Supervised Feature Selection Based on Generalized Matrix Learning Vector Quantization

Zetao Chen

S2061244

September 2012

Master Project

Artificial Intelligence

University of Groningen, the Netherlands

Internal Supervisor:

Dr. Marco Wiering (Artificial Intelligence,University of Groningen)

External Supervisor:

Prof. Michael Biehl (Computer Science, University of Groningen)

university of groningen / faculty of mathematics and natural sciences / artificial intelligence

Contents

Abstract

Data mining involves the use of data analysis tools to discover and extract information from a data set and transform it into an understandable expression. One of its central problems is to identify a representative subset of features from which a learning model can be constructed. Feature selection is an important pre-processing step before data mining which aims to select a representative subset of features with high predictive information and eliminate irrelevant features with little importance for classification. By reducing the dimensionality of the data, feature selection helps to decrease the time for training and by selecting the most relevant features and removing the irrelevant and noisy data, the classification performance may be improved. Besides, with a smaller feature subset, the learned model may be more intuitive and easier to interpret.

This thesis investigates the extension of Generalized Matrix LVQ (GMLVQ) model on feature selection. Generalized Matrix LVQ employs a full matrix as the distance metric in training. The diagonal and off-diagonal elements of the distance matrix respectively measure the contribution of each feature and feature pair for classification; therefore, their distribution can provide a quantitative measurement of feature weight. More steps and analysis are performed to force a more effective feature selection result and remove the weighting ambiguity. Besides, compared to other methods which perform feature ranking first and learning a model after selecting the feature subset, GMLVQ based methods can combine the process of feature ranking and classification together which helps to decrease the computation time.

Experiments in this thesis were performed on data sets collected from the UCI Machine Learning Repository [29]. The GMLVQ based feature weight algorithm is compared with other state-of-the-art methods: Information Gain, Fisher and Relieff. All these four feature ranking methods are evaluated using both GMLVQ and RBF based Support Vector Machine (RBF-SVM) methods by increasing the size of the selected feature subset with a stepsize rate. The results indicate that the performance of GMLVQ based feature selection method is comparable to other methods and on some of the data sets, it consistently outperforms the other methods.

Chapter 1

1 Introduction and Background

1.1 Motivation

For a machine learning algorithm to be successful on a given task, the representation and quality of the data are the first and most important. With the advancing of database technology, data is easier to assess and more features can be gathered for a specific task. However, more features do not necessarily result in more discriminative classifiers. Instead, when there are too many redundant or irrelevant features, the computation can be much more expensive and the classifier may have a poor generalization performance due to the interference of noises; therefore, proper data preprocessing is essential for the successful training of machine learning algorithms.

Feature selection is one of the most important and frequently used preprocessing techniques [5] which aims to identify and select the most discriminative subset from the original features while eliminating irrelevant, redundant and noisy data. Some studies have shown that irrelevant features can be removed without significant performance downgrade [6]. The application of feature selection can have some benefits:

1. It reduces the data dimensionality which helps the learning algorithms to work faster and more effectively;

2. In some cases, the classification accuracy can be improved by using a subset of all features;

3. The selected feature subset is usually a more compact result which can be interpreted more easily;

To perform feature selection, the training data can be with or without label information, corresponding to supervised or unsupervised feature selection. In unsupervised tasks [1, 2], without considering the label information, feature relevance can be evaluated by measuring some intrinsic properties within the data, such as the separability or covariance. In practice, unlabeled data is easier to obtain compared to labelled ones, thereby indicating the significance of unsupervised algorithms. However, these methods ignore label information, which may lead to performance deterioration when the label information is available. Supervised feature selection is proposed to take the label information into account. It can be generally divided into two major frameworks: the filter model [14, 15, 16, 17] and the wrapper model [18, 19, 20]. The filter model performs the feature selection as a pre-processing step, independent of the choice of the classifier. The wrapper model, on the other hand, evaluates subsets of features according to their usefulness to a given predictor.

Feature selection techniques can be further categorized into feature ranking and feature subset selection. Feature ranking methods assign a weight to each

feature, indicating their importance in terms of some criterion. It is the user to select the subset of features by choosing a threshold and eliminate all features which do not achieve that score. Feature subset selection searches for the optimal subset which collectively has the best performance with respect to some predictor. In this thesis, a new method for feature ranking will be investigated and compared with other state-of-the-art ones.

Learning Vector Quantization is one of the most famous prototype-based supervised learning methods. It was first introduced by Kohonen [3]. After that, several advanced cost functions were proposed to improve the performance, one example being Generalized LVQ [4] which is only based on Euclidean distance. To model the different contributions of features for classification, Generalized Relevance LVQ is proposed [4, 7] to extend the Euclidean distance with scaling or relevance factors for all features. The recently introduced Generalized Matrix LVQ (GMLVQ) [33] extends the distance measurement further to account for pairwise contribution of features. The distance matrix in GMLVQ contains some information which may be useful for feature selection. For example, the diagonal element Λ_{ii} of the dissimilarity matrix can be regarded as a measurement of the overall relevance of feature i for classification and the off-diagonal element Λ_{ij} can be interpreted as the contribution of feature pair i and j. A high absolute value indicates the existence of a highly relevant relationship while an absolute value closer to zero may suggest that is is not that important for classification.

The above discussion illustrates the potential application of GMLVQ in feature ranking which has not yet been fully investigated. Early studies include applying GMLVQ to select the best feature in the classification of lung disease [39] and select the most discriminative marker in the diagnosis of Adrenal Tumor [41]. In this thesis, a further investigation will be conducted and experiments on more data sets will be carried out.

1.2 Research Questions

This thesis will attempt to answer the following questions:

1. Can GMLVQ method be extended to perform feature ranking?
2. How well does the feature ranking perform? In this thesis, the GMLVQ based feature ranking technique will be compared with three other state-of-the-art feature ranking methods. All these four methods will be evaluated by GMLVQ and RBF-SVM in terms of their AUC metric.
3. Can GMLVQ combine the feature ranking and classification into one single process and how well does the classification perform compares to other methods in which feature ranking and classification are performed in two steps?

1.3 Thesis Outline

This thesis has six chapters and is organized as follows. Chapter 2 presents the basic concepts in machine learning and the algorithm details of the Sup-

port Vector Machine (SVM) and GMLVQ which will be used to evaluate the performance of various feature ranking algorithms in a later stage. Chapter 3 discusses the idea of feature selection, its general framework and three state-of-the-art feature ranking techniques which will be compared with the GMLVQ based ranking method. Chapter 4 gives a description about the GMLVQ based feature ranking method. In this chapter, details will be given to extract feature ranking from GMLVQ, the waypoint averaging algorithm and how to obtain a unique feature ranking result. Chapter 5 elaborates on the experiments conducted to compare the four feature ranking techniques discussed above and is followed by Chapter 6 that states the conclusion and future work for this thesis.

Chapter 2

2 Machine Learning

In this chapter, we firstly give a brief introduction to machine learning and some of its basic concepts. The data representation, classification and learning algorithms are further presented. We further present some specific learning algorithms which are RBF-based SVM, basic LVQ and its two other variations. The learning algorithms introduced in this chapter will be later utilized to evaluate the feature selection algorithms introduced in Chapter 3.

2.1 Basic Concept of Machine Learning

2.1.1 Definition of learning

What is learning? Learning is generally referred to the mutual interaction between the environment and the person through which one gains or modifies knowledge or skills. A more formal definition was given by Runyon in 1977 [36]: "Learning is a process in which behavior capabilities are changed as the result of experience provided the change cannot be accounted for by native response tendencies, maturation, or temporary states of the organism due to fatigue, drugs, or other temporary factors."

One of the examples in learning is the association between events. For example, if a normal person tastes an apple for the first time and finds it very delicious, he will assume that the next apple he meets will also be delicious although he has not eaten it and that apple is different from the one he ate. The important discovery here is the association of the facts that the apple is tasty. This association is the knowledge someone gains by the experience to eat an apple.

2.1.2 Definition of machine learning

Learning for computers falls into the field of machine learning. A widely accepted definition is: A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E [37]. The experience here usually refers to the data which demonstrates the relationship between observed variables.

There are many example applications in Machine Learning. One of the largest groups lies in the categorization of objects into a set of pre-specified classes or labels. Some of the practical examples are:

1. Optical Character Recognition: classify images of handwritten characters to the specific letters;

2. Face Recognition: categorize facial images to the person it belongs to;

Instance #	Features				Class
	Outlook	Temperature	Humidity	Wind	
1	sunny	hot	high	false	Don't play
2	sunny	hot	high	true	Don't Play
3	overcast	hot	high	false	Play
4	rain	mild	high	false	Play
5	rain	cool	normal	false	Play
6	rain	cool	normal	true	Don't Play
7	overcast	cool	normal	true	Play
8	sunny	mild	high	false	Don't Play
9	sunny	cool	normal	false	Play
10	rain	mild	normal	false	Play
11	sunny	mild	normal	true	Play
12	overcast	mild	high	true	Play
13	overcast	hot	normal	false	Play
14	rain	mild	high	true	Don't Play

Figure 1: The "Golf" example demonstrating the data representation in machine learning.

3. Medical Diagnosis: determine whether or not a patient suffers from some disease;

4. Stock Prediction: predict whether a stock goes up or down

2.1.3 Data representation

In the field of machine learning, data is represented by a table where each row corresponds to one sample or instance and each column describes one attribute or feature. In the case of supervised learning, there will be another column containing the label information for each instance. One of the examples is shown in Figure 1. There are 14 instances in this example and each instance consists of the data with four features: "Outlook", "Temperature", "Humidity", "Wind" and the label information specifying whether or not to play.

The mathematical expressions of the data and labels are presented here to serve as the notations in this thesis. Let $\{x_i, y_i\}$ denote the i^{th} instance where $x_i \in R^N$ denotes the data in the N dimensional space and y_i is the corresponding label information with C different possible values. To be brief, the combination of data and label are expressed as below:

$$\{x_i, y_i\} \in R^N \times C \tag{1}$$

2.2 Classification

As discussed in the previous section, the major task in machine learning is to learn how to classify objects into one of the pre-defined set of labels. In such task,

it is crucial to identify the common characteristics from a set of representative objects in a class. For example, to identify whether a fruit is a banana, people have to check its color, size, shape and infer its label from this information.

2.2.1 Unsupervised and supervised learning

The classification task discussed above is generally referred to as supervised learning where the labels of training data are provided and the learning algorithm tries to generalize from the training instances to enable novel objects to be classified to correct categories. In contrast to supervised learning, unsupervised learning refers to the learning in which the labels of training data are unknown. Its goal is to group the training data into different clusters by evaluating some intrinsic properties within the data, such as the separability or covariance; therefore, the quality of the data provided for training is crucial. If irrelevant or noisy data are provided, misclassifications will happen on novel data.

2.3 Learning Algorithms

In this section, two supervised learning algorithms will be described which are the SVM algorithm with RBF kernel and the LVQ algorithm with its two variants: GRLVQ and GMLVQ. The GMLVQ and RBF-SVM will be later utilized to evaluate the performance of four feature ranking methods.

2.3.1 SVM with RBF kernel

The Support Vector Machine (SVM) was originally proposed by Vapnik for classification and regression [25, 24, 26, 27] and then it was also extended for other application [28]. It has attracted large attention in recent years due to its superior performance and soundly developed theoretical foundation. As a result, it also serves as an evaluation method for the feature selection results in this thesis.

The SVM is a method to find an optimal hyperplane to separate training data of two or more classes and at the same time, maximizing its margin. The linear Support Vector Machine, as the simplest and most basic case, will be introduced first. Then we will show how it can classify non-linearly separable data in a feature space in higher dimensions.

Linear SVM and separating hyperplane maximization The linear SVM is a supervised learning method which is built upon a group of labelled samples and that performs binary classification in the feature space. Let's denote the data and labels as (x_i, y_i) where $x_i \subseteq R^N$ is a N-dimensional feature vector and y_i is the label of sample x_i. In a two-class problem, $y_i \in \{+1, -1\}$. The classification process of a supervised learning algorithm can then be regarded as a mapping process $f(x_i)$:$R^N \rightarrow R$ which maps the feature vector from a N dimensional space to the class membership of the vector. Without loss of

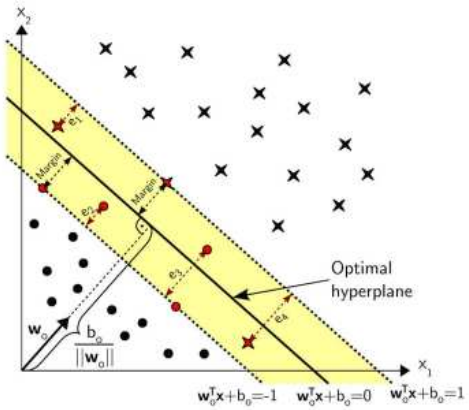

Figure 2: Linear Support Vector Machine. Function f(x) divides the feature space into two halves

generality, it is assumed that $f(x_i) > 0$ and $y_i = 1$ indicate the feature vector belongs to class 1 and $f(x_i) < 0$ and $y_i = -1$ specify the class 2. Then a formal definition about linearly separable data can be given as: a data set is linearly separable if the following equations hold:

$$\forall\, y_i = 1 :\ f(x_i) > 0 \tag{2}$$

$$\forall\, y_i = -1 :\ f(x_i) \leq 0 \tag{3}$$

An illustration example is shown in Figure 2. As can be seen from the figure, all the points with $y_i = 1$ are classified into the positive side of the hyperplane and others with $y_i = -1$ are in the opposite side.

The discriminant function in Figure 2 is a linear model and can be expressed as:

$$f(x) = w^T x + b \tag{4}$$

where w indicates the weight vector and b is the bias. The hyperplane which divides the plane into two half-planes is expressed as:

$$f(x) = w^T x + b = 0$$

The discriminant function $f(x)$ can also help to measure the distance of a data point to the hyperplane. Consider the point x_d and its normal projection x_0 on the hyperplane in Figure 2. The coordinates of the point x_d can then be expressed as:

$$x_d = x_0 + d\frac{w}{\|w\|} \tag{5}$$

where d describes the algebraic distance between the point x_d and x_0. Because x_0 is on the hyperplane, $f(x_0) = 0$. We have:

$$f(x_d) = f(x_0 + d\frac{w}{\|w\|}) = w^T(x_0 + d\frac{w}{\|w\|}) + b \quad (6)$$

$$= f(x_0) + d\frac{w^T w}{\|w\|} = d\,\|w\| \quad (7)$$

It follows that: $d = \frac{f(x_i)}{\|w\|}$ and to enforce that d is always positive under correct classification, we define:

$$d_i = \frac{y_i f(x_i)}{\|w\|} \quad (8)$$

Then the term margin p can be defined here as the distance between the hyperplane and the closest data points from both sides:

$$p = \frac{\underset{i=1,2\cdots n}{min}\, y_i f(x_i)}{\|w\|} \quad (9)$$

where n is the number of examples in the training data set. The linear SVM is trained to find an optimal hyperplane to maximize the margin p. As shown in the formula above, this can be achieved by either maximizing the value of $y_i f(x_i)$ of the closest points or by minimizing $\|w\|$. Since $w^T x + b$ can be scaled without changing its sign, it is reasonable to impose the constraint that:

$$y_i(w^T x_i + b) \geq 1 \quad (10)$$

$$i = 1, 2, \cdots n \quad (11)$$

Therefore, the optimization problem can be formulated as [25]: given a set of training samples $\{x_i, y_i\}_{i=1}^n$, try to find the optimal parameters w and b which satisfies the constraint that:

$$y_i(w^T x_i + b) \geq 1 \quad (12)$$

$$i = 1, 2, \cdots, n \quad (13)$$

and minimizes the following function:

$$L = \frac{1}{2} w^T w \quad (14)$$

This is called the primary problem and can be solved by constructing the Lagrange function [30] as below:

$$J(w, b, a) = \frac{1}{2} w^T w - \sum_{i=1}^{n} a_i [y_i(w^T x_i + b) - 1] \quad (15)$$

The a_i here are called the Lagrange multipliers and the solution of this optimization problem should be minimized with respect to w and b and maximized with respect to a_i. As a result, it follows that

$$\frac{\partial J(w,b,a)}{\partial w} = w - \sum_{i=1}^{n} a_i y_i x_i = 0 \tag{16}$$

and

$$\frac{\partial J(w,b,a)}{\partial b} = \sum_{i=1}^{n} a_i y_i = 0 \tag{17}$$

which gives rise to

$$w = \sum_{i=1}^{n} a_i y_i x_i \tag{18}$$

and

$$\sum_{i=1}^{n} a_i y_i = 0 \tag{19}$$

Then by substituting the above two equations into equation (15), the equation becomes:

$$Q(a) = \sum_{i=1}^{n} a_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j y_i y_j x_i^T x_j \tag{20}$$

The corresponding problem is called the dual problem and is formulated as below: given training samples $\{x_i, y_i\}_{i=1}^n$, try to find the optimal Lagrange multipliers $\{a_i\}_{i=1}^n$ which maximize the objective function above and also satisfy the following constrains:

1. $$\sum_{i=1}^{n} a_i y_i = 0;$$

2. $a_i \geq 0$ for $i = 1, 2, \ldots, n$

After the Lagrange multipliers are determined, the weight vector can be easily determined by

$$w = \sum_{i=1}^{n} a_i y_i x_i \tag{21}$$

and the bias b can be determined by arbitrarily choosing a labeled sample $\{x_i, y_i\}$ and calculate:

$$y_i(w^T x_i + b) = 1 \tag{22}$$

$$\forall\, y_i = 1: \quad b = 1 - w^T x_i \tag{23}$$

$$or \tag{24}$$

$$\forall\, y_i = -1: \quad b = -1 - w^T x_i \tag{25}$$

It is also important to state the Karush-Kuhn-Tucker theorem [25, 30] which gives the following constraint on the saddle point of the Lagrange:

$$a_{i0}[y_i(w_0^T x_i + b_0) - 1] = 0 \; for \; i = 1, 2, \ldots, n \tag{26}$$

It states that $a_{i0} \neq 0$ only for the points which satisfy $y_i(w_0^T x_i + b_0) = 1$. These points are called the support vectors.

