101	0.0649	144	0.0633	175	0.0611	112	0.0523
113	227	0.0651	149	0.0649	235	0.0648	235	0.0648	85	0.0631	135	0.0611	113	0.0525
114	119	0.0649	247	0.0648	166	0.0647	149	0.0647	158	0.0630	205	0.0609	114	0.0526
115	120	0.0648	211	0.0646	149	0.0646	211	0.0645	253	0.0628	123	0.0607	115	0.0526
116	153	0.0647	255	0.0646	101	0.0644	137	0.0644	107	0.0628	167	0.0607	116	0.0525
117	208	0.0646	115	0.0644	168	0.0643	17	0.0643	247	0.0626	164	0.0606	117	0.0523
118	164	0.0644	182	0.0643	131	0.0642	238	0.0642	67	0.0626	234	0.0606	118	0.0522
119	90	0.0643	60	0.0642	137	0.0641	60	0.0640	104	0.0626	20	0.0604	119	0.0522
120	211	0.0642	31	0.0641	209	0.0639	115	0.0639	139	0.0623	35	0.0603	120	0.0522
121	49	0.0641	101	0.0639	164	0.0638	183	0.0638	4	0.0623	89	0.0602	121	0.0523

122	247	0.0640	256	0.0639	190	0.0637	89	0.0637	151	0.0623	120	0.0602	122	0.0522
123	115	0.0639	235	0.0638	89	0.0636	180	0.0636	48	0.0622	134	0.0601	123	0.0521
124	184	0.0637	137	0.0636	183	0.0635	190	0.0634	131	0.0621	176	0.0599	124	0.0521
125	59	0.0636	89	0.0635	165	0.0634	158	0.0633	65	0.0621	223	0.0598	125	0.0520
126	31	0.0635	165	0.0634	158	0.0632	45	0.0632	192	0.0621	227	0.0597	126	0.0519
127	101	0.0634	190	0.0633	60	0.0631	165	0.0630	113	0.0621	3	0.0597	127	0.0521
128	165	0.0633	131	0.0632	36	0.0630	35	0.0629	94	0.0621	147	0.0598	128	0.0523
129	209	0.0632	209	0.0630	182	0.0629	122	0.0628	205	0.0619	181	0.0597	129	0.0525
130	110	0.0631	59	0.0629	45	0.0628	131	0.0627	227	0.0618	174	0.0596	130	0.0528
131	131	0.0629	208	0.0628	59	0.0627	182	0.0626	240	0.0618	31	0.0596	131	0.0528
132	235	0.0628	180	0.0627	13	0.0625	31	0.0625	137	0.0616	45	0.0595	132	0.0526
133	89	0.0627	35	0.0626	122	0.0624	237	0.0623	124	0.0615	68	0.0594	133	0.0525
134	60	0.0626	36	0.0625	134	0.0623	36	0.0622	184	0.0614	142	0.0593	134	0.0524
135	33	0.0625	158	0.0624	180	0.0622	123	0.0621	60	0.0613	188	0.0592	135	0.0525
136	180	0.0624	45	0.0622	35	0.0621	59	0.0620	37	0.0612	131	0.0592	136	0.0525
137	137	0.0623	122	0.0621	37	0.0619	134	0.0619	173	0.0611	48	0.0591	137	0.0525
138	190	0.0622	134	0.0620	208	0.0619	37	0.0618	92	0.0611	21	0.0590	138	0.0524
139	35	0.0621	184	0.0619	226	0.0617	173	0.0616	148	0.0610	169	0.0589	139	0.0523
140	36	0.0619	254	0.0618	69	0.0616	209	0.0615	196	0.0608	170	0.0588	140	0.0522
141	187	0.0618	37	0.0617	110	0.0615	138	0.0614	13	0.0607	203	0.0587	141	0.0521
142	241	0.0617	238	0.0616	248	0.0614	69	0.0613	118	0.0606	51	0.0586	142	0.0520
143	32	0.0616	110	0.0615	118	0.0613	226	0.0611	69	0.0605	29	0.0584	143	0.0521
144	45	0.0615	13	0.0614	187	0.0612	19	0.0610	166	0.0604	30	0.0583	144	0.0521
145	69	0.0614	69	0.0613	124	0.0611	13	0.0609	207	0.0604	34	0.0582	145	0.0524
146	37	0.0613	19	0.0612	52	0.0609	124	0.0608	115	0.0603	38	0.0581	146	0.0528

147	52	0.0612	52	0.0611	184	0.0609	118	0.0607	106	0.0603	165	0.0580	147	0.0529
148	74	0.0611	123	0.0609	238	0.0608	23	0.0606	244	0.0601	208	0.0580	148	0.0529
149	253	0.0610	187	0.0608	173	0.0606	157	0.0604	97	0.0602	168	0.0579	149	0.0528
150	134	0.0609	226	0.0607	15	0.0605	236	0.0603	120	0.0601	60	0.0578	150	0.0528
151	158	0.0608	118	0.0606	123	0.0604	244	0.0602	141	0.0600	23	0.0577	151	0.0528
152	181	0.0607	124	0.0605	237	0.0603	52	0.0601	254	0.0599	245	0.0577	152	0.0529
153	122	0.0605	240	0.0604	19	0.0602	187	0.0600	190	0.0598	124	0.0576	153	0.0529
154	148	0.0604	74	0.0603	74	0.0601	208	0.0599	154	0.0597	53	0.0575	154	0.0528
155	13	0.0603	173	0.0602	236	0.0600	154	0.0598	212	0.0596	44	0.0574	155	0.0527
156	169	0.0602	49	0.0602	23	0.0599	184	0.0597	22	0.0595	88	0.0573	156	0.0526
157	160	0.0601	237	0.0600	85	0.0598	85	0.0596	236	0.0593	140	0.0572	157	0.0525
158	19	0.0600	148	0.0599	22	0.0596	248	0.0595	59	0.0593	233	0.0572	158	0.0525
159	238	0.0599	248	0.0598	148	0.0595	22	0.0594	6	0.0593	239	0.0571	159	0.0526
160	124	0.0598	169	0.0597	212	0.0594	74	0.0593	134	0.0592	204	0.0570	160	0.0525
161	58	0.0597	138	0.0596	32	0.0594	148	0.0593	90	0.0591	2	0.0570	161	0.0527
162	118	0.0596	236	0.0595	138	0.0592	155	0.0591	123	0.0590	64	0.0571	162	0.0532
163	199	0.0595	85	0.0594	154	0.0591	206	0.0590	172	0.0589	125	0.0570	163	0.0533
164	176	0.0594	15	0.0593	30	0.0590	169	0.0589	52	0.0588	196	0.0569	164	0.0533
165	85	0.0593	32	0.0592	206	0.0589	205	0.0588	169	0.0587	139	0.0568	165	0.0532
166	34	0.0592	181	0.0591	21	0.0588	139	0.0587	138	0.0586	81	0.0568	166	0.0532
167	226	0.0591	34	0.0590	254	0.0588	255	0.0587	33	0.0586	24	0.0567	167	0.0532
168	185	0.0591	154	0.0589	181	0.0587	181	0.0586	248	0.0585	99	0.0567	168	0.0532
169	123	0.0590	206	0.0588	204	0.0586	222	0.0585	209	0.0584	100	0.0567	169	0.0531
170	173	0.0589	23	0.0587	14	0.0585	212	0.0584	36	0.0584	116	0.0566	170	0.0531
171	53	0.0588	22	0.0586	222	0.0584	34	0.0583	204	0.0583	133	0.0564	171	0.0530

172	15	0.0587	212	0.0585	157	0.0582	254	0.0582	19	0.0582	150	0.0564	172	0.0529
173	248	0.0586	160	0.0585	169	0.0582	172	0.0581	74	0.0581	160	0.0564	173	0.0529
174	125	0.0585	53	0.0584	142	0.0581	15	0.0581	208	0.0581	187	0.0563	174	0.0529
175	237	0.0584	222	0.0583	34	0.0580	110	0.0580	181	0.0580	74	0.0562	175	0.0529
176	154	0.0583	244	0.0582	176	0.0579	21	0.0579	31	0.0579	226	0.0562	176	0.0529
177	196	0.0582	176	0.0581	155	0.0578	53	0.0578	110	0.0579	65	0.0562	177	0.0531
178	212	0.0581	157	0.0580	172	0.0577	142	0.0577	24	0.0577	55	0.0561	178	0.0534
179	224	0.0580	204	0.0579	53	0.0576	204	0.0576	140	0.0576	85	0.0560	179	0.0535
180	236	0.0579	30	0.0578	249	0.0575	196	0.0575	88	0.0575	59	0.0560	180	0.0535
181	22	0.0578	21	0.0577	49	0.0575	30	0.0574	239	0.0575	148	0.0559	181	0.0534
182	204	0.0577	142	0.0576	205	0.0574	160	0.0574	220	0.0573	126	0.0558	182	0.0534
183	186	0.0576	205	0.0575	125	0.0573	88	0.0573	142	0.0572	166	0.0558	183	0.0534
184	239	0.0576	155	0.0574	255	0.0573	239	0.0572	156	0.0571	87	0.0557	184	0.0533
185	23	0.0575	172	0.0573	196	0.0572	249	0.0571	170	0.0570	183	0.0557	185	0.0533
186	170	0.0574	58	0.0572	244	0.0571	140	0.0570	206	0.0569	69	0.0556	186	0.0532
187	138	0.0573	14	0.0571	160	0.0570	20	0.0569	21	0.0568	249	0.0555	187	0.0532
188	198	0.0572	196	0.0571	88	0.0569	58	0.0568	58	0.0568	91	0.0556	188	0.0532
189	206	0.0571	125	0.0570	20	0.0569	49	0.0568	249	0.0567	190	0.0555	189	0.0532
190	172	0.0570	33	0.0569	171	0.0568	141	0.0567	256	0.0567	209	0.0555	190	0.0531
191	126	0.0569	249	0.0568	141	0.0567	32	0.0566	49	0.0566	90	0.0555	191	0.0532
192	88	0.0569	139	0.0567	58	0.0566	253	0.0566	199	0.0565	76	0.0555	192	0.0532
193	155	0.0568	88	0.0566	139	0.0565	33	0.0565	53	0.0565	220	0.0554	193	0.0533
194	171	0.0567	253	0.0566	170	0.0564	176	0.0564	160	0.0564	70	0.0553	194	0.0535
195	30	0.0566	20	0.0565	240	0.0564	170	0.0564	30	0.0563	185	0.0552	195	0.0536
196	142	0.0565	239	0.0564	140	0.0563	14	0.0563	116	0.0562	26	0.0551	196	0.0535

197	21	0.0564	171	0.0563	185	0.0562	156	0.0562	34	0.0562	40	0.0550	197	0.0534
198	244	0.0564	141	0.0562	199	0.0562	125	0.0561	15	0.0561	84	0.0550	198	0.0534
199	222	0.0563	170	0.0562	126	0.0561	221	0.0560	186	0.0560	186	0.0549	199	0.0533
200	157	0.0562	199	0.0561	38	0.0560	240	0.0560	125	0.0559	218	0.0548	200	0.0532
201	205	0.0561	185	0.0560	253	0.0559	171	0.0559	222	0.0559	219	0.0546	201	0.0531
202	14	0.0560	140	0.0559	156	0.0558	199	0.0558	86	0.0558	182	0.0546	202	0.0530
203	20	0.0559	223	0.0559	223	0.0557	223	0.0557	171	0.0557	75	0.0546	203	0.0529
204	249	0.0559	126	0.0558	239	0.0557	38	0.0557	198	0.0556	244	0.0545	204	0.0529
205	225	0.0558	38	0.0557	225	0.0556	24	0.0556	225	0.0555	19	0.0545	205	0.0528
206	54	0.0557	225	0.0556	186	0.0555	185	0.0555	185	0.0555	184	0.0545	206	0.0527
207	55	0.0556	156	0.0555	24	0.0554	225	0.0554	20	0.0554	58	0.0544	207	0.0527
208	70	0.0555	186	0.0555	70	0.0553	220	0.0553	197	0.0553	73	0.0543	208	0.0527
209	223	0.0554	221	0.0553	221	0.0552	126	0.0552	221	0.0552	117	0.0542	209	0.0527
210	56	0.0554	54	0.0553	54	0.0552	116	0.0551	32	0.0552	201	0.0541	210	0.0528
211	141	0.0553	198	0.0552	203	0.0551	186	0.0551	38	0.0551	217	0.0540	211	0.0528
212	73	0.0552	24	0.0551	33	0.0550	198	0.0550	87	0.0550	1	0.0540	212	0.0527
213	38	0.0551	203	0.0550	198	0.0550	54	0.0549	126	0.0549	202	0.0539	213	0.0526
214	139	0.0550	70	0.0549	116	0.0549	73	0.0548	117	0.0548	155	0.0539	214	0.0525
215	140	0.0549	73	0.0549	55	0.0548	203	0.0548	176	0.0548	54	0.0538	215	0.0524
216	203	0.0549	116	0.0548	73	0.0547	197	0.0547	73	0.0547	43	0.0537	216	0.0523
217	57	0.0548	86	0.0547	86	0.0546	86	0.0546	241	0.0546	49	0.0537	217	0.0521
218	242	0.0547	55	0.0546	44	0.0545	70	0.0545	203	0.0545	225	0.0536	218	0.0520
219	197	0.0546	220	0.0545	87	0.0544	87	0.0544	14	0.0545	243	0.0535	219	0.0519
220	200	0.0545	44	0.0544	220	0.0543	256	0.0544	25	0.0544	240	0.0535	220	0.0518
221	156	0.0544	87	0.0543	41	0.0543	117	0.0543	252	0.0543	253	0.0534	221	0.0518

222	71	0.0544	197	0.0542	117	0.0542	44	0.0542	54	0.0542	18	0.0534	222	0.0517
223	86	0.0543	117	0.0542	197	0.0541	243	0.0541	223	0.0541	25	0.0533	223	0.0516
224	252	0.0542	41	0.0541	133	0.0540	57	0.0541	71	0.0541	32	0.0533	224	0.0516
225	41	0.0541	133	0.0540	71	0.0539	25	0.0539	213	0.0540	42	0.0532	225	0.0515
226	116	0.0540	71	0.0539	25	0.0538	133	0.0539	44	0.0539	86	0.0531	226	0.0515
227	87	0.0539	57	0.0538	256	0.0538	55	0.0538	243	0.0538	199	0.0531	227	0.0515
228	44	0.0539	56	0.0537	56	0.0537	252	0.0537	133	0.0537	39	0.0530	228	0.0516
229	24	0.0538	224	0.0537	57	0.0536	241	0.0537	57	0.0536	198	0.0529	229	0.0517
230	221	0.0537	25	0.0536	250	0.0535	224	0.0536	70	0.0535	213	0.0528	230	0.0517
231	243	0.0536	250	0.0535	213	0.0534	71	0.0535	27	0.0534	17	0.0528	231	0.0518
232	117	0.0535	243	0.0534	224	0.0534	213	0.0534	224	0.0534	71	0.0527	232	0.0519
233	133	0.0534	241	0.0534	243	0.0533	41	0.0533	55	0.0533	132	0.0526	233	0.0520
234	220	0.0533	213	0.0533	252	0.0532	250	0.0532	219	0.0532	56	0.0526	234	0.0520
235	40	0.0532	252	0.0532	39	0.0531	56	0.0532	29	0.0531	216	0.0525	235	0.0520
236	251	0.0531	39	0.0531	40	0.0530	219	0.0531	250	0.0530	250	0.0524	236	0.0520
237	202	0.0530	40	0.0530	132	0.0529	242	0.0530	41	0.0529	215	0.0523	237	0.0519
238	42	0.0530	132	0.0529	251	0.0529	39	0.0529	242	0.0528	254	0.0523	238	0.0519
239	72	0.0529	251	0.0528	219	0.0527	27	0.0528	132	0.0527	33	0.0522	239	0.0518
240	250	0.0528	242	0.0527	26	0.0526	251	0.0527	56	0.0527	214	0.0521	240	0.0518
241	43	0.0527	200	0.0527	200	0.0526	132	0.0526	216	0.0526	41	0.0521	241	0.0518
242	213	0.0526	219	0.0526	242	0.0525	29	0.0525	26	0.0525	57	0.0520	242	0.0517
243	201	0.0525	26	0.0525	27	0.0524	26	0.0524	251	0.0524	101	0.0520	243	0.0516
244	39	0.0524	43	0.0524	43	0.0523	200	0.0523	218	0.0523	27	0.0519	244	0.0516
245	25	0.0523	42	0.0523	42	0.0522	218	0.0522	39	0.0522	200	0.0518	245	0.0516
246	132	0.0522	27	0.0522	241	0.0522	43	0.0521	28	0.0521	224	0.0518	246	0.0516

247	216	0.0521	218	0.0521	218	0.0521	40	0.0520	200	0.0520	197	0.0517	247	0.0516
248	217	0.0520	29	0.0520	29	0.0520	216	0.0520	43	0.0519	251	0.0516	248	0.0516
249	218	0.0519	217	0.0519	217	0.0519	72	0.0519	217	0.0518	149	0.0516	249	0.0515
250	215	0.0518	216	0.0518	216	0.0518	217	0.0518	72	0.0517	252	0.0515	250	0.0514
251	219	0.0517	72	0.0517	72	0.0517	42	0.0517	215	0.0516	242	0.0515	251	0.0514
252	214	0.0516	202	0.0516	202	0.0516	28	0.0516	40	0.0516	28	0.0514	252	0.0513
253	26	0.0515	28	0.0515	28	0.0515	215	0.0515	214	0.0515	241	0.0513	253	0.0513
254	27	0.0514	201	0.0514	201	0.0514	202	0.0514	42	0.0514	255	0.0513	254	0.0513
255	29	0.0513	215	0.0513	215	0.0513	214	0.0513	202	0.0513	72	0.0512	255	0.0512
256	28	0.0512	214	0.0512	214	0.0512	201	0.0512	201	0.0512	256	0.0512	256	0.0512
	AVG	0.0677	AVG	0.0675	AVG	0.0673	AVG	0.0661	AVG	0.0618	AVG	0.0641	AVG	0.0523

