MSc Project
Feature Selection using Information Theoretic Techniques

Adam Pocock
pococka4@cs.man.ac.uk

15/08/2008

Abstract

This document presents a investigation into 3 different areas of feature selection, using information theoretic methods.

The first area of research is an investigation into the selection of the first feature in common feature selection algorithms. This step is often overlooked in the construction of feature selection algorithms, with the assumption that the most informative feature is the best first choice. This can be proven to be untrue, and so an investigation into how to select a better suited feature forms the first part of the research. New methods for selecting the first feature are proposed and empirically tested to see if they offer an improvement over the standard method.

The second area of research is an investigation into applying the Rényi extension to information theory to standard feature selection techniques. This requires the development of a Rényi mutual information measure, and two different measures are proposed. The Rényi extension provides a positive real parameter, α, which can be varied. The new Rényi feature selection techniques are empirically tested, varying the measure and value of α used.

The third area of research is an investigation into a different method of estimating the Rényi entropy of a variable, by using a function based upon the length of the minimal spanning tree of the variable. This enables a high dimensional estimate of the entropy to be constructed in $O(s \log s)$ time and can be used to implement the Max Dependency criterion. This research investigates the use of this estimator as a feature selection criterion and proposes a new genetic algorithm based method for selecting features, in contrast to the traditional forward search used in other algorithms.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institution of learning.

Copyright

Acknowledgements

I would like to thank my parents for their continued support, both personal and financial. I would also like to thank my supervisor Gavin Brown for his guidance which has made this thesis possible.

Contents

List of Figures

List of Tables

Chapter 1

Project Aims & Contributions

1.1 Motivation

Feature selection has become an area of interest due to the increasingly large amount of data that is generated by new scientific methods like genomics. It provides a mechanism for stripping out extraneous and noisy data to avoid masking the informative data. Different mechanisms for feature selection can be found such as linear correlation, and other statistical tests, but an emerging field is the use of information theoretic methods to select informative features using a quantifiable measure of information.

The use of such techniques provides a way of comparing the use of different features with different properties, and determining how useful each feature is to the current problem. Further information theoretic techniques enable the combination of features to be evaluated, and this enables informative feature sets to be constructed. Information theory can be applied when there are small numbers of samples in a dataset, which has applications in text-mining and genomics as there is usually a paucity of samples in biological fields. Improvements to feature selection techniques enable biologists to test large numbers of genes and extract ones which are relevant to a particular classification problem, which can lead to the inference of a link between a particular gene and a problem such as a disease. This knowledge can then either be used to aid the diagnosis of difficult cases or to indicate which genes need further investigation to generate more knowledge about a condition.

More generally, feature selection techniques provide a way of reducing the complexity and increasing the accuracy of classification and regression tasks as the stripping out of extraneous data enables a more streamlined classifier that is less likely to over-fit on certain data. The reduction of a feature set to its most useful subset enables further data to be collected in a less expensive way, by reducing the amount of data collected, and also to reduce

the classification time for new data because of the reduction in information that needs to be processed.

Research that improves the current set of feature selection techniques therefore has wide ranging applications in many fields of data analysis, and it is the analysis and development of feature selection techniques that forms the basis of this research.

1.2 Description of Document

Background The background chapter contains a detailed look into the process of feature selection and why it is becoming increasingly important as the size and complexity of datasets increases. It contains a review of information theory including several extensions to the theory which parametrise it. Then there is a review of the current state of the art feature selection algorithms, and how these use information theoretic techniques to select useful feature sets for further analysis.

Testing Framework This chapter details the datasets used for testing the various feature selection algorithms used throughout the research. It describes an optimisation to two of the techniques to bring an improvement in execution time, and a discrepancy found in the reference implementation of mRMR which can alter the outcome of the algorithm. Then the various implementations of algorithms are described and the classifiers used to test them are analysed. Finally the testing framework for the research is constructed, detailing how the selected feature sets are tested.

Comparing CMIM, DISR and mRMR An initial study is performed testing the state of the art algorithms against the selected datasets chosen for this work. Additionally a comparison of the execution times of the implementations of the algorithms is performed to give a baseline for the complexity of the algorithms modified for the research.

Investigating the first feature This chapter forms an investigation into the properties of the feature selection algorithms and how they select the first feature. The greedy assumption that the feature with the highest mutual information is the best feature to select first is questioned and a number of different methods for selecting the first feature are proposed. These novel methods are then empirically tested against the standard method using all the feature selection algorithms, and across all the datasets. Finally the analysis of the results leads to the conclusion that the greedy assumption is not optimal in the first stage of the feature selection algorithms, and selecting the

first feature using a criterion that promotes feature independence provides an increase in classification accuracy.

Investigating the Rényi measures of information This chapter forms an investigation into applying the Rényi entropy extension to information theory, and deriving a measure of mutual information. It is found that two methods can be used to measure the mutual information using the Rényi entropy, and this leads to two measures of mutual information parametrised by a positive real number α. The feature selection algorithms analysed previously are then modified to use these different measures of mutual information. The new algorithms are empirically tested using a range of values for the α parameter and the two different measures of the mutual information, across all the datasets and classifiers. Finally conclusions are drawn showing that the use of the Rényi mutual information in these algorithms can provide an increase in classification performance but it adds an extra layer of model inference and so is more prone to over-fitting the data.

Graph-based Entropy Estimation This chapter forms an investigation into a different method of estimating the Rényi entropy than the standard method using histogram bins. This method is adapted for use with feature selection in [1] and an extension to that work is developed which, instead of estimating the Shannon entropy, works directly from the Rényi entropy to derive measures of the Rényi mutual information as discussed in chapter 6. Then a genetic algorithm is developed to replace the forward search used in the feature selection algorithm, as the new entropy estimate is less prone to dimensionality problems found in other estimates. This genetic algorithm uses the new entropy estimator as the fitness function for stochastic hill-climbing to search for an optimal solution. The new feature selection techniques developed are empirically tested using the datasets and classifiers used elsewhere in the research. Finally conclusions are drawn about the performance of the new methods, with a functional flaw leading to poor performance of the new entropy estimator. However the performance of the genetic algorithm appears promising and deserves further study.

Conclusions This chapter provides a summary of the work performed in this research, and provides a list of further directions for research in this area to take.

1.3 Project Contributions

1.3.1 Knowledge Contributions

- A comparison of four state of the art feature selection algorithms, using a variety of datasets and two classifiers
- The creation of several different methods for selecting the initial features used in the 3 selection algorithms, including a thorough empirical investigation
- An theoretical study into the differing constructions of the mutual information using the Rényi entropy and generalised divergence, including a thorough empirical investigation
- A thorough investigation of the performance of the different feature selection algorithms when using different values of the Rényi α parameter
- A critique of the feature selection technique from [1]
- An investigation into using a genetic algorithm to replace the forward feature selection, with the graph-based entropy estimator
- A comparison between the graph-based forward feature selection algorithm, the graph-based genetic feature selection algorithm, and the Rényi versions of CMIM, DISR and mRMR, using a variety of datasets

1.3.2 Implementation Contributions

- An optimisation of the DISR and mRMR algorithms to improve execution time
- The development of a set of Rényi entropy based MATLAB/C++ functions to enable further work
- The recreation of CMIM, DISR and mRMR to use the Rényi based mutual information
- The creation of a MATLAB/C++ implementation of the Graph-based entropy feature selection technique from [1]
- The development of a genetic algorithm in MATLAB/C++ to replace the forward feature selection used in the graph-based feature selection algorithm

Chapter 2

Background

2.1 Machine Learning

Machine Learning is the use of statistical processes to derive information and rules used to interpret new data. It is designed to automate the process of analysing data and extracting rule-based information from it. This leads to several different areas of work. Machine learning systems can be trained to respond to certain stimulus to provide control frameworks for machinery or to automate tasks that have been demonstrated. Data-based systems can be trained to classify data into different groups based upon the values in each datapoint, or they can be trained to generate real values based upon the values in the datapoint to provide estimates of the value of new unseen datapoints. Both these approaches require labelled samples of data, with the end value, or class defined so the algorithm can learn by example. It is the classification of data that this research will focus upon.