To sum up, we have:

$$f(x) = \sum_{i=1}^{m} a_{i0} y_i x_i^T x + b_0 \tag{27}$$

where $\{x_i\}_{i=1}^m$ are the support vectors and $\{a_{i0}\}_{i=1}^m$ are the corresponding Lagrange multipliers.

Non-linear separable data and soft margin In practical applications, many of the data sets are non-linearly separable which makes the algorithm in the previous section infeasible. One example is shown in Figure 3. As can be seen from the figure, although most of the points are classified into the correct side, there are still some points which violate the hyperplane. These points either cross the boundary of the margin but are still located on the correct half-space, or have been misclassified onto the incorrect half-space. In such cases, it is impossible to find a hyperplane which completely removes the errors; instead, a solution can be proposed to minimize the errors on the training data.

Slack variables are introduced to solve this problem. For a data set with n samples, there are n slack variables $\{\varepsilon_i\}_{i=1}^n$ which satisfy:

$$\forall\, y_i = 1: \; w^T x_i + b \geqslant 1 - \varepsilon_i \tag{28}$$

$$\forall\, y_i = -1: \; w^T x_i + b \leqslant -1 + \varepsilon_i \tag{29}$$

The slack variable ε_i here is a measure of the violation to the margin. If $0 < \varepsilon_i < 1$, then the sample violates the margin but is still correctly classified. When $\varepsilon_i > 1$, the sample is classified into the wrong half-space. Since the goal is to have fewer training samples misclassified, a penalty force can be added:

$$\eta(\varepsilon) = \sum_{i=1}^{n} \varepsilon_i \tag{30}$$

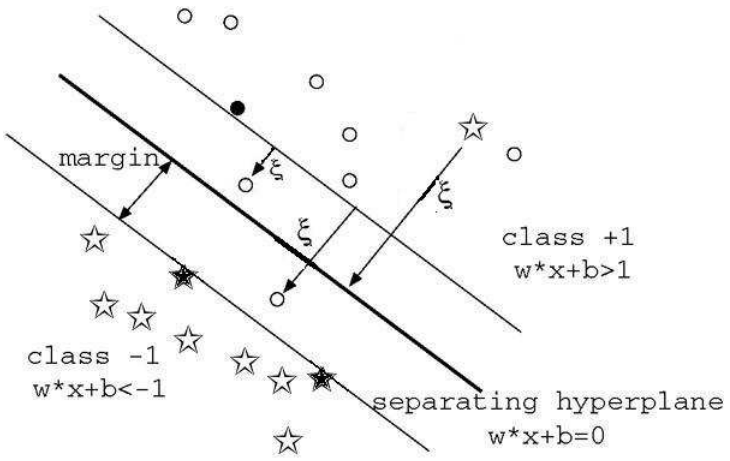

Figure 3: Non-linearly separable situation in SVM

which should be minimized. It can be incorporated into the cost function in the previous section as:

$$f = \frac{1}{2} w^T w + C \sum_{i=1}^{n} \varepsilon_i \tag{31}$$

The parameter C here controls the trade-off between the margin rigidity enforcement and the number of errors it can tolerate during training. A larger value of C will produce a more accurate model while at the same time increasing the risk of over-fitting; therefore, the value of C has to be optimized by the user during the experiment.

The corresponding Lagrange function for this problem is:

$$J(w, b, a, u, \varepsilon) = \frac{1}{2} w^T w + C \sum_{i=1}^{n} \varepsilon_i - \sum_{i=1}^{n} \mu_i \varepsilon_i - \sum_{i=1}^{n} \alpha_i [y_i (w^T x_i + b) - 1 + \varepsilon_i] \tag{32}$$

where μ_i is the Lagrange multiplier for the slack variables.

Kernel trick Consider the typical XOR problem which tries to separate four examples in four corners of a rectangle such that the two examples connected by a diagonal belong to the same class. It is impossible to make this in a two-dimensional space but when projecting it to a three-dimensional space, it becomes much easier. This example indicates that a non-linearly separable data set may become linearly separable in a higher dimensional space. This kind of mapping increases the separability of the data set.

Let the function θ defines the non-linear mapping:

$$\theta : R^N \to H \tag{33}$$

Therefore, the discriminant function can be formulated as:

$$f(x) = \sum_{i=1}^{n} a_i y_i \theta(x_i)^T \theta(x) + b \tag{34}$$

The kernel function is defined here by:

$$K(x, y) = \theta(x)^T \theta(y) \tag{35}$$

and the discriminant function turns into:

$$f(x) = \sum_{i=1}^{n} a_i y_i K(x_i, x) + b \tag{36}$$

This expression avoids providing the exact representation in a higher dimensional space. Numerous kernels have been proposed to solve various kinds of problems. One of the most popular kernels is the RBF kernel which is used in this thesis. The RBF kernel can be expressed as:

$$K(x, y) = e^{(-\frac{1}{2\sigma^2}\|x-y\|^2)} \tag{37}$$

σ indicates the kernel width. A larger σ indicates a smoother function to avoid overfitting and also avoid reproducing the noises in the training data; On the other hand, a smaller σ implies a more flexible function to produce highly irregular decision boundaries. Hence, it is important to determine the optimal value for σ by means of cross validation.

2.3.2 LVQ

Learning Vector Quantization is one of the most famous prototype-based supervised learning methods which was first introduced by Kohonen [3]. It has some advantages over the other methods. Firstly, this method can be easily implemented and the complexity of the classifier can be controlled and determined by the user. Secondly, multi-class problems can be naturally tackled by the classifier without modifying the learning algorithm or decision rule. Lastly, the resulting classifier is intuitive and easy to interpret due to its assignment of class prototypes and intuitive classification mechanism of new data points to the closest prototype. The resulting prototypes can then provide class-specific attributes for the data. This is a big advantage over the methods such as SVM or Neural Networks which suffer from the drawback of being like a black box and because of that, LVQ has been applied into many fields, such as bioinformatics, satellite remote sensing and image analysis [34, 35, 39]

Training data for LVQ can be denoted as:

$$\{x_i, y_i\}_{i=1}^{n} \in R^N \times \{1, 2, \ldots C\} \tag{38}$$

where x_i denotes the data in N dimensional space and y_i is the label with C different classes.

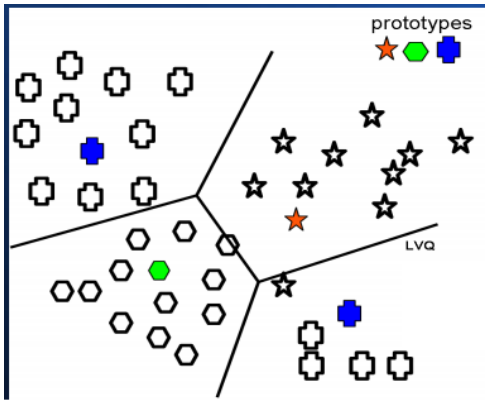

Figure 4: Example for LVQ with 3 Different Prototypes

LVQ can be parameterized by a set of prototypes representing the classes in feature space and the distance measurement which may be a traditional Euclidean distance or a full matrix trained from the data. One of the examples can be seen in Figure 4 where there are 4 different prototypes representing 3 different classes.

Traditional LVQ employs Euclidean distance measurement and is based on nearest prototype classification. To be more specific, a set of prototypes are defined to represent the different classes. If one prototype per class is defined, the prototypes can be represented as: $W = \{w_j, c(w_j)\} \in R^N \times \{1, 2, \ldots, C\}$. Each unseen example x_{new} will be assigned a label whose prototype has the closest distance to it with respect to the distance measurement:

$$c(x_{new}) \leftarrow c(w_k)\, with\, w_k = \underset{j}{argmin}\, d(w_j, x_{new}) \tag{39}$$

It is called a winner-takes-all strategy.

Training of this model is guided by the minimization of the cost function:

$$F = \sum_{i=1}^{n} \phi(\varepsilon_i)\, with\, \varepsilon_i = \frac{d(x_i, w_H) - d(x_i w_M)}{d(x_i, w_H) + d(x_i w_M)} \tag{40}$$

where ϕ is any monotonic function and in this thesis, $\phi(x) = x$; w_H and w_M are respectively the closest prototype with the same and different label to sample x_i:

$$w_H = \underset{j}{argmin}\, d(x_i, w_j)\, \forall c(w_j) = c(x_i) \tag{41}$$

$$w_M = \underset{j}{argmin}\, d(x_i, w_j)\, \forall c(w_j) \neq c(x_i) \tag{42}$$

In traditional LVQ systems, only the locations of the prototypes are updated during the training to minimize the errors. w_H is pushed toward the sample x_i

and w_M is pushed away from it. Their derivatives to the cost function F are expressed as:

$$\triangle w_H = -\alpha \cdot \phi^{'}(\varepsilon_i) \cdot \varepsilon^{'}_{i,H} \cdot \nabla_{w_H} d(x_i, w_H) \tag{43}$$

$$\triangle w_M = \alpha \cdot \phi^{'}(\varepsilon_i) \cdot \varepsilon^{'}_{i,M} \cdot \nabla_{w_M} d(x_i, w_M) \tag{44}$$

where α is the learning rate; $\phi^{'}(\varepsilon_i) = 1$ because $\phi(x) = x$; $\varepsilon^{'}_{i,H} = 2 \cdot d(x_i w_M)/[d(x_i, w_H)+d(x_i w_M)]^2$ and $\varepsilon^{'}_{i,M} = 2 \cdot d(x_i w_H)/[d(x_i, w_H)+d(x_i w_M)]^2$; $\nabla_{w_H} d(x_i, w_H)$ and $\nabla_{w_M} d(x_i, w_H)$ are respectively the derivatives of w_H and w_M to the distance $d(x_i, w_{H\,or\,M})$ and therefore depend on the distance measurement.

2.3.3 Two variants of LVQ: GRLVQ and GMLVQ

How the distance is calculated is very important in the LVQ system. One of the most popular metrics is the Euclidean distance which is a special case of Minkowski distance. The Euclidean distance from a data point x_i to a prototype w can be expressed as:

$$d(w, x_i) = \sqrt{\sum_{j=1}^{N} (x_i^j - w^j)^2} \tag{45}$$

The Euclidean distance assigns the same weight for each feature, indicating that each feature has the same contribution for classification. However, in practical applications, it is usually observed that different features contribute differently toward the classification. Therefore, relevance learning [7, 4] is proposed to assign adaptive weight values for different feature inputs:

$$d(w, x_i) = \sqrt{\sum_{j=1}^{N} \lambda_j (x_i^j - w^j)^2} \tag{46}$$

The corresponding LVQ system is called GRLVQ [7, 4].

Each feature, besides their individual contribution for the classification, will also correlate with the others to influence the performance. Generalized Matrix LVQ (GMLVQ) [38] is proposed to extend the previous methods. A full matrix of adaptive relevance is employed as the similarity metric and the distance is calculated as:

$$d(w, x_i) = (x_i - w)^T \Lambda (x_i - w) \tag{47}$$

where Λ is a full $N \times N$ matrix whose off-diagonal element $\Lambda_{i,j}$ account for the contribution of feature pair i and j for classification. The matrix Λ has to be positive definite to keep the distance result positive. Its positive definiteness is achieved by constructing:

$$\Lambda = \Omega^T \Omega \tag{48}$$

where Ω is an arbitrary real $M \times N$ matrix with $M \leqslant N$. However, in this thesis, we only consider the case : $M = N$. Substituting Eq. (42) into Eq. (41), obtain:

$$(x_i - w)^T \Lambda (x_i - w) = (x_i - w)^T \Omega^T \Omega (x_i - w) = [\Omega(x_i - w)]^2 \geqslant 0 \tag{49}$$

It is noticed the GRLVQ is a special case of GMLVQ with $diag(\Lambda) = \{\lambda_i\}_{i=1}^N$.

The derivative of the distance $d(w, x_i)$ with respect to prototype w is:

$$\nabla_w d(w, x_i) = -2\Lambda(x_i - w) \tag{50}$$

Substituting Eq. (50) into Eq. (43) and Eq. (44), we can obtain the update rule for closest correct and incorrect prototype.

In the model of GMLVQ, the update rule of the distance matrix Ω also need to be computed. The derivative of $d(w, x_i)$ with respect to one single element Ω_{lm} is:

$$\begin{aligned} \nabla_{\Omega_{lm}} d(w, x_i) &= \sum_k (x_i^m - w^m)\Omega_{lk}(x_i^k - w^k) + \sum_j (x_i^j - w^j)\Omega_{lj}(x_i^m - w^m) \quad (51) \\ &= 2 \cdot (x_i^m - w^m)[\Omega(x_i - w)]_l, \quad (52) \end{aligned}$$

The derivative of the cost function F with respect to one single element Ω_{lm} can then be expressed as:

$$\begin{aligned} \triangle\Omega_{lm} &= \triangle\Omega_{lm}^H + \triangle\Omega_{lm}^M \quad (53) \\ &= -\beta \cdot 2 \cdot \phi^{'}(\varepsilon_i) \cdot \varepsilon_{i,H}^{'} \cdot \nabla_{\Omega_{lm}^H} d(x_i, w_H) + \beta \cdot 2 \cdot \phi^{'}(\varepsilon_i) \cdot \varepsilon_{i,M}^{'} \cdot \nabla_{\Omega_{lm}^M} d(x_i, w_M) \end{aligned}$$

where β is the learning rate for Ω.

Chapter 3

3 Feature Selection

3.1 Challenge

In this section, two topics about the challenges in feature selection will be discussed. The first issue about the curse of dimensionality and the second one is the relevance and redundancy of features.

3.1.1 Curse of dimensionality

In machine learning, the term curse of dimensionality was initially defined by Richard Bellman [10] when he conducted the work on dynamic optimization [9, 10] and found it quite difficult to tackle the problem of the curse of dimensionality. He stated:

> "In view of all that we said in the foregoing sections, the many obstacles we appear to have surmounted, what casts the pall over our victory celebration? It is the curse of dimensionality, a malediction that has plagued the scientist from the earliest days." [10]

Up to date, there are already many definitions about it, but generally it refers to the problem incurred by adding extra features to the space. The reliability of the learning model depends on the density of training examples in the feature space. The increase of data dimensionality will sparse the feature space and thus deteriorate the generalization performance.

It states that the predictive performance of a learning algorithm will deteriorate with the increase of data dimensionality. With the increase of the feature space, the feature space will become more sparse and more training examples are required. For example, if 5 samples are enough in each dimension, then 25 samples are sufficient to fill a two-dimensional cube. However, this number will increase to 5^{20} for a 20-dimensional hypercube.

It is also observed that it becomes more difficult to estimate the kernel in a higher dimension [11]. Table 1 illustrates the number of samples required to estimate a kernel at density 0 with a certain accuracy.