Table 7.13: Correlations between Error Rates and AD Means

	Datasets	ITID (ILIA1)	ITID (ILIA2)	AVG
1	Diabetes	-0.548740	-0.598152	-0.605679
2	Cancer	-0.540790	-0.786446	-0.666998
3	Glass	-0.352446	-0.334416	-0.359648
4	Thyroid	-0.079300	-0.219379	-0.129115
5	Semeion	-0.834786	-0.421807	-0.628297

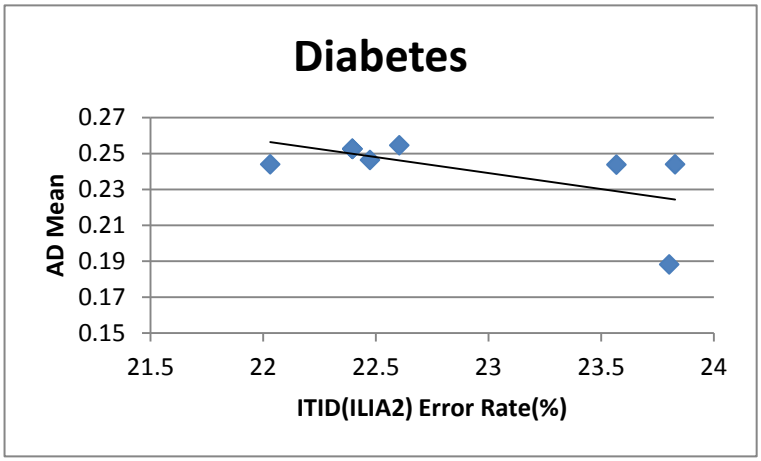


Figure 7.6: Correlations between Error Rates and AD Means (Diabetes)

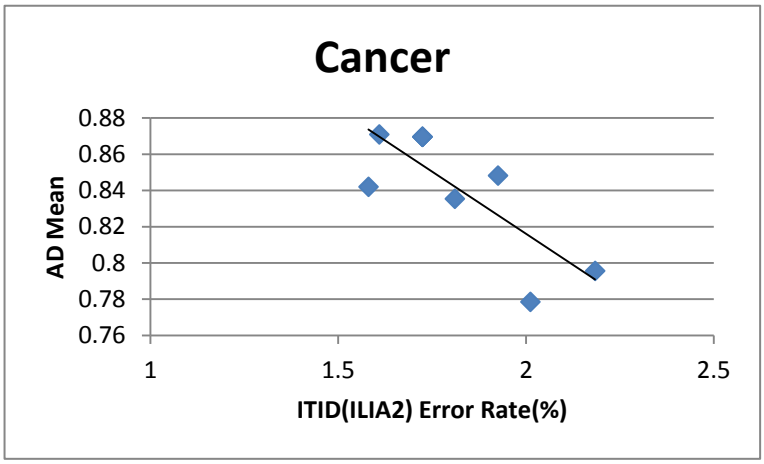


Figure 7.7: Correlations between Error Rates and AD Means (Cancer)

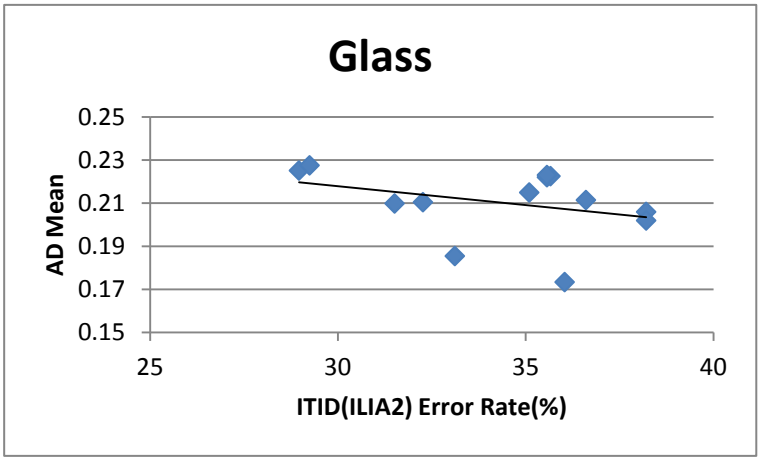


Figure 7.8: Correlations between Error Rates and AD Means (Glass)

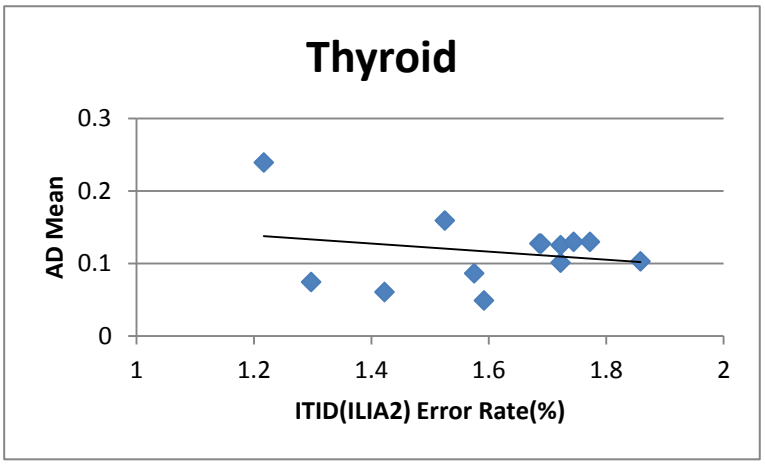


Figure 7.9: Correlations between Error Rates and AD Means (Thyroid)

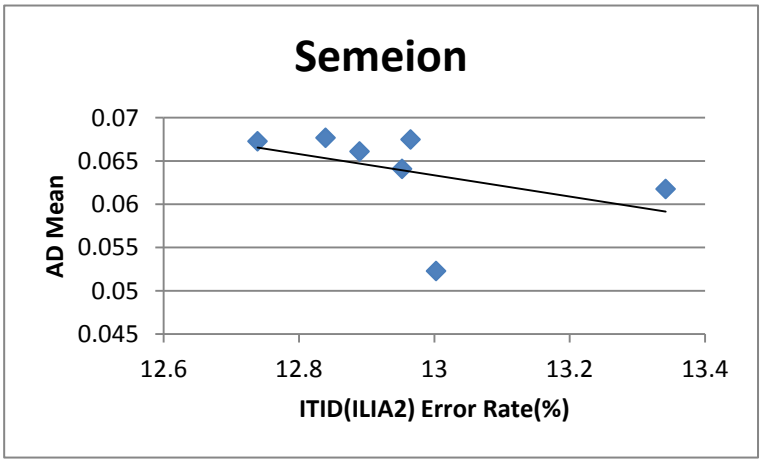


Figure 7.10: Correlations between Error Rates and AD Means (Semeion)

7.4 Comparisons with State of the Art Results

In this section, the best classification results derived from the feature ordering approaches for IAL, which has been presented in this thesis, are compared with some state of the art results from other researchers all over the world.

Here, the best results in this study are selected from two aspects. One is the best result from the whole study including all the approaches developed in this thesis; and the other is the best result from AD, the most applicable and well-performed feature ordering approach in this study.

All the state of the art results are selected from the papers published in the recent two years. Moreover, it is required that all the approaches employed in these paper should be neural networks or some varieties of neural networks.

The result comparison is shown in Table 7.14. According to the table, results derived from this study performs well comparing to some other results. For example, in Diabetes, best results in this study and AD get the third and the fourth place; in Cancer, they obtain the fourth and the fifth place, respectively; furthermore, result from SD and AD also exhibit the third and the fourth good performance in Glass; in Thyroid and Semeion, this study overcomes all the other studies, and obtains the lowest error rates.

In addition, comparing the prediction mechanism of this study with the others show in the table, the neural networks in our prediction system is purer than some of the others. Apart from neural networks and corresponding preprocessing works, few other techniques are employed in this study like PSO and SVM, while in some other studies, neural networks often work together with some other methodologies. For example, in line 15, pPCA is carried out with both neural networks and Principal Component Analysis (PCA). Such a combination of neural networks and some other approaches is able to enhance the pattern recognition performance. Therefore, in the future, feature ordering approaches developed in this study should fuse with some other integrated machine learning techniques. It seems such a method can bring better results for pattern recognition.

Table 7.14: Result Comparison with State of the Art Results

	Approaches	Error Rates (%)				
		Diabetes	**Cancer**	**Glass**	**Thyroid**	**Semeion**
1	**Best Result and its approach in this study**	**21.33** Integrated Correlation-based	**1.55** AD	**28.96** SD	**1.22** AD	**12.74** FS
2	**AD Best Result**	**21.61**	**1.55**	**29.25**	**1.22**	**12.84**
3	Bootstrapping [67]	-	-	-	-	18.63
4	SVDD [68]	-	-	-	-	13.0
5	CCA [69]	-	-	-	-	19.1
6	CMAC NN [70]	25.57	3.94	35.22	-	-
7	SGHS [71]	-	3.00	35.02	6.59	-
8	ACFNNA [72]	20.87	1.08	-	1.58	-
9	ISO-FLANN [73]	20.37	2.64	-	-	-
10	RBFN [74]	27.30	4.54	30.99	-	-
11	PGFN [74]	23.96	6.25	31.16	-	-
12	AT_CasPer [75]	22.88	1.29	28.42	1.53	-
13	A_CasPer [75]	23.14	1.15	27.68	1.67	-
14	Layered_CasPer [75]	23.91	2.13	30.38	4.33	-
15	pPCA [76]	25.00	1.78	32.07	5.87	-
16	MDEGL [77]	24.84	4.34	37.13	-	-

7.5 Summary

This chapter compared all the feature ordering approaches and their classification results with each other. According to the experimental results and comparing with each other, AD-based feature ordering exhibited very well and very stable. Therefore, such a feature ordering method can be regarded as the optimum feature ordering approach. The reason why AD-based feature ordering is chosen as the optimum feature ordering approach is that classification errors can be forecasted by AD. According to the analysis about AD values calculated for different feature orderings, the classification error rates are negative correlated to the AD values. Therefore the greater AD values of a ordering are, the lower error rates it will get. Comparing classification results derived by our best feature ordering and AD-based feature orderings with some state of the art results which are also based on neural networks, the performance of our approaches is acceptable, and the AD-based feature ordering method MAMFO can also be definitely treated as a preferred approach for IAL feature ordering. However, because the methodology is different, the state of the art results only can be treated as a reference to this study. In the feature, it is likely to obtain better classification results, if IAL feature ordering approaches can be successfully applied in some sophisticated pattern classification approaches.

Chapter 8

Feature Ordering with Feature Selection

8.1 Overview

Along with the fast development of computer hardware, more and more enormous data problems can be solved with the aid of computers. At present, as a necessity, computers are employed to cope with these data problems with many widely used techniques, especially those in the area of machine learning, pattern recognition and data mining. Enormous data problems have three types: the first one is enormous in the pattern or instance size, the second one is enormous in the size of features, and the last one is enormous in both pattern and feature size. Feature selection, as one of the inevitable preprocesses of feature reduction, concentrates on the aspect of feature size big data, and this is also the aspect which is focused on by IAL.

Feature selection is a commonly used technique to solve problems with high-dimensional features. Intuitively, the larger the feature number, the easier is the classification. However, a byproduct of large feature number is interference. Thus, some researchers believe that, for high-dimensional problems with a large number of features, “the less is more” should be a strategy in feature number reduction process [78]. In fact, the meaning of “the less is more” is to obtain the most useful parts and get rid of the interference part for prediction in pattern recognition. According to this, a significant step before machine learning, pattern recognition and data mining is to select the most useful feature subset for the datasets.

In previous studies of IAL, approaches of feature selection have already been studied with

contribution-based wrapper approaches [5, 7] for a long time. Experimental results confirm that feature selection not only can work very well with IAL, but also can cope with both classification and regression problems. However, because there was no filter feature ordering research in previous studies, some filter feature ordering approaches can not only save preprocessing time, but also enhance the accuracy of final results. Therefore, filter feature selection with sorted ordering gradually becomes a necessity for researchers.

In this chapter, feature selection for IAL is implemented with filter feature ordering approaches. In the Section 8.2, a model of dynamic feature selection is presented, and experiments and corresponding analysis is illustrated in Section 8.3.

8.2 Dynamic Feature Selection

As it is mentioned before, IAL is a "divide-and-conquer" machine learning strategy which gradually trains input features one after another. Thus feature ordering is required for the solution about almost all problems solved by IAL. Because of feature ordering, when features are introduced to the IAL systems, the dimension of feature space is increasing. The growing feature space impacts on the training process. Therefore, in IAL, if some subsets of features should be selected, it is necessary to consider from the aspect of growing process of feature space, and the influence brought by the growing. Such a dynamic process is quite different from the stable feature selection process in conventional approaches.

Naturally, because AD is accurate, effective and stable, feature selection in this study of IAL will be carried out based on feature ordering derived by AD and Maximum Mean Criterion. Features will be sequentially imported in training. In each step, an error rate will be obtained. According to these error rates, if the classification error rate of the later imported feature is greater than that of previous one, then the later imported feature should be discarded from the dataset. After that, useful features will be selected eventually. This feature selection technique has been presented in [5]. Here, a new metric called Reduction Rate is employed to decide which feature should be discarded, and which one should not. The equation of Reduction Rate of ($i+1$)-th feature is:

$$Reduction\_Rate_{f_{i+1}} = \frac{\left(Error\_Rate_{f_{i+1}} - \min\left(Error\_Rate_{f_1}, \cdots, Error\_Rate_{f_i}\right)\right)}{\min\left(Error\_Rate_{f_1}, \cdots, Error\_Rate_{f_i}\right)} \times 100\% \quad (8.1)$$

Obviously, if the result of $\text{Reduction\_Rate}_{f_{i+1}}$ is not larger than 0, then feature *i+1* can be employed, otherwise, it should be discarded.

Feature selection in IAL is dynamic. During the process, features are gradually introduced into the training in some orders, it is difficult to rank features or select feature subsets simultaneously in one batch. Therefore, features should be gradually ranked or selected into a subset according to the regulation of IAL, and such a dynamic process can improve the performance of IAL. Figure 8.1 illustrates the feature selection process in IAL.

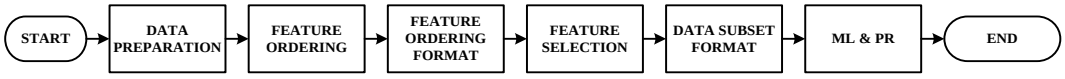


Figure 8.1: Feature selection process in IAL[5].

8.3 Experiments

In this study, experiments on feature selection based on AD feature ordering by neural IAL approaches is launched. Four datasets from UCI Machine Learning Repository, Diabetes, Cancer, Glass and Thyroid are selected for this study. Moreover, all the experimental results are compared with SD and contribution-based IAL feature selection approaches [2, 5]. The results derived by these two kinds of feature ordering with feature selection are employed for comparison, because AD is improved from SD, and contribution- based feature selection is a representative which has been studied in previous research.

8.3.1 Diabetes

In Diabetes, the feature selection procedure is shown in Table 8.1, where the feature ordering derived by AD, error rates obtained in each step, and the reduction rate of error rates have been shown for the demonstration of selection procedure. During the procedure, if the reduction rate is positive, the corresponding feature should be discarded. In the meanwhile, the prediction system

[5] ML&PR denotes the step of machine learning and pattern recognition.

should return to the previous step, and restart the prediction process with the next feature.

Figure 8.2 shows the change of error rates presented in Table 8.1, and Table 8.2 compares feature selection results with those without selection that have been shaded in the table. The results show that, in Diabetes, AD feature ordering without feature selection obtains the lowest classification error rates.

Table 8.1: Feature Selection Procedure with AD Feature Ordering (Diabetes)

	Feature Ordering	Error Rates in Each Step (%)	Reduction Rate (%)	Discard or Not
1	2	23.56770		
2	6	22.81251	-3.20436	
3	7	22.89064	0.34251	X
4	8	22.50000	-1.36988	
5	5	21.74479	-3.35649	
6	4	21.74479	0.00000	
7	1	21.61459	-0.59879	
8	3	21.61458	-0.00002	

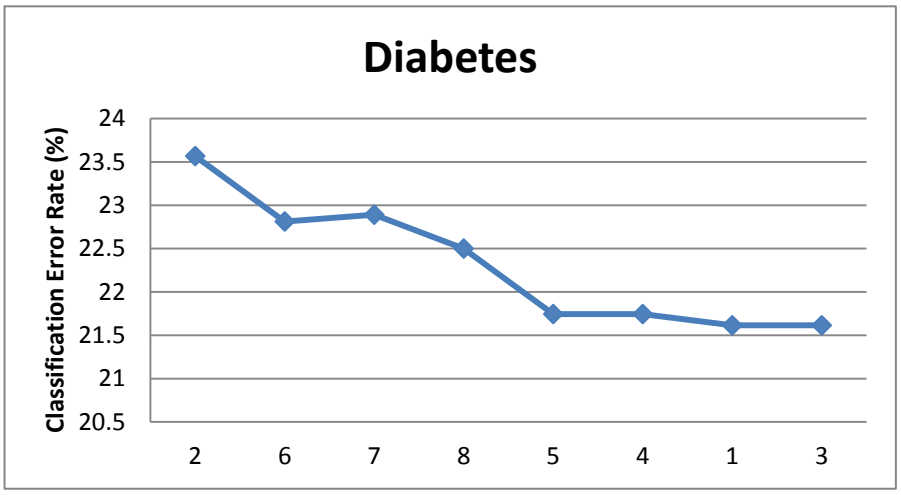


Figure 8.2: Error Rates Change with ITID and AD Feature Ordering (Diabetes)

Table 8.2: Feature Selection Result Comparison (Diabetes)

	Approach	Selected Features with Ordering	Classification Error (%)		
			ITID (ILIA1)	ITID (ILIA2)	AVG
1	AD	2-6-8-5-4-1-3	21.64063	22.78645	22.21354
2	SD	2-6-8-7-1-5-3	21.66667	22.76042	22.21355
3	Contribution-based	2-6	22.81251	23.69790	23.25520
4	AD	2-6-7-8-5-4-1-3	21.61458	22.60416	22.10937
5	Conventional Method	No Feature Ordering	23.93229		

8.3.2 Cancer

In Cancer, the feature selection procedure is presented in Table 8.3. During the feature selection procedure based on AD feature ordering, features 5, 1, 8, 4 and 9 are discarded. Figure 8.3 shows the change of error rates presented in Table 8.3, and Table 8.4 compares feature selection results with those without selection that have been shaded in the table. The results show that, in this dataset, feature selection exhibits a very good performance, where both SD and AD feature orderings with feature selection obtain the lowest classification error rates.