Classification systems are being used to analyse large datasets generated in the fields of text analysis and genomics provide preprocessing for human researchers to remove extraneous data, and to perform complex numerical analysis to generate rules and trend information that human researchers would find difficult to extract from the surrounding noise of the dataspace. Simple classification tasks involve the placement of data samples into different categories based upon a function of the data contained in the sample. These tasks can be blurred by large quantities of data, as it becomes more difficult to sift out the informative pieces of data, from the surrounding noise.

2.2 Feature Selection

Feature Selection techniques have become increasingly important within Machine Learning as the increase in processing power of computers has enabled ever more complex datasets to be created, manipulated and analysed, and especially with the recent growth of Genomics as a scientific discipline. Feature Selection is a process where a group of features from a dataset are selected with the aim of improving classification accuracy and decreasing computational complexity [6]. It is closely related to Feature Extraction which is a process where feature vectors are created from the original dataset through manipulations of the data space, which can be considered to be a superset of Feature Selection techniques.

In addition to its use as a preprocessor for classification tasks, Feature Selection can also be used to identify features which are informative about different classes, so these features can be further analysed [6]. This is extremely useful in the analysis of gene expression data, which can contain many thousands of features, with each one taken from the expression values of an individual gene. The techniques can then be applied to find genes that appear indicative of certain classes, which can then be used for further experimentation.

Feature Selection techniques come in 2 main groups: filters and wrappers [6]. Filter techniques use classifier independent methods to select features from the feature space. These techniques are based on a number of different statistical tests, or on the information theoretic concept of Mutual Information. Wrapper techniques embed the feature selection into a classifier, where the classification performance is used to measure the quality of the currently selected feature set [8]. A further group of embedded techniques are similar to wrappers, but instead of using the classifier as a black box it integrates the classifier into the feature selection algorithm. It is similar to the training of a neural network which can gradually reduce the influence of useless inputs by giving it a lower weight.

The different techniques have differing advantages and drawbacks. Filter techniques rely upon measuring information values which have to be defined separately from classification performance, and do not necessarily translate to a set of features which is the optimal set for a particular type of classifier. However they are faster than wrappers, and find a set of features which can be shown to have a high level of information about the target irrespective of the classifier. Wrapper techniques are extremely dependent on the classifier used to test the performance of the feature set. They find feature sets which are highly linked to the choice of classifier, so the features selected do not generalise well to other classifiers. Due to the wrapping process the selected features are well suited for use with the chosen classifier, so may produce a more accurate class prediction. Also, due to the process of constantly retraining and testing a classifier, wrapper techniques have a high computational complexity

when compared to filter techniques [6].

2.2.1 Search Strategies

Both types of technique require a search strategy to be selected, to determine how the selection space is searched. As the search space for feature sets is exponential in the number of features, the most commonly used strategy is a greedy search, choosing the best option at each current iteration rather than taking a sub-optimal option to increase the solution performance in later iterations. The two main kinds of search strategy are forward search and backward search. Forward search starts from an empty set of selected features, and then proceeds to add features to the set until a specific requirement has been reached. This can be stated as, at each iteration m, the feature x_n is found which maximises the condition F, which is then added to the solution set S (equation (2.1)).

$$S = S \cup \{\arg \max_{X_n \in X_{\setminus S}} F(X_{\setminus S})\} \tag{2.1}$$

Backward search starts from a set of all the features then proceeds to eliminate features from the set until a specific requirement has been reached. This can be stated as, at each iteration m, the feature x_n is found which minimises the condition F, which is then removed from the solution set S (equation (2.2)).

$$S = S \setminus \{\arg \min_{X_n \in S} F(S)\} \tag{2.2}$$

Backwards selection can be computationally more intensive when selecting a small number of features from a large feature set, as there are more iterations before the search reaches the desired number of features.

Search strategies can also be constructed that combine both types, so features are added to the set, and then features are removed, and the process is iterated until the set has met the requirements. The stopping conditions for the algorithms are usually defined as being a set number of selected features, but in the case of wrappers the condition can be the classification performance of the selected features using the internal classifier. This helps stop over-fitting of data by not adding extraneous information.

2.3 Document Notation

Throughout the rest of this document the following notation will hold. Capital letters will denote random variables, and lower case letter denote a value from that random variable, e.g. X, Y and Z denotes random variables, with x, y and z being a particular value from their respective variables. XY denotes the joint random variable composed from X and Y, and $X|Y$ denotes the conditioning of X by Y. $p(x)$ denotes the probability of the event recorded in x occurring. S denotes the selected set of features, and F denotes the set of all features. k denotes the number of desired features output by a feature selection algorithm, n denotes the number of features input into the algorithm (otherwise known as the cardinality of F).

2.4 Information Theory

To have a way of classifying features by information level, first information itself must be quantified in a way that is not subjective. This was proposed by Shannon in 1948 [20], and is known as Shannon's Information Theory. This formulation is the standard method for quantifying information but other methods have been proposed, namely by Rényi in 1961 [18].

Information Theory is constructed from analysis of information flows in systems, and how much data is required to transmit information [20]. A more general explanation of the topic of information theory can be found in [3]. The main concept in Information Theory is Entropy, which measures the degree of uncertainty in a single variable. It gives high values to variables where each possible value has an equal probability and low values where values have an unequal probability distribution (e.g. for the boolean random variable {0,1,1,0} either 0 or 1 is equally likely thus it has a high entropy, for the boolean random variable {0,0,0,1}, 0 is much more likely than 1, thus it has a low entropy). Entropy is valued between 0 and 1 when there are 2 possible values, with the maximum possible entropy increasing depending upon the number of possible values in the variable.

Shannon's Information Theory uses the logarithm as the method for deriving the amount of information held in a variable.

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2(p(x_i)) \tag{2.3}$$

Where X is a random variable with n possible values, $H(X)$ is the entropy of X, and $p(x)$

is the probability of the value x occurring in X.

It sums over the possible values of X taking the probability of each value of X multiplied by the logarithm of the probability. The logarithmic base is usually chosen to be 2, which results in an entropy value measured in bits, though other bases are possible.

Other bi-variate entropies are also defined such as the joint entropy and the conditional entropy. The joint entropy is the sum of the uncertainty contained in two variables, and is bounded above by the sum of the two individual entropies, and bounded below by the highest individual entropy.

$$H(XY) = -\sum_{i=1}^{n}\sum_{j=1}^{m} p(x_i, y_j)\log_2(p(x_i, y_j)) \tag{2.4}$$

$$\max(H(X), H(Y)) \leq H(XY) \leq H(X) + H(Y) \tag{2.5}$$

The lower equality is true when X is completely dependent upon Y, or vice versa. The upper equality is true when X & Y are independent of each other. This is because adding a variable to a system can only increase the uncertainty present, but not decrease it.