Dimensionality	Sample Size
1	4
2	19
5	786
7	10,700
10	842,000

Table 1: Sample size required for kernel estimation [11].

3.1.2 Irrelevance and redundancy

There are some controversies in the definition of feature relevance. There is a review [8] which introduces the different relevance definitions that have been proposed in the literature. The authors then present an example to indicate that all the other relevance definitions produce unexpected results and based on that, the authors suggest that two different degrees of relevance are required: strong relevance and weak relevance. The definition of weak relevance can also be regarded as the definition of redundancy.

Let $< X, Y >$ denote the training examples where $X \in R^N$ is the data and Y indicates the labels. Let F be the full feature set and F_i is the i^{th} feature; therefore each instance X is one element of the combination of the set $F_1 \times F_2 \times \cdots F_N$. Let $S_i = F - \{F_i\}$ denote the feature subset with all features except for F_i and s_i denote one value instantiation of S_i. Let P denote the conditional probability of the label Y given a feature subset.

Strong relevance A feature F_i is strongly relevant iff

$\exists x_i \in F_i, y \in Y$ where $P(x_i, s_i) > 0$ and $P(Y = y|S_i = s_i, F_i = x_i) \neq P(Y = y|S_i = s_i)$

Weak relevance A feature F_i is weakly relevant iff it is not strongly relevant and

$\exists x_i \in F_i, s_i \subseteq S_i, y \in Y$, such that $P(Y = y|S_i = s_i, F_i = x_i) \neq P(Y = y|S_i = s_i)$

A feature F_i is called relevant if it is either strongly or weakly relevant to the class label; otherwise it is irrelevant. A feature F_i which is weakly relevant can become strongly relevant after removing a certain feature subset. The weak relevance can be interpreted as the existence of other relevant features which can provide similar prediction power as the one we are measuring. This is also what we call redundant. It is important to note that the feature F_i which is weakly relevant or redundant should not be removed if the feature subset whose removal makes F_i strongly relevant has been removed by the feature selection algorithm.

3.2 General Framework

The framework in Figure 5 shows that a typical feature selection system usually consists of four components. They include: feature subset generation, feature subset evaluation, stopping criterion and feature subset validation. As indicated in the figure, the complete feature set is firstly sent to the “Generation" model which produces different feature subset candidates based on some search strategy. Each subset candidate will then be evaluated in the “Evaluation" model by a certain evaluation measurement. A new subset which turns out to be better will replace the previous best one. This subset generation and evaluation will be repeated over and over until the given stopping criterion is met. After that,

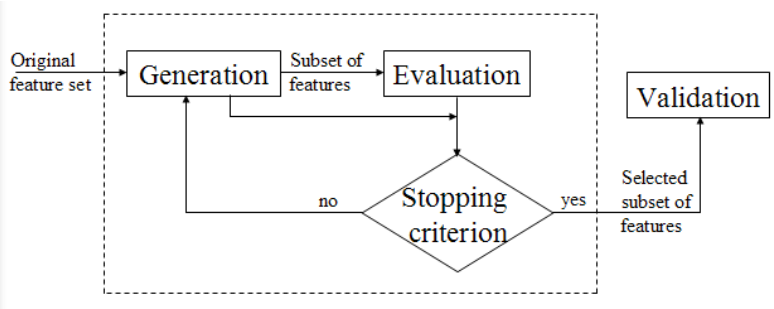

Figure 5: Framework of Feature Selection

the ultimately selected feature subset will be sent to the "Validation" model for validation by certain learning algorithms.

Two basic issues have to be addressed in the "Generation model". Those are: Starting point and search strategy.

- Starting Point. Choose a point to start the search in the feature space. One choice is to begin with no feature and then for each iteration, expand the current feature subset with each feature that is not yet in the subset. The feature whose addition produces the best evaluation performance is added to the current subset. This is called forward selection. Another option is to do it conversely. The search starts with a full feature set and then successively eliminates the feature whose removal results in the best evaluation performance. This search is called backward selection. A third alternative is to start by selecting a random feature subset [13] and then successively add or remove features depending on the performance. This random approach can avoid being trapped into local optima.

- Search Strategy. There are three different search strategies: complete, heuristic and random. The complete strategy examines all the possible feature subsets and guarantees to find the optimal one. When there are N features, the search will examine 2^N subsets which makes it unrealistic for large N. Heuristic search is guided by some heuristic. It is less computationally demanding but the optimal subset is not guaranteed. The guideline determines whether or not a better subset can be found. The random strategy just simply chooses the next feature at random; therefore, the probability to find the optimal subset depends on how many epochs are tried.

3.3 Wrapper and Filter Approach

The evaluation methods in feature selection can be generally divided into two basic models: the filter model [14, 15, 16, 17] and the wrapper model [18, 19, 20].

The filter model selects a feature subset as a pre-processing step, without considering the predictor performance. It is usually achieved by designing an

evaluation function and then choosing a set of features to maximize it. Some evaluation functions that are frequently used are distance measures, information measures, dependency measures and consistency measures. The filter model does not involve any training of learning algorithm and is thus much faster which makes it suitable to be applied on large data sets.

In the wrapper model, a predetermined data mining algorithm is utilized to evaluate the feature subset and the candidate with highest prediction performance will be selected as the final subset. The wrapper model can usually select a feature subset with superior performance because it selects features better suited to the predetermined algorithm. However, because the algorithm has to be trained and tested for each subset candidate, the wrapper model tends to be very computationally expensive, especially with large feature size.

3.4 Feature Ranking Technique

3.4.1 Information gain

Information gain [21] measures the dependency between a feature X_i and the class label Y. It is a very popular technique in feature selection because it is easy to understand and compute. Information gain can also be regarded as a measure of the reduction in uncertainty about a feature X_i when the value of Y is known. Uncertainty is usually measured by Shannon's entropy:

Entropy Entropy measures the amount of uncertainty that a feature X_i contains. It is given by

$$H(X_i) = -\sum_{j=1}^{p} P(j) log_2 P(j) \tag{54}$$

Where p is the number of possible values in X_i and $P(j)$ indicates the observation probability of the value j. Form this formula, a more uniform distribution tends to produce a higher entropy. For example, if you toss a fair coin, there are two possible values each with equal probability 0.5. Its entropy value is

$$H(coinToss) = -2 \times (0.5 \times log_2 0.5) = 1$$

In another example, if you toss a die, there are six possible outcomes, each with probability 1/6. Its entropy value is

$$H(diceToss) = -6 \times ((1/6) \times log_2(1/6)) = 2.585$$

Therefore, the higher the entropy is, the more uncertainty it contains and the more difficult to predict the output.

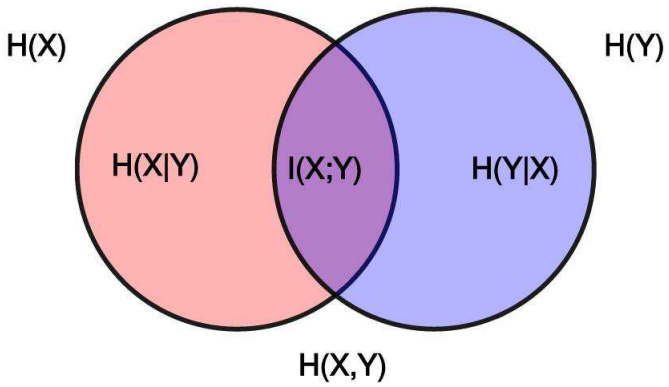

Figure 6: Example to illustrate the algorithm Information Gain

Information Gain The information gain of a feature X_i and the label Y is:

$$I(X_i, Y) = H(X_i) - H(X_i \mid Y) \tag{55}$$

Where $H(X_i)$ and $H(X_i \mid Y)$ are respectively the entropy of feature X_i and the entropy of X_i after knowing the value of Y. $H(X_i \mid Y)$ is calculated as

$$H(X_i \mid Y) = -\sum_j P(Y_j) \sum_k P(x_k \mid Y_j) log_2 P(x_k \mid Y_j) \tag{56}$$

A better understanding can be gained from Figure 6. As can be seen in Figure 6, $H(X)$ and $H(Y)$ respectively measure the entropy of X and Y. The information gain $I(X, Y)$ is a measure of the information shared by X and Y. $H(X, Y)$ is the information that X and Y collectively contain.

$$H(X,Y) = -\sum_x \sum_y p(x,y) log_2 p(x,y) = H(X \mid Y) + I(X,Y) + H(Y \mid X) = H(X) + H(Y) - I(X,Y) \tag{57}$$

If X and Y are highly correlated, then the information they share is very high, indicating a large value of $I(X, Y)$. Then if Y is known, much of the information about X can be "guessed" from Y, suggesting a low $H(X \mid Y)$ and vice versa for $H(Y \mid X)$.

3.4.2 Relieff

Relief [22] is a univariate feature weighting algorithm in the filter model. It is based on the principle that the attribute which can better separate similar instances but with different classes is more important and should be assigned a larger weight. The three basic steps to compute the feature weight are:

Algorithm 1 Pseudo code of Relief algorithm

Description: There are P instances described by N features and there are C different classes: $x \in R^N\, c(x) \in \{1, 2, \ldots C\}$; T iterations are performed.

1. Set all the feature weights to 0: $\forall i, w(i) = 0$;
2. For t = 1 to T, do:
3. Randomly pick an instance $x \in R^N$;
4. Find nearest hit $NH(x)$ and nearest miss $NM(x)$:
5. $NH(x) \leftarrow x_h, with\, x_h = \underset{j}{argmin}\, d(x, x_j) \forall c(x_j) = c(x)$
6. $NM(x) \leftarrow x_m\, with\, x_m = \underset{k}{argmin}\, d(x, x_k) \forall c(x_k) \neq c(x)$
7. For i =1 to N, do:
8. $w(i) = w(i) + d(x_i, NM(x)_i)/(P \times T) - d(x_i, NH(x)_i)/(P \times T)$
9. end do.
10. end do.

1. Find the nearest miss and nearest hit where nearest hit is the closest sample with the same class as the test sample and nearest miss specifies the closest sample with a different label as the test sample;
2. Calculate the weight of a feature;
3. Return a ranked list of feature weights or the top k features according to a given threshold;

The algorithm starts by initializing all the feature weights to be zero and it randomly select an instance from the samples and calculates its nearest hit NH and nearest miss NM. Each feature weight is then updated based on its ability to discriminate NH and NM. The detailed pseudo code is given in Algorithm 1.

Relieff [23] extends the original Relief algorithm to deal with the multi-class situation. It incorporates two important improvements. First, the result is more robust to noises because of the consideration of k nearest neighbourhoods. Second, it can deal with the multi-class problem. The detailed pseudo code is shown in Algorithm 2.

3.4.3 Fisher

Fisher [40] is an effective supervised feature selection algorithm which aims to select features that assign similar values to the same class and different values to different classes. The evaluation score of Fisher's algorithm is:

Algorithm 2 Pseudo code of Relieff algorithm

Description: Instances described by N features and there are C different classes: $x \in R^N$, $c(x) \in \{1, 2, \ldots C\}$; Look for k nearest neighbours; perform T iteration; $p(y)$ the class probability specifying the probability of an instance being from the class y.

1. Set all the feature weights to 0: $\forall i, w(i) = 0$;
2. For t = 1 to T, do:
3. Randomly pick an instance $x \in R^N$ with label y_x;
4. for y = 1 to C, do
5. find k nearest instances of x from class y: $x(y, l)$ where $l = 1, 2, \cdots k$
6. for i = 1 to N, do:
7. for l = 1 to k, do:
8. if $y = y_x$ (nearest hit), then
9. $w(i) = w(i) - \frac{d(x_i - x(y,l)_i)}{T \times n}$
10. else (nearest miss),
11. $w(i) = w(i) + \frac{p(y)}{1 - p(y_x)} \times \frac{d(x_i - x(y,l)_i)}{T \times n}$
12. end if.
13. end for.
14. end for.
15. end for.
16. end for.

$$Fisher(f_i) = \frac{\sum_{j=1}^{c} n_j(\mu_{i,j} - \mu_i)}{\sum_{j=1}^{c} n_j \sigma_{i,j}^2} \tag{58}$$

where f_i is the i^{th} feature to be evaluated, n_j is the number of instances in class j, μ_i is the mean of feature i, $\mu_{i,j}$ and $\sigma_{i,j}$ are respectively the mean and the variance of feature i on class j. Fisher algorithm is computationally effective and widely applied in many applications, however, because it considers the features individually, it has no ability to deal with redundant features.

Chapter 4

4 GMLVQ Based Feature Selection Algorithms

4.1 Entropy Enforcement for Feature Ranking Results

It is stated that the element $\Lambda_{i,j}$ in matrix Λ measures the correlation between feature i and j and the diagonal element $\Lambda_{i,i}$ quantifies the contribution of feature i for classification. The above statement only makes sense when the features have similar magnitude, therefore a z-score transformation is always performed on the data before the training starts. One example is shown in Figure 7 where 32 features are ranked with respect to their value of the diagonal elements. The 19^{th} feature contains the highest value, indicating that it has the largest correlation with the classification. Another constraint is added so that after each adaption, the sum of the diagonal elements is equal to zero:

$$\sum_{i=1}^{N} \Lambda_{i,i} = 1 \tag{59}$$

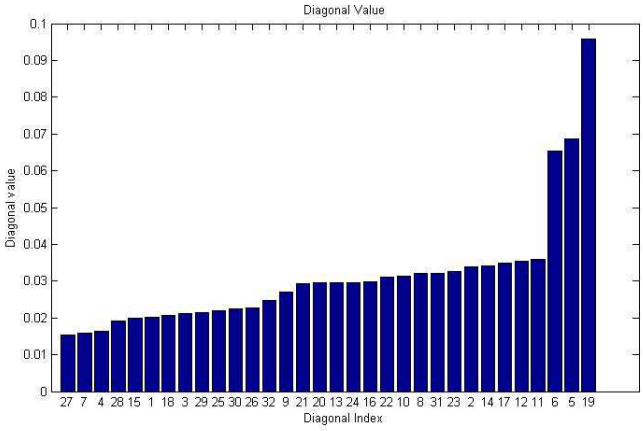

Figure 7: One example of diagonal elements of Λ

One of the ideal situations in feature ranking is that some of the features are much more important than others and the least important features can therefore be removed from the feature set without deteriorating the classification performance. An external entropy force is added to the cost function to push the diagonal elements to this ideal situation.

The definition of the entropy force is:

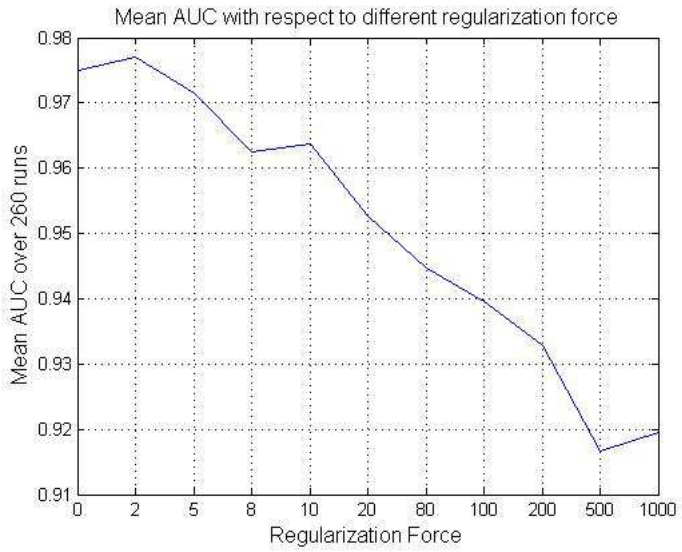

Figure 8: Classification performance with respect to different values of regularization force

$$Entropy(\Lambda_{diag}) = -\sum_{i=1}^{N} \Lambda_{i,i} log_2 \Lambda_{i,i} \tag{60}$$

where N is the data dimension. This external force will reach the maximum when all the diagonal elements are equal, i.e. all the features are equally important for classification. Its minimization will, on the other hand, push to generate a discriminative feature relevance and at the extreme, only one feature is identified as relevant for classification and the relevances of other features are zero.

It is integrated into the cost function by:

$$F^{new} = F + \alpha \times Entropy(\Lambda_{diag}) \tag{61}$$

where the regularization parameter α controls the trade-off between the classification accuracy and the discrimination between features. A larger value of α will produce a more discriminative feature ranking result by sacrificing the classification performance. Their mutual relation on one of the data sets is visualized in Figure 8.