Table 8.3: Feature Selection Procedure with AD Feature Ordering (Cancer)

	Feature Ordering	Error Rates in Each Step (%)	Reduction Rate (%)	Discard or Not
1	3	6.89655		
2	2	4.56896	-33.75000	
3	6	3.01724	-33.96220	
4	7	0.60345	-80.00000	
5	5	1.35058	123.80970	X
6	1	1.60920	166.66680	X
7	8	1.58046	161.90500	X
8	4	1.63793	171.42870	X
9	9	1.55173	157.14300	X

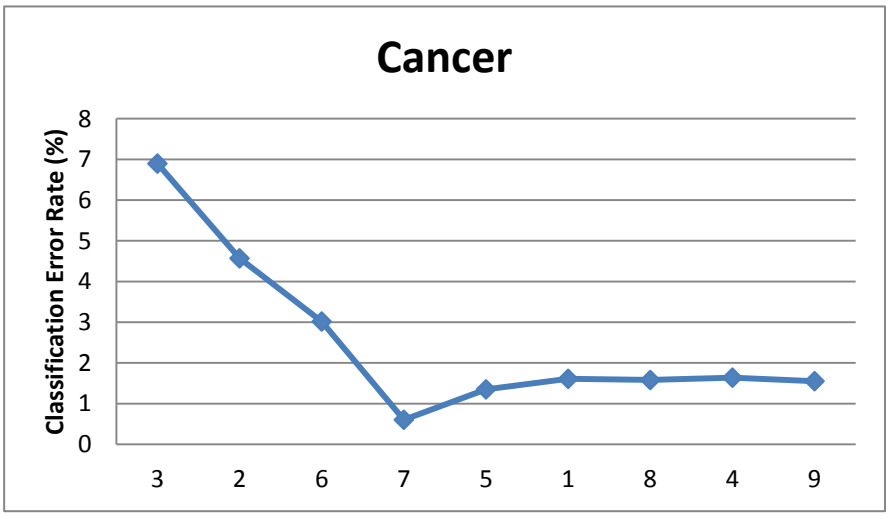


Figure 8.3: Error Rates Change with ITID and AD Feature Ordering (Cancer)

Table 8.4: Feature Selection Result Comparison (Cancer)

	Approach	Selected Features with Ordering	Classification Error (%)		
			ITID (ILIA1)	ITID (ILIA2)	AVG
1	AD	3-2-6-7	0.60345	0.89081	0.74713
2	SD	3-2-6-7	0.60345	0.89081	0.74713
3	Contribution-based	2-3-5-6-1	1.20690	1.35058	1.27874
4	AD	3-2-6-7-5-1-8-4-9	1.55173	1.60920	1.58046
5	Conventional Method	No Feature Ordering	1.86782		

8.3.3 Glass

In Glass, the feature selection procedure is presented in Table 8.5. During the feature selection procedure based on AD feature ordering, features 5 and 9 are discarded. Figure 8.4 shows the change of error rates presented in Table 8.5, and Table 8.6 compares feature selection results with those without selection that have been shaded in the table. The results show that, in this dataset, feature selection exhibits a very good performance with ITID (ILIA1), where AD feature ordering with feature selection obtains the lowest classification error rates. However, the situation is quite different in ITID (ILIA2), where AD feature ordering without feature selection gets the lowest classification error rates.

Table 8.5: Feature Selection Procedure with AD Feature Ordering (Glass)

	Feature Ordering	Error Rates in Each Step (%)	Reduction Rate (%)	Discard or Not
1	3	61.88684		
2	8	58.11324	-6.09758	
3	2	58.11324	0.00000	
4	4	32.92455	-43.34420	
5	6	31.13210	-5.44413	
6	7	31.03776	-0.30303	
7	1	30.56606	-1.51975	
8	5	31.13210	1.85184	X
9	9	34.33964	12.34565	X

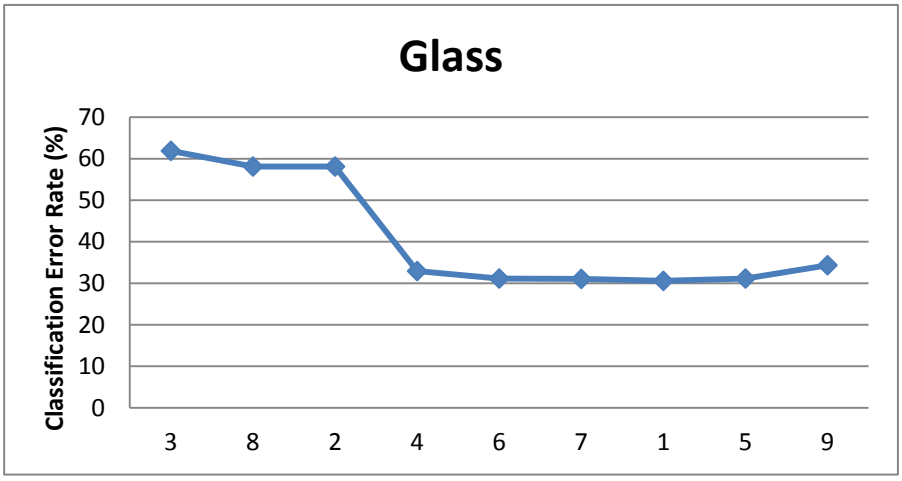


Figure 8.4: Error Rates Change with ITID and AD Feature Ordering (Glass)

Table 8.6: Feature Selection Result Comparison (Glass)

	Approach	Selected Features with Ordering	Classification Error (%)		
			ITID (ILIA1)	ITID (ILIA2)	AVG
1	AD	3-8-2-4-6-7-1	30.56606	33.20758	31.88682
2	SD	3-8-4-6	31.41512	33.96230	32.68871
3	Contribution-based	4-8-3-2	31.69814	34.33962	33.01888
4	AD	3-8-2-4-6-7-1-5-9	34.33964	29.24530	31.79247
5	Conventional Method	No Feature Ordering	41.22641		

8.3.4 Thyroid

In Thyroid, the feature selection procedure is presented in Table 8.7. Based on AD feature ordering, in the feature selection procedure, feature 21, 17, 13, 7, 8, 3, 1 and 2 are selected. Figure 8.5 shows the fluctuation of error rates presented in Table 8.7, and Table 8.6 compares feature selection results with those without selection that have been shaded in the table. The results in Thyroid are similar to those in Glass, where feature selection exhibits a very good performance with ITID (ILIA1), and AD feature ordering with feature selection obtains the lowest classification error rates. Nevertheless, in ITID (ILIA2), AD feature ordering without feature selection gets the lowest classification error rates.

Table 8.7: Feature Selection Procedure with AD Feature Ordering (Thyroid)

	Feature Ordering	Error Rates in Each Step (%)	Reduction Rate (%)	Discard or Not
1	21	6.43611		
2	18	6.47222	0.56108	X
3	19	6.48333	0.73372	X
4	15	6.48333	0.73372	X
5	20	6.47500	0.60424	X
6	17	3.02778	-52.95640	
7	13	2.97500	-1.74311	
8	7	2.96111	-0.46689	
9	12	2.98611	0.84435	X
10	5	2.97500	0.46908	X
11	4	2.99444	1.12574	X
12	8	2.93611	-0.84426	
13	3	1.91111	-34.91010	
14	9	2.06667	8.13962	X
15	16	2.06389	7.99431	X
16	6	2.07500	8.57570	X
17	14	2.09722	9.73842	X
18	1	1.87500	-1.88953	
19	11	1.89167	0.88891	X
20	10	1.88055	0.29629	X
21	2	1.52500	-18.66660	

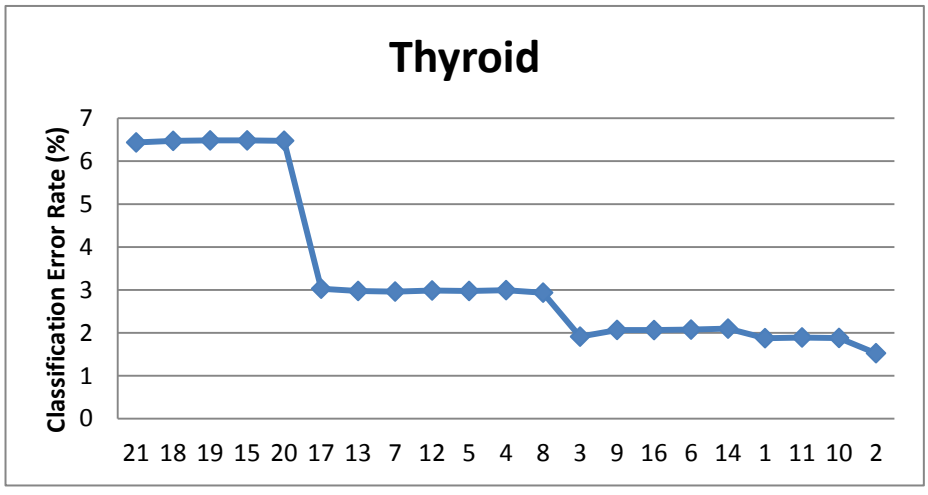


Figure 8.5: Error Rates Change with ITID and AD Feature Ordering (Thyroid)

Table 8.8: Feature Selection Result Comparison (Thyroid)

	Approach	Selected Features with Ordering	Classification Error (%)		
			ITID (ILIA1)	ITID (ILIA2)	AVG
1	AD	21-17-13-7-8-3-1-2	1.43889	1.37222	1.40556
2	SD	21-19-17-18-3-13-20-10-8	1.86667	1.45555	1.66111
3	Contribution-based	17-21-19-18-1	3.60833	3.03333	3.32083
4	AD	21-18-19-15-20-17-13-7-12-5-4-8-3-9-16-6-14-1-11-10-2	1.52500	1.21667	1.37083
5	Conventional Method	No Feature Ordering	1.86389		

8.4 Summary

According to the experimental results presented in this study, it seems that feature selection does not always perform well in IAL with ITID. Based on AD and the MMDC, some optimum feature orderings have been obtained. According to these optimum feature orderings, useful feature subsets are selected. Experimental results showed that, comparing with the other approaches presented in the previous studies, IAL using such optimum feature orderings and feature selection sometimes can achieve lowest classification error rates, but mostly in ITID (ILIA1), because error rates obtained in each round of ITID are in ILIA1 style. In ILIA2, the neural networks improve with a new layer, which makes the prediction structure much different from that of original ILIA1. Therefore, ITID (ILIA1) in feature selection is more likely to obtain better results than ITID (ILIA2). Moreover, compared with results derived by AD feature ordering without feature selection, it seems that it is difficult to obtain better performance with feature selection in ITID (ILIA2). Consequently, if feature selection is necessary to be employed in IAL, ITID (ILIA1) has more probabilities to exhibit better performance than ITID (ILIA2).

Chapter 9

Feature Ordering for Regression

9.1 Overview

In statistics and pattern recognition, regression analysis is widely used in prediction and forecasting. It estimates the values of some variables depending on some other variables. Therefore, regression analysis is a technique to evaluate the relations among variables. Moreover, as an important technique in pattern recognition and statistics, machine learning is often employed along with regression analysis. Actually, apart from classification, regression is another application aspect of IAL.

The research of regression analysis in IAL has been studied for a long time, which formally starts at the same time of the research on classification. Firstly, regression analysis is carried out in ILIA algorithms, where it is found applicable for IAL [21]. Secondly, regression analysis is employed with feature ordering and output partition in ITID [2]. Both of these studies confirm that the regression analysis based on IAL with a proper feature ordering can obtain better performance than conventional batch training approaches. Besides this, regression analysis is also implemented in feature selection [5] and feature grouping[6]. All these studies are based on neural networks, and all the calculations on feature ordering and partition are based on features' single contribution to outputs.

It is necessary to mention that relationships existing among variables do not indicate that regression analysis can be employed to infer causal relationships between variables. There is a phrase in science and statistics called “Correlation does not imply causation”, which emphasizes that one correlation between two variables does not necessarily imply that one causes the other

[79]. Anyway, it is manifested that the correlation can be used in regression analysis, although sometimes the relations between two variables are unexplainable.

However, the same as classification in previous studies of IAL, regression analysis is also often carried out according to some feature ordering. More specifically, feature orderings for regression analysis in previous studies are usually derived from contribution-based approaches. Nevertheless, correlations among variables are seldom employed for pattern value estimation and prediction. Thus in this chapter, correlations among features will be sorted for feature ordering, and experimental results will be compared and analyzed. Moreover, in some real world datasets, because some features have more than one attributes, it is necessary to calculate the relations among the feature with one or more attributes. In Section 9.2, the method about how to sort features with one or more attributes is presented, and in Section 9.3 experiments on IAL regression analysis is demonstrated.

9.2 Ordered Feature Grouping

Occasionally, one feature has more than one attributes. For example, as it is mentioned in Chapter 2, colour consists of red, green, and blue. Sometimes, this three kinds of colour attributes can be employed simultaneously, while sometimes these three colours should be used independently. This will depend on the problems or the actual situation. Anyway, in pattern recognition process, when such a multiple attribute feature appears, there are two different ways to cope with the problems. One is that these attributes can be packaged as a united feature, while the other approach is that these attributes can be divided and each one can be treated as a single feature.

In the first approach, such a packaged feature can be treated as a feature partition or attribute group. In the phase of preprocess, such a grouped feature should be used as a single feature. After the calculation like feature ordering and selection, the original attributes will be employed instead of integrative feature in prediction, because integrative feature has less information than the original multiple attributes after attribute integration, where the original multiple attributes are closer to the original problem by themselves. Therefore, in the preprocess, the multiple attributes

in the grouped feature should be integrated, so that it can be easily used in feature ordering and feature selection. Thus. the strategy to integrate multiple attributes into one feature is very crucial to the prediction.

The second approach to tackle with multiple attribute approach is to make all attributes to be single features. Namely, they are not grouped as a united feature any more. Every attributes will be treated as a single feature in all the calculation step like preprocessing and prediction.

These two kinds of approaches will be used for different types of data. For different problems, the data structure of different datasets may be different. Some features in the problems may be categorical, while others may be continuous. Actually, according to different types of data, different approach should be chosen. In this study, feature with multiple attributes is integrated into one value when the data is categorical. If the data is continuous, multiple attributes will be used as single features with the second approach.

Obviously, the second approach is much easier than the first one. In this study, the first approach is carried out according to the weight of each category.

Assuming that **x** is the integrated feature value, $x_1, \cdots, x_n$ are the categorical attributes of a feature, and $\omega_0, \omega_1, \omega_2, \cdots, \omega_n$ are the weights of each attributes in the feature, thus the integrated feature value can be estimated as:

$$\mathbf{x} = \omega_1 x_1 + \omega_2 x_2 + \cdots + \omega_7 x_7 + \omega_0 \tag{9.1}$$

Usually, if the feature **x** has some categorical attributes for each pattern, only one category will the feature belongs to. Mathematically, only one attribute will be marked 1 and all the rest attributes will be 0. Therefore, the formula Eq.(9.1) can be simplified according to the attribute values of different patterns. The simplified formula is shown in Eq.(9.2):

$$\begin{cases} \mathbf{x} = \omega_1 x_1 + \ \omega_0 (\text{if } x_1 \neq 0) \\ \qquad\qquad \vdots \\ \mathbf{x} = \omega_n x_n + \omega_0 \ (\text{if } x_n \neq 0) \end{cases} \tag{9.2}$$

where

$$\omega_i = \frac{\text{the number of pattens in category i}}{\text{total number of patterns}} \tag{9.3}$$

To simplify the calculation, ω_0 in Eq.(9.2) can be assumed as 0, thus Eq.(9.2) can be written as:

$$\begin{cases} \mathbf{x} = \omega_1 x_1 (\text{if } x_1 \neq 0) \\ \qquad\qquad \vdots \\ \mathbf{x} = \omega_n x_n \ (\text{if } x_n \neq 0) \end{cases} \tag{9.4}$$

therefore, feature with multiple attributes can be calculated.

9.3 Experiments

In this study, experiments are carried out based on UCI machine learning repository. Four datasets have been employed. They are Flare, Building, Hearta and Housing. Two kinds of correlation based approaches have been employed for comparison, one is with feature group, and the other is merely based on each single attributes. The former uses formulae to integrate multiple attributes into one feature, while the latter treats every attribute as a single feature and directly employs each attributes for estimation. Obviously, all the attributes in one feature stay together in the grouped correlation based approach, while they are distributed in the single correlation based approach. Anyway, all the features or attributes are sorted according to some features orderings or grouped feature orderings. Furthermore, there are two kinds of correlations, positive and negative. However, the correlative level is irrelevant to the correlation sign. Thus, all the correlation values are presented with their mathematical absolute value. In this study, both of the results derived by Correlation-based feature ordering approaches will be compared with the results derived by other approaches like Contribution-based, Original Ordering, and Conventional Method.

9.3.1 Flare

Flare has 10 features, where the first, the second and the third feature have seven, six and four attributes, respectively. Therefore, when all the attributes are employed as single feature in preprocessing and prediction, the total number of features is 24. Moreover, all the values of the first and the last feature are 0. Thus these two features have little discrimination ability on prediction, and in correlation based feature ordering, it is impossible to compute the correlation values of this two features when they are independently used as a single feature. Therefore, in Grouped Correlation-based feature ordering for regression, there are 9 features, while in Single Correlation-based feature ordering, there are 22 features. Table 9.1 shows the correlations for grouped attribute feature ordering of Flare, while Table 9.2 shows the correlations for single attribute feature ordering. Table 9.3 and Figure 9.1 is the comparison of the final regression testing error derived from these two correlation feature ordering approaches with those results derived by some other feature ordering approaches. Final results show that, in Flare, all the

results obtained in ITID (ILIA2) have much lower testing error rates than the conventional batch training mode. In addition, the correlation-based feature ordering cannot exhibit better performance in ITID (ILIA1) than conventional method, whereas contribution-based and original feature ordering is able to.