The conditional entropy is the uncertainty remaining in one variable when another variable has a known value. It is bounded below by 0, and above by the entropy of the original variable. The lower bound is true when X is completely dependent upon Y, and the upper bound is true when X & Y are independent of each other. In general conditioning on entropy reduces the value of the entropy, as it is rare that two variables are completely independent.

$$H(X|Y) = \sum_{j=1}^{n} p(y_j)H(X|Y = y_j) \tag{2.6}$$

$$0 \leq H(X|Y) \leq H(X) \tag{2.7}$$

The interactions between all these values can be modelled in the two variable case by a Venn diagram, showing how the values relate to each other and to the quantity of mutual information. In the diagram (figure 2.1) the term $H(XY)$ relates to the total area of both $H(X)$ and $H(Y)$. Also it can be seen that the conditional entropy is also equal to the joint entropy minus the entropy of the conditioning variable.

$$H(X|Y) = H(XY) - H(Y) \tag{2.8}$$

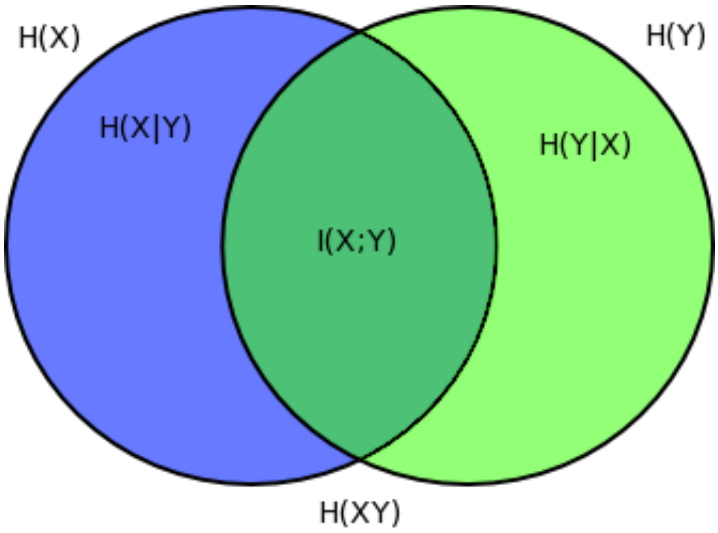

Figure 2.1: How entropy and mutual information relate

2.4.1 Mutual Information

Mutual Information (MI) is defined as a measure of how much information is jointly contained in two variables, or how much knowledge of one variable determines the other variable. It forms the basis of information theoretic feature selection as it provides a function for calculating the relevance of a variable to the target class, irrespective of correlations and other assumptions.

Mutual Information can be defined in two different but equivalent ways, either in terms of entropies, or in terms of probability distributions, and it is symmetric with respect to the ordering of X and Y.

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ I(X;Y) &= H(Y) - H(Y|X) \\ I(X;Y) &= H(X) + H(Y) - H(X,Y) \end{aligned} \tag{2.9}$$

These differing formulations of the mutual information all result in the same value, and are related by figure 2.1. The mutual information is the overlap between the variables, and can thus be obtained as the sum of the two entropies, minus the joint entropy which removes all information that is not contained twice in the variables. This gives rise to an equation for the mutual information. Another equation can be obtained from the entropy minus the conditional entropy as the amount of uncertainty that is lost through the conditioning of the variable by another variable, and lost uncertainty is equivalent to a gain in information. This conditional equation is symmetric, so the ordering of the variables can be changed.

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log(\frac{p(x,y)}{p(x)p(y)}) \tag{2.10}$$

$$I(X;Y) = D_{KL}(p(x,y)||p(x)p(y)) \tag{2.11}$$

The probability based definition is defined as the Kullback-Liebler divergence between the joint probability distribution and the sum of the marginal distributions. The Kullback-Liebler divergence is similar to a distance metric between two probability distributions, however it is not a true distance metric as it is not symmetric.

It can also be defined in the continuous case, where the discrete probability values become continuous probability densities.

$$I(X;Y) = \int_Y \int_X p(x,y) \log(\frac{p(x,y)}{p(x)p(y)}) \tag{2.12}$$

In general the mutual information when used for feature selection tasks is constructed in the discrete form, or in the continuous form upon a discrete class. This leads to a summation over the discrete class space, and an integral over the continuous feature space.

Conditional forms of the mutual information can be formed but unlike with entropies, conditioning the mutual information does not always reduce it. This is because the knowledge of a third variable can make the two original variables dependent upon each other, (e.g. in the case of the XOR problem detailed in Chapter 5, conditioning by the other variable makes the mutual information equal to the entropy of the class).

$$I(X;Y|Z) = \sum_{z \in Z} p(z) I(X;Y|Z=z) \tag{2.13}$$

In the multivariate case the mutual information becomes a more complex measure, and can no longer be represented by a simple diagram. This is because the presence of an additional variable can increase the information between the original variables more than the sum of their parts.

$$I(X_{1,2};Y) = I(X_1;Y) + I(X_2;Y) + \underbrace{(I(X_1;X_2|Y) - I(X_1;X_2))}_{\text{increases or decreases information}} \tag{2.14}$$

In the example above it is possible for $I(X_{1,2};Y) > I(X_1;Y) + I(X_2;Y)$ or $I(X_{1,2};Y) \leq I(X_1;Y) + I(X_2;Y)$ depending upon the interaction of the variables X_1 and X_2. Therefore an information system cannot be treated as a strict sum of its various components, as they interact in counter-intuitive ways.

2.5 Information Theory and Feature Selection

Once a metric has been decided for quantifying information then a number of feature selection techniques can be created. The most simple and intuitive one is to rank the features by their mutual information upon the class, and then select the top k features to use for classification. This technique is the simplest to implement, and requires the fewest evaluations of the mutual information (n evaluations, where n is the number of features), but it leads to some problems. Firstly it has the potential to introduce large amounts of redundant information, that is information already held in a selected feature, and secondly it does not take into account the multivariate interactions of a set of features. Therefore more complex techniques have been devised to try and reduce the flaws inherent in this selection algorithm, with an accompanying increase in computational complexity and the number of mutual information evaluations. These techniques try and generate an optimal feature set, but before they can be created the notion of an optimal feature set must be defined.

Optimality for feature sets can be defined in numerous ways, based upon the number of features selected, and various techniques for measuring the information content of a feature set. Estimating the information content of a feature set becomes computationally intractable the more features that are added to the set, as the mutual information operator scales badly with dimension. One accepted definition of optimality in feature selection comes from the analysis of Markov blankets [9].

A Markov blanket is defined as a set of features, M, such that all the information held in a particular feature X not in the set is captured. This enables the removal of the feature X from the selected feature set as it is redundant. Koller & Sahami then prove that if a previously removed feature (F_i) had a Markov blanket within the feature set (G), and another feature (F_j) that is about to be removed has a Markov blanket in the set G, then F_i has a Markov blanket in $G - F_j$. This gives rise to a backward search of the feature space, removing only features whose information is completely contained in the remaining set.

The Markov blanket result forms an ideal method for calculating an optimal feature set in terms of minimal redundancy. It does not aim to produce a feature set of an expected size, though the algorithm could easily have such a criterion introduced by modifying it to calculate approximate Markov blankets. This would mean that a feature would be discarded if enough of its information was held in the current feature set, where this value could be modified. However the calculation of Markov blankets is computationally intractable, as it has an exponential complexity.

From this notion of a redundant feature several algorithms have been developed which explicitly punish redundancy in the selected feature set. These algorithms are all formulated

using the standard definition of entropy and mutual information. Additionally they are all greedy forward searches, instead of the backward search proposed in [9]. The techniques chosen for study in this research have been detailed below.

2.6 Feature Selection Algorithms

2.6.1 mRMR

From the work done by Koller & Sahami which provides a formalisation for the redundancy of a feature set, and the use of the mutual information on the class as a measure of relevance, several different ways have been used to combine the two [17, 23]. The accepted method is detailed in [17], as this also provides a useful selection algorithm that compares well with other techniques.

The framework given in [17] defines the terms relevance, redundancy, and introduces the maximum dependency criterion, and the minimal-Redundancy Maximal-Relevance (mRMR) criterion. It defines the relevance of a feature as its mutual information on the class, and thus the maximal relevance will select features based upon the highest mutual information values on the class. Redundancy is defined as the amount of information held about a feature by the feature set and approximated by averaging the bi-variate mutual informations between the feature and each individual feature in the feature set (equation (2.15)) where X_i is the candidate feature and S is the feature set).

$$R = \frac{1}{|S|^2} \sum_{X_j \in S} I(X_i; X_j) \tag{2.15}$$

The maximum dependency criterion is to find the feature set which maximises the mutual information of the feature set upon the class (equation (2.16)) where F is the feature set).

$$D = \max_{S \subseteq F} I(S; Y) \tag{2.16}$$

Maximum dependency is computationally intractable with the standard method of computing the mutual information, as multi-variate mutual informations are too computationally intensive to calculate, and also it is an exponential search over the feature space [16]. Because of this problem the paper then proposes the mRMR criterion as an acceptable tradeoff between calculating the maximum dependency and the performance of the algorithm. The criterion used in the algorithm is designed to simultaneously minimise the shared information within the feature set, and maximise the information on the class. The mRMR criterion

is developed to approximate the maximum dependency whilst remaining computationally tractable for large datasets.