The choice of the regularization value depends on how important the accuracy and the discrimination are for the user and differs per data set. A safe way is to post their relation for each data set and choose the optimal point based on the requirement. A more efficient way in this thesis is to choose the value by $\frac{N^2}{10}$ where N is the data dimension. On all of the data sets conducted in this thesis,

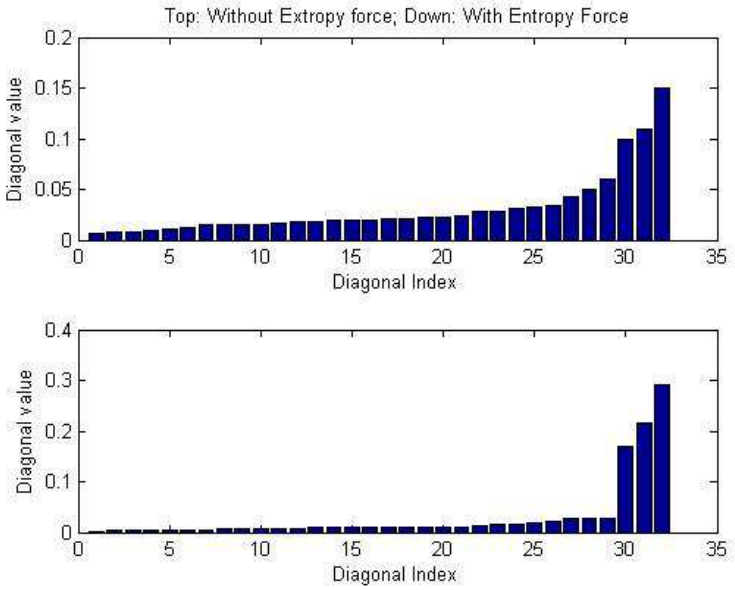

Figure 9: Comparison of feature ranking result with and without entropy force

such a value can generate a considerable discriminative feature ranking result without deteriorating the performance to a large extent. One of the examples with and without entropy force can be seen in Figure 9.

4.2 Way-Point Average Algorithm

Gradient based minimization is a popular and powerful method in non-linear optimization [31]. In this thesis, batch gradient descent is employed to train the GMLVQ model. One of the critical choices in gradient descent methods is the appropriate choice of the step size. Too small step sizes will slow the convergence, however large steps can result in oscillatory or even divergent behavior.

In this section, a modification of batch gradient descent [32] is introduced which aims at better convergence behavior. The idea is that, during the training procedure, we compare the cost function of normal descent adaption with that of the gliding average over the most recent steps and if the latter produces a lower optimization value, minimization jumps to the latter configuration and decreases the step size at the same time. A more detailed description is described below.

Consider we want to minimize an objective function F with respect to a N-dimensional vector $x \in R^N$. A gradient descent process is started at x_0 and proceeds to generate a sequence of steps iteratively:

$$x_{t+1} = x_t - a_t \frac{\nabla F}{|\nabla F|} \tag{62}$$

Note that the gradient has been normalized by $\frac{\nabla F}{|\nabla F|}$ and therefore, a_t here is exactly the step size length during adaption: $|x_{t+1} - x_t| = a_t$.

The waypoint averaging algorithm starts at x_0 with initial step size a_0 and performs k steps with gradient steps unchanged:

$$x_{t+1} = x_t - a_t \frac{\nabla F}{|\nabla F|} \quad for\, t = 0, 1, 2 \ldots k-1\, with\, a_t = a_0 \tag{63}$$

After that ($t \geqslant k$), the procedure proceeds as below:

1. perform the normal gradient descent step and evaluate the corresponding cost function:

$$x^*_{t+1} = x_t - a_t \frac{\nabla F}{|\nabla F|_t}\, and\, calculate\, F(x^*_{t+1}) \tag{64}$$

2. perform the waypoint average over the previous j steps:

$$\overline{x_{t+1}} = \frac{1}{j} \sum_{i=0}^{j-1} x_{t-i}\, and\, calculate\, F(\overline{x_{t+1}}) \tag{65}$$

3. determine the new step size and adaption position by comparison:

$$\begin{cases} x_{t+1}{=}x^*_{t+1}\, and\, a_{t+1}{=}a_t\, if\, F(x^*_{t+1}){\leqslant}F(\overline{x_{t+1}}) \\ x_{t+1}{=}\overline{x_{t+1}}\, and\, a_{t+1}{=}\lambda a_t\, else \end{cases} \tag{66}$$

with the parameter $0 < \lambda < 1$.

As can be seen from the algorithm, as long as the normal gradient descent procedure produces a position with lower cost than the waypoint average algorithm, the iteration proceeds as a normal gradient descent algorithm.

On the other hand, $F(\overline{x_{t+1}}) < F(x^*_{t+1})$ indicates the potential existence of oscillatory behavior because under oscillatory condition, the position fluctuates around the local minimum and it is expected that the average over the previous steps may provide a closer estimate to the minimum than the normal gradient descent adaption. It also indicates that the step size may be too large to get to the minimum and should be decreased for better convergence.

An intuitive example is shown in Figure 10 [32] which visualizes the adaption steps of both the normal gradient descent and waypoint averaging algorithm. The dotted lines mark the update trajectory of normal gradient descent algorithm with constant step sizes which display strong oscillatory behavior. The waypoint averaging algorithm shares the same trajectory with the normal gradient descent in the first four steps. However, after that it jumps to the average position over the previous steps and reduces the step size at the same time which enables it to move closer to the minimum in the middle.

When considering its application in GMLVQ, since the cost function in GMLVQ has to be optimized with respect to both the prototype w and the matrix Ω, two independent waypoint averaging algorithms about w and Ω have to be performed. The typical scheme is formulated as below:

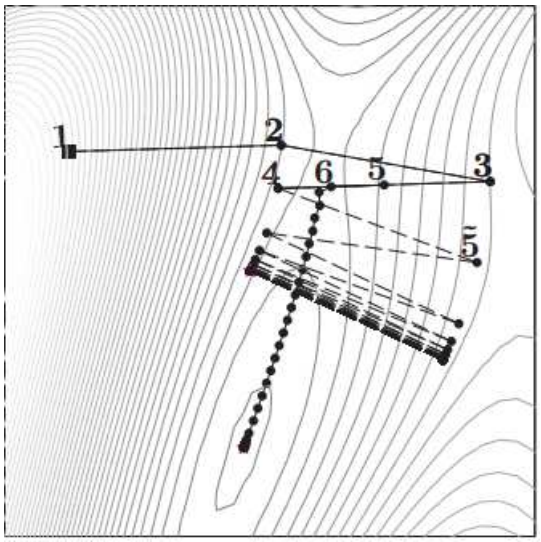

Figure 10: Comparison between way-point average algorithm and normal gradient descent. From [32]

Given a GMLVQ system represented by Ω and a set of prototypes $\{w_k\}_{k=1}^{M}$ with cost function represented by F, respectively choose the start points Ω^0 and $\{w_k^0\}_{k=1}^{M}$ and initial step sizes a_0^Ω and a_0^w for Ω and w;

1. perform k (k=3 in this thesis) steps with gradient steps unchanged:

$$\Omega_{t+1} = \Omega_t - a_t^\Omega \frac{\nabla F}{|\nabla F|} \quad for\, t = 0, 1, 2 \ldots k-1\, with\, a_t^\Omega = a_0^\Omega \tag{67}$$

$$w_{t+1} = w_t - a_t^w \frac{\nabla F}{|\nabla F|} \quad for\, t = 0, 1, 2 \ldots k-1\, with\, a_t^w = a_0^w \tag{68}$$

After that ($t \geqslant k$), the procedure proceeds as below:

2. perform the normal gradient descent step and evaluate the corresponding cost function for both Ω and w:

$$\Omega_{t+1}^* = \Omega_t - a_t^\Omega \frac{\nabla F}{|\nabla F|_t}\, and\, calculate\, F(\Omega_{t+1}^*) \tag{69}$$

$$w_{t+1}^* = w_t - a_t^w \frac{\nabla F}{|\nabla F|_t}\, and\, calculate\, F(w_{t+1}^*) \tag{70}$$

3. perform the waypoint average over the least previous j (j=3 in this thesis) steps:

$$\overline{\Omega_{t+1}} = \frac{1}{j} \sum_{i=0}^{j-1} \Omega_{t-i}\, and\, calculate\, F(\overline{\Omega_{t+1}}) \tag{71}$$

$$\overline{w_{t+1}} = \frac{1}{j}\sum_{i=0}^{j-1} w_{t-i} \, and \, calculate \, F(\overline{w_{t+1}}) \tag{72}$$

4. determine the new step size and adaption position for both Ω and w:

$$\begin{cases} \Omega_{t+1}{=}\Omega^*_{t+1} \, and \, a^{\Omega}_{t+1}{=}a^{\Omega}_t \, if \, F(\Omega^*_{t+1}){\leqslant}F(\overline{\Omega_{t+1}}) \\ \Omega_{t+1}{=}\overline{\Omega_{t+1}} \, and \, a^{\Omega}_{t+1}{=}\lambda a^{\Omega}_t \, else \end{cases} \tag{73}$$

$$\begin{cases} w_{t+1}{=}w^*_{t+1} \, and \, a^{w}_{t+1}{=}a^{w}_t \, if \, F(w^*_{t+1}){\leqslant}F(\overline{w_{t+1}}) \\ w_{t+1}{=}\overline{w_{t+1}} \, and \, a^{w}_{t+1}{=}\lambda a^{w}_t \, else \end{cases} \tag{74}$$

with the parameter $\lambda = 2/3$.

4.3 Feature Ranking Ambiguity Removal

Up to this step, we have the input vectors and class labels:

$$\{x, y_i\}_{i=1}^{n} \qquad with \, x_i \in R^N, \quad y_i \in \{1, 2, \ldots C\} \tag{75}$$

associated with a set of prototypes:

$$\{w_k\}_{k=1}^{M} \quad where \, M \geqslant C \tag{76}$$

And the distance is calculated as:

$$d(x_i, w_k) = (x_i - w_k)^T \Lambda (x_i - w_k) = (x_i - w_k)^T \Omega^T \Omega (x_i - w_k) = \mid \Omega(x_i - w_k) \mid^2 \tag{77}$$

where $\Lambda, \Omega \in R^{N \times N}$ and $\Omega = [z_1, z_2, \cdots z_N]^T$ where $\{z_i\}_{i=1}^{N}$ are column vectors with dimension N. The feature ranking results can be obtained from the values of diagonal elements in matrix Λ. However, an issue is raised whether there is another matrix Λ which can keep the distance measurement unchanged. It that matrix exists, the feature ranking results can be different without modifying the classifier, which means the feature ranking results we have obtained in previous steps are not unique.

Consider a vector v_j which satisfies the following constraints:

$$\forall \, i : \, v_j^T x_i = 0 \tag{78}$$

$$\forall \, k : \, v_j^T w_k = 0 \tag{79}$$

If we add such a vector v_j to any row z_i^T of the matrix Ω, consider, for instance, i = 1:

$$\Omega_{new} = [z_1 + v_j, z_2, \cdots z_N]^T \tag{80}$$

we can easily verify that the following mappings keep unchanged:

$$\forall\, i:\ \Omega_{new}\, x_i = \Omega x_i \tag{81}$$

$$\forall\, k:\ \Omega_{new}\, w_k = \Omega w_k \tag{82}$$

Therefore, the distances between any pair of input samples and prototypes will keep the same:

$$d(x_i, w_k) = \mid \Omega(x_i - w_k) \mid^2 = \mid \Omega_{new}(x_i - w_k) \mid^2 \quad for\, all\, i, k \tag{83}$$

Since the mapping and distance calculation are the same between Ω and Ω_{new}, the cost functions, classification errors and classifiers they produce will also stay the same. However, the feature ranking results may vary between Ω and Ω_{new}, because there is no constraint on the consistency of their diagonal elements in matrix Λ and Λ^{new}.

Without loss of generality, we assume that there are J such spurious vectors $\{v_j\}_{j=1}^{J}$ and as it will be proved in later stages all such vectors are actually eigenvectors of a construction matrix, we can additionally assume that all the vectors $\{v_j\}_{j=1}^{J}$ are orthonormal:

$$v_j \bullet v_k = \delta_{jk} = \begin{cases} 1 & if\ j{=}k \\ 0 & otherwise \end{cases} \tag{84}$$

The proposed solution is to project out all the spurious directions $\{v_j\}_{j=1}^{J}$ from a given matrix Ω:

$$\Omega_{new}^T = [I - \sum_{j=1}^{J} v_j v_j^T]\Omega^T \tag{85}$$

It follows that:

$$\Omega_{new}^T(x_i - w_k) = \Omega^T(x_i - w_k) - \sum_{j=1}^{J} \underbrace{v_j(x_i - w_k)}_{0} v_j^T \Omega^T = \Omega^T(x_i - w_k) \tag{86}$$

$$v_k^T \Omega_{new} = v_k^T \Omega - \sum_{j=1}^{J} \underbrace{v_j v_k}_{\delta_{jk}} v_j^T \Omega = v_k^T \Omega - v_k^T \Omega = 0 \tag{87}$$

Hence, we can interpret the resulting matrix Ω_{new} as the minimal representation of the mapping which contains no contribution of the spurious direction v_j.

The next question is how to find all these vectors $\{v_j\}_{j=1}^{J}$. The conditions that $\forall\, i:\ v_j^T x_i = 0$ and $\forall\, k:\ v_j^T w_k = 0$ can be rewritten as

$$X^T v_j = 0 \quad where\, X = [x_1, x_2, \cdots x_L, w_1, w_2, \cdots w_M] \tag{88}$$

This in turn, is equivalent to

$$[X^T v_j]^2 = 0 \Leftrightarrow v_j^T C v_j = 0 \quad where\, C = XX^T = \sum_{i=1}^{L} x_i x_i^T + \sum_{k=1}^{M} w_k w_k^T \tag{89}$$

The matrix C here is a positive (semi-) definite matrix. Let's assume that the set of its orthonormal eigenvectors $\{g_i\}N_{i=1}$ with eigenvalues $\{\gamma_i\}_{i=1}^{N} \geqslant 0$ form a basis of R^N. Then any vector $v_j \in R^N$ can be written as a linear combination of all the eigenvectors $v_j = \sum_{i=1}^{N} a_i g_i$ with coefficients $a_i \in R^N$ and we obtain:

$$v_j^T G v_j = \sum_{i,j} a_i g_i^T G a_j g_j = \sum_{i,j} a_i a_j \gamma_j \underbrace{g_i^T g_j}_{=\delta_{ij}} = \sum_{j} a_j^2 \gamma_j \tag{90}$$

Hence, except for the nontrivial solutions in which $v = 0$, the other vectors v which satisfy $v^T G v = 0$ should meet the requirement $\begin{cases} a_j = 0 \\ or\ a_j \neq 0, for\ \gamma_j = 0 \end{cases}$. Combined together, the solution $\{v_j\}_{j=1}^{J}$ are those eigenvectors with zero eigenvalues. Since in practical applications, it is difficult to obtain exactly zero eigenvalues, the J smallest eigenvalues are selected here.

To sum up, the typical scheme to obtain unique feature ranking results for GMLVQ is formulated as below:

Given a GMLVQ system represented by Ω and a set of prototypes $\{w_k\}_{k=1}^{M}$, the training examples are $\{x, y_i\}_{i=1}^{n}$, construct the matrix X as:

$$X = [x_1, x_2, \cdots x_L, w_1, w_2, \cdots w_M] \tag{91}$$

and then calculate its eigenvalues and eigenvectors and perform the projection as:

$$\Omega_{new} = [I - \sum_{j=1}^{J} v_j v_j^T]\Omega \tag{92}$$

where $\{v_j\}_{j=1}^{J}$ are the J eigenvectors of X with smallest eigenvalues.

It is therefore important to determine the number J. It can be represented as the delete rate which is the ratio between the value of J and dimension N. The delete rate is determined by experiment and differs per data set. Since no additional training is required when testing with multiple delete rates, the choice of an optimal delete rate will not significantly increase the computation time.

Chapter 5

5 Experiments and Results

5.1 Data Set Description

Table 2 summarizes all the datasets that are chosen in our experiments for feature selection evaluation. Except for the first data "Adrenal Tumor", all the other datasets in the table are selected from the UCI data set repository [29]. The selected datasets describe diverse real-world problems and thus show a variety of characteristics. They are utilized to assist the comparison of different feature selection algorithms. The table below provides a summary about the various characteristics of the datasets, including the number of instances and the number of input attributes and output classes.