Table 9.1: Correlations for Grouped Attribute Feature Ordering (Flare)

Feature	1,...,7	8,...,13	14,.,17	18	19	20	21	22	23	24
1,...,7	1.0000									
8,...,13	0.4622	1.0000								
14,...,17	0.0300	0.0579	1.0000							
18	0.2576	0.1185	0.3119	1.0000						
19	0.0283	0.1181	0.0233	0.0198	1.0000					
20	0.1505	0.0939	0.2084	0.4482	0.0637	1.0000				
21	0.2363	0.0767	0.2842	0.2713	0.2296	0.2042	1.0000			
22	0.4487	0.2712	0.1024	0.1557	0.0172	0.0836	0.0872	1.0000		
23	0.2431	0.2391	0.3678	0.3333	0.0574	0.2906	0.1815	0.0569	1.0000	
24	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Output 1	0.2331	0.1219	0.2285	0.2742	0.0797	0.1526	0.1754	0.0930	0.0926	NaN
Output 2	0.0751	0.0739	0.2084	0.0872	0.0746	0.0297	0.0947	0.0546	0.2517	NaN
Output 3	0.0284	0.1107	0.1648	0.1336	0.0578	0.0927	0.0728	0.0228	0.4010	NaN
AVG	0.1122	0.1021	0.2006	0.1650	0.0707	0.0917	0.1143	0.0568	0.2484	NaN
Ordering	5	6	2	3	8	7	4	9	1	NaN

Table 9.2: Correlations for Single Attribute Feature Ordering (Flare)

F[6]	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
2	1.0000																					
3	0.1961	1.0000																				
4	0.2020	0.2496	1.0000																			
5	0.1210	0.1495	0.1540	1.0000																		
6	0.0787	0.0972	0.1001	0.0600	1.0000																	
7	0.2869	0.3545	0.3651	0.2187	0.1422	1.0000																
8	0.9920	0.1946	0.2004	0.1200	0.0780	0.2846	1.0000															
9	0.1903	0.2640	0.0210	0.1277	0.0689	0.0020	0.1888	1.0000														
10	0.3216	0.0507	0.1398	0.1164	0.0846	0.4119	0.3300	0.3989	1.0000													
11	0.2043	0.1010	0.2669	0.2763	0.1176	0.2037	0.2027	0.2450	0.4283	1.0000												
12	0.0522	0.0278	0.0058	0.0398	0.1274	0.0285	0.0518	0.0626	0.1094	0.0672	1.0000											
13	0.0846	0.0813	0.0984	0.2012	0.1524	0.1334	0.0839	0.1014	0.1774	0.1089	0.0278	1.0000										
14	0.2869	0.3545	0.3651	0.2187	0.1422	1.0000	0.2846	0.0020	0.4119	0.2037	0.0285	0.1334	1.0000									
15	0.3666	0.4706	0.0470	0.0683	0.1088	0.6155	0.3624	0.1290	0.1810	0.1390	0.0354	0.1628	0.6155	1.0000								
16	0.0802	0.1205	0.3772	0.1639	0.2670	0.3651	0.0779	0.1226	0.1965	0.3586	0.0664	0.1670	0.3651	0.4334	1.0000							
17	0.0723	0.0894	0.0126	0.4057	0.0765	0.1307	0.0717	0.0867	0.1516	0.1173	0.0238	0.4344	0.1307	0.1551	0.0920	1.0000						
18	0.1288	0.0464	0.1857	0.2364	0.1836	0.2380	0.1272	0.1685	0.0825	0.2309	0.0571	0.3110	0.2380	0.1025	0.2619	0.3292	1.0000					
19	0.0099	0.0298	0.1441	0.0090	0.0798	0.1173	0.0045	0.0460	0.1109	0.0384	0.0822	0.0439	0.1173	0.0370	0.0753	0.0398	0.0198	1.0000				
20	0.0850	0.0025	0.0808	0.1362	0.1183	0.1322	0.0843	0.1019	0.0543	0.1158	0.0280	0.2547	0.1322	0.0487	0.0682	0.3371	0.4482	0.0637	1.0000			
21	0.1956	0.1027	0.1475	0.2827	0.2468	0.1624	0.1924	0.1942	0.0505	0.2812	0.0444	0.2276	0.1624	0.1464	0.2905	0.1831	0.2713	0.2296	0.2042	1.0000		
22	0.1562	0.1930	0.1988	0.1190	0.0774	0.5444	0.1549	0.1304	0.3011	0.1059	0.0772	0.0561	0.5444	0.3351	0.1988	0.0712	0.1557	0.0172	0.0836	0.0872	1.0000	
23	0.0578	0.0715	0.0409	0.2407	0.2491	0.1045	0.0574	0.0693	0.1212	0.0095	0.0190	0.5537	0.1045	0.1241	0.0082	0.6496	0.3333	0.0574	0.2906	0.1815	0.0569	1.0000

[6] Attribute 1 and 24 are NaN attributes, thus they are not appeared in the table.

Out 1	0.1028	0.0036	0.1204	0.2370	0.1214	0.2125	0.1013	0.0663	0.1114	0.1905	0.0133	0.1947	0.2125	0.0819	0.2818	0.1566	0.2742	0.0797	0.1526	0.1754	0.0930	0.0926
Out 2	0.0554	0.0490	0.1403	0.0968	0.0132	0.1002	0.0550	0.0466	0.0534	0.0809	0.0182	0.1607	0.1002	0.0876	0.1211	0.2387	0.0872	0.0746	0.0297	0.0947	0.0546	0.2517
Out 3	0.0232	0.0287	0.0667	0.0521	0.0115	0.0419	0.0230	0.0278	0.0486	0.0299	0.0076	0.2741	0.0419	0.0498	0.0295	0.3207	0.1336	0.0578	0.0927	0.0728	0.0228	0.4010
AVG	0.0605	0.0271	0.1091	0.1286	0.0487	0.1182	0.0598	0.0469	0.0711	0.1004	0.0130	0.2098	0.1182	0.0731	0.1441	0.2387	0.1650	0.0707	0.0917	0.1143	0.0568	0.2484
FO	16	21	10	6	19	7	17	20	14	11	22	3	8	13	5	2	4	15	12	9	18	1

Table 9.3: Regression Result Comparison (Flare)

	Approach	**Feature Ordering**	**Testing Error (%)**		
			ITID (ILIA1)	**ITID (ILIA2)**	**AVG**
1	Grouped Correlation-based	23-(14,15,16,17)-18-21-(1,2,3,4,5,6,7)-(8,9,10,11,12,13)-20-19-22[7]	0.59421	0.52991	0.56206
2	Single Correlation-based	23-17-13-18-16-5-7-14-21-4-11-20-15-10-19-2-8-22-6-9-3-12[8]	0.59369	0.53260	0.56314
3	Contribution-based	(1,2,3,4,5,6,7)-(8,9,10,11,12,13)-(14,15,16,17)-18-21-23-22-20-19-24	0.52911	0.53627	0.53269
4	Original Ordering	(1,2,3,4,5,6,7)-(8,9,10,11,12,13)-(14,15,16,17)-18-19-20-21-22-23-24	0.52580	0.53116	0.52848
5	Conventional Method	No Feature Ordering	0.55000		

[7] The attribute 24 is an NaN attribute, so it is discarded.

[8] Both attribute 1 and 24 are NaN attributes, so they are discarded.

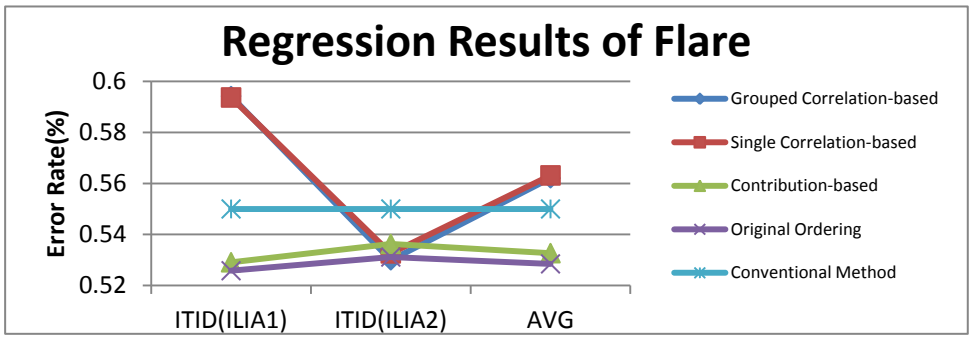


Figure 9.1: Regression Results Comparison (Flare)

9.3.2 Building

The Building dataset has 8 features, where the first feature has seven categorical attributes. Thus, if all the attributes are also regarded as single features, the total number of feature is 14. Tables 9.4 and 9.5 individually present the correlation values among features with grouped attributes and single attribute. Corresponding feature orderings are obtained according to these correlation values. Regression results are compared with different approaches and these are shown in Table 9.6. Figure 9.2 compares results in graph. Obviously, all the results derived by ITID (ILIA1) are not better than the conventional batch training method. On the contrary, in the approaches using ITID (ILIA2), except Single Correlation-based feature ordering, all the other feature orderings produce lower error rates as compared to the conventional batch training method.

Table 9.4: Correlations for Grouped Attribute Feature Ordering (Building)

Feature	1,...,7	8	9	10	11	12	13	14
1,...,7	1.0000							
8	0.0021	1.0000						
9	0.0023	0.1256	1.0000					
10	0.0015	0.1445	0.8664	1.0000				
11	0.0435	0.1065	0.3037	0.3925	1.0000			
12	0.0773	0.0214	0.2334	0.2988	0.3622	1.0000		
13	0.0130	0.6969	0.2125	0.3945	0.3613	0.2327	1.0000	
14	0.1119	0.1594	0.1903	0.2540	0.0115	0.1467	0.2335	1.0000
Output 1	0.2764	0.5336	0.3391	0.4017	0.1381	0.2347	0.5519	0.2648
Output 2	0.0529	0.1518	0.1502	0.1961	0.7999	0.5708	0.2702	0.0129
Output 3	0.0315	0.0383	0.1954	0.2404	0.8618	0.4184	0.2275	0.0531
AVG	0.0422	0.0951	0.1728	0.2182	0.8309	0.4946	0.2489	0.0330
Ordering	7	5	6	4	1	2	3	8

Table 9.5: Correlations for Single Attribute Feature Ordering (Building)

Feature	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	1.0000													
2	0.1735	1.0000												
3	0.1735	0.1741	1.0000											
4	0.1715	0.1721	0.1721	1.0000										
5	0.1655	0.1662	0.1662	0.1643	1.0000									
6	0.1655	0.1662	0.1662	0.1643	0.1586	1.0000								
7	0.1655	0.1662	0.1662	0.1643	0.1586	0.1586	1.0000							
8	0.0028	0.0015	0.0015	0.0043	0.0014	0.0014	0.0014	1.0000						
9	0.0052	0.0009	0.0009	0.0096	0.0009	0.0009	0.0009	0.1256	1.0000					
10	0.0035	0.0008	0.0008	0.0073	0.0008	0.0008	0.0008	0.1445	0.8664	1.0000				
11	0.0781	0.0056	0.0728	0.0922	0.0806	0.0181	0.1823	0.1065	0.3037	0.3925	1.0000			
12	0.0820	0.0427	0.0847	0.0997	0.0353	0.0076	0.1924	0.0214	0.2334	0.2988	0.3622	1.0000		
13	0.0154	0.0233	0.0007	0.0182	0.0227	0.0368	0.0417	0.6969	0.2125	0.3945	0.3613	0.2327	1.0000	
14	0.0729	0.0062	0.0158	0.0711	0.0078	0.1338	0.0449	0.1594	0.1903	0.2540	0.0115	0.1467	0.2335	1.0000
Output 1	0.0752	0.3077	0.3291	0.1320	0.1596	0.1572	0.1288	0.5336	0.3391	0.4017	0.1381	0.2347	0.5519	0.2648
Output 2	0.0251	0.1041	0.0431	0.0946	0.0852	0.1036	0.1077	0.1518	0.1502	0.1961	0.7999	0.5708	0.2702	0.0129
Output 3	0.0500	0.0486	0.0435	0.0872	0.1156	0.0580	0.1162	0.0383	0.1954	0.2404	0.8618	0.4184	0.2275	0.0531
AVG	0.0501	0.1535	0.1385	0.1046	0.1201	0.1062	0.1176	0.2412	0.2283	0.2794	0.5999	0.4080	0.3499	0.1103
Ordering	14	7	8	13	9	12	10	5	6	4	1	2	3	11

Table 9.6: Regression Result Comparison (Building)

	Approach	Feature Ordering	Testing Error (%)		
			ITID (ILIA1)	ITID (ILIA2)	AVG
1	Grouped Correlation-based	11-12-13-10-8-9-(1,2,3,4,5,6,7)-14	1.00303	0.77954	0.89128
2	Single Correlation-based	11-12-13-10-8-9-2-3-5-7-14-6-4-1	0.97301	0.99915	0.98608
3	Contribution-based	11-12-13-9-14-10-8-(1,2,3,4,5,6,7)	0.99377	0.88031	0.93704
4	Original Ordering	(1-2-3-4-5-6-7)-8-9-10-11-12-13-14	1.04815	0.82405	0.93610
5	Conventional Method	No Feature Ordering	0.93966		

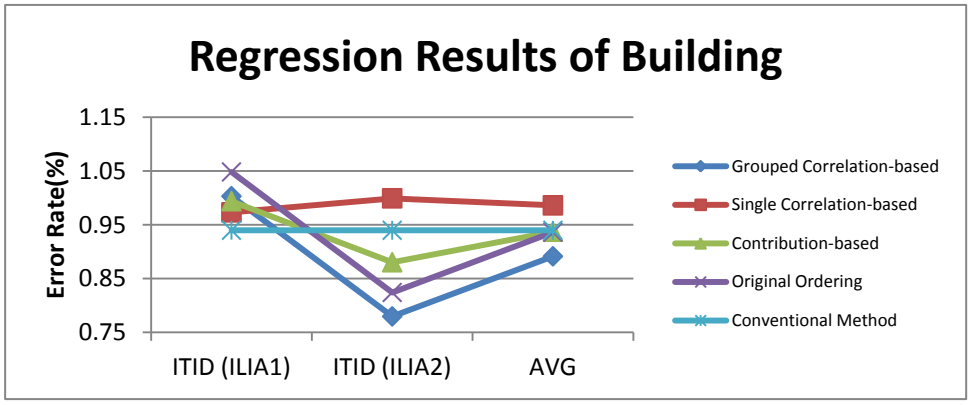


Figure 9.2: Regression Results Comparison (Building)

9.3.3 Hearta

Hearta is the Heart dataset for function analogy. It is a very complicated dataset. Due to the fact that there are many missing values in this dataset, some features have one or more attributes, one of which is a symbol for missing value marking. Tables 9.7 and 9.8 demonstrate the correlation values of Grouped Attribute Feature Ordering and Single Attribute Feature Ordering, respectively. In the meanwhile, the feature orderings is obtained for IAL regression. Result comparison is illustrated in Table 9.9 and Figure 9.3, whereby both ITID (ILIA1) and ITID (ILIA2) exhibit better performance than conventional batch training method in terms of error rates.

Table 9.7: Correlations for Grouped Attribute Feature Ordering (Hearta)

Feature	1	2	3,...,7	8,9	10,11	12,...,14	15,...,18	19,20	21,...,23	24,25	26,...,29	30,31	32,...,35
1	1.0000												
2	0.0873	1.0000											
3,...,7	0.1223	0.1309	1.0000										
8,9	0.0819	0.0919	0.0463	1.0000									
10,11	0.0581	0.1634	0.1493	0.0529	1.0000								
12,...,14	0.1756	0.0821	0.0513	0.0524	0.1825	1.0000							
15,...,18	0.1776	0.0087	0.0000	0.0333	0.1235	0.0546	1.0000						
19,20	0.3432	0.1681	0.0936	0.7362	0.1056	0.0145	0.0360	1.0000					
21,...,23	0.1924	0.1398	0.4137	0.1276	0.0131	0.0039	0.0196	0.0830	1.0000				
24,25	0.1118	0.0645	0.1595	0.7267	0.0541	0.0454	0.0101	0.7221	0.2326	1.0000			
26,...,29	0.0962	0.0538	0.0737	0.2323	0.0431	0.0353	0.1686	0.3277	0.0825	0.3429	1.0000		
30,31	0.1463	0.1334	0.0335	0.1405	0.3323	0.0749	0.3066	0.2551	0.0559	0.2149	0.4971	1.0000	
32,...,35	0.0588	0.1226	0.0027	0.1924	0.0916	0.0273	0.2476	0.2566	0.0529	0.1576	0.4314	0.6820	1.0000
Output	0.3415	0.3315	0.4817	0.0073	0.2628	0.1634	0.0427	0.2415	0.4242	0.0916	0.0649	0.0097	0.0284
Ordering	3	4	1	13	5	7	10	6	2	8	9	12	11

Table 9.8: Correlations for Single Attribute Feature Ordering (Hearta)

Feature[9]	1	2	3	4	5	6	8	9	10	11	13	14	15	16	17	18
1	1.0000															
2	0.0873	1.0000														
3	0.1090	0.0218	1.0000													
4	0.2176	0.1223	0.1041	1.0000												
5	0.0012	0.0592	0.1211	0.2637	1.0000											
6	0.1227	0.1364	0.2345	0.5105	0.5939	1.0000										
8	0.1053	0.0454	0.1237	0.0646	0.0492	0.0404	1.0000									
9	0.2119	0.1093	0.0065	0.0153	0.1161	0.0840	0.5526	1.0000								
10	0.0256	0.1717	0.0600	0.0956	0.0305	0.0736	0.0551	0.0188	1.0000							
11	0.0674	0.1490	0.0842	0.1141	0.0234	0.1440	0.0507	0.0465	0.7721	1.0000						
13	0.2119	0.0516	0.1272	0.0403	0.0234	0.0020	0.1022	0.0177	0.0999	0.0719	1.0000					
14	0.0406	0.0673	0.0693	0.0537	0.0866	0.1443	0.0355	0.0760	0.3640	0.5357	0.1251	1.0000				
15	0.1859	0.0221	0.0797	0.0739	0.0052	0.0286	0.0525	0.0425	0.0496	0.0398	0.1121	0.0545	1.0000			
16	0.0957	0.0367	0.0505	0.0042	0.0391	0.0510	0.0015	0.1082	0.1278	0.1369	0.0581	0.0435	0.5911	1.0000		
17	0.1437	0.0145	0.1216	0.0885	0.0373	0.0137	0.0660	0.0501	0.1863	0.1880	0.0842	0.1232	0.6292	0.2413	1.0000	
18	0.0649	0.0364	0.1438	0.0315	0.0366	0.0046	0.0056	0.0158	0.0193	0.0490	0.0261	0.0938	0.0820	0.0315	0.0335	1.0000
19	0.4036	0.2008	0.0934	0.1853	0.0342	0.2123	0.2062	0.5259	0.1466	0.1907	0.0044	0.0598	0.0015	0.1552	0.1439	0.0226
20	0.2087	0.0993	0.0002	0.0017	0.0922	0.0769	0.5166	0.9349	0.0167	0.0140	0.0309	0.0710	0.0489	0.1033	0.0378	0.0148
22	0.1753	0.1312	0.1339	0.2920	0.1345	0.3975	0.2134	0.1303	0.0056	0.0190	0.0149	0.0184	0.0189	0.0872	0.0533	0.0471
23	0.2087	0.0993	0.0002	0.0017	0.0922	0.0769	0.5166	0.9349	0.0167	0.0140	0.0309	0.0710	0.0489	0.1033	0.0378	0.0148
24	0.0876	0.0319	0.0014	0.1832	0.1516	0.2720	0.3828	0.4855	0.0565	0.0812	0.0062	0.0095	0.0247	0.0802	0.1116	0.0270
25	0.2210	0.1156	0.0340	0.0003	0.0799	0.0823	0.4537	0.8376	0.0129	0.0355	0.0910	0.0470	0.1018	0.1909	0.0577	0.0165

[9] Feature 7, 12, and 21 are NaN features, so they do not appear in Single Attribute Feature Ordering.