The mRMR criterion is a bi-variate measure of information held in a single feature, penalised by an average of the inter-feature dependence. The criterion is proved to be equivalent to the maximum dependency in the first-order case, where features are added to the set one by one.

$$X_{\text{mRMR}} = \arg \max_{X_n \in F_{\setminus S}} I(X_n; Y) - \frac{1}{|S| - 1} \sum_{X_k \in S} I(X_n; X_k) \tag{2.17}$$

The mRMR criterion has no tri-variate estimation, meaning that it is an efficient algorithm when implemented. However this reduces the selection power of the criterion as it has no method for determining any form of variable complementarity, making it ill-suited for feature selection on datasets that have high levels of complementarity.

2.6.2 CMIM

There is a limit to the level of performance that can be extracted from bi-variate feature selection methods. This is due to the issue of feature complementarity, where more information can be found through the use of multiple features at once than can be used by taking features separately.

One of the first algorithms that can be classified as using feature complementarity is the Conditional Mutual Information Maximisation (CMIM), presented in [4] (with erratum issued here `http://www.idiap.ch/~fleuret/papers/fleuret-erratum-jmlr2004.pdf`). The criterion used in the algorithm is designed to minimise the shared information within the selected feature set. This is to prevent redundant information accumulating in the feature set at the expense of more specific information.

The CMIM criterion is a tri-variate measure of the information held in a single feature about the class, conditioned upon an already selected feature. It loops over the selected features and assigns each candidate feature a score based upon the lowest conditional mutual information, between the selected features the candidate feature and the class. The feature then selected is the one with the maximum score.

$$X_{\text{CMIM}}(1) = \arg \max_n I(Y; X_n) \tag{2.18}$$

$$\forall k, 1 \leq k \leq K, X_{\text{CMIM}}(k+1) = \arg \max_n \min\{I(Y; X_n) \min_{l \geq k} I(Y; X_n | X_{\text{CMIM}}(l))\} \tag{2.19}$$

As the algorithm takes a pessimistic view of the data, by taking the minimum value as the

score for a particular feature, this enables a shortcut in the algorithm implementation. At each iteration the score for a particular feature can only decrease so scores that are already below the current best score are not updated in the current run through, as they do not affect the calculation. Also as the scores are maintained through iterations it means there is no need to recalculate the conditional mutual information between already selected features and the remaining features as these values persist throughout the algorithm.

The CMIM algorithm is precisely defined with explicit conditions in the initial loop, and so is not ambiguous in how the algorithm proceeds. The fast implementation of CMIM makes the algorithm efficient and quick to execute. The implementation also loses no accuracy over the direct implementation of equation (2.19). The implementation used is detailed in chapter 3. As the CMIM algorithm only considers conditional mutual information it fails to cope well with complementary variables as the pessimistic assumption does not improve the scores of highly complementary variables. This means it may not be well suited to working on datasets with a high complementarity such as gene expression data.

2.6.3 DISR

Feature complementarity can have a large effect upon the information level of the resulting feature set, and so an algorithm was developed by Bontempi & Meyer in [12] to explicitly use complementarity to select features. The resulting criterion, called the Double Input Symmetrical Relevance (DISR) is a tri-variate measure of the information held in two features about the class. It is based on the Symmetrical Relevance (SR) which is defined as the mutual information of the feature on the class, normalised by the joint entropy of the feature and the class (see equation (2.20)). In DISR this is always used in the 3 variable case where it takes the joint mutual information of the two features on the class, normalised by the joint entropy of the 2 features and the class.

$$SR(X;Y) = \frac{I(X,Y)}{H(X,Y)} \tag{2.20}$$

At each stage of the algorithm the selected feature is then the feature that maximises the sum of the SR's over all the currently selected features (see equation (2.21)).

$$X_{\text{DISR}} = \arg \max_{X_n \in F_{\setminus S}} \left(\sum_{X_k \in S} SR(X_{n,k};Y) \right) \tag{2.21}$$

The DISR criterion works well when variable complementarity is a feature of the dataset, outperforming CMIM in most cases and performing equivalently with mRMR, but its more

computationally intensive than either due to the tri-variate mutual information evaluations. One flaw in the algorithm as presented is that it fails to state explicit initial conditions for the algorithm, implicitly assuming that the equation (2.21) reduces to selecting the feature with the highest mutual information in X.

These techniques form the state of the art with respect to information theoretic feature selection, using Shannon's measure of entropy. The DISR criterion takes into account two feature complementarity, and this technique captures the most information about the feature space. Unfortunately none of the techniques described above can perform multi-variate feature analysis in a computationally tractable amount of time. They are all tricks in information theory to approximate the multi-variate estimations. As computation power increases it should be possible to extend DISR to take 3 feature variables at once, but this will produce a significant increase in computation time, as it adds another order of complexity to the algorithm.

2.7 Rényi's Information Theory

2.7.1 Rényi Entropy

Rényi in a paper analysing the Shannon entropy [18] proposed a new class of entropies which form a generalisation of the Shannon entropy. This definition of entropy has a parameter α and is defined $\forall \alpha \geq 0, \alpha \neq 1$.

$$H_\alpha(X) = \frac{1}{1-\alpha} \log(\sum_{x \in X} p(x)^\alpha) \tag{2.22}$$

Additionally the Shannon entropy is the limiting case of the Rényi entropy as α tends to 1.

$$\lim_{\alpha \to 1} H_\alpha(X) = H(X) \tag{2.23}$$

From this entropy the joint entropy is easily formed, and holds as the joint probability distribution. However it is not possible to conventionally form the conditional entropy. An expression for the conditional entropy can be created but it does not hold to the relation established in equation (2.8), as demonstrated by figure 2.2. This created conditional entropy does however take the Shannon conditional entropy as its limiting case.

$$H_\alpha(X|Y) = \sum_{y \in Y} p(y) \frac{1}{1-\alpha} \log(\sum_{x \in X} p(x|Y=y)^\alpha) \tag{2.24}$$

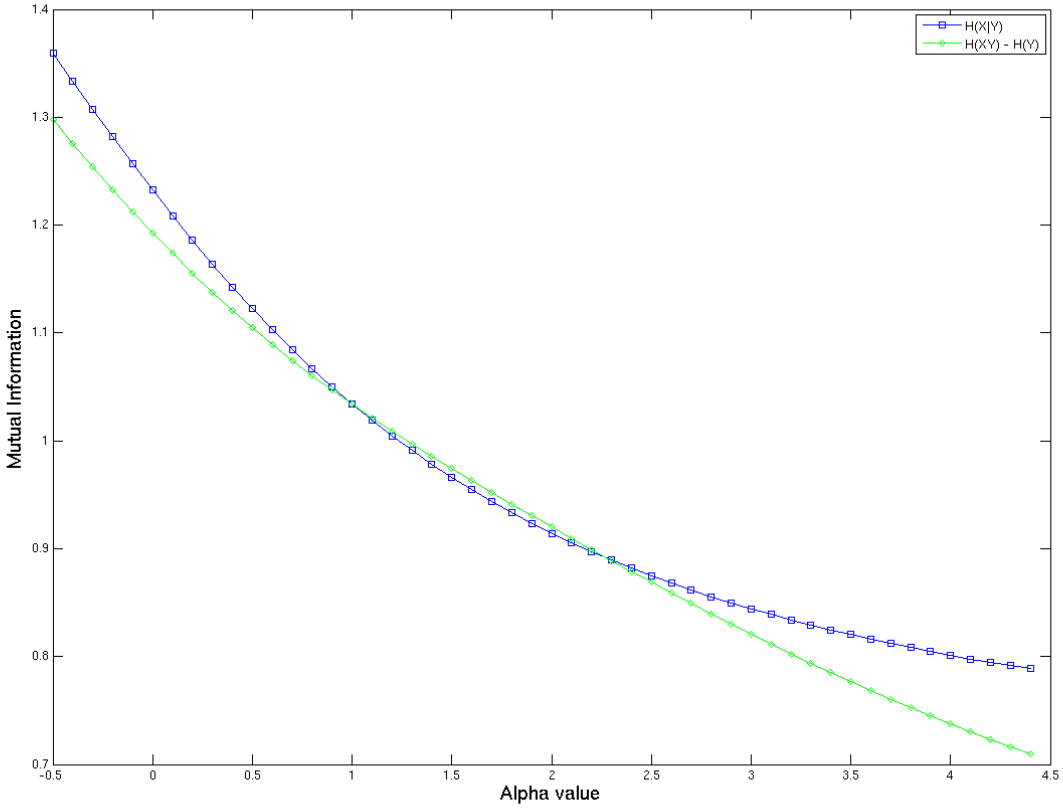

Figure 2.2: Difference between $H_\alpha(X|Y)$ and $H_\alpha(XY) - H_\alpha(X)$, using the Lung dataset, and X = feature 1, Y = class

The implications of this are explained in detail in chapter 6.