Name of Datase	Number of Instances	Attribute Type	Number of Attributes	Number of Classes
Adrenal Tumor	147	Real	32	2
Breast Cancer Wisconsin	569	Real	30	2
Ionosphere	351	Integer and Real	34	2
SPECTF Heart	267	Integer	44	3
Connectionist Bench Sonar	208	Real	60	2

Table 2: The five datasets above are selected for the experiments. For each dataset, its number of instances, attribute type, number of attributes and number of output classes are illustrated in the table.

5.2 Experiment Design

We now turn to the experimental procedure to perform feature selection and evaluate the results. The various algorithms used in our experiments include the four feature selection algorithms (information gain (IG), Relieff (RFF), fisher (FR) and GMLVQ-based (GMLVQFS) feature selection and two evaluation methods (GMLVQ and rbf-kernel SVM). A five-fold cross-validation is used in the experiment. As shown in Figure 11, the data is divided into five folders. In each cross-validation, one folder is chosen as the test set, one for validation and the other three as the training set. Feature selection algorithms are performed on the dataset constituting of the training and validation set. For the GMLVQ-based feature selection algorithm, the GMLVQ model is trained on the training set and validated on the validation set to prevent overfitting. As long as the feature ranking result is obtained, new data sets are constructed by selecting different numbers of features, in term of their importance indicated from the feature ranked list. For each new data set, which has the same distribution of training, validation and testing set, only different in the number of features, the two evaluation methods are trained on the training and validation

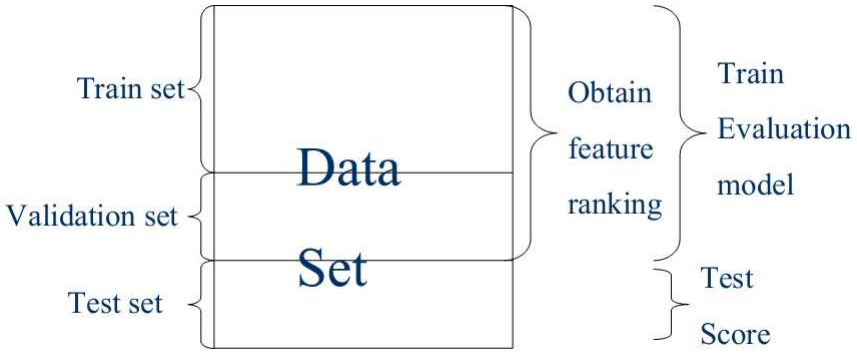

Figure 11: Experiment design of data set.

set and then tested on the test data. Their performances on the test set are evaluated by the ROC curve and the AUC.

5.3 Results and Discussion

In this section, the experimental results on the five datasets are presented and discussed. Each dataset will serve as a case-study. The abilities of the four feature selection algorithms to deal with irrelevant and redundant features will also be demonstrated in independent sections.

5.3.1 Case Study 1: Adrenal Tumor

As shown in Table 2, the Adrenal Tumor data set consists of 32 features and 2 classes. There are a few missing values in this data set and those missing values are replaced by the mean of that feature. Four different feature ranking techniques (IG,RFF,FR,GMLVQ) have been used on the Adrenal Tumor dataset. After that, two different evaluation methods, GMLVQ and RBF based SVM, are built to evaluate the feature selection results. The performance is evaluated in terms of the AUC metric.

Feature Ranking Results Comparison The differences between feature ranking algorithms are essentially measured by the different feature ranking lists they generate; therefore it is important to compare the different feature ranking lists they have produced. Figure 12 illustrates the average feature ranking results over 125 epochs from the four feature ranking methods. It is shown that the features 5, 6 and 19 are ranked as the top three most important features by all the four algorithms. Closer inspections on the feature distributions of the top 3 features are demonstrated in Figure 13. It is obvious that although the average ranking results of the top 3 features are the same among the four feature ranking algorithms, their top three feature distributions are different.

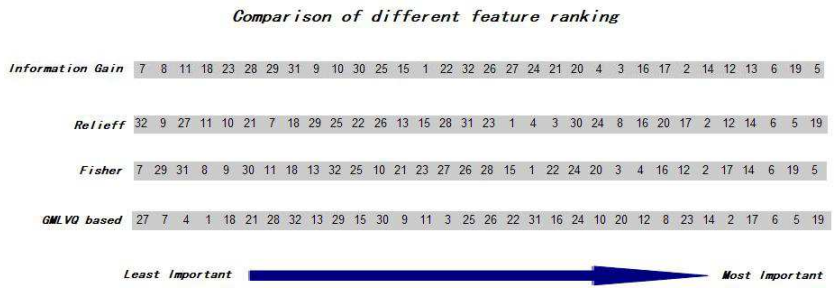

Figure 12: Different feature ranking results on data set Adrenal Tumor.

For example, while the GMLVQ based feature ranking algorithm always chooses features 5, 6 and 19 as the top 3 features, the Information Gain algorithm also occasionally ranks feature 2, 12, or 14 as the top three most important. It is these differences in feature distribution that causes the performance differences during the valuation even when the average ranking results are the same. This analysis also demonstrates the more stable ranking results from GMLVQ based method compared to other algorithms.

Feature Ranking Algorithm Evaluation In this section, different feature ranking algorithms are evaluated by both the GMLVQ and SVM with RBF kernel in terms of the AUC metric. Figure 14 describes the average evaluation performance over 125 epochs using GMLVQ as the evaluation method on the four feature ranking algorithms. It is seen from the figure that GMLVQ based method outperforms other feature ranking algorithms when the feature subset contains less than 13 features. This is essential for feature selection algorithms because we always aim to achieve relatively better results by using smaller feature subsets. For example, when using only the top six most important features, GMLVQ based method can achieve a AUC metric of 0.956 which is already quite close to the 0.961 when all the features are included.

Furthermore, the line with squares illustrates the evaluation performance when we directly utilize the prediction model from the feature ranking training process to test on unseen data. In this way, the feature ranking training and model learning process are combined together as one process which saves much computation time. From Figure 14, it is shown that its performance is even better than other feature ranking algorithms by using the top eight or less most important features.

The evaluation result by RBF-based SVM in Figure 15 demonstrates that GMLVQ based feature ranking method consistently outperforms methods "Fisher" and "Information Gain" no matter how many features are selected. The method "Relieff " performs slightly better when in the range of (16,20) and (24,26) than the GMLVQ based method. Another notable feature about GMLVQ based ranking method is that it achieves the maximum performance when the number of features is 6 and then decreases when more features are added into the sub-

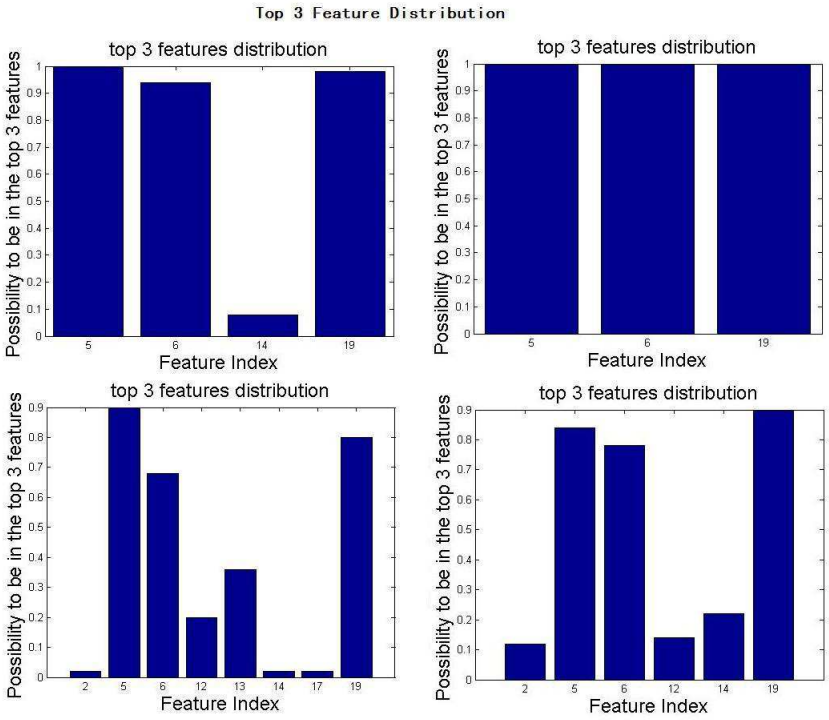

Figure 13: Top three feature distribution among methods Fisher (top left), GMLVQ based (top right), Information Gain (bottom left) and Relieff (bottom right)

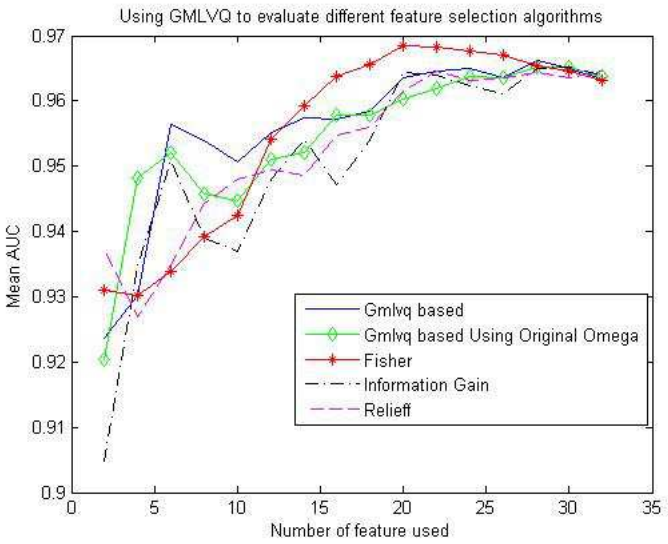

Figure 14: Evaluation on test set using GMLVQ on different feature selection methods. Data set: Adrenal Tumor.

set. This is the ideal situation for feature selection algorithms because a better performance can be achieved with less features and thus less computation time.

Irrelevant Features Irrelevant features are added to the original data set to test their ability to filter out irrelevant features. To be more specific, three uniformly random boolean attributes, three uniformly random 4-valued attributes and three uniformly random 8-value attributes are added to the dataset. Figure 16 demonstrates the average number of irrelevant features included in the subset with various feature subset sizes. The ideal method will include no irrelevant feature in the top 32 features, leaving all the irrelevant attributes ranked the least important ones. It is shown from Figure 15 that Information Gain method can generate such ideal result while for the GMLVQ based method, its 32-feature subset will contain 2 irrelevant features by average, filtering out 7 irrelevant features. The performance difference here depends on how we construct the irrelevant features. It is therefore expected that different methods of irrelevant feature generation will produce different filtering out performance.

Redundant Features 9 Redundant features were added to the original data set to form a data of 41 dimensions and test their ability to filter out redundant features. Specifically, the 5^{th},20^{th} and 25^{th} feature in the original data set were selected as they represented different importances to the classification. Each of these three features was duplicated for three times to generate three more redundant features and for these three redundant features, Gaussian noises with

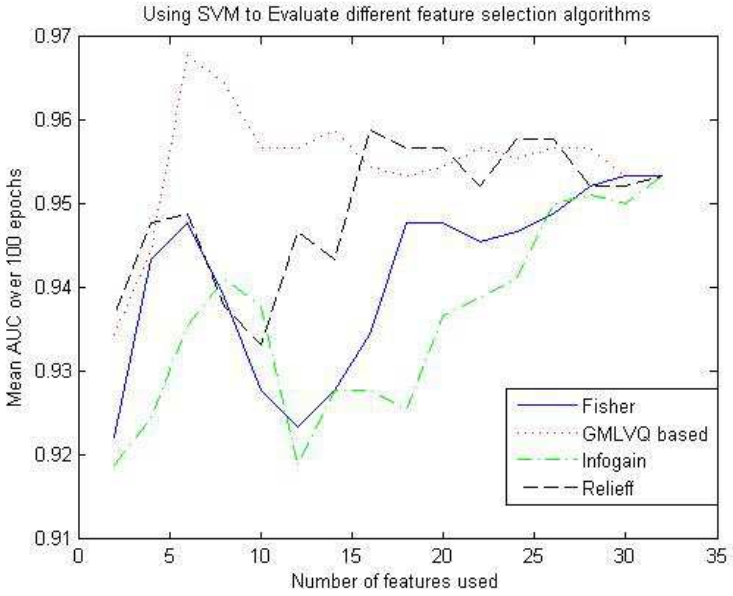

Figure 15: Evaluation on test set using SVM on different feature selection methods. Data set: Adrenal Tumor

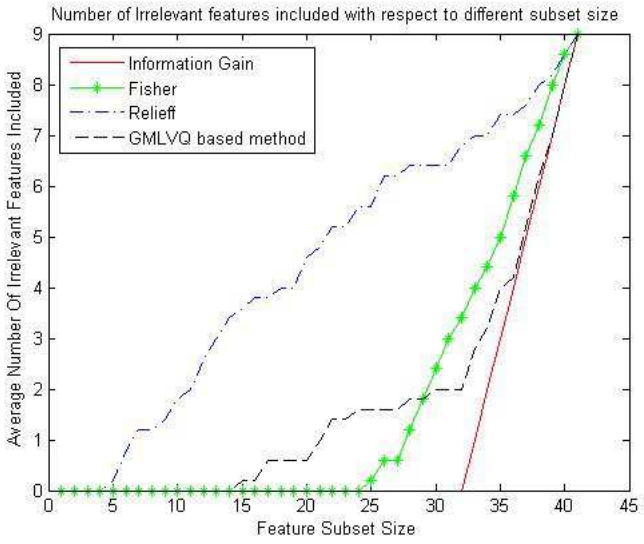

Figure 16: Irrelevant feature detection performance between different feature selection methods. Data set: Adrenal Tumor

different signal-to-noise ratios of respective 10, 50 and 80 were added. The feature ranking results are shown in Figure 17. The top figure illustrates the average number of redundant features that a feature subset contains. Although the GMLVQ based method performs relatively better than the Fisher and Relieff methods, the overall performances of all four methods are poor. To be more specific, with a feature subset of 32 features which is the same size of the original feature sets before adding redundant features, even the method with best performance already contains 7 redundant features, filtering only 2 features. A closer inspection about their exact ranking results was conducted to investigate the reason of bad performance. It reveals that the redundant features can not be efficiently removed by all these four methods. For example, the 33^{th},34 and 35^{th} feature is the noisy copy of the 5^{th} feature. In all the feature ranking results here these three features are all ranked in the top 9 features. A similar situation can also be observed for the 20^{th} and 25^{th} feature and their noisy copies.

5.3.2 Case Study 2: Ionosphere

The "Ionosphere" data set describes the radar data collected by a system in Goose Bay. As shown in Table 2, it consists of 34 features and 2 classes. The second feature contains all zeros and is removed during the experiment. There are 351 instances in total from which 210 random ones are selected for training, the other 70 randomly picked samples for validation and the remaining 71 instances used for testing. Four different feature ranking techniques (IG,RFF,FR,GMLVQ) have been used on this dataset. After that, two different evaluation methods, GMLVQ and RBF based SVM, are built to evaluate the feature selection results. The performance is evaluated in terms of the AUC metric.

Feature Ranking Comparison Figure 18 illustrates the average feature ranking results over 125 epochs from the four feature ranking methods. The methods "Fisher" and "GMLVQ based" have closer average feature ranking results and both rank the feature 1,4 and 2 as the most important features. Method "Relieff" and "Information Gain", on the other hand, display quite different average ranking results. For example, while the Relieff feature ranking algorithm chooses features 26, 23 and 7 as the top 3 features, the Information Gain algorithm ranks features 5, 4, and 32 as the top three most important.