26	0.0520	0.0961	0.0561	0.0373	0.0930	0.1315	0.0112	0.1296	0.0137	0.0003	0.0459	0.0632	0.0434	0.1100	0.1514	0.0433
27	0.1798	0.0417	0.0007	0.2250	0.1089	0.2679	0.1934	0.1690	0.0056	0.0381	0.0298	0.0321	0.0160	0.0291	0.0558	0.0520
28	0.1077	0.0899	0.0189	0.0861	0.0507	0.1180	0.0030	0.0279	0.0927	0.0538	0.0923	0.0913	0.0596	0.0201	0.0945	0.0181
29	0.1991	0.0063	0.0392	0.2476	0.0577	0.2253	0.1928	0.3070	0.0569	0.0686	0.0402	0.0729	0.0880	0.1397	0.2452	0.0252
30	0.1981	0.0260	0.0615	0.1074	0.0021	0.0596	0.0456	0.0876	0.1445	0.1655	0.0179	0.0902	0.0608	0.1739	0.2449	0.0241
31	0.1023	0.1570	0.1015	0.0675	0.0989	0.0740	0.0658	0.1759	0.2675	0.3580	0.0307	0.2168	0.1274	0.3376	0.4872	0.0485
32	0.0611	0.2768	0.1044	0.0454	0.1753	0.1573	0.0333	0.1304	0.0837	0.1190	0.0175	0.1184	0.0921	0.1922	0.3024	0.0359
33	0.0297	0.0866	0.0065	0.0400	0.0197	0.0506	0.0716	0.0575	0.0192	0.0226	0.1019	0.0905	0.0032	0.0341	0.0264	0.0158
34	0.1679	0.1618	0.0323	0.1281	0.0617	0.1657	0.0851	0.0968	0.0067	0.0161	0.0074	0.0844	0.0267	0.1281	0.1610	0.0333
35	0.0972	0.0631	0.0588	0.1592	0.0887	0.0239	0.1288	0.2134	0.0669	0.0767	0.0375	0.1263	0.1002	0.2489	0.3714	0.0641
Output	0.3415	0.3315	0.0477	0.3838	0.1928	0.4825	0.0982	0.0888	0.1908	0.2743	0.0887	0.1594	0.0663	0.0760	0.0007	0.0396
Ordering	5	7	30	3	14	1	19	21	15	10	22	18	25	24	32	31

(Continued on the next page)

(Continued from the previous page)

Feature	19	20	22	23	24	25	26	27	28	29	30	31	32	33	34	35
19	1.0000															
20	0.5583	1.0000														
22	0.2008	0.1598	1.0000													
23	0.5583	1.0000	0.1598	1.0000												
24	0.1751	0.5513	0.4312	0.5513	1.0000											
25	0.5052	0.8975	0.1584	0.8975	0.6143	1.0000										
26	0.3277	0.1211	0.1764	0.1211	0.0578	0.1350	1.0000									
27	0.0912	0.1764	0.3378	0.1764	0.4174	0.1966	0.4255	1.0000								
28	0.0251	0.0613	0.2389	0.0613	0.3073	0.0683	0.1478	0.2153	1.0000							
29	0.1848	0.3254	0.3236	0.3254	0.5498	0.3626	0.3723	0.5422	0.1883	1.0000						
30	0.0905	0.0819	0.0128	0.0819	0.2017	0.0912	0.1025	0.1755	0.0419	0.2516	1.0000					
31	0.3363	0.1645	0.0603	0.1645	0.1976	0.1832	0.3661	0.1632	0.0143	0.5054	0.4978	1.0000				
32	0.2716	0.1219	0.1685	0.1219	0.0054	0.1358	0.3128	0.0408	0.0469	0.2970	0.1266	0.5457	1.0000			
33	0.0166	0.0537	0.0117	0.0537	0.0195	0.0217	0.0608	0.0283	0.0475	0.0009	0.0426	0.0050	0.1304	1.0000		
34	0.0160	0.0875	0.1908	0.0875	0.1623	0.0796	0.0255	0.1940	0.0100	0.2192	0.2938	0.3342	0.2738	0.1207	1.0000	
35	0.2334	0.1970	0.0171	0.1970	0.1347	0.1682	0.2556	0.2030	0.0259	0.4260	0.3618	0.7291	0.5607	0.2471	0.5190	1.0000
Output	0.3815	0.0554	0.4162	0.0554	0.3061	0.0594	0.1903	0.3392	0.1638	0.2717	0.1990	0.0941	0.2528	0.0855	0.2797	0.0513
Ordering	4	27	2	28	8	26	16	6	17	11	13	20	12	23	9	29

Table 9.9: Regression Result Comparison (Hearta)

	Approach	Feature Ordering	Testing Error (%)		
			ITID (ILIA1)	ITID (ILIA2)	AVG
1	Grouped Correlation-based	(3,4,5,6,7)-(21,22,23)-1-2-(10,11)-(19,20)-(12,13,14)-(24,25)-(26,27,28,29)-(15,16,17,18)-(32,33,34,35)-(30,31)-(8,9)	4.38038	4.60504	4.49271
2	Single Correlation-based	6-22-4-19-1-27-2-24-34-11-29-32-30-5-10-26-28-14-8-31-9-13-33-16-15-25-20-23-35-3-18-17-7-12-21[10]	4.45005	4.58240	4.51622
3	Contribution-based	(3,4,5,6,7)-(21,22,23)-(24,25)-(19,20)-(10,11)-(32,33,34,35)-(26,27,28,29)-(30,31)-1-2-(12,13,14)-(15,16,17,18)-(8,9)	4.45479	4.60945	4.53212
4	Original Ordering	1-2-(3,4,5,6,7)-(8,9)-(10,11)-(12,13,14)-(15,16,17,18)-(19,20)-(21,22,23)-(24,25), (26,27,28,29) -(30,31)-(32,33,34,35)	4.52306	4.59911	4.56109
5	Conventional Method	No Feature Ordering	4.70893		

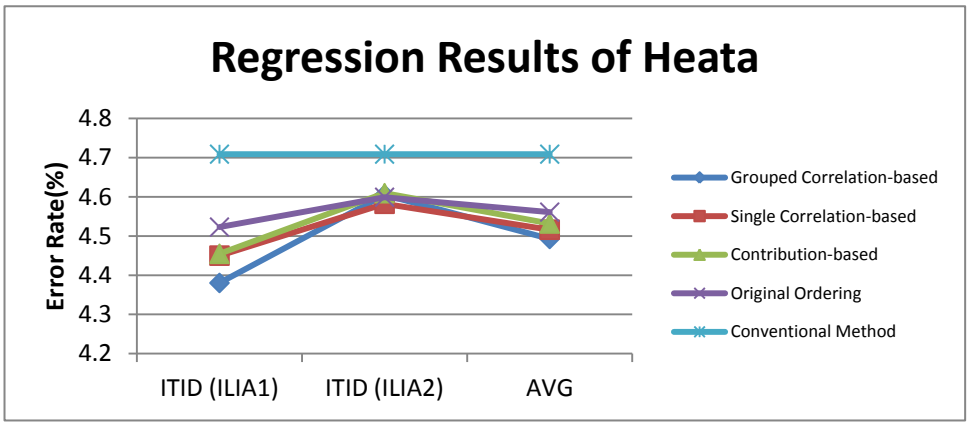


Figure 9.3: Regression Results Comparison (Hearta)

9.3.4 Housing

Compared with previous datasets, Housing is much simpler. All the features in this dataset has only one attribute. Table 9.10 shows feature correlations for ordering. In Table 9.11 and Figure 9.4, the results derived by feature correlations are compared with those derived by Contribution-based feature ordering, Original Ordering, and one-batch conventional method. It is

[10] Feature 7, 12, and 21 are NaN features.

manifested that the performance of Housing is very complex, where Contribution-based feature ordering produces lower error rates than conventional method in ITID (ILIA1) and both Correlation-based and Original Ordering exhibit better performance than conventional method in ITID (ILIA2).

Table 9.10: Correlations for Feature Ordering (Housing)

F	1	2	3	4	5	6	7	8	9	10	11	12	13
1	1.0000												
2	0.2005	1.0000											
3	0.4066	0.5338	1.0000										
4	0.0559	0.0427	0.0629	1.0000									
5	0.4210	0.5166	0.7637	0.0912	1.0000								
6	0.2192	0.3120	0.3917	0.0913	0.3022	1.0000							
7	0.3527	0.5695	0.6448	0.0865	0.7315	0.2403	1.0000						
8	0.3797	0.6644	0.7080	0.0992	0.7692	0.2052	0.7479	1.0000					
9	0.6255	0.3119	0.5951	0.0074	0.6114	0.2098	0.4560	0.4946	1.0000				
10	0.5828	0.3146	0.7208	0.0356	0.6680	0.2920	0.5065	0.5344	0.9102	1.0000			
11	0.2899	0.3917	0.3832	0.1215	0.1889	0.3555	0.2615	0.2325	0.4647	0.4609	1.0000		
12	0.3851	0.1755	0.3570	0.0488	0.3801	0.1281	0.2735	0.2915	0.4444	0.4418	0.1774	1.0000	
13	0.4556	0.4130	0.6038	0.0539	0.5909	0.6138	0.6023	0.4970	0.4887	0.5440	0.3740	0.3661	1.0000
Out	0.3883	0.3604	0.4837	0.1753	0.4273	0.6954	0.3770	0.2499	0.3816	0.4685	0.5078	0.3335	0.7377
FO	7	10	4	13	6	2	9	12	8	5	3	11	1

Table 9.11: Regression Result Comparison (Housing)

	Approach	Feature Ordering	Testing Error (%)		
			ITID (ILIA1)	ITID (ILIA2)	AVG
1	Simple Correlation-based	13-6-11-3-10-5-1-9-7-2-12-8-4	0.01117	0.01097	0.01107
2	Contribution-based	10-9-13-3-1-11-6-8-2-5-4-7-12	0.00669	0.01149	0.00909
3	Original Ordering	1-2-3-4-5-6-7-8-9-10-11-12-13	0.01192	0.01077	0.01135
4	Conventional Method	No Feature Ordering	0.01103		

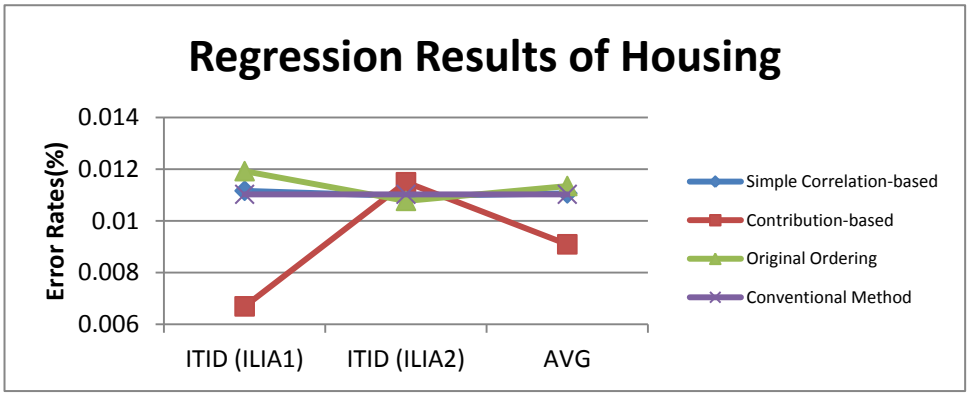


Figure 9.4: Regression Results Comparison (Housing)

9.4 Summary

IAL regression experiments presented in previous subsections show that the results derived by different feature orderings. Table 9.12 summarizes all the performance of ordering in IAL regression comparing with batch training conventional method. According to this table, in correlation-based feature ordering, ITID (ILIA1) failed in most of orderings for IAL regression, whereas ITID (ILIA2) is better than the conventional method using Grouped Correlation-based feature ordering. Furthermore, Contribution-based feature ordering failed in Building dataset using ITID (ILIA1) and in Housing dataset based on ITID (ILIA2). Therefore, the performance of Contribution-based feature ordering is unstable. In addition, although the original feature ordering performance is better than the conventional method with lower error rates in ITID (ILIA2), it is still uncertain that the original feature ordering can always exhibit better performance than the conventional method. The reason of that is the essence of original feature ordering is a random feature ordering, which merely depends on the format by the dataset donors. Accordingly, Grouped Correlation-based feature ordering can be treated as a more stable and applicable feature ordering approach for IAL regression problems. Table 9.13 shows the regression error rate reduction comparing with the error rate derived by conventional batch training approach.

Table 9.12: Regression Performance Compared with Conventional Method

	Approaches		Flare			Building			Hearta			Housing		
			ITID (ILIA1)	ITID (ILIA2)	AVG	ITID (ILIA1)	ITID (ILIA2)	AVG	ITID (ILIA1)	ITID (ILIA2)	AVG	ITID (ILIA1)	ITID (ILIA2)	AVG
1	Correlation-based	Grouped	X		X	X						X		X
		Single	X		X	X	X	X						
2	Contribution-based					X							X	
3	Original Ordering					X						X		X

Table 9.13: Regression Error Rate Reduction Compared with Conventional Method

	Approaches		Flare			Building			Hearta			Housing		
			ITID (ILIA1)	ITID (ILIA2)	AVG	ITID (ILIA1)	ITID (ILIA2)	AVG	ITID (ILIA1)	ITID (ILIA2)	AVG	ITID (ILIA1)	ITID (ILIA2)	AVG
1	Correlation-based	Grouped	8.04%	-3.65%	2.19%	6.74%	-17.04%	-5.15%	-6.98%	-2.21%	-4.59%	1.27%	-0.54%	0.36%
		Single	7.94%	-3.16%	2.39%	3.55%	6.33%	4.94%	-5.50%	-2.69%	-4.09%			
2	Contribution-based		-3.80%	-2.50%	-3.15%	5.76%	-6.32%	-0.28%	-5.40%	-2.11%	-3.75%	-39.35%	4.17%	-17.59%
3	Original Ordering		-4.40%	-3.43%	-3.91%	11.55%	-12.30%	-0.38%	-3.95%	-2.33%	-3.14%	8.07%	-2.36%	2.90%

Chapter 10

Conclusions and Future Work

This thesis demonstrated the significance of preprocessing in machine learning and pattern recognition, especially in IAL. It directly showed that proper preprocessing work can effectively enhance the performance of final results, even if there is little improvement or change in prediction or classification approaches. Therefore, it is obvious that good preprocessing methods are as important as prediction approaches.

This thesis mainly focused on the research of IAL feature ordering approaches and the applications of IAL on feature selection, classification and regression. It firstly reviewed the literature of IAL, including neural algorithms and preprocessing methods such as feature ordering, feature selection and feature grouping. Secondly, it presented methodologies about the experiments implemented in this study. Thirdly, compared with contribution-based feature ordering, three novel feature ordering approaches have been developed, which are based on statistical correlations, mutual information and linear discriminant, respectively. Based on these feature ordering approaches, the usage of feature ordering is extended to the combinative using with feature selection. Moreover, experimental results showed that feature ordering is useful for enhancing the performance in both classification and regression problems.

According to the experimental results in this study, the following conclusions can be drawn:

1. In IAL, feature ordering should be regarded as an independent preprocessing step before formal pattern recognition process.
2. It is able to improve final classification and regression results, if features are sequentially imported into the prediction system according to some special orderings.
3. Except features' single contribution, applicable feature orderings can be derived by

some other metrics. In this thesis, more than ten feasible feature ordering approaches were presented based on some metrics like statistical correlations, mutual information, linear discriminant, and so on.

4. Some feature selection filtering approaches, like mRMR, can be employed in the feature ranking for feature ordering and sorting.
5. There is no such a feature ordering approach which can always produce the best results with the lowest error rate. However, because of the stable performance and very low error rates, AD can be treated as a candidate of optimum feature ordering approach when it is employed with ITID (ILIA2) in classification. In this study, classification error rates derived by ITID (ILIA2) based on AD feature ordering of Diabetes, Cancer, Glass, Thyroid and Semeion reduced by 5.55%, 13.85%, 29.06%, 34.72% and 3.68%, respectively. Moreover, the MMDC and the MAMFO Algorithm are useful to the search of the optimum feature ordering for IAL.
6. According to the experimental results, the final classification results are negatively correlated with the AD means, which indicates that final classification results can be forecasted by AD means. Moreover, the feature ordering with maximum AD means has more probabilities to produce lower error rates.
7. According to the comparison of results between AD and SD feature ordering approaches, feature ordering in IAL should be calculated based on a dynamic feature space, so that the greatest feature discrimination ability can be guaranteed.
8. Feature ordering can be used with feature selection, and ITID (ILIA1) is more stable to exhibit better performance.
9. Feature ordering can be applied in both classification and regression problems. Experimental results confirmed that in most of the situations, IAL with feature ordering has more probability to obtain lower error rates than conventional batch machine learning approach.
10. Feature grouping also plays a very important role in IAL regression with feature ordering, and Grouped Correlation-based feature ordering can be treated as a more stable and applicable feature ordering approach for IAL regression problems with ITID (ILIA2) also produce competitive results. Based on this approach, the error rates of

Flare, Building, Hearta, and Housing reduced by 3.65%, 17.04%, 2.21% and 0.54%, respectively.