2.7.2 Rényi Generalised Divergence

Additionally Rényi specifies a series of divergence measures that generalise the Kullback-Leibler divergence [11], providing a range of metrics for finding the divergence between two probability distributions. These are also parametrised by α defined $\forall \alpha \neq 1$.

$$D_\alpha(P||Q) = \frac{1}{\alpha - 1} \log(\sum_{k=1}^{n} p_k^\alpha q_k^{1-\alpha}) \tag{2.25}$$

Again the Kullback-Leibler divergence forms the limiting case of the Rényi generalised divergence as α tends to 1.

$$\lim_{\alpha \to 1} D_\alpha(P||Q) = D_{KL}(P||Q) \tag{2.26}$$

These divergences can be used to construct a mutual information like value similar to the Shannon one. This is achieved by substituting into the α divergence the values $p(x, y)$ and $p(x)p(y)$. This forms a divergence between the joint probability density and the sum of the marginals like the mutual information, and is thus called the Rényi mutual information, and will be denoted throughout this paper as $I_\alpha(X; Y)$.

$$D_\alpha(p(x, y)||p(x)p(y)) = \frac{1}{\alpha - 1} \log(\sum_{x \in X} \sum_{y \in Y} p(x, y)^\alpha p(x)^{1-\alpha} p(y)^{1-\alpha}) \tag{2.27}$$

This value, like the Rényi entropy, tends to the standard mutual information as $\alpha \to 1$, and can thus be used as a basis for analysing the usefulness of the Renyi measure of information with respect to feature selection, and analysing the different values of the α parameter, and how this affects performance.

$$\lim_{\alpha \to 1} I_\alpha(X; Y) = I(X; Y) \tag{2.28}$$

2.7.3 Rényi entropy estimation

There are various different ways of estimating the entropy of a variable. The standard method used with discrete entropy is to derive the probability for a given event from the number of times the event occurs divided by the number of observed events, as seen in [19]. This can be extended to cope with unseen events, but introduces uncertainty. The standard method for estimating the continuous entropy uses a Parzen Window [15] with a suitable interpolation function to provide an estimate for the probability density function [21]. A new method of estimating the continuous entropy was developed by Hero et al in [7], and was extended in [13]. This graph-based method for estimating the entropy is not subject to the dimensionality constraints of the Parzen Window estimator and enables the entropy of a multidimensional variable to be estimated. The applications of this estimator are explored in detail in chapter 7

2.8 Summary

This chapter has provided an introduction to the field of Feature Selection and Information theory. It has:

- Detailed why feature selection is used
- Explained the basics of information theory

- Shown how information theory is used to construct feature selection algorithms
- Explained how information theory is expanded and complicated by Rényi's work

Chapter 3

Testing Framework

3.1 Datasets

The datasets used for this research comprise a selection of microarray experiments recording gene expression levels related to various different types of cancer. These datasets are taxing for classifiers due to their small number of samples compared with the high number of features, and is one of the reasons feature selection techniques have been an area of research, as they alleviate some of the problems caused by the high number of features.

3.1.1 Public Datasets

The first 5 datasets used in the research are taken from various sources detailed in [17], which collates and uses them in the analysis of the mRMR-D and mRMR-Q feature selection algorithms. All the features in these algorithms are discretised into three states, with a varying number of classes and examples per class.

Title	Features	Classes	Examples
Colon	2000	2	62
Leukaemia	7070	2	72
Lung	325	7	73
Lymphoma	4026	9	96
NCI 9	9712	9	60

Table 3.1: Dataset properties

3.1.2 Adenocarcinoma Dataset

A dataset has been provided by Prof. Andrew Brass of the University of Manchester which contains gene expression data taken from cancer patients. The dataset contains 41672 continuous features with 2 resultant classes (correlating to two different forms of lung cancer), and 59 samples. Before being used in this research the dataset was discretised along the mean into two states, this reduces some of the problems when trying to generalise from the continuous data.

3.2 Classifiers

Classifiers are algorithms that take an amount of training data, and can then be used to predict the class of any new data given. They can be used with a set of known data to provide an estimate for the information contained in the features given by training on a portion of the data, then testing the prediction given against the rest of the data. The correct classification rate can then be used as a measure of how much information the classifier could extract from the chosen features.

Different classifiers have different inherent biases. This means that a set of features that works well with one classifier is not guaranteed to work well with another, so to judge the usefulness of different feature selection algorithms then multiple classifiers must be used.

There were 2 chosen classifiers used in this work for testing feature selection algorithms, they were a linear Support Vector Machine (SVM), and a 3-Nearest Neighbour classifier.

3.2.1 Support Vector Machine

Support Vector Machines are a way of generating an optimal classification boundary in many dimensional space. The linear variant acts much like a perceptron, and fits a linear separating boundary to the dataset, that correctly classifies the maximum amount of training data. The algorithm can be modified with a kernel to map the boundary into a higher dimensional space, to give a non-linear separating boundary when mapped back into the original space.

3.2.2 k-Nearest Neighbour

The k-NN classifier is the simplest way of classifying data. It takes the nearest k neighbours of the new sample and classifies it with the largest group of neighbours. Effectively this creates a non-linear boundary separating the classes depending upon the original training data. An odd value should be chosen for k, as otherwise it introduces ambiguity into the

classification process as there may be an equal number of examples in all the classes. In the case where two classes are tied for the highest number of examples, the algorithm generates a random number to select which is the predicted class. This can still be a problem when an odd value of k is chosen if the number of classes is greater than 2 (e.g. in the case where you have a 3 class problem, and a 3-NN classifier, you could have an example of each class as the 3 nearest neighbours, leading to ambiguity).

The k-NN classifier, despite being one of the simplest, provides good classification accuracy, and is simple to construct and understand which is why it was chosen for this research.

3.3 Feature Selection Algorithms

The three chosen algorithms (CMIM, DISR, mRMR) were all recoded in C++/MEX with a MATLAB wrapper script to improve their performance. This was due to the time constraints and the number of tests that were performed using each algorithm. The basic Shannon entropy and mutual information functions are taken from H. Peng's mutual information toolbox, constructed for the work in [17], and then reverted to C++ and streamlined. The basic Rényi entropy and mutual informations are constructed using the probability functions from the mutual information toolbox, and developed in C++, with MATLAB wrappers to enable their use in other algorithms.

The accompanying code will be released with source at a later date.

3.3.1 CMIM implementation

The CMIM implementation is a C++ implementation of the optimised pseudocode provided in [4]. This pseudocode exploits the minimisation property of the CMIM equation to avoid calculating results that can only be worse than the current optimum. The resulting implementation is extremely optimised and was not suitable for further algorithm optimisation.