Feature Ranking Algorithm Evaluation In this section, different feature ranking algorithms are evaluated by both the GMLVQ and SVM with RBF kernel in terms of the AUC metric. Figure 19 describes the average evaluation performance over 125 epochs using GMLVQ as the evaluation method on the four feature ranking algorithms. It is seen from the figure that GMLVQ based method consistently outperforms other feature ranking algorithms in the top 30 features. Considering there are only 33 features in total, this phenomenon demonstrates the big advantage of the GMLVQ based feature ranking method

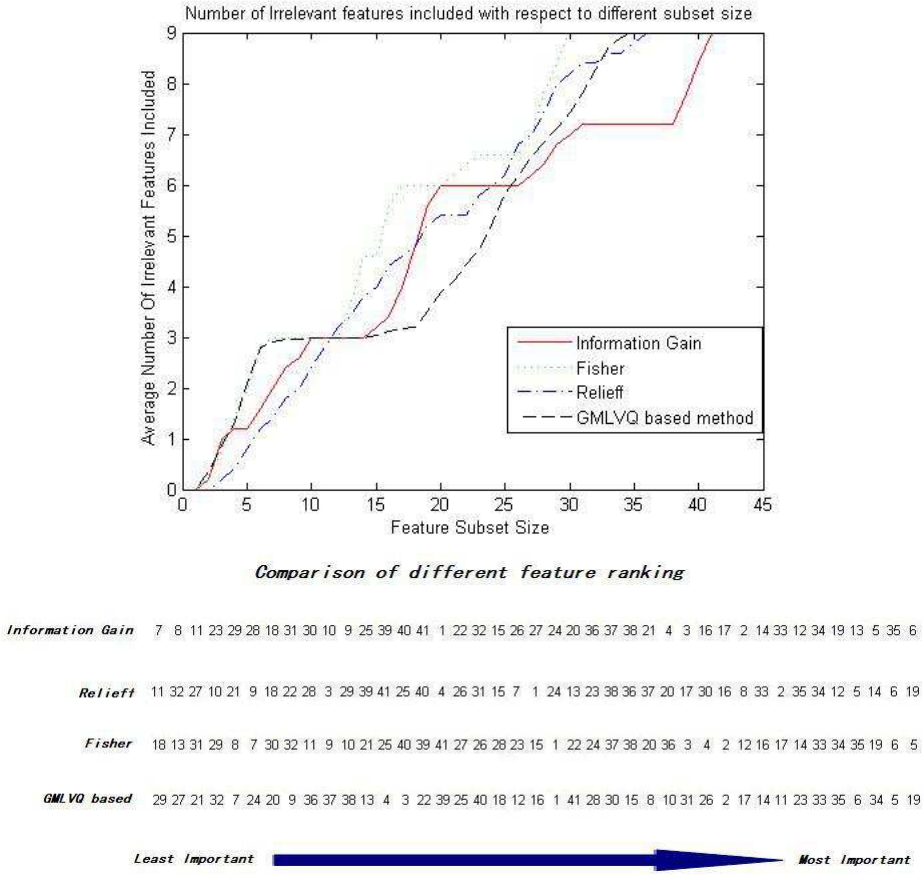

Figure 17: Feature ranking results after adding 9 redundant features. Data set: Adrenal Tumor

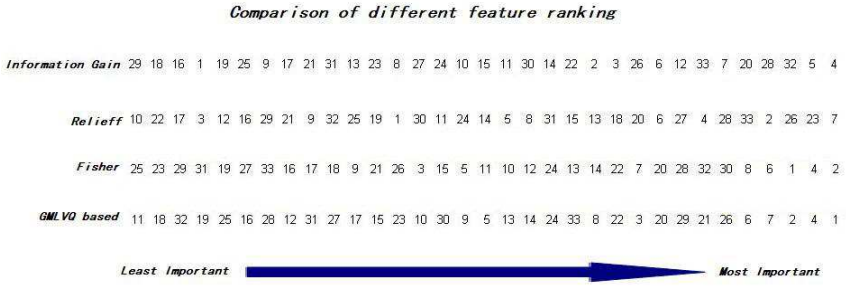

Figure 18: Differnt feature ranking results. Data set: Ionosphere.

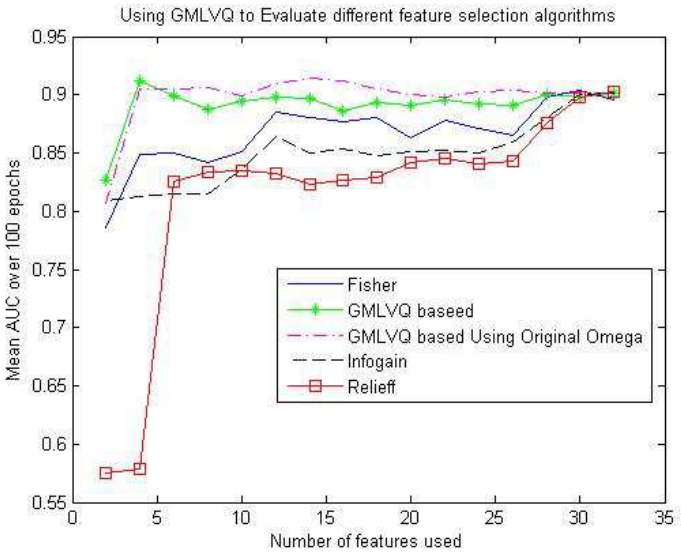

Figure 19: Evaluation on test set using GMLVQ on different feature selection methods. Data set: Ionosphere.

on this data set. Furthermore, the dash-dot line illustrates the evaluation performance by the classifier directly from the feature ranking training process to test on unseen data. It means the feature ranking training and model learning process are combined together as one process and as long as the feature ranking results are obtained, the classification can be performed without retraining the learning model again. Figure 19 shows it outperforms other feature ranking algorithms when the number of feature selected is larger than 4.

Figure 20 present the evaluation results on the four feature ranking algorithm using RBF-based SVM. It is shown that in the top 6 features, GMLVQ based feature ranking results can provide a better AUC measurement and after the methods "Relieff" and "Information Gain" take turns to lead the performance.

Irrelevant Features 20 irrelevant features were added to the original data set to test their ability to filter out irrelevant features. The way to construct the irrelevant features here is different from that of data set "Adrenal Tumor". In this experiment, the irrelevant features added are not discrete uniformly distributed values but truly random continuous signals. From Figure 21, it is observed that Information Gain method outperforms other methods by containing about only 1 irrelevant feature in the top 32 features, compared to 2 irrelevant features from method Relieff and GMLVQ. Fisher algorithm performs worst on this data set.

Redundant Features 9 Redundant features were added to the original data set to form a data of 42 dimensions and test their ability to filter out redundant

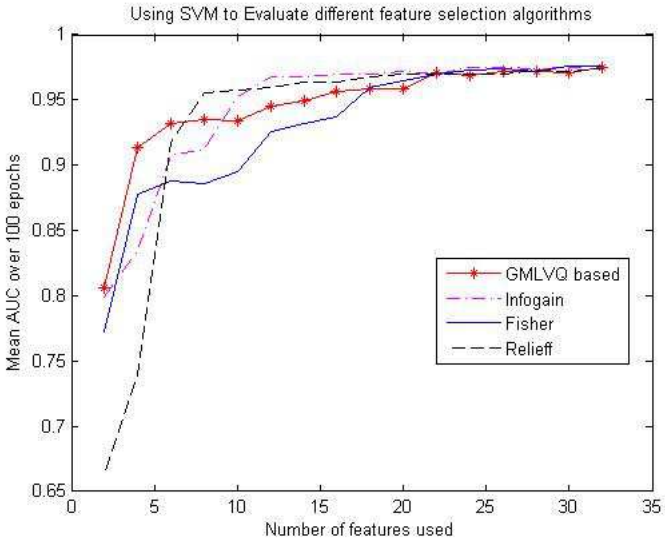

Figure 20: Evaluation on test set using SVM on different feature selection methods. Data set: Ionosphere.

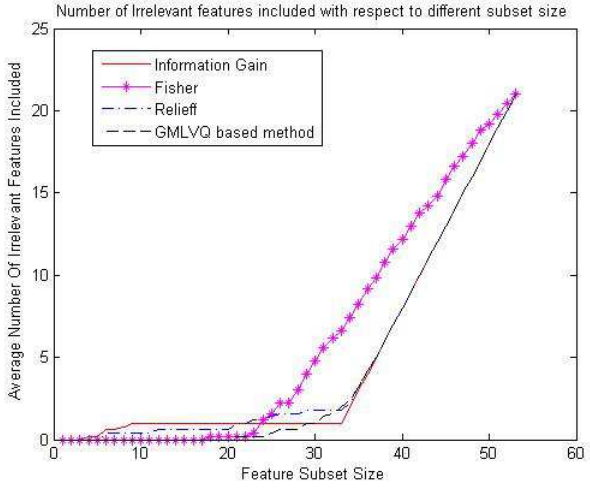

Figure 21: Irrelevant feature detection performance between different feature selection methods. Data set: Ionosphere.

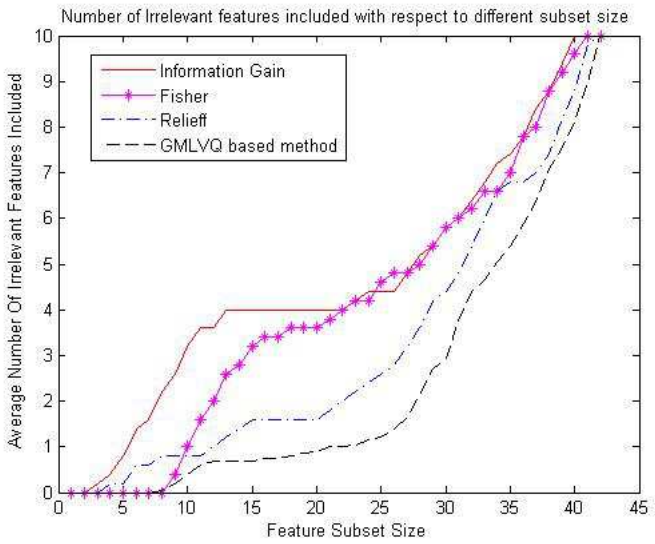

Figure 22: Feature ranking results after adding 9 redundant features on data set "Ionosphere". Data set: Ionosphere.

features. Specifically, the 5^{th},20^{th} and 25^{th} features in the original data set were selected as they represented different importances to the classification. Each of these three features was duplicated for three times to generate three more redundant features and for these three redundant features, Gaussian noises with different signal-to-noise ratios of respective 10, 50 and 80 were added. The results are shown in Figure 22. The figure indicates that in the top 32 features, the GMLVQ based method consistently outperforms other methods no matter how many features are selected in the feature subset.

5.3.3 Case Study 3: Connectionist Bench Sonar

The "Connectionist Bench Sonar" data set describes features of the cell nuclei present from the digitized images of a breast mass. As shown in table 2, it consists of 208 instances with 60 features and 2 different classes. 125 random ones are selected for training, the other 42 randomly picked samples for validation and the remaining 42 instances used for testing. Four different feature ranking techniques (IG,RFF,FR,GMLVQ) have been used on this dataset. After that, two different evaluation methods, GMLVQ and RBF based SVM, are built to evaluate the feature selection results. The performance is evaluated in terms of the AUC metric.

Feature Ranking Comparison Figure 23 describes the average feature ranking results over 100 epochs from the four feature ranking methods. Since it is

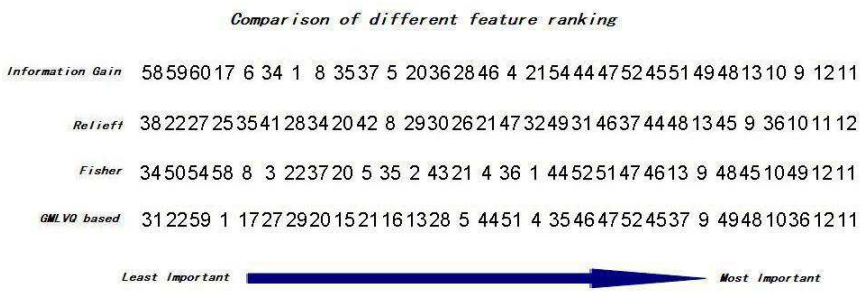

Figure 23: Feature ranking results on data set Connectionist Bench Sonar.

difficult to display all the 60 features clearly, only the top 30 features are visualized here for the ease of observation. It is observed that these four feature ranking methods have similar results on this data set. For example, in the top 8 most important features ranked by these four methods, 6 of them are the same which are respectively the $11^{th}, 12^{th}, 10^{th}, 9^{th}, 10^{th}\, and\, 48^{th}$ feature.

Feature Ranking Algorithm Evaluation In this section, different feature ranking algorithms are evaluated by both the GMLVQ and SVM with RBF kernel in terms of the AUC metric. Figure 24 describes the average evaluation performance over 100 epochs using GMLVQ as the evaluation method on the four feature ranking algorithms. It is observed that in the top six features, the "Relieff" method outperforms other methods and achieves the maximal performance at number of features being 4. Thereafter, the GMLVQ based feature ranking methods take the lead. It is also noticed that the model which combines the feature ranking training and evaluation process together still demonstrates some considerable performance, indicating the advantage of GMLVQ feature ranking method.

The evaluation results by RBF-based SVM in Figure 25 demonstrates that the evaluation performances of all these four feature ranking methods consistently improve with the increase of the number of features. Specifically, the GMLVQ method starts at 0.82 when the number of features is 2 and achieves the maximal performance among all these feature ranking methods at 0.95 with the number of features being 36. Besides, the GMLVQ based method has a similar performance with the "Relieff" method and consistently outperforms the other two methods.

Irrelevant Features 20 irrelevant features were added to the original data set to test their ability to filter out irrelevant features. The irrelevant features added are truly random continuous signals. From Figure 26, it is observed that Information Gain method outperforms other methods by containing about only 1 irrelevant feature in the top 60 features, compared to 2 irrelevant features from method Relieff and GMLVQ based one. Fisher algorithm, on this data set,

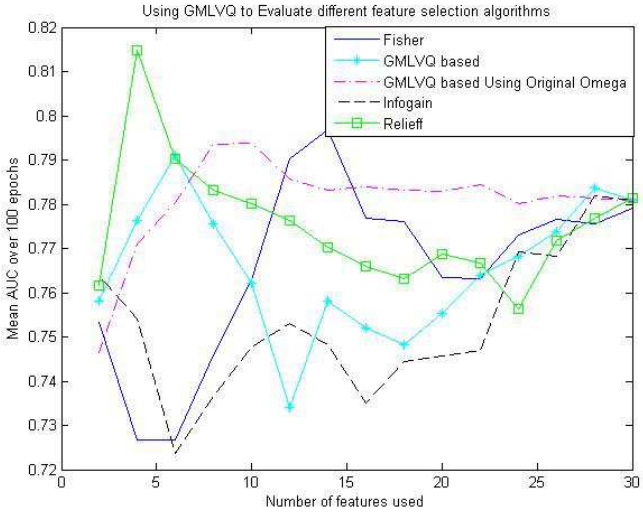

Figure 24: Evaluation on test set using GMLVQ on different feature selection methods. Data set: Connectionist Bench Sonar.

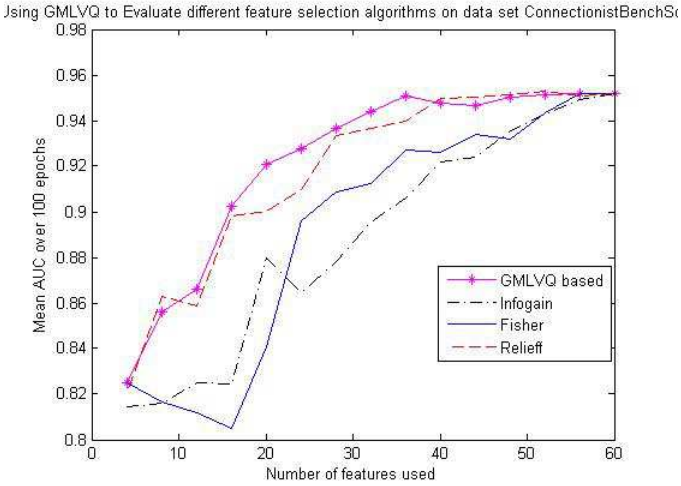

Figure 25: Evaluation on test set using SVM on different feature selection methods. Data set: Connectionist Bench Sonar.

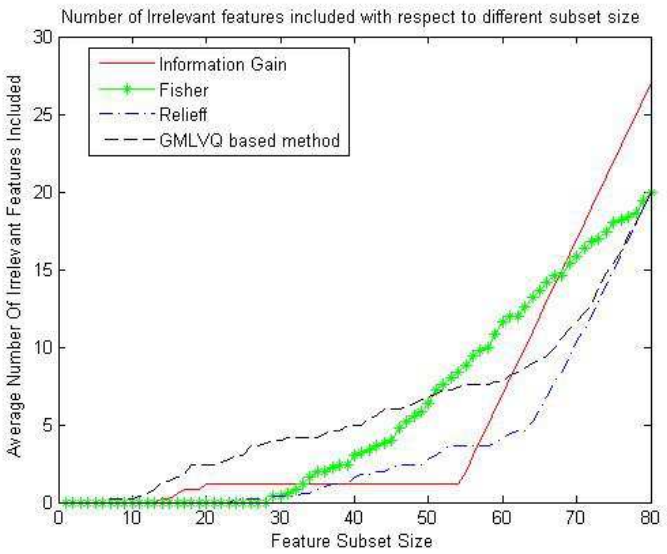

Figure 26: Irrelevant feature detection performance between different feature selection methods. Data set: Connectionist Bench Sonar.

performs worst.