In the future, the development of feature ordering approaches should be continued. Whether there is any better feature ordering method existing for IAL is still an important question in the continuous research of this study. Moreover, corresponding metrics, criteria and algorithms are still crucial to the research for improving final results of classification and regression. Further, data distribution, preprocessing methods and predictive algorithms are three important elements which may influence the final classification results. What is the relations among these elements and classification performance will be an issue to be researched in the future. Furthermore, influence brought by output and its division should be considered as an element for IAL. Whether IAL feature ordering and final results will be influenced by output is worthy of being researched in the future. In addition, some fusions of different feature ordering approaches will be carried out in future. Whether better results can be obtained is also a very interesting topic. Actually, in this aspect, relevant research is carried on, where SD and Entropy is combined as a new metric for IAL feature ordering [80]. Last but not the least, IAL with optimum feature ordering approach will be promoted to solve some big and difficult real world problems as its applications in the future.

Appendix A

Data Description

The UCI Machine Learning Repository is a very famous dataset collection resource for machine learning, pattern recognition, data mining and artificial intelligence research. It was firstly created as an FTP archive by researchers of UCI in 1987, and then it became more and more popular, and was widely employed by researchers, students, and educators all over the world as a primary source of machine learning data sets. In the last two decades, because these datasets are widely common used, researchers in machine learning pattern recognition, data mining and artificial intelligence have noticed that these datasets can be used as benchmarks. Research experimental results derived from these benchmark problems can be compared with each other. The comparison of different approaches are suggested to using the same datasets, which makes the results be more convincing than those using different datasets. At present, UCI Machine Learning Repository is managed by the Centre for Machine Learning and Intelligent Systems. New datasets are warmly welcomed to be donated into the archive.

Most datasets employed in the research of this thesis have been used in previous IAL studies. Thus the results derived in this study can be compared smoothly with those calculated in previous research. Obviously, if the dataset and the predictive algorithms are kept stable in the experiments, the only reason why experimental results is changing can be obviously observed, which is from different preprocessing approaches. Then which preprocessing approach is more applicable to exhibit better performance can be easily found.

In the following several subsections, the practical meaning of benchmark datasets used in this research and presented in this thesis is shown. This information is irrelevant to machine learning or pattern recognition directly. The introduction aims to make readers get more perceptual knowledge about what kind of classification and regression has been made in this

thesis. It is believed that the approaches presented in this thesis can be easily understood after they have been read. The information and introduction about these datasets in the aspect of machine learning and pattern recognition can be found in Section 3.2 and 3.3.

1. Pima Indians Diabetes Data Set

Pima Indians Diabetes Dataset was donated by National Institute of Diabetes and Digestive and Kidney Diseases, USA, in 1990. This dataset aims to diagnose whether a Pima Indian has diabetes or not. The diagnosis results were obtained according to 8 continuous input features. The descriptions of these features are as follows: 1. number of times pregnant; 2. plasma glucose concentration; 3. diastolic blood pressure; 4. triceps skin fold thickness; 5. 2-Hour serum insulin; 6. body mass index; 7. diabetes pedigree function; 8. age. The output of this dataset is univariate.

2. Breast Cancer Wisconsin (Original) Data Set

This breast cancer databases was donated by Dr. William H. Wolberg from the University of Wisconsin Hospitals, Madison. This dataset tries to diagnose whether a patient has got a breast cancer and try to classify whether a tumor is either benign or malignant based on cell descriptions gathered by microscopic examinations. The decision can be made according to 9 different input features: 1. Clump Thickness, 2. Uniformity of Cell Size, 3. Uniformity of Cell Shape, 4. Marginal Adhesion, 5. Single Epithelial Cell Size, 6. Bare Nuclei, 7. Bland Chromatin, 8. Normal Nucleoli, 9. Mitoses. As the same as Diabetes, breast cancer is also an univariate output dataset.

3. Glass Identification Data Set

This dataset aims to classify 6 different types of glass in terms of their oxide content. It was donated by Dr. Vina Spiehler for Forensic Science. The original motivation about this dataset aims to classify the types of glass at the scene of the crime. It is believed that the glass left can be used as evidence, if it is correctly identified. Glass has 9 input features, which are: 1. RI: refractive index, 2. Na: Sodium (unit measurement: weight percent in corresponding oxide, as are attributes 3-9), 3. Mg: Magnesium, 4. Al: Aluminum, 5. Si: Silicon, 6. K: Potassium, 7. Ca: Calcium, 8. Ba: Barium and 9. Fe: Iron. It also has 6 outputs: 1. building windows float processed, 2. building windows non float processed, 3 vehicle windows float processed, 4.

containers, 5. tableware, and 6. headlamps.

4. Thyroid Disease Data Set

Thyroid dataset aims to diagnose whether a patient's thyroid is overfunction, normal function, or underfunction based on patient query and examination data. This dataset was from Garavan Institute in Sydney, Australia. It has 21 input features and 3 outputs. Three statuses of a patient's thyroid are three types of outputs, while input features are: 1. age, 2. sex, 3. on thyroxine, 4. query on thyroxine, 5. on antithyroid medication, 6. sick, pregnant, 7. thyroid surgery, 8. I313 treatment, 9. query hypothyroid, 10. query hyperthyroid, 11. lithium, 12. goitre, 13. tumor, 14. hypopituitary, 15. psych, 16. TSH, 17. T3, 18. TT4, 19. T4U, 20. FTI, and 21. TBG.

5. Solar Flare Data Set

The dataset Flare aims to predict the number of solar flares of small, medium, and large size that will happen during the next 24-hour period in a fixed active region of the sun surface. This dataset was donated by Gary Bradshaw from University of Colorado Boulder. The features of this dataset are: 1. Code for class (modified Zurich class) (7 attributes: A,B,C,D,E,F,H), 2. Code for largest spot size (6 attributes: X,R,S,A,H,K), 3. Code for spot distribution (4 attributes: X,O,I,C), 4. Activity (1=reduced, 2=unchanged), 5. Evolution (1=decay, 2=no growth, 3=growth), 6. Previous 24 hour flare activity code (1=nothing as big as an M1, 2=one M1, 3=more activity than one M1), 7. Historically-complex (1=Yes, 2=No),8. Did region become historically complex (1=yes, 2=no) , on this pass across the sun's disk, 9. Area (1=small, 2=large),10. Area of the largest spot (1= <=5, 2 = >5). The outputs have three dimensions: common, moderate, and severe flares. They represent three different types of flares production number in the fixed region in the following 24 hours.

6. Building Data Set

In Proben1, Building aims to predict the energy consumption in a building. Users try to estimate the hourly consumption of electrical energy, hot water, and cold water, based on the date, time of day, outside temperature, outside air humidity, solar radiation, and wind speed. The dataset was created in 1993 for the ASHRAE meeting in Denver, Colorado. It has 14 input features and 3

outputs.

7. Heart Disease Data Set (analog)

The analogy version of Heart Disease Data Set is shorted as Hearta in Proben1. The datasets were denoted by Cleveland Clinic Foundation, Hungarian Institute of Cardiology, V.A. Medical Centre Long Beach, and University Hospital Zurich. It employs a single continuous output that represents by the magnitude of its activation the number of vessels that are reduced.

8. Boston Housing Data Set

Housing dataset aims to estimate housing values in suburbs of Boston. This dataset is from the StatLib library , Carnegie Mellon University. It has 13 input features and 1 output prediction. More specifically, the input features are: 1. CRIM: per capita crime rate by town, 2. ZN: proportion of residential land zoned for lots over 25,000 sq. ft. , 3. INDUS: proportion of non-retail business acres per town, 4. CHAS: Charles River dummy variable (= 1 if tract bounds river; 0 otherwise), 5. NOX: nitric oxides concentration (parts per 10 million), 6. RM: average number of rooms per dwelling, 7. AGE: proportion of owner-occupied units built prior to 1940, 8. DIS: weighted distances to five Boston employment centres, 9. RAD: index of accessibility to radial highways, 10. TAX: full-value property-tax rate per $10,000, 11. PTRATIO: pupil-teacher ratio by town, 12. B: 1000 (Bk - 0.63)^2 where Bk is the proportion of blacks by town, 13. LSTAT: % lower status of the population. MEDV(Median value of owner-occupied homes in $1000's) is the output prediction.

9. Semeion Handwritten Digit Data Set

This dataset was created by Tactile Srl, Brescia, Italy (http://www.tattile.it/) and donated to Semeion Research Centre of Sciences of Communication, Rome, Italy (http://www.semeion.it/), in 1994 for machine learning research. Semeion dataset contains 1593 handwritten digits from around 80 persons. These handwritten digits were scanned, stretched in a rectangular box 16x16 in a gray scale of 256 values. Then each pixel of each image was scaled into a Boolean (1/0) value using a fixed threshold. Each person wrote on a paper all the digits from 0 to 9, twice. The commitment was to write the digit the first time in the normal way (trying to write each digit

accurately) and the second time in a fast way (with no accuracy). Semeion is a multivariate classification problem without missing values. This dataset consists of 1593 records (rows), 256 attributes (columns), and 10 outputs. Each record represents a handwritten digit, originally scanned with a resolution of 256 greys scale. Each pixel of the each original scanned image was first stretched, and after scaled between 0 and 1 (setting to 0 every pixel whose value was under the value 127 of the grey scale (127 included) and setting to 1 each pixel whose original value in the grey scale was over 127). Finally, each binary image was scaled again into a 16x16 square box (the final 256 binary attributes).

Appendix B

Results of Contribution-based Feature Ordering

Apart from the original feature orderings naturally given by each dataset, the real feature orderings were firstly researched according to each feature's contribution to different outputs. This has been studied by Guan and his colleagues [5] in their studies on contribution-based feature selection, where the error rates derived by feature's single contribution can be treated as a measurement to discrimination ability of each feature. Feature's single contribution can be measured by the error rates derived by the prediction using each single feature for classification or regression. The process of contribution-based feature ordering was reviewed in section 2.3.3.

- **Classification**

Experimental results on contribution-based feature ordering are presented in Tables B.1~B.10. More specifically, all the features are solely used to predict classification results one by one in the first place, and sorted according to their contribution to the prediction. Obviously, the lower the error rate is, the greater the contribution. Therefore, all the features are ascending sorted according to their classification error rates derived from their single contribution. The feature sorting results of UCI Benchmarks Diabetes, Cancer, Glass, thyroid and Semeion are shown in Tables B.1, B.3, B.5, B.7 and B.9, respectively. All those final results are compared with Conventional Method, which trains all the features simultaneously in one batch, and shown in Tables B.2, B.4, B.6, B.8 and B.10. According to the results shown in these tables, it is obvious that not all contribution-based feature ordering can obtain better results than conventional method. Therefore, contribution-based feature ordering approach cannot be regarded as an optimum

method for feature ordering in IAL.

1. Diabetes

Table B.1: Ordering Index derived from Feature Single Contribution (Diabetes)

	Feature No.	Contribution Classification Error (%)	Feature Ordering Index
1	1	35.4167	3
2	2	23.5677	1
3	3	36.4583	7
4	4	35.9635	6
5	5	35.4688	4
6	6	37.9948	8
7	7	35.8073	5
8	8	31.8490	2

Table B.2: Contribution-based Feature Ordering (Diabetes)

	Approach	Feature Ordering	Classification Error (%)		
			ITID (ILIA1)	ITID (ILIA2)	AVG[11]
1	Contribution-based	2-8-1-5-7-4-3-6	22.31772	22.03125	22.17448
2	Conventional Method	No Feature Ordering	23.93229		

2. Cancer

Table B.3: Ordering Index derived from Feature Single Contribution (Cancer)

	Feature No.	Contribution Classification Error (%)	Feature Ordering Index
1	1	13.7931	8
2	2	5.3736	1
3	3	6.8966	2
4	4	11.4943	7
5	5	8.6207	3
6	6	9.7701	5
7	7	10.3448	6
8	8	9.1954	4
9	9	21.2644	9

[11] AVG stands for an average number.

Table B.4: Contribution-based Feature Ordering (Cancer)

	Approach	Feature Ordering	Classification Error (%)		
			ITID (ILIA1)	ITID (ILIA2)	AVG
1	Contribution-based	2-3-5-8-6-7-4-1-9	2.50000	1.92529	2.21264
2	Conventional Method	No Feature Ordering	1.86782		

3. Glass

Table B.5: Ordering Index derived from Feature Single Contribution (Glass)

	Feature No.	Contribution Classification Error (%)	Feature Ordering Index
1	1	70.8490	7
2	2	56.4151	2
3	3	61.8868	4
4	4	38.4906	1
5	5	73.4906	9
6	6	63.0189	5
7	7	70.8490	8
8	8	58.4906	3
9	9	67.6415	6

Table B.6: Contribution-based Feature Ordering (Glass)

	Approach	Feature Ordering	Classification Error (%)		
			ITID (ILIA1)	ITID (ILIA2)	AVG
1	Contribution-based	4-2-8-3-6-9-1-7-5	36.41510	33.11322	34.76416
2	Conventional Method	No Feature Ordering	41.22641		

4. Thyroid

Table B.7: Ordering Index derived from Feature Single Contribution (Thyroid)

	Feature No.	Contribution Classification Error (%)	Feature Ordering Index
1	1	7.2778	5
2	2	7.2778	6
3	3	7.2778	7
4	4	7.2778	8
5	5	7.2778	9
6	6	7.2778	10
7	7	7.2778	11
8	8	7.2778	12
9	9	7.2778	13
10	10	7.2778	14
11	11	7.2778	15
12	12	7.2778	16
13	13	7.2778	17
14	14	7.2778	18
15	15	7.2778	19
16	16	7.2778	20
17	17	4.2694	1
18	18	7.2722	4
19	19	6.6833	3
20	20	7.2778	21
21	21	6.4361	2

Table B.8: Contribution-based Feature Ordering (Thyroid)

	Approach	Feature Ordering	Classification Error (%)		
			ITID (ILIA1)	ITID (ILIA2)	AVG
1	Contribution-based	17-21-19-18-1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-20	2.50556	1.72222	2.11389
2	Conventional Method	No Feature Ordering	1.86389		

5. Semeion

Table B.9: Ordering Index derived from Feature Single Contribution (Semeion)

	Feature No.	Contribution Classification Error (%)	Feature Ordering Index
1	1	85.9799	212
2	2	84.9497	161
3	3	84.1960	127
4	4	82.8141	53
5	5	82.9648	63
6	6	81.9096	27
7	7	81.1558	11
8	8	80.8041	8
9	9	81.8090	18
10	10	82.5880	44
11	11	83.6683	90
12	12	83.6432	88
13	13	83.9196	106
14	14	83.6683	91
15	15	83.6809	101
16	16	83.8065	104
17	17	86.6834	231
18	18	86.4322	222
19	19	85.8291	205
20	20	84.1709	119
21	21	84.4975	138
22	22	82.3995	36
23	23	84.8367	151
24	24	85.1759	167
25	25	86.4322	223
26	26	85.6784	196
27	27	87.1357	244
28	28	87.8392	252
29	29	84.6734	143
30	30	84.6734	144
31	31	84.3091	131
32	32	86.4322	224
33	33	86.9473	239
34	34	84.6734	145
35	35	84.1709	120
36	36	82.6633	45
37	37	82.4121	37
38	38	84.6734	146
39	39	86.5327	228

40	40	85.6784	197
41	41	87.0352	241
42	42	86.4322	225
43	43	86.1809	216
44	44	84.9246	155
45	45	84.4221	132
46	46	83.6683	92
47	47	82.7010	51
48	48	84.4724	137
49	49	86.1809	217
50	50	83.9699	111
51	51	84.5603	142
52	52	82.4121	38
53	53	84.8995	154
54	54	86.1558	215
55	55	85.2136	178
56	56	86.6960	234
57	57	87.0980	242
58	58	85.9296	207
59	59	85.2513	180
60	60	84.8241	150
61	61	83.7186	103
62	62	83.4171	78
63	63	81.9221	28
64	64	84.9749	162
65	65	85.2010	177
66	66	83.9196	107
67	67	82.4121	39
68	68	84.4221	133
69	69	85.4271	186
70	70	85.6658	194
71	71	86.6834	232
72	72	88.6935	255
73	73	85.9296	208
74	74	85.1885	175
75	75	85.7035	203
76	76	85.5527	192
77	77	83.6934	102
78	78	82.6633	46
79	79	80.8669	9
80	80	81.9095	22
81	81	85.1508	166
82	82	81.9095	23

83	83	83.4045	77
84	84	85.6784	198
85	85	85.2387	179
86	86	86.4322	226
87	87	85.3518	184
88	88	84.9246	156
89	89	84.1709	121
90	90	85.5276	191
91	91	85.4397	188
92	92	83.4171	79
93	93	83.5176	86
94	94	83.1658	66
95	95	80.7287	5
96	96	80.7287	6
97	97	82.8769	54
98	98	82.1859	32
99	99	85.1759	168
100	100	85.1759	169
101	101	87.1357	243
102	102	83.8693	105
103	103	82.9272	61
104	104	82.9146	57
105	105	83.3668	74
106	106	82.4121	40
107	107	82.9020	55
108	108	82.6633	47
109	109	82.1608	30
110	110	82.4121	41
111	111	81.2814	12
112	112	81.1181	10
113	113	81.8342	19
114	114	81.8844	21
115	115	83.6683	93
116	116	85.1759	170
117	117	85.9296	209
118	118	83.6683	94
119	119	83.1658	67
120	120	84.1709	122
121	121	82.4121	42
122	122	82.6633	48
123	123	84.0830	115
124	124	84.8869	153
125	125	84.9875	163