3.3.2 DISR implementation

The DISR implementation is a modified C++ implementation of the criterion provided in [12]. The paper presents an algorithm based around the DISR criterion. A simple implementation of this algorithm proved to be inefficiently recalculating past values. A decision was made to sacrifice some of the space complexity of the algorithm in exchange for an decrease in computation time. This involves storing SR values between selected features and candidate features, and reloading this value rather than recalculating it from scratch.

```
for i = 2 to k
  for j = 1 to totalFeatures
    if not selected j
      for m = 1 : selectedFeatures
        currentScore =  currentScore + SR(m,j,class);
      end
      if currentScore > score
        score = currentScore;
        currentHighestFeature = j;
      end
    end
  end
  answerFeatures(i) = currentHighestFeature;
end
```

Figure 3.1: Original DISR pseudocode

The pseudocode given in figure 3.1 has $\frac{k^2*n^2}{2}$ calls to the SR function.

The optimisation, given in figure 3.2, adds one operation to each iteration of the loop, but only has $k * n$ calls to the SR function, as it stores the previously computed values. This leads to a large improvement in computation time, whilst not altering the output of the algorithm in any way. This does result in an increase in the memory used of $k * n$ doubles, but due to the performance increase this was found to be acceptable.

Table 3.2 shows the increase in performance using the optimised version of the DISR algorithm over the standard naive implementation. The final C++ implementation is included to show the speed increase of MEX implementation over a MATLAB script. The optimised implementation provides a 43 fold improvement on execution time, on average, with the C++ implementation providing a further 9 fold increase, on average.

Dataset	DISR (MATLAB)	DISR Optimised (MATLAB)	Optimisation Performance Increase
Colon	2259.88	50.24	44.98x
Leukaemia	8300.95	183.63	45.20x
Lung	316.00	7.39	42.76x
Lymphoma	4973.54	109.56	45.39x
NCI 9	10328.08	258.15	40.01x

Table 3.2: DISR optimisation performance

```
for i = 2 to k
  for j = 1 to totalFeatures
    if not selected j
      for m = 1 to iMinus
        if not calculated SR(m,j,class)
          calculate SR(m,j,class)
          store in SRMatrix
        end
        currentScore =  currentScore + featureSRMatrix(m,j);
      end
      if currentScore > score
        score = currentScore;
        currentHighestFeature = j;
      end
    end
  end
  answerFeatures(i) = currentHighestFeature;
end
```

Figure 3.2: Optimised DISR pseudocode

3.3.3 mRMR implementation

The reference implementation of mRMR contains an modification that is designed to speed up the calculation time on large datasets (over 1,000 features). It only labels the top 1000 features ranked by their mutual information as possible candidates for selection even if the dataset is much larger. This modification is not detailed in the paper that details the algorithm [17] and changes the criterion, meaning that other experiments performed where the algorithm has been reconstructed cannot be compared with the original experiments given in [17]. It changes the set F used in equation 3.1 so instead of containing all the features, it contains the top 1000 ranked by the mutual information on the class.

$$X_{\text{mRMR}} = \arg \max_{X_n \in F_{\setminus S}} I(X_n; Y) - \frac{1}{|S| - 1} \sum_{X_k \in S} I(X_n; X_k) \tag{3.1}$$

This could lead to inaccuracies when the algorithm is executed on a dataset where the standard mutual information is a bad measure for differentiating between features in the dataset, when selecting a number of features close to 1000, or when selecting features from a dataset with an extremely large number of features. As a consequence of this all results for mRMR use a version of the algorithm where this modification has been removed, due to the

possible bias consequences when comparing between different feature selection techniques, and so the algorithm used is a strict implementation of the mRMR criterion.

In addition to this flaw with the reference implementation, which was corrected in the modified C++ implementation of mRMR used for this research, a similar optimisation to the one used in the DISR implementation was added. Where the DISR implementation only calculates the SR if it hasn't previously calculated this value then the mRMR implementation only calculates the intra feature mutual information if this value hasn't been previously calculated. This results in a similar complexity decrease to the modified DISR implementation. Both the Quotient and Difference variants of mRMR were coded, and both were optimised and corrected in the same way. It is to be noted that this optimisation does not change the performance of the algorithm when compared to a strict implementation of the mRMR criterion, as compared with the modification in the reference implementation.

3.3.4 Graph implementation

The implementation of the graph-based entropy estimator was done in C++, with a MATLAB wrapper. It uses Kruskal's algorithm [10] for finding the minimal spanning tree of a graph, as implemented in the Boost C++ library ver 1.35 (`http://www.boost.org/`). The remainder of the code was implemented around the minimal spanning tree algorithm in C++.

3.4 Test Construction

There are several problems in analysing datasets with low numbers of samples and high numbers of features. The high number of features raises the probability that any particular strongly correlated feature has only achieved this correlation through random chance, and is unrelated to the classification of the variable. The method of detection for this problem is to acquire more data, but this is often not feasible.

The low number of samples causes different issues, as it leads to an over-fitting of the classifier to the data. This is one of the problems that feature selection tries to mitigate as it reduces the complexity of the task, though it is still possible for the feature selection algorithm to over-fit to the task due to the high number of features. Additionally the paucity of samples leads to problems when splitting the data into training and testing sets for the construction of classifiers, as there is not enough data to ensure a satisfactory level of reliability in the results. To alleviate this problem cross-validation is used, where the data is broken up into several folds, with one folds selected for testing, and the remainder used

for training. This is repeated until all the folds have been used for testing, and the resultant errors are averaged to give a measure of the performance.

Due to the nature of the datasets used for this research leave one out cross-validation is used, where the size of each fold is reduced to 1. Then the test error is the number of incorrect classifications divided by the number of samples. It is not possible to construct error bars with this technique, as the overall error is only determined after all folds have been tested, so there is no mean calculated.

The first area of research, into the selection of the first feature in feature selection algorithms uses the 3 different algorithms constructed to select 50 features, with the first feature being selected by a variety of different methods. These are then tested using the 2 selected classifiers, and graphed.

The research related to various investigations of the Rényi entropy select 100 features in each of the tests, which are then tested using the 2 selected classifiers.

Each generated feature set is tested using the first k features selected by the algorithm, $1 \leq k \leq m$ where m is the total number of features selected. This generates a graph of the performance to see which subset of the selected features produces the best classification accuracy. The graphs display the classification error rate, ranging between 0 and 0.5 for the Rényi investigation, and between 0 and 0.6 for the first feature investigation. This however means the performance of the first few features can be missing from the graph of the performance of the Rényi algorithms, but the scale would cause visibility issues with the smaller fluctuations in performance if a larger scale was used.

Chapter 4

Comparing CMIM, DISR and mRMR

4.1 Introduction

Before the various investigations into different properties of the three feature selection algorithms a comparative test will be performed between the unmodified versions of these algorithms on the datasets detailed in chapter 3.

4.2 Results

Each of the tests were performed according to the strategy given in chapter 3. The discrete jumps in error rate present in the graphs are due to the low sample number and the technique of leave one out cross validation, as the error rate can only take values of the form $\frac{i}{n}$ where n is the number of samples, and i is an integer.

4.2.1 Lymphoma Dataset

This dataset has 4026 features, discretised into 3 states. There are 9 classes, and 96 examples. The graphs are shown in figure 4.1.

4.2.2 NCI 9 Dataset

This dataset has 9712 features, discretised into 3 states. There are 9 classes and 60 examples. The graphs are shown in figure 4.2.