Redundant Features 9 Redundant features were added to the original data set to form a data set with 69 dimensions and test their ability to filter out redundant features. The results are shown in Figure 27. It indicates that the GMLVQ based feature ranking method contains about 3.5 redundant features in the top 44 features which is the best performance compared to other methods. It is observed that the GMLVQ based method consistently outperforms the Fisher and Relieff method for the whole range of subset size. When the size of feature subset is over 52, the Information Gain method performs a little better than GMLVQ based method.

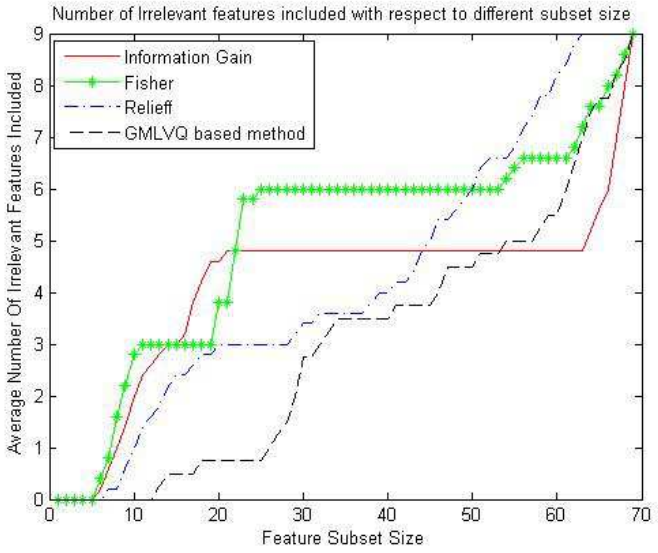

Figure 27: Feature ranking results after adding 9 redundant features. Data set: Connectionist Bench Sonar.

5.3.4 Case Study 4: Breast Cancer

The "Breast Cancer" data set describes characteristics of cell nuclei from digitized images of fine needle aspirates of breast masses. It consists of 30 features and 569 instances. There are two possible labels for each instance, indicating whether the cancer is benign or malignant.

Feature Ranking Results Comparison Figure 28 describes the average feature ranking results over 100 epochs from the four feature ranking results on data set "Breast Cancer". It shows that these four feature ranking methods have similar ranking results on this data set. For example, all these fours methods rank the nine features: $23^{th}, 21^{th}, 28^{th}, 24^{th}, 8^{th}, 3^{th}, 1^{th}, 4^{th}, 7^{th}$ as the top 10 most important ones.

Feature Ranking Results Evaluation In this section, all these four feature ranking methods are evaluated by GMLVQ and RBF-SVM in terms of their AUC metric. Figure 29 describes the performance evaluated by GMLVQ. A closer inspection reveals that there is no dominant method on this data set with the methods of Relieff, Information Gain and Fisher take turn to lead. GMLVQ based method performs a little worse than others on this data set.

Figure 30 describes the evaluation performance of RBF-SVM. It is seen from the figure that all these four methods perform quite similarly while the GMLVQ based method slightly outperforms others when the subset size is between 14 to 24.

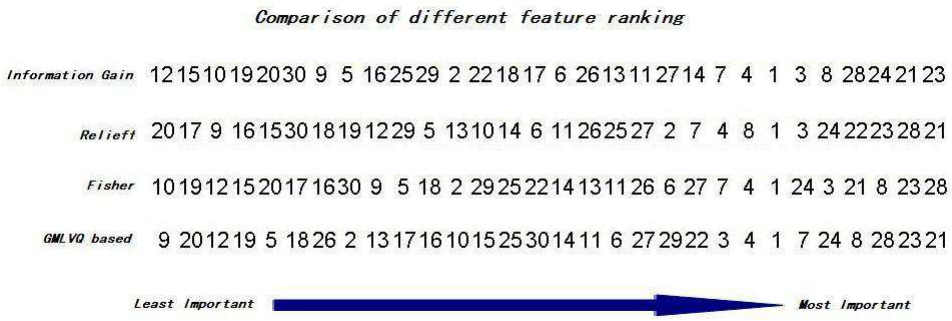

Figure 28: Feature Ranking Results on data set Breast Cancer

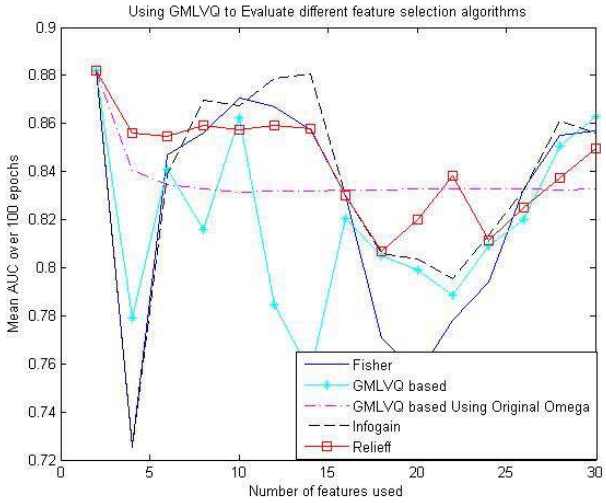

Figure 29: Evaluation on test set using GMLVQ on different feature selection methods. Data set: Breast Cancer

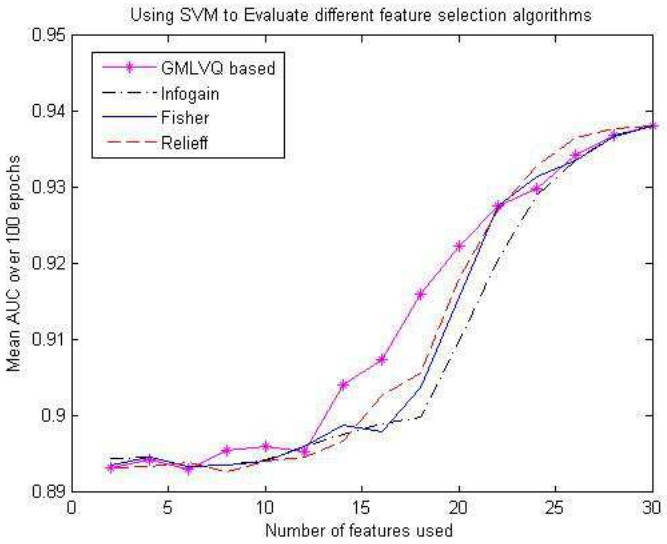

Figure 30: Evaluation on test set using SVM on different feature selection methods. Data set: Breast Cancer

Irrelevant Features 20 more random features are added to the original data to form a 50 dimensional data set. It is seen from Figure 31 that the Information Gain method performs best followed by the GMLVQ based method. Specifically, when selecting the top 30 features, the GMLVQ based method contains about 3 irrelevant features on average, filtering out the other 18 noisy features.

Redundant Features 9 redundant features are added into the original data to form a data set of 39 features. All the 9 features are duplicates of one of the features in the original data set and added with Gaussian noises. It is shown in Figure 32 that the GMLVQ based method performs consistently better than the others when the size of subset is smaller than 30. However, even the best performance still contains 6 redundant features in the top 30 important list, indicating that all these four methods can not effectively filter out redundant features on this data set.

5.3.5 Case Study 5: SPECTF Heart

The SPECTF Heart data set describes the diagnosis of the Single Proton Emission Computed Tomography (SPECT) images. It consists of 267 instances and 44 dimensions with each dimension describing one feature extracted from the SPECT images. Each instance is labeled as normal or abnormal, indicating whether the patient suffers from the disease. There are no missing values in this data set. In each run of training, 160 examples are randomly selected for

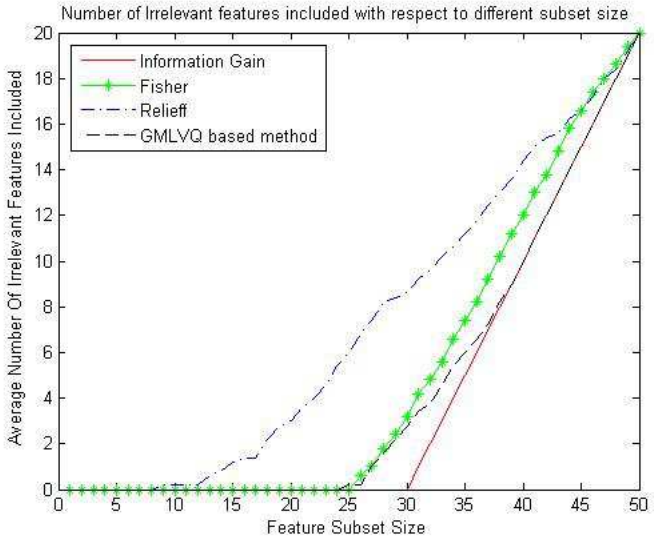

Figure 31: Irrelevant feature detection performance between different feature selection methods. Data set: Breast Cancer

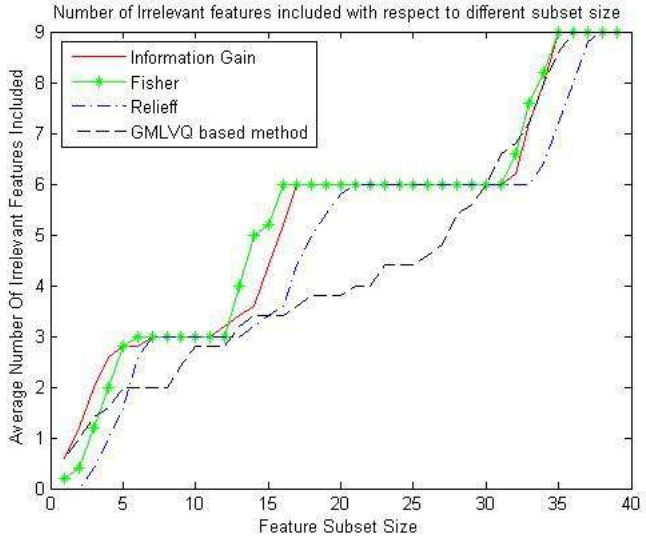

Figure 32: Feature ranking results after adding 9 redundant features. Data set: Breast Cancer

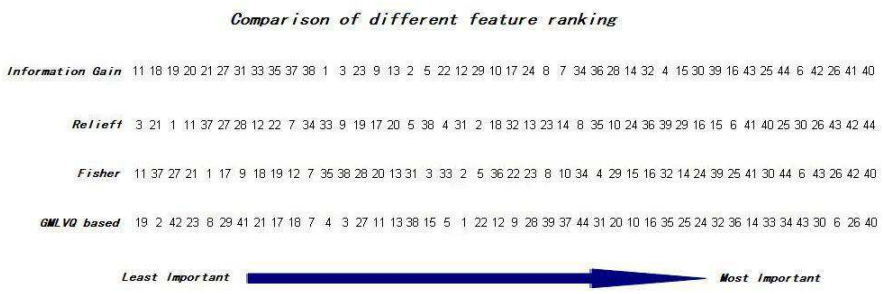

Figure 33: Feature ranking results on data set SPECTF Heart.

training, the other 53 are randomly selected for validation and the remaining 54 examples serve as the test data.

Feature Ranking Results Comparison Figure 33 describes the average feature ranking results over 100 epochs on data set SPECTF Heart. In the top 12 most important features, 7 of them are the same which are the 40^{th}, 41^{th}, 26^{th}, 6^{th}, 25^{th}, 43^{th} and 30^{th} features. It indicates that on this data set, these feature ranking results have similar ranking results.

Feature Ranking Algorithm Evaluation All the feature ranking methods are evaluated in this section. Figure 34 illustrates the evaluation performance by GMLVQ. It is seen from the figure that when selecting the top eight most important features, GMLVQ based method performs the best and after that the fisher method takes the lead and in the end they converge to about 0.77 in terms of AUC. It is noticed that the GMLVQ based method achieves the maximum performance by selecting only the top six features which means that these six features may be most informative and strongly relevant for classification.

Figure 35 illustrates the evaluation performance by SVM. As seen from the figure, the Relieff method takes the lead in the top 8 features and after that the GMLVQ based method performs better than others when the number of features is larger than 16.

Irrelevant Features 9 continuous random features are added to the original data to form a 53 dimensional data set. The feature ranking results are shown in Figure 36. It is noticed that the Information Gain method performs the best in this case. On the other hand, GMLVQ based method performs worst. Specifically, the number of irrelevant features it contains increases linearly with the size of feature subset, indicating that most of the irrelevant features can not be detected and filtered out.

Redundant Features The 21^{th},10^{th} and 43^{th} feature in the original data set were selected to generate redundant features. The ranking results are shown in

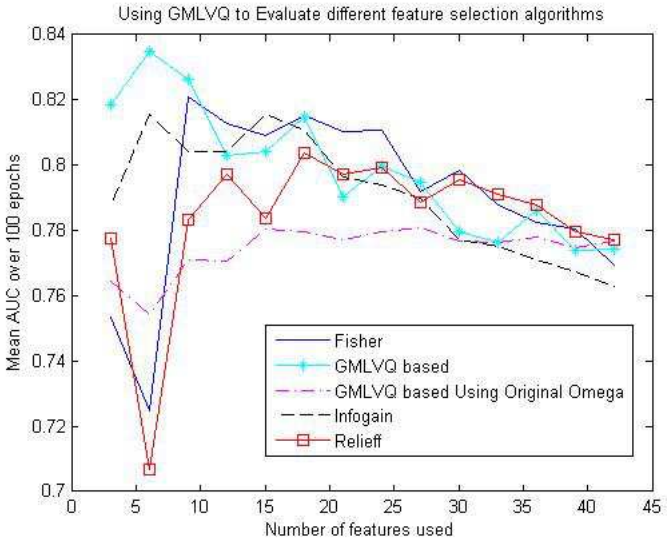

Figure 34: Evaluation on test set using GMLVQ on different feature selection methods. Data set: SPECTF Heart

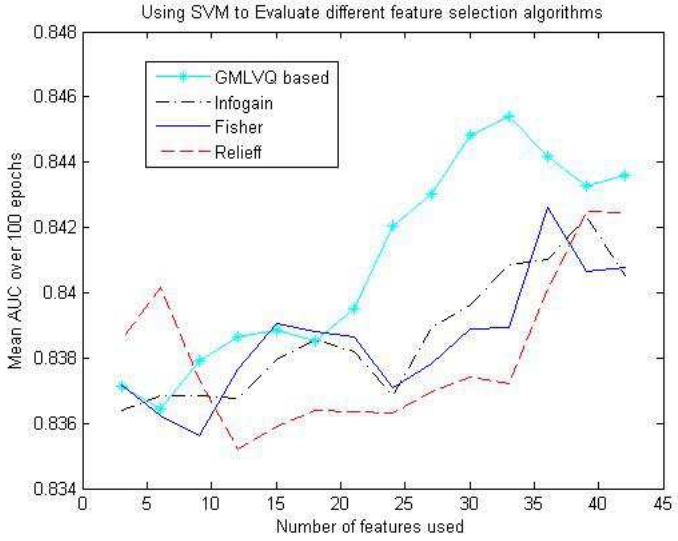

Figure 35: Evaluation on test set using SVM on different feature selection methods. Data set: SPECTF Heart

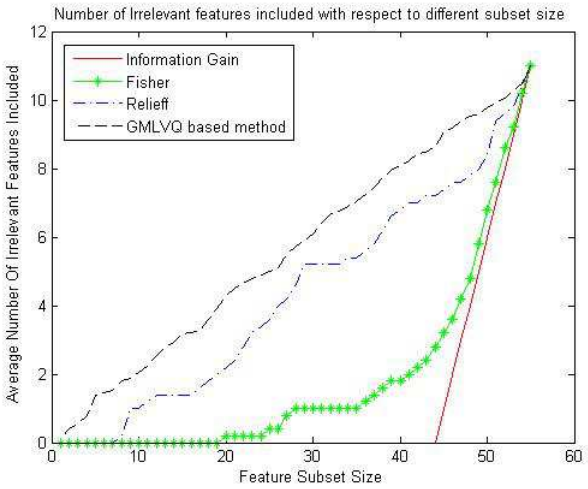

Figure 36: Irrelevant feature detection performance between different feature selection methods. Data set: SPECTF Heart

Figure 37. On this data set, the GMLVQ based method performs much better than the others. For example, by choosing a subset of 40 features, only 1.5 redundant features on average are included, indicating that the GMLVQ based method can filter out most of the redundant features and assigning them as the least important ones.