126	126	85.3015	182
127	127	81.8844	20
128	128	82.3493	35
129	129	81.3568	13
130	130	81.9095	26
131	131	84.4598	136
132	132	86.6834	233
133	133	85.1759	171
134	134	84.1709	123
135	135	84.0453	113
136	136	83.1156	65
137	137	83.1658	68
138	138	83.9196	108
139	139	85.1382	165
140	140	84.9246	157
141	141	83.4171	80
142	142	84.4221	134
143	143	81.7337	16
144	144	83.2035	72
145	145	80.7287	4
146	146	80.6784	3
147	147	84.2086	128
148	148	85.2638	181
149	149	87.4623	249
150	150	85.1759	172
151	151	83.6683	95
152	152	83.1658	69
153	153	83.4171	81
154	154	83.6683	96
155	155	86.0804	214
156	156	83.6683	97
157	157	82.4121	43
158	158	83.9196	109
159	159	83.0653	64
160	160	85.1759	173
161	161	81.4824	15
162	162	79.8995	1
163	163	82.9272	62
164	164	84.1457	117
165	165	84.6734	147
166	166	85.3517	183
167	167	84.1206	116
168	168	84.7111	149

169	169	84.5226	139
170	170	84.5352	140
171	171	83.4799	85
172	172	83.6683	98
173	173	82.9146	58
174	174	84.2839	130
175	175	83.9825	112
176	176	84.1709	124
177	177	82.2111	33
178	178	80.2513	2
179	179	82.0226	29
180	180	83.4171	82
181	181	84.2714	129
182	182	85.6910	202
183	183	85.3769	185
184	184	85.9171	206
185	185	85.6658	195
186	186	85.6784	199
187	187	85.1759	174
188	188	84.4221	135
189	189	83.5176	87
190	190	85.4523	189
191	191	81.7588	17
192	192	83.1658	70
193	193	82.3367	34
194	194	80.7412	7
195	195	82.6884	50
196	196	85.0503	164
197	197	87.3744	247
198	198	86.6080	229
199	199	86.4448	227
200	200	87.1482	245
201	201	85.9296	210
202	202	86.0301	213
203	203	84.5603	141
204	204	84.9246	160
205	205	84.0704	114
206	206	83.1658	71
207	207	82.1859	31
208	208	84.6734	148
209	209	85.5150	190
210	210	83.3417	73
211	211	82.7136	52

212	212	83.9196	110
213	213	86.6457	230
214	214	86.9598	240
215	215	86.9347	237
216	216	86.7337	235
217	217	85.9296	211
218	218	85.6784	200
219	219	85.6784	201
220	220	85.5905	193
221	221	82.9146	59
222	222	82.9146	60
223	223	84.1709	125
224	224	87.3116	246
225	225	86.1809	218
226	226	85.1885	176
227	227	84.1960	126
228	228	82.6633	49
229	229	81.9095	24
230	230	81.4321	14
231	231	81.9095	25
232	232	83.3920	76
233	233	84.9246	158
234	234	84.1458	118
235	235	83.3919	75
236	236	82.9146	56
237	237	83.4674	84
238	238	83.6683	99
239	239	84.9246	159
240	240	86.3065	220
241	241	88.2412	253
242	242	87.7387	251
243	243	86.1809	219
244	244	85.7035	204
245	245	84.8367	152
246	246	83.6557	89
247	247	83.4171	83
248	248	83.6683	100
249	249	85.4271	187
250	250	86.9095	236
251	251	87.4372	248
252	252	87.6382	250
253	253	86.4071	221
254	254	86.9347	238

255	255	88.2915	254
256	256	89.9120	256

Table B.10: Contribution-based Feature Ordering (Semeion)

	Approach	Feature Ordering	Classification Error (%)		
			ITID (ILIA1)	ITID (ILIA2)	AVG
1	Contribution-based	162-178-146-145-95-96-194-8-79-112-7-111-129-230-161-143-191-9-113-127-114-80-82-229-231-130-6-63-179-109-207-98-177-193-128-22-37-52-67-106-110-121-157-10-36-78-108-122-228-195-47-211-4-97-107-236-104-173-221-222-103-163-5-159-136-94-119-137-152-192-206-144-210-105-235-232-83-62-92-141-153-180-247-237-171-93-189-12-246-11-14-46-115-118-151-154-156-172-238-248-15-77-61-16-102-13-66-138-158-212-50-175-135-205-123-167-164-234-20-35-89-120-134-176-223-227-3-147-181-174-31-45-68-142-188-131-48-21-169-170-203-51-29-30-34-38-165-208-168-60-23-245-124-53-44-88-140-233-239-204-2-64-125-196-139-81-24-99-100-116-133-150-160-187-74-226-65-55-85-59-148-126-166-87-183-69-249-91-190-209-90-76-220-70-185-26-40-84-186-218-219-182-75-244-19-184-58-73-117-201-217-1-202-155-54-43-49-225-243-240-253-18-25-32-42-86-199-39-198-213-17-71-132-56-216-250-215-254-33-214-41-57-101-27-200-224-197-251-149-252-242-28-241-255-72-256	18.25378	12.95226	15.60302
2	Conventional Method	No Feature Ordering	13.32915		

- **Regression**

Tables B.11~B.18 illustrate the regression process with contribution-based feature ordering. Four datasets from UCI machine learning repository are employed in the experiments. They are Flare, Building, Hearta and Housing. In each experiments, feature ordering derived by features' single contribution is presented. Obviously, it is similar to classification, where the lower testing error,

the greater the contribution. Thus according to each feature's testing error, feature ordering is indexed. In the next step, all the datasets are reformatted according the new feature ordering, and trained by ITID for regression. Tables B.11, B.13, B.15 and B.17 show the feature orderings of these four datasets, and Tables B.12, B.14, B.16 and B.18 present the regression results and compare the results with conventional batch training method.

1. Flare

Table B.11: Ordering Index derived from Feature Single Contribution (Flare)

	Feature/Attribute No.	Testing Error (%)	Feature Ordering Index
1	(1,2,3,4,5,6,7)	0.5301	1
2	(8,9,10,11,12,13)	0.5730	2
3	(14,15,16,17)	0.5830	3
4	18	0.5944	4
5	19	0.6207	9
6	20	0.6175	8
7	21	0.6111	5
8	22	0.6150	7
9	23	0.6148	6
10	24	0.6212	10

Table B.12: Contribution-based Feature Ordering (Flare)

	Approach	Feature Ordering	Testing Error (%)		
			ITID (ILIA1)	ITID (ILIA2)	AVG
1	Contribution-based	(1,2,3,4,5,6,7)-(8,9,10,11,12,13)-(14, 15,16,17)-18-21-23-22-20-19-24	0.52911	0.53627	0.53269
2	Conventional Method	No Feature Ordering	0.55000		

2. Building

Table B.13: Ordering Index derived from Feature Single Contribution (Building)

	Feature/Attribute No.	Testing Error (%)	Feature Ordering Index
1	(1,2,3,4,5,6,7)	1.2576	8
2	8	1.1557	7
3	9	1.1108	4
4	10	1.1202	6
5	11	1.0347	1
6	12	1.0441	2
7	13	1.0839	3
8	14	1.1198	5

Table B.14: Contribution-based Feature Ordering (Building)

	Approach	Feature Ordering	Testing Error (%)		
			ITID (ILIA1)	ITID (ILIA2)	AVG
1	Contribution-based	11-12-13-9-14-10-8-(1,2,3,4,5,6,7)	0.99377	0.88031	0.93704
2	Conventional Method	No Feature Ordering	0.93966		

3. Hearta

Table B.15: Ordering Index derived from Feature Single Contribution (Hearta)

	Feature/Attribute No.	Testing Error (%)	Feature Ordering Index
1	1	7.9160	9
2	2	8.1560	10
3	(3,4,5,6,7)	5.7435	1
4	(8,9)	8.8820	13
5	(10,11)	7.7485	5
6	(12,13,14)	8.3577	11
7	(15,16,17,18)	8.5978	12
8	(19,20)	7.3646	4
9	(21,22,23)	6.9031	2
10	(24,25)	7.3018	3
11	(26,27,28,29)	7.8785	7
12	(30,31)	7.8931	8
13	(32,33,34,35)	7.7543	6

Table B.16: Contribution-based Feature Ordering (Hearta)

	Approach	Feature Ordering	Testing Error (%)		
			ITID (ILIA1)	ITID (ILIA2)	AVG
1	Contribution-based	(3,4,5,6,7)-(21,22,23)-(24,25)-(19,20)-(10,11)-(32,33,34,35)-(26,27,28,29)-(30,31)-1-2-(12,13,14)-(15,16,17,18)-(8,9)	4.45479	4.60945	4.53212
2	Conventional Method	No Feature Ordering	4.70893		

4. Housing

Table B.17: Ordering Index derived from Feature Single Contribution (Housing)

	Feature No.	Testing Error (%)	Feature Ordering Index
1	1	0.0106	5
2	2	0.0128	9
3	3	0.0102	4
4	4	0.0131	11
5	5	0.0130	10
6	6	0.0119	7
7	7	0.0161	12
8	8	0.0125	8
9	9	0.0095	2
10	10	0.0068	1
11	11	0.0110	6
12	12	0.1212	13
13	13	0.0102	3

Table B.18: Contribution-based Feature Ordering (Housing)

	Approach	Feature Ordering	Testing Error (%)		
			ITID (ILIA1)	ITID (ILIA2)	AVG
1	Contribution-based	10-9-13-3-1-11-6-8-2-5-4-7-12	0.00669	0.01149	0.00909
2	Conventional Method	No Feature Ordering	0.01103		

- **Discussion**

According to above tables, it is manifest that contribution-based feature ordering cannot always obtain better performance than conventional methods in pattern classification problems. For example, in classification, error rates in Cancer (both ILIA1 and ILIA2), Thyroid (ILIA1) and Semeion (ILIA1) are higher than those derived from conventional methods. Furthermore, in regression problems, Housing cannot obtained lower testing error rate in ITID (ILIA2) than in conventional method. Such a phenomenon indicates that, firstly, if there is no proper feature ordering, IAL cannot always bring better results compared with conventional methods; secondly, feature ordering is important to obtain good final results. Therefore, are there any metrics and approaches existing for optimum feature ordering is an urgent question needed to be solved in our studies, and these questions have not been deeply studied in previous studies before this thesis is written.

Appendix C

Parameter Setting and Stop Criteria

The performance of RPROP is relatively insensitive to the values selected. The RPROP algorithm sets the following parameters: $\eta^+ = 1.2$, $\eta^- = 0.5$, $\Delta_0 = 0.1$, $\Delta_{\max} = 50$, $\Delta_{\min} = 1.0e-6$ with initial weights from –0.25 … 0.25 randomly. In order to produce random weight, a random number generator ran 20 times and randomly produced 20 different seeds in previous experiments. Because we want to compare different preprocessing work effects, these 20 random produced seeds are required to be unchangeable in all the experiments once they were produced. The produced seeds are 500, 1720, 2440, 3215, 4810, 5311, 6777, 8550, 9070, 9870, 1173, 2173, 3173, 4173, 5173, 6173, 7173, 8173, 9173, and 10173. Therefore, there is a 20-time training, validation and testing. At last, the average of the results produced in each time will be regarded as the final results.

During the learning process, the set of available patterns is divided into three sets: a training set is used to train the network, a validation set is used to evaluate the quality of the network during training and to measure overfitting, and a test set is used at the end of training to evaluate the resultant network. The size of the training, validation, and test set is 50%, 25% and 25% of the problem's total available patterns.

The error measure E used is the squared error percentage, derived from the normalization of the mean squared error to reduce the dependency on the number of coefficients in the problem representation and on the range of output values used:

$$E = 100 \cdot \frac{o_{\max} - o_{\min}}{K \cdot P} \sum_{p=1}^{P} \sum_{k=1}^{K} (o_{pk} - t_{pk})^2$$

where $o_{\max}$ and $o_{\min}$ are the maximum and minimum values of output coefficients in the problem representation.

$E_{tr}(t)$ is the average error per pattern of the network over the training set, measured after epoch t. The value $E_{va}(t)$ is the corresponding error on the validation set after epoch t and is used by the stopping criterion. $E_{te}(t)$ is the corresponding error on the test set; it is not known to the training algorithm but characterizes the quality of the network resulting from training.

The value $E_{opt}(t)$ is defined to be the lowest validation set error obtained in epochs up to epoch t:

$$E_{opt}(t) = \min_{t' \le t} E_{va}(t')$$

The generalization loss [36] at epoch t is defined as the relative increase of the validation error over the minimum so far (in percent):

$$GL(t) = 100 \cdot (\frac{E_{va}(t)}{E_{opt}(t)} - 1)$$

A high generalization loss is one candidate reason to stop training because it directly indicates overfitting.

To formalize the notion of training progress, a training strip of length k is defined to be a sequence of k epochs numbered $n+1 \ldots n+k$ where n is divisible by k. The training progress measured after a training strip is:

$$P_k(t) = 1000 \cdot (\frac{\sum_{t' \in t-k+1..t} E_{tr}(t')}{k \cdot \min_{t' \in t-k+1..t} E_{tr}(t')} - 1)$$

It is used to measure how much larger the average training error is than the minimum training error during the training strip.

During the process of growing and training sub-networks, we adopted the following heuristic overall stopping criteria: $E_{opt} < E_{th}$ **OR** (Reduction of training set error due to the last new hidden unit is less than 0.01% **AND** Validation set error increased due to the last new

hidden unit). The first part ($E_{opt} < E_{th}$) means that the optimal validation set error is below the threshold and the result has been acceptable. The other part means the last insertion of a hidden unit resulted in hardly any progress. The criteria for adding a new hidden unit are as follows: At least 25 epochs reached for the current network **AND** (Generalization loss $GL(t)$ >5 **OR** Training progress $P_k(t)$ <0.1). The first part means that the current network should be trained for at least a certain number of epochs before a new hidden unit is installed because the error curves will be turbulent in the beginning. The second part means that the current network has been overfitted or training has little progress.

Appendix D

Neural Network Program

Before formal machine learning process starts, all the datasets should be prepared in a regular style. Data should be divided into three sets, training, validation, and testing. The proportion of patterns in these datasets should strictly according to 50%, 25%, 25%, respectively. They are presented in three different files: *.trn, *.val, *.tst. When the machine learning prediction system launched, all these three files will be sequentially imported, and a result file *.rst will be created in the hard disk. The machine learning prediction system needs parameter initialization in the first place, and then starts training. During the whole machine learning process, all the results will be saved in this result file. Figure D.1 is the interface of prediction system. Table D.1 shows parameter description in this neural network system, and Figure D.2 is the parameter setting interface in this software. All these parameter settings are the same as previous IAL studies[4, 6, 21].

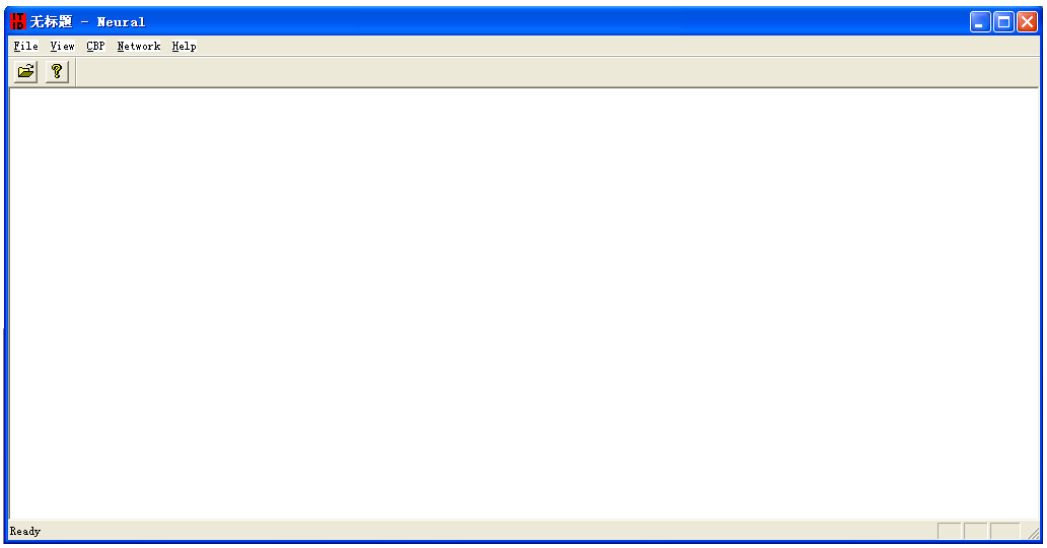


Figure D.1: Interface of Neural IAL Prediction System

Table D.1: Parameter Initialization Description of Neural IAL Prediction System

Parameters	Description
Output Max	the maximum value of output coefficients in the problem representation, i.e., the maximum values of output units for all the patterns. For the classification problems, this value is set to 1. For the regression problems, it may be not equal to 1
Output Min	the minimum value of output coefficients in the problem representation, i.e., the minimum values of output units for all the patterns. For the classification problems, this value is set to 0. For the regression problems, it may be not equal to 0.
Strip Length	the interval (number of epochs) between two measurements of the validation set error is called the strip length. In our experiments, we set 5.
Generalization Loss	The generalization loss at epoch t is defined as the relative increase of the validation error over the minimum so far (in percent): $GL(t) = 100 \cdot (\frac{E_{va}(t)}{E_{opt}(t)} - 1)$. In our experiments, it is set to 5.
Training Progress	The training progress measured after a training strip is: $P_m(t) = 1000 \cdot (\frac{\sum_{t' \in t-m+1..t} E_{tr}(t')}{m \cdot \min_{t' \in t-m+1..t} E_{tr}(t')} - 1)$. It is used to measure how much larger the average training error is than the minimum training error during the training strip.
Accuracy	E_{th} (error threshold)
Training Constraints	True
Max Epoch	The maximum epochs. It is one of the overall stopping criteria.
Training Mode	Freezing
Growing Mode	Batch by batch
Number of Hidden Units	1 for each