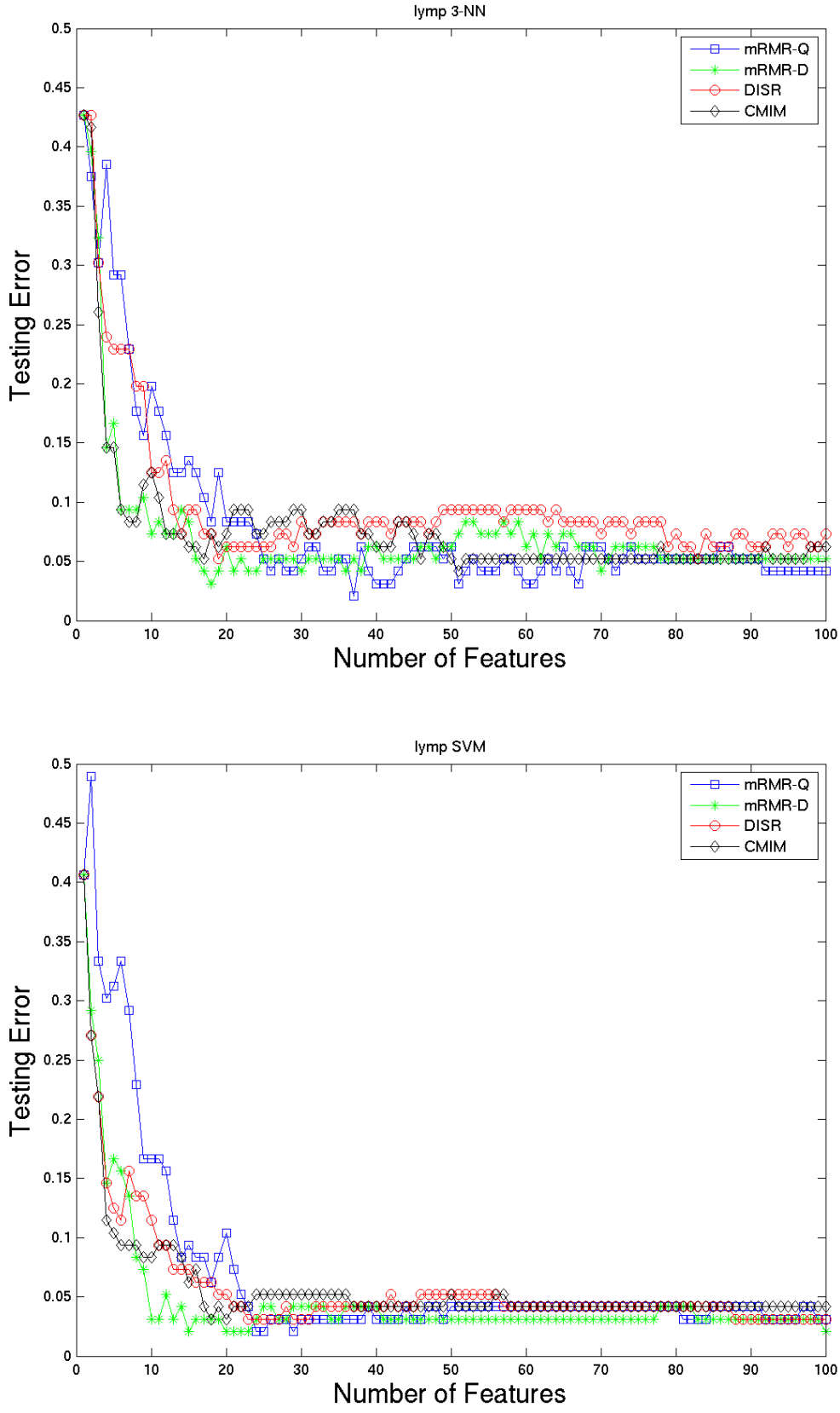

Figure 4.1: Lymphoma Dataset, 3-NN and SVM Classifiers

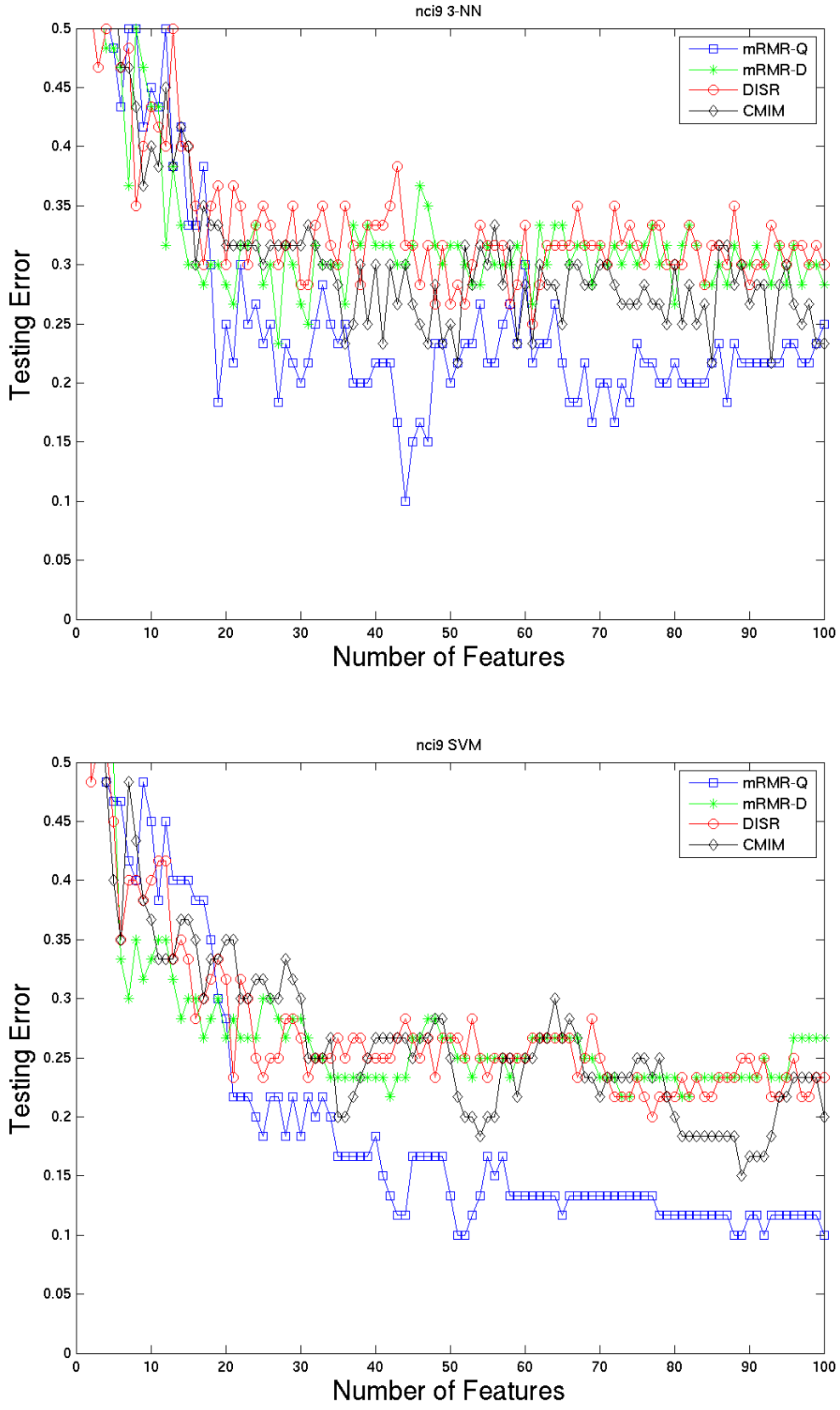

Figure 4.2: NCI9 Dataset, 3-NN and SVM Classifiers

4.2.3 Colon Dataset

This dataset has 2000 features discretised as 3 states. There are 2 classes and 62 examples. The graphs are shown in figure 4.3.

4.2.4 Leukaemia Dataset

This dataset has 7070 features discretised into 3 states. There are 2 classes and 72 examples. The graphs are shown in figure 4.4.

4.2.5 Lung Cancer Dataset

This dataset has 325 features discretised into 3 states. There are 7 classes and 73 examples. The graphs are shown in figure 4.5.

4.2.6 Adenocarcinoma Dataset

This dataset has 41672 features, discretised into 2 states. There are 2 classes and 59 examples. The graphs are shown in figure 4.6.

4.3 Analysis

4.3.1 Adenocarcinoma Dataset

The mRMR-Q algorithm has exceptional performance on this dataset, after 6 features for the 3-NN classifier and 9 features for the SVM it is able to select features which perfectly predict the result. This performance is due to the low number of samples, and the way the dataset is discretised, as it removes large amounts of data which could confuse the classifier. The remaining 3 feature selection techniques converge to accuracies below 5%. With the 3-NN classifier CMIM and mRMR-D converge to the same result with a classification error of 1.6%, and DISR converges to a classification error of 3.3%. With the SVM classifier CMIM and DISR converge to the higher error rate, and mRMR-D tends towards the lower error rate.

4.3.2 Colon Dataset

This dataset has best classification performance with the 3-NN classifier, as the SVM error rates increase as more features are added (within the testing range). The CMIM algorithm

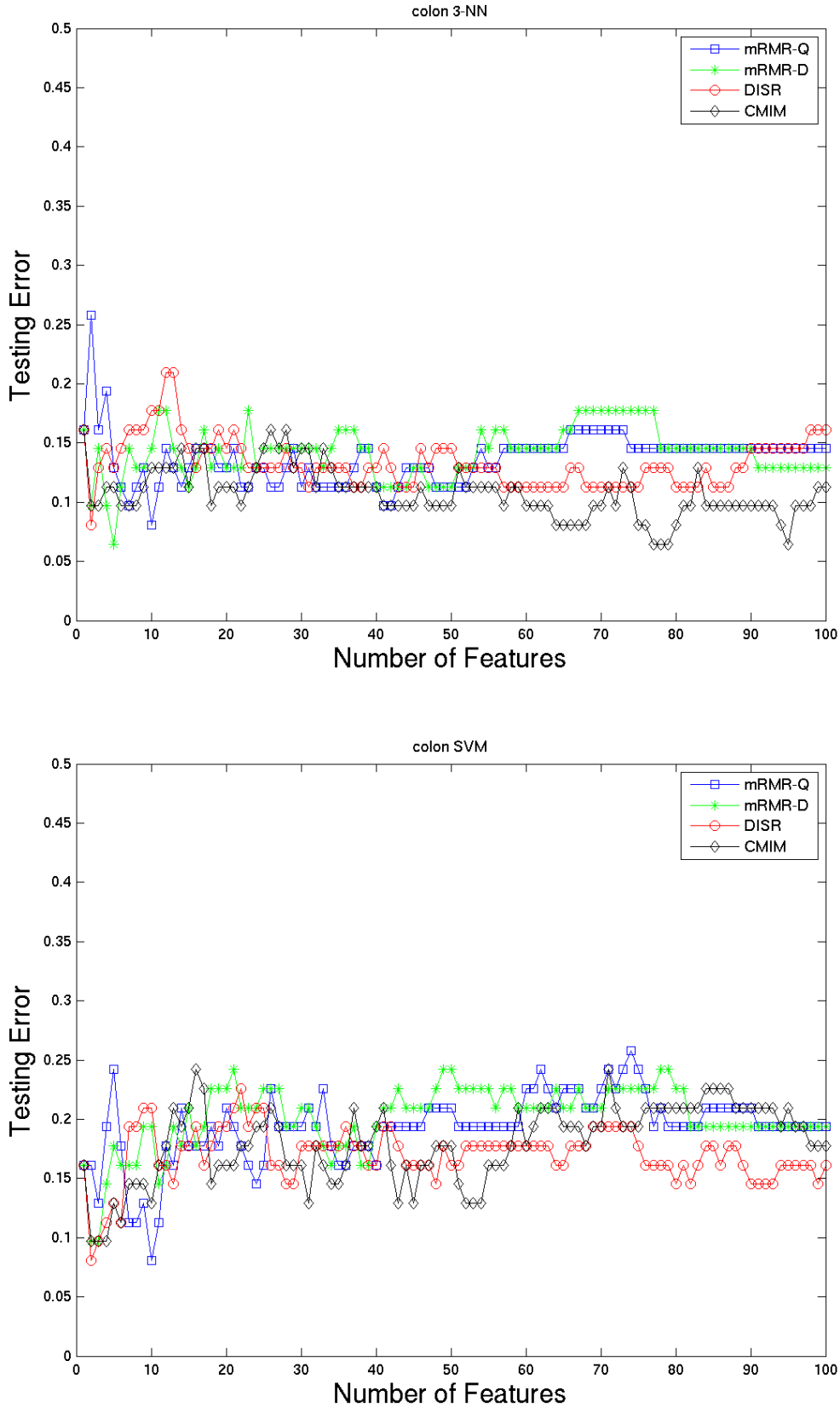

Figure 4.3: Colon Dataset, 3-NN and SVM Classifiers

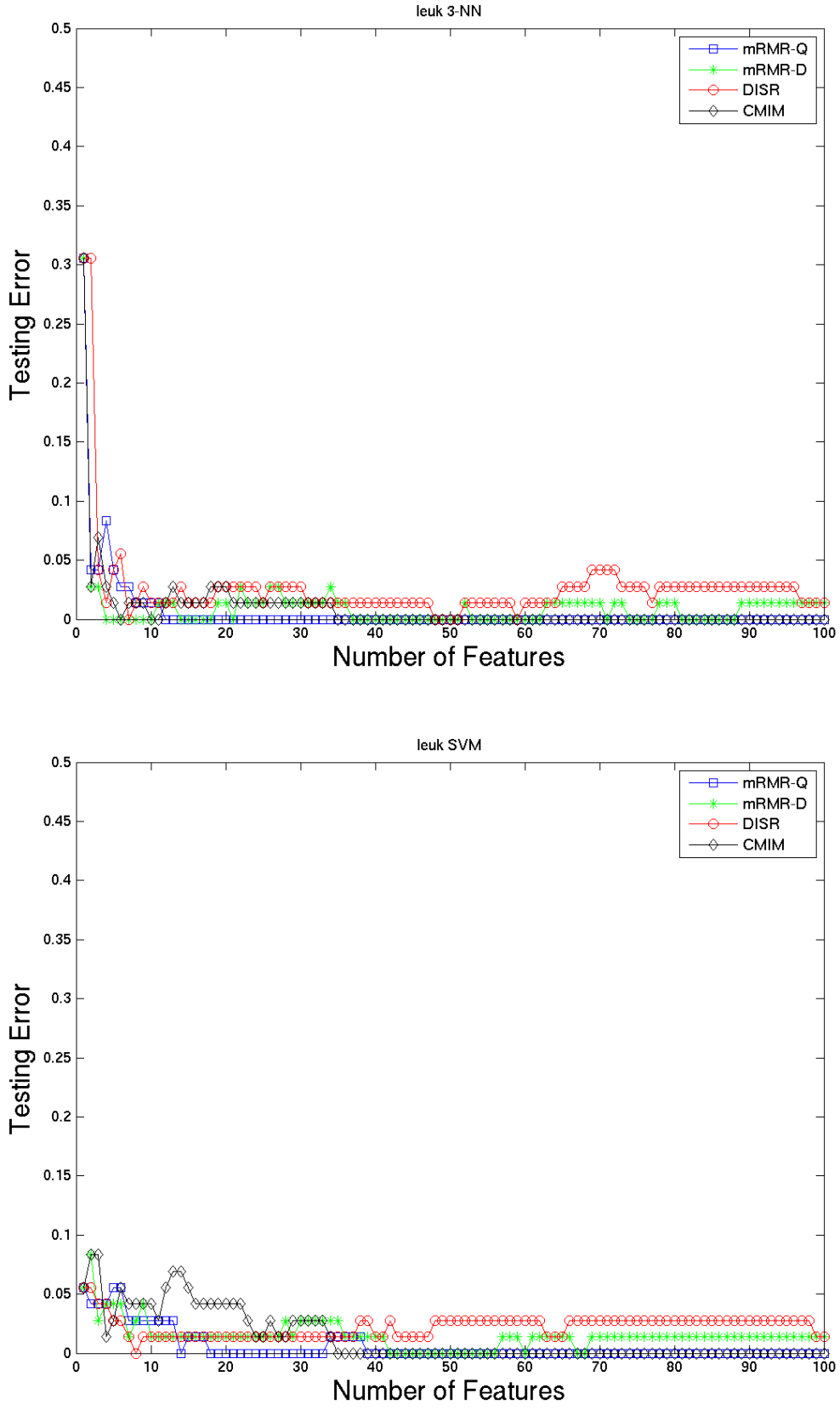

Figure 4.4: Leukaemia Dataset, 3-NN and SVM Classifiers