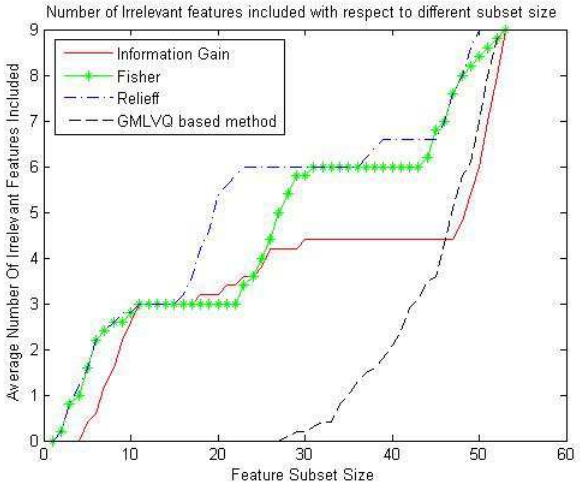

Figure 37: Feature Ranking Results After Adding 9 Redundant Features. Data set: SPECTF Heart

5.4 Discussion and Summary

Five experiments have been conducted in this chapter to compare the performances of the four features ranking algorithms. Their performances differ per data set, and a summary is provided here to conclude their comparison.

The performances are evaluated by GMLVQ and RBF-SVM in terms of their AUC metric. An easy way to compare their evaluation performance may be to count the percentage that each method dominates among these four ranking algorithms. An algorithm which dominates on more features may be regarded as providing a better ranking results with respect to the evaluation methods. The summaries can be seen in Table 3 and Table 4. It is obvious from Table 3 that GMLVQ related feature ranking methods, including the first and second columns, dominate the performances, indicating the advantage of GMLVQ based methods. Table 4 also demonstrates the advantage of the GMLVQ based feature ranking method. To be more specific, on three of these five data sets, the GMLVQ based method has a dominance over 50%.

To compare their performances on filtering out irrelevant features, we count the number of irrelevant features they contain when the sizes of subset are equal to the original feature set before adding irrelevant features. The results are shown in Table 5. It is shown from the table that the Information Gain method performs best among the five experiments and in three of these experiments, the GMLVQ based method performs in the second place. On average, the Information Gain method has the best performance. The GMLVQ based method is in the second place. The Relieff and Fisher methods are respectively in the third and fourth place.

The performances on filtering out redundant features are also compared by counting the number of redundant features the four algorithms contain when the size of subset is equal to the original data set before adding redundant features. The results are shown in Table 6. It is shown that the GMLVQ based method contains 29.08 redundant features in total over the five data sets. This performance is close to that of the method Information Gain which contains 28.8 redundant features in total. On the other hand, the Relieff and Fisher methods respectively contain 34.6 an 34 redundant features in total.

Name of Dataset	GMLVQ Based	GMLVQ Based Using Original Omega	Information Gain	Fisher	Relieff
Adrenal Tumor	31.25%	12.50%	43.75%	0%	12.50%
Ionosphere	12.50%	75.00%	6.25%	0%	6.25%
Connectionist Bench Sonar	13.33%	53.33%	13.33%	6.67%	13.33%
Breast Cancer Wisconsin	6.67%	33.33%	13.33%	26.67%	20.00%
SPECTF Heart	28.57%	0%	42.86%	7.14%	21.43%

Table 3: Percentage of dominance of each method, evaluated by GMLVQ

Name of Dataset	**GMLVQ Based**	**Information Gain**	**Fisher**	**Relieff**
Adrenal Tumor	**56.25%**	**0%**	**0%**	**43.75%**
Ionosphere	**25.00%**	**56.25%**	**6.25%**	**12.50%**
Connectionist Bench Sonar	**66.66%**	**0%**	**0%**	**33.33%**
Breast Cancer Wisconsin	**20.00%**	**33.33%**	**20.00%**	**26.67%**
SPECTF Heart	**71.43%**	**0%**	**14.29%**	**14.29%**

Table 4: Percentage of dominance of each method, evaluated by RBF-SVM

Name of Dataset	GMLVQ Based	Information Gain	Fisher	Relieff
Adrenal Tumor	2	0	6.8	3.4
Ionosphere	1.6	1	6.2	1.8
Connectionist Bench Sonar	7.2	1.2	8.0	3.6
Breast Cancer Wisconsin	2.8	0	3.2	8.6
SPECTF Heart	8.6	0	2.8	7.2

Table 5: Comparing the performances on irrelevant features

Name of Dataset	GMLVQ Based	Information Gain	Fisher	Relieff
Adrenal Tumor	8.28	7.2	9.0	8.4
Ionosphere	5.8	6.4	6.2	5.4
Connectionist Bench Sonar	5.5	4.8	6.6	8.2
Breast Cancer Wisconsin	6	6	6	6
SPECTF Heart	3.5	4.4	6.2	6.6

Table 6: Comparing the performances on redundant features

Chapter 6

6 Conclusion and Future Work

This thesis investigates the application of GMLVQ model on the the feature ranking problems. The basic concepts in classification and feature selection are discussed in the first three chapters as background information. Three state-of-the-art feature ranking techniques are then described and introduced to work as comparison methods for the GMLVQ based method. The GMLVQ based feature ranking technique is intensively described in Chapter 4 and then followed by experimental results on the data sets collected from the UCI repository [29].

The experimental results, evaluated by GMLVQ and RBF-SVM, indicate that GMLVQ based feature ranking method is comparable with other state-of-the-art methods. Sometimes it consistently outperforms other methods. For example, on the data set "Ionosphere" evaluated by "GMLVQ", the GMLVQ based method consistently has superior performance to others.

Another noticeable feature about GMLVQ based feature ranking method is that it can combine the processes of feature ranking and classification together which can help to save much computation time. Because the feature ranking result is extracted from the distance metric from the training, after a specific feature subset is selected the distance metric can then directly be obtained by collecting the corresponding columns and removing the others. In this way, there is no need to retrain the learning model and the classification result can be directly obtained. The experimental results in this thesis demonstrate that its performance is comparable to other results which perform feature ranking first and model training in two steps.

The ability of the feature ranking methods to deal with irrelevant and redundant feature is also tested and the results demonstrate that these four feature methods have better ability to tackle irrelevant features than redundant features. On average, these four methods will contain more redundant features than irrelevant features when a specific feature subset is evaluated.

To answer the three research questions proposed in Chapter 1:

1. Can GMLVQ method be extended to perform feature ranking?

Yes, GMLVQ can be extended for feature ranking. The algorithm is detailed in Chapter 4. The feature ranking results are obtained by measuring the diagonal elements of the relevance matrix in GMLVQ. The diagonal elements are regarded as a measurement of the contributions of the features for classification. External force is incorporated to enforce more discriminative ranking results and to obtain a unique ranking list, the feature-pair linear dependency in the relevance matrix is removed.

2. How well does the feature ranking perform?

The performances of the GMLVQ based method are comparable to the other three state-of-the-art feature ranking methods. It is shown that on some of the

data sets, the GMLVQ based method consistently performs better than other ranking algorithms and on average, the GMLVQ based method performs better than any of the other three algorithms. For example, when the performances are evaluated by RBF-SVM, the GMLVQ based method demonstrates its performance dominance on three out of five data sets that have been tested. When the GMLVQ model is used for evaluation, the combination of the two GMLVQ based methods also illustrates the performance superior to other methods on three out of five experiments.

3. Can GMLVQ combine the feature ranking and classification into one single process and how well does the classification perform compares to other methods in which feature ranking and classification are performed in two steps?

Yes, the process of feature ranking and classification can be combined together under the framework of GMLVQ model. Its performances are compared with four other feature ranking algorithms and it indicates the performances are still comparable with other methods which perform feature ranking and classification in two steps. Among all the five data sets that have been tested, it has dominant performances on three of them which is a quite promising results.

Some challenges and future work can still be extended after this thesis.

Feature Selection with Larger Data Set Some data sets in practical applications contain hundreds or thousands of features. Such data sets require efficient feature selection methods to select the most representative feature subset. Experiments on such data sets may also demonstrate some new characteristics of the feature ranking methods in this thesis.

Feature Selection with Active Data Selection In this thesis, the training, validation and test data are selected randomly from the original data set, ignoring the different data characteristic that different instances can have. Active data selection means to explore the data characteristics first and then actively select the instances with higher probability to be informative for the training of feature selection. Such active training may improve the stability and performance of feature selection.

Investigate The GMLVQ Based Method on Feature Redundancy Although some of the experiments in this thesis demonstrate that the GMLVQ based method has better performance to filter out redundant features compared to other state-of-the-art methods, its theoretical foundation to deal with redundancy has not yet been investigated. Since the GMLVQ model already accounts for feature pair contribution for classification, the study may be extended to investigate the feature pair correlation and try to reduce their influence on the classification.

References

[1] Jennifer G. Dy and Carla E. Brodley and Avinash C. Kak and Lynn S. Broderick and Alex M. Aisen. “Unsupervised Feature Selection Applied to Content-Based Retrieval of Lung Images.” IEEE Transactions on Pattern Analysis and Machine Intelligence, 25:373-378, 2003.

[2] Jennifer G. Dy and Carla E. Brodley. “Feature selection for unsupervised learning.” Journal of Mach. Learn. Res., 5:845-889, 2004.

[3] Teuvo Kohonen. “Self-Organizing Maps.” Second. Berlin, Heidelberg: Springer, 1997.

[4] A Sato, K Yamada. “Generalized learning vector quantization.” In: G. Tesauro, D. Touretzky, & T. Leen, (eds.), Advances in Neural Information Processing System 7, (1995), 423-429. MIT Press, 1995.

[5] Avrim L. Blum and Pat Langley. “Selection of Relevant Features and Examples in Machine Learning." Artificial Intelligence, vol. 97, pp. 245-271, 1997.

[6] Manoranjan Dash and Huan Liu. “Feature selection for classification.” Intelligent Data Analysis: An International Journal, 1(3):131-156, 1997.

[7] Barbara Hammera, Thomas Villmann. “Generalized relevance learning vector quantization.” Neural Networks 15(2002), Nr. 8-9, S. 1059-1068.

[8] George H. John, Ron Kohavi and Karl Pfleger. “Irrelevant features and the subset selection problem." International Conference on Machine Learning, pp. 121-129, 1994.

[9] Richard Ernest Bellman; Rand Corporation. "Dynamic programming." Princeton University Press. ISBN 978-0-691-07951-6, 1957.

[10] Richard Ernest Bellman. "Adaptive control processes: a guided tour." Princeton University Press. 1961.

[11] Bernard W. Silverman. “Density estimation for statistics and data analysis." London, Chapman and Hall, 1986.

[12] Michel Verleysen and Damien François. “The curse of dimensionality in data mining and time series prediction." International Work-Conference on Artificial and Natural Neural Networks, volume 3512, pages 758-770. Springer-Verlag, 2005.

[13] Justin Doak. “An evaluation of feature selection methods and their application to computer security.” technical report, Univ. of California at Davis, Dept. Computer Science, 1992.

[14] Manoranjan Dash, Kiseok Choi, Peter Scheuermann and Huan Liu. "Feature Selection for Clustering - A Filter Solution." IEEE International Conference on Data Mining, pp. 115-122, 2002.

[15] Mark A. Hall. "Correlation-based Feature Selection for Discrete and Numeric Class Machine Learning." International Conference on Machine Learning, pp. 359-366, 2000.

[16] Lei Yu and Huan Liu. "Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution." International Conference on Machine Learning, pp. 856-863, 2003.

[17] Huan Liu and Rudy Setiono. "A Probabilistic Approach to Feature Selection - A Filter Solution." International Conference on Machine Learning, pp. 319-327, 1996.

[18] Jennifer G. Dy and Carla E. Brodley. "Feature Subset Selection and Order Identification for Unsupervised Learning." International Conference on Machine Learning, pp. 247-254, 2000.

[19] YeongSeog Kim and Filippo Menczer. "Feature selection in unsupervised learning via evolutionary search." Knowledge Discovery and Data Mining, pp. 365-369, 2000.

[20] Ron Kohavi and George H. John. "Wrappers for feature subset selection." Artificial Intelligence, vol. 97, nos. 1-2, pp. 273-324, 1997.

[21] Thomas M. Cover and Joy A. Thomas. "Elements of information theory." Wiley, 1991.

[22] Kenji Kira and Larry A. Rendell. "A Practical Approach to Feature Selection." International Conference on Machine Learning, pp. 249-256. Morgan Kaufmann, 1992.

[23] Igor Kononenko. "Estimating Attributes: Analysis and Extensions of RELIEF." The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, pp. 171-182, 1994. Berlin: Springer-verlag, 1995.

[24] Nello Cristianini and John Shawe-Taylor. "An Introduction to Support Vector Machines and Other Kernel based Learning Methods." Cambridge University Press, 2000.

[25] S. Haykin. " Neural Networks: A Comprehensive Foundation." Prentice-Hall, 2nd edition, 1999.

[26] Vladimir N. Vapnik. "The Nature of Statistical Learning Theory." Springer-Verlag, New York, USA, 1995.

[27] Vladimir N. Vapnik. "The support vector method of function estimation." Neural Networks and Machine Learning, pp. 239–268, 1998.

[28] Bernhard Schölkopf and Alex Smola. "Learning with Kernels." MIT Press, Cambridge, MA, 2002.

[29] Catherine Blake and Christopher J. Merz (1998). "UCI repository of machine learning databases." http://www.ics.uci.edu/~mlearn/MLRepository.html.

[30] L.Rade and B. Westergren. "Mathematics Handbook for Science and Engineering." 4th ed: Springer-Verlag: Berlin, 1999.

[31] Roger Fletcher. "Practical methods of optimization." John Wiley & Sons, New York, 2nd edition, 1987.

[32] G. Papari, Kerstin Bunte, and Michael Biehl. "Waypoint averaging and step size control in learning by gradient descent." In F.-M. Schleif and T. Villmann, eds., MIWOCI 2011, Mittweida Workshop on Computational Intelligence. Univ. of Bielefeld, pages 16–26, 2011.

[33] Petra Schneider, Michael Biehl and Barbara Hammer. "Adaptive Relevance Matrices in Learning Vector Quantization." Neural Computation, vol. 21, pp. 3532-3561, 2009.

[34] Michael Biehl and Rainer Breitling and Yang Li. "Analysis of Tiling Microarray Data by Learning Vector Quantization and Relevance Learning." Intelligent Data Engineering and Automated Learning, pp. 880-889, 2007.

[35] Barbara Hammer and Marc Strickert and Thomas Villmann. "Prototype Based Recognition of Splice Sites." Bioinformatic using Computational Intelligence Paradigms, Springer-Verlag, pp. 25-26, 2005.

[36] K.E. Runyon. "Consumer behavior and the Practice of Marketing." Charles E. Merrill Publishing Company, Columbus, Ohio, 1977 .

[37] Tom M. Mitchell. "Machine Learning." McGraw Hill. ISBN 0-07-042807-7, 1997.

[38] Petra schneider, Michael Biehl and Barbara hammer. "Distance Learning in Discriminative Vector Quantization." Neural Computation, vol. 21, no. 10, 2009.

[39] Markus B. Huber, Kerstin Bunte, Mahesh B. Nagarajan, Michael Biehl, Lawrence A. Ray, Axel Wismuller. "Texture feature selection with relevance learning to classify interstitial lung disease patterns." In Medical Imaging 2011: Computer Aided Diagnostics. 2011.

[40] R.O. Duda, P.E. Hart, and D.G. Stork. "Pattern Recognition." John Wiley & Sons, New York, 2 edition, 2001.

[41] W. Arlt, M. Biehl, A.E. Taylor, S. Hahner, R. Libe, B.A. Hughes, P. Schneider, D.J. Smith, H. Stiekema, N. Krone, E. Porfiri, G. Opocher, J. Bertherat, F. Mantero, B. Allolio, M. Terzolo, P. Nightingale, C.H.L. Shackleton, X. Bertagna, M. Fassnacht, P.M. Stewart. "Urine steroid metabolomics as a biomarker tool for detecting malignancy in adrenal tumors." Journal of Clinical Endocrinology and Metabolism 96: 3775-3784, 2011.