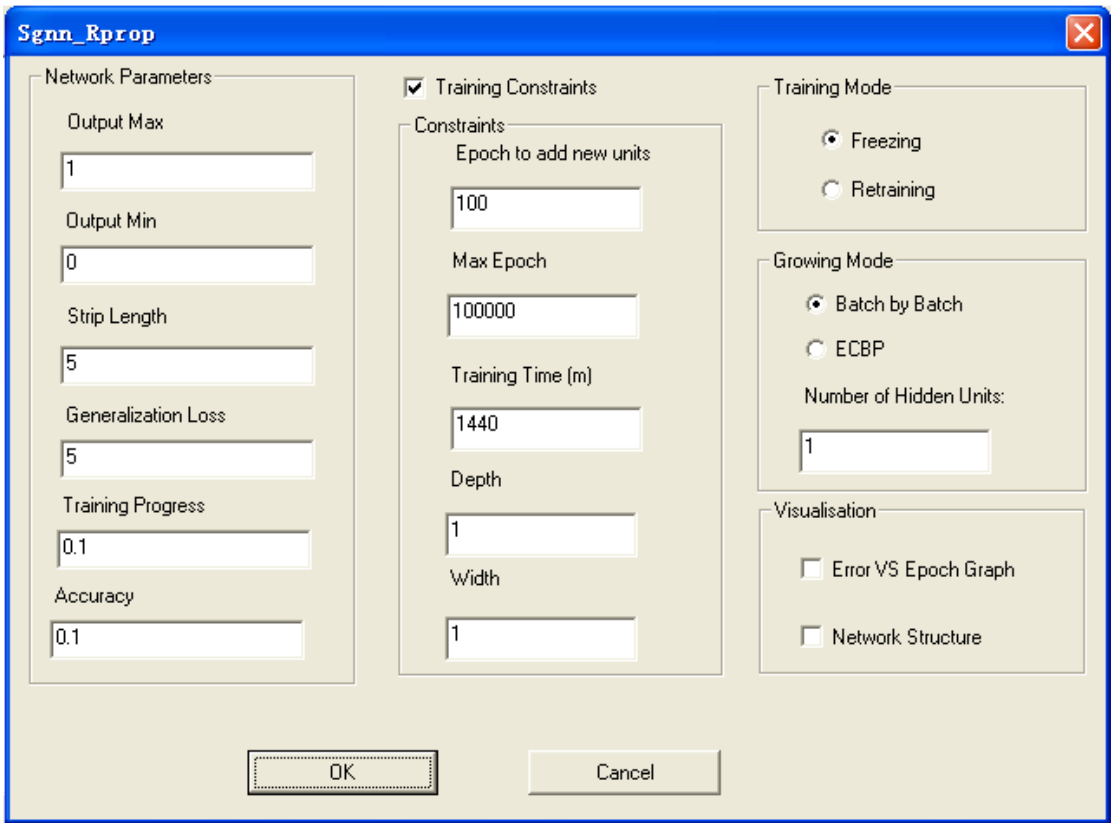


Figure D.2: Parameter Initialization of Neural IAL Prediction System

Bibliography

1. Knuth, D.E., *The art of computer programming, volume 3: (2nd ed.) sorting and searching*1998: Addison Wesley Longman Publishing Co., Inc. 780.
2. Guan, S.U. and J. Liu, *Incremental ordered neural network training.* Journal of Intelligent Systems, 2002. **12**(3): p. 137-172.
3. Guan, S.U. and S.C. Li, *Parallel growing and training of neural networks using output parallelism.* Ieee Transactions on Neural Networks, 2002. **13**(3): p. 542-550.
4. Guan, S.-U. and J. Liu, *Incremental neural network training with an increasing input dimension.* Journal of Intelligent Systems, 2004. **13**(1): p. 45-69.
5. Guan, S.-U., J. Liu, and Y. Qi, *An incremental approach to contribution-based feature selection.* Journal of Intelligent Systems, 2004. **13**(1): p. 15-44.
6. Guan, S.-U. and J.H. Ang, *Incremental training based on input space partitioning and ordered attribute presentation with backward elimination.* Journal of Intelligent Systems, 2005. **14**(4): p. 321-351.
7. Guan, S.-U. and J. Liu, *Feature selection for modular networks based on incremental training.* Journal of Intelligent Systems, 2005. **14**(4): p. 353-383.
8. Ang, J.H., et al., *Interference-less neural network training.* Neurocomputing, 2008. **71**(16-18): p. 3509-3524.
9. Kawewong, A. and O. Hasegawa. *Fast and incremental attribute transferring and classifying system for detecting unseen object classes*. in *20th International Conference on Artificial Neural Networks, ICANN 2010, September 15, 2010 - September 18, 2010*. 2010. Thessaloniki, Greece: Springer Verlag.
10. Kankuekul, P., et al. *Online incremental attribute-based zero-shot learning*. in *2012 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2012, June 16, 2012 - June 21, 2012*. 2012. Providence, RI, United states: IEEE Computer Society.
11. Sheng-Uei, G. and M. Wenting, *Incremental evolution strategy for function optimization.* International Journal of Hybrid Intelligent Systems, 2006. **3**(4): p. 187-203.
12. Chen, Q. and S.-U. Guan, *Incremental multiple objective genetic algorithms.* IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 2004. **34**(3): p. 1325-1334.
13. Zhu, F. and S.-U. Guan, *Ordered incremental training with genetic algorithms.* International Journal of Intelligent Systems, 2004. **19**(12): p. 1239-1256.
14. Guan, S.U. and F.M. Zhu, *An incremental approach to genetic-algorithms-based classification.* Ieee Transactions on Systems Man and Cybernetics Part B-Cybernetics, 2005. **35**(2): p. 227-239.
15. Zhu, F.M. and S.U. Guan, *Cooperative co-evolution of GA-based classifiers based on input decomposition.* Engineering Applications of Artificial Intelligence, 2008. **21**(8): p. 1360-1369.
16. Zhu, F.M. and S.U. Guan, *Enhanced Cooperative Co-evolution Genetic Algorithm for Rule-Based Pattern Classification*, in *Hybrid Artificial Intelligence Systems*, E. Corchado, A. Abraham, and W. Pedrycz, Editors. 2008, Springer-Verlag Berlin: Berlin. p. 113-123.

17. Agrawal, R.K. and R. Bala, *Incremental Bayesian classification for multivariate normal distribution data.* Pattern Recognition Letters, 2008. **29**(13): p. 1873-1876.

18. Chao, S. and F. Wong. *An incremental decision tree learning methodology regarding attributes in medical data mining*. in *2009 International Conference on Machine Learning and Cybernetics, July 12, 2009 - July 15, 2009*. 2009. Baoding, China: IEEE Computer Society.

19. Bai, W., et al. *Incremental attribute based particle swarm optimization*. in *2012 8th International Conference on Natural Computation, ICNC 2012, May 29, 2012 - May 31, 2012*. 2012. Chongqing, China: IEEE Computer Society.

20. Liu, X., et al. *An incremental feature learning algorithm based on least square support vector machine*. in *2nd International Frontiers in Algorithmics Workshop, FAW 2008, June 19, 2008 - June 21, 2008*. 2008. Changsha, China: Springer Verlag.

21. Guan, S.-U. and S. Li, *Incremental learning with respect to new incoming input attributes.* Neural Processing Letters, 2001. **14**(3): p. 241-260.

22. Guan, S.-U. and P. Li, *A hierarchical incremental learning approach to task decomposition.* Journal of Intelligent Systems, 2002. **12**(3): p. 201-223.

23. Bao, C. and S.-U. Guan. *Reduced training for hierarchical incremental class learning*. in *2006 IEEE Conference on Cybernetics and Intelligent Systems, June 7, 2006 - June 9, 2006*. 2006. Bangkok, Thailand: Inst. of Elec. and Elec. Eng. Computer Society.

24. Guan, S.-U., C. Bao, and R.-T. Sun, *Hierarchical incremental vlass learning with reduced pattern training.* Neural Processing Letters, 2006. **24**(2): p. 163-177.

25. Guan, S.-U. and K. Wang, *Hierarchical incremental class learning with output parallelism.* Journal of Intelligent Systems, 2007. **16**(2): p. 167-193.

26. Guan, S.-U. and P. Li, *Incremental learning in terms of output attributes.* Journal of Intelligent Systems, 2004. **13**(2): p. 95-122.

27. Guan, S.U., C. Bao, and T. Neo, *Reduced pattern training based on task decomposition using pattern distributor.* Ieee Transactions on Neural Networks, 2007. **18**(6): p. 1738-1749.

28. Bao, C.Y., T.N. Neo, and S.U. Guan, *Reduced pattern training in pattern distributor networks.* Journal of Research and Practice in Information Technology, 2007. **39**(4): p. 273-286.

29. Baluja, S., *Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning*, 1994, Carnegie Mellon University.

30. Han, J., *Data Mining: Concepts and Techniques*2005: Morgan Kaufmann Publishers Inc.

31. Freitas, A.A., *Data Mining and Knowledge Discovery with Evolutionary Algorithms*2002: Springer-Verlag New York, Inc. 280.

32. Bermejo, P., et al., *Fast wrapper feature subset selection in high-dimensional datasets by means of filter re-ranking.* Knowledge-Based Systems, 2012. **25**(1): p. 35-44.

33. Frank, A. and A. Asuncion. *UCI Machine Learning Repository*. 2010 [cited 2012; Available from: http://archive.ics.uci.edu/ml/.

34. Wolberg, W.H. and O.L. Mangasarian, *Multisurface method of pattern separation for medical diagnosis applied to breast cytology.* Proceedings Of The National Academy Of Sciences Of The United States Of

America, 1990. **87**(23): p. 9193-9196.

35. Harrison, D., Jr. and D.L. Rubinfeld, *Hedonic Housing Prices and the Demand for Clean Air.* Journal of Environmental Economics and Management, 1978. **5**(1): p. 81-102.

36. Prechelt, L., *Proben1 - A Set of Neural Network Benchmark Problems and Benchmarking Rules*, in *Technical Report 21/94*1994, Fakultät für Informatik, Universität Karlsruhe: Karlsruhe, Germany.

37. Wang, T., S.-U. Guan, and F. Liu. *Feature discriminability for pattern classification based on neural incremental attribute learning*. in *Foundations of Intelligent Systems: Proceedings of the Sixth International Conference on Intelligent Systems and Knowledge Engineering, Shanghai, China, Dec 2011 (ISKE2011)*. 2011. Tiergartenstrasse 17, Heidelberg, D-69121, Germany: Springer Verlag.

38. Wang, T., et al. *Evolving linear discriminant in a continuously growing dimensional space for incremental attribute learning*. in *9th IFIP International Conference on Network and Parallel Computing, NPC 2012, September 6, 2012 - September 8, 2012*. 2012. Gwangju, Korea, Republic of: Springer Verlag.

39. Su, L., S.U. Guan, and Y.C. Yeo, *Growing cascade correlation networks in two dimensions: A heuristic approach.* Journal of Intelligent Systems, 2001. **11**(4): p. 249-267.

40. Schapire, R.E., *The Strength of Weak Learnability.* Mach. Learn., 1990. **5**(2): p. 197-227.

41. Freund, Y. and R.E. Schapire, *A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting.* Journal of Computer and System Sciences, 1997. **55**(1): p. 119-139.

42. Breiman, L., *Bagging predictors.* Machine Learning, 1996. **24**(2): p. 123-140.

43. Tsymbal, A. and S. Puuronen, *Bagging and Boosting with Dynamic Integration of Classifiers*, in *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*2000, Springer-Verlag. p. 116-125.

44. Ripley, B.D., *Pattern Recognition and Neural Networks*1996, UK:Cambridge: Cambridge University Press.

45. Littlestone, N. *LEARNING QUICKLY WHEN IRRELEVANT ATTRIBUTES ABOUND: A NEW LINEAR-THRESHOLD ALGORITHM*. in *28th Annual Symposium on Foundations of Computer Science*. 1987. Los Angeles, CA, USA: IEEE.

46. Blum, A., A. Kalai, and J. Langford. *Beating the hold-out: bounds for K-fold and progressive cross-validation*. in *Proceedings of the 1999 12th Annual Conference on Computational Learning Theory (COLT'99), July 6, 1999 - July 9, 1999*. 1999. Santa Cruz, CA, USA: ACM.

47. Kim, J.-H., *Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap.* Computational Statistics and Data Analysis, 2009. **53**(11): p. 3735-3745.

48. Fahlman, S.E. and C. Lebiere, *The cascade-correlation learning architecture.* 1989.

49. Duda, R.O., P.E. Hart, and D.G. Stork, *Pattern Classification (2nd Edition)*2000: Wiley-Interscience.

50. Parekh, R., J. Yang, and V. Honavar, *Constructive neural-network learning algorithms for pattern classification.* Ieee Transactions on Neural Networks, 2000. **11**(2): p. 436-451.

51. Riedmiller, M. and H. Braun. *Direct adaptive method for faster backpropagation learning: The RPROP algorithm*. in *1993 IEEE International Conference on Neural Networks, March 28, 1993 - April 1, 1993*. 1993. San Francisco, CA, USA: Publ by IEEE.

52. Prechelt, L., *Investigation of the CasCor family of learning algorithms.* Neural Networks, 1997. **10**(5): p. 885-896.

53. Lehtokangas, M., *Modelling with constructive backpropagation.* Neural Networks, 1999. **12**(4-5): p. 707-716.

54. Reed, R., *Pruning algorithms - a survey.* Ieee Transactions on Neural Networks, 1993. **4**(5): p. 740-747.

55. Blum, A. and R.L. Rivest, *Training a 3-node neural network is NP-complete.* Neural Networks, 1992. **5**(1): p. 117-127.

56. Fu, L., H.-H. Hsu, and J.C. Principe, *Incremental backpropagation learning networks.* Ieee Transactions on Neural Networks, 1996. **7**(3): p. 757-761.

57. Kwok, T.-Y. and D.-Y. Yeung, *Objective functions for training new hidden units in constructive neural networks.* Ieee Transactions on Neural Networks, 1997. **8**(5): p. 1131-1148.

58. Rogers, J., *Object-oriented neural networks in C++*1997, San Diego, CA: Academic Press.

59. Hall, M.A., *Correlation-based Feature Selection for Machine Learning*, in *Department of Computer Science*1999, University of Waikato: New Zealand.

60. Huang, J., et al. *A method for feature selection based on the correlation analysis*. in *2012 International Conference on Measurement, Information and Control, MIC 2012, May 18, 2012 - May 20, 2012*. 2012. Harbin, China: IEEE Computer Society.

61. Peng, H., F. Long, and C. Ding, *Feature selection based on mutual information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005. **27**(8): p. 1226-1238.

62. Wang, T. and Y. Wang. *Pattern classification with ordered features using mRMR and neural networks*. in *2010 International Conference on Information, Networking and Automation, ICINA 2010, October 17, 2010 - October 19, 2010*. 2010. Kunming, China: IEEE Computer Society.

63. Fisher, R.A., *THE USE OF MULTIPLE MEASUREMENTS IN TAXONOMIC PROBLEMS.* Annals of Eugenics, 1936. **7**(2): p. 179-188.

64. Rao, C.R., *The Utilization of Multiple Measurements in Problems of Biological Classification.* Journal of the Royal Statistical Society - Series B, 1948. **10**(2).

65. Zhao, Z., et al. *Advancing Feature Selection Research: ASU Feature Selection Repository*. Feature Selection Technical Report, 2010.

66. Mitchell, T.M., *Machine Learning*1997: McGraw-Hill, Inc. 432.

67. Horak, V.C., T. Berka, and M. Vajteršic, *AN INTRINSICALLY PARALLEL META-CLASSIFIER FOR HIGH-DIMENSIONAL SPACES.* ParNum 11: p. 48.

68. Chen, Y.-L., Y.F. Zheng, and Y. Liu, *Margin and Domain Integrated Classification for Images.* International Journal of Information Acquisition, 2011. **8**(01): p. 1-16.

69. Shin, Y.J. and C.H. Park, *Analysis of correlation based dimension reduction methods.* International Journal of Applied Mathematics and Computer Science, 2011. **21**(3): p. 549-558.

70. Wu, J.-Y. *MIMO CMAC neural network classifier for solving classification problems*. 2011. Langford Lane, Kidlington, Oxford, OX5 1GB, United Kingdom: Elsevier Ltd.

71. Kulluk, S., L. Ozbakir, and A. Baykasoglu, *Training neural networks with harmony search algorithms*

for classification problems. Engineering Applications of Artificial Intelligence, 2012. **25**(1): p. 11-19.

72. Puma-Villanueva, W.J., E.P. dos Santos, and F.J. Von Zuben, *A constructive algorithm to synthesize arbitrarily connected feedforward neural networks.* Neurocomputing, 2012. **75**(1): p. 14-32.

73. Dehuri, S., et al., *An improved swarm optimized functional link artificial neural network (ISO-FLANN) for classification.* Journal of Systems and Software, 2012. **85**(6): p. 1333-1345.

74. Yang, X., S. Chen, and B. Chen, *Plane-Gaussian artificial neural network.* Neural Computing and Applications, 2012. **21**(2): p. 305-317.

75. Shen, T. and D. Zhu. *Layered-CasPer: Layered cascade artificial neural networks*. in *2012 Annual International Joint Conference on Neural Networks, IJCNN 2012, Part of the 2012 IEEE World Congress on Computational Intelligence, WCCI 2012, June 10, 2012 - June 15, 2012*. 2012. Brisbane, QLD, Australia: Institute of Electrical and Electronics Engineers Inc.

76. Tsoy, Y. *Evolving linear neural networks for features space dimensionality reduction*. in *2012 Annual International Joint Conference on Neural Networks, IJCNN 2012, Part of the 2012 IEEE World Congress on Computational Intelligence, WCCI 2012, June 10, 2012 - June 15, 2012*. 2012. Brisbane, QLD, Australia: Institute of Electrical and Electronics Engineers Inc.

77. Mineu, N.L., A.J. Da Silva, and T.B. Ludermir. *Evolving neural networks using differential evolution with neighborhood-based mutation and simple subpopulation scheme*. in *2012 Brazilian Conference on Neural Networks, SBRN 2012, October 20, 2012 - October 25, 2012*. 2012. Curitiba, Parana, Brazil: IEEE Computer Society.

78. Liu, H. and H. Motoda, *Computational Methods of Feature Selection (Chapman \\& Hall/Crc Data Mining and Knowledge Discovery Series)*2007: Chapman \\& Hall/CRC.

79. Aldrich, J., *Correlations Genuine and Spurious in Pearson and Yule*, 1995, Economics Division, School of Social Sciences, University of Southampton, Discussion Paper Series In Economics And Econometrics.

80. Wang, T., S.-U. Guan, and F. Liu. *Entropic feature discrimination ability for pattern classification based on neural IAL*. in *9th International Symposium on Neural Networks, ISNN 2012, July 11, 2012 - July 14, 2012*. 2012. Shenyang, China: Springer Verlag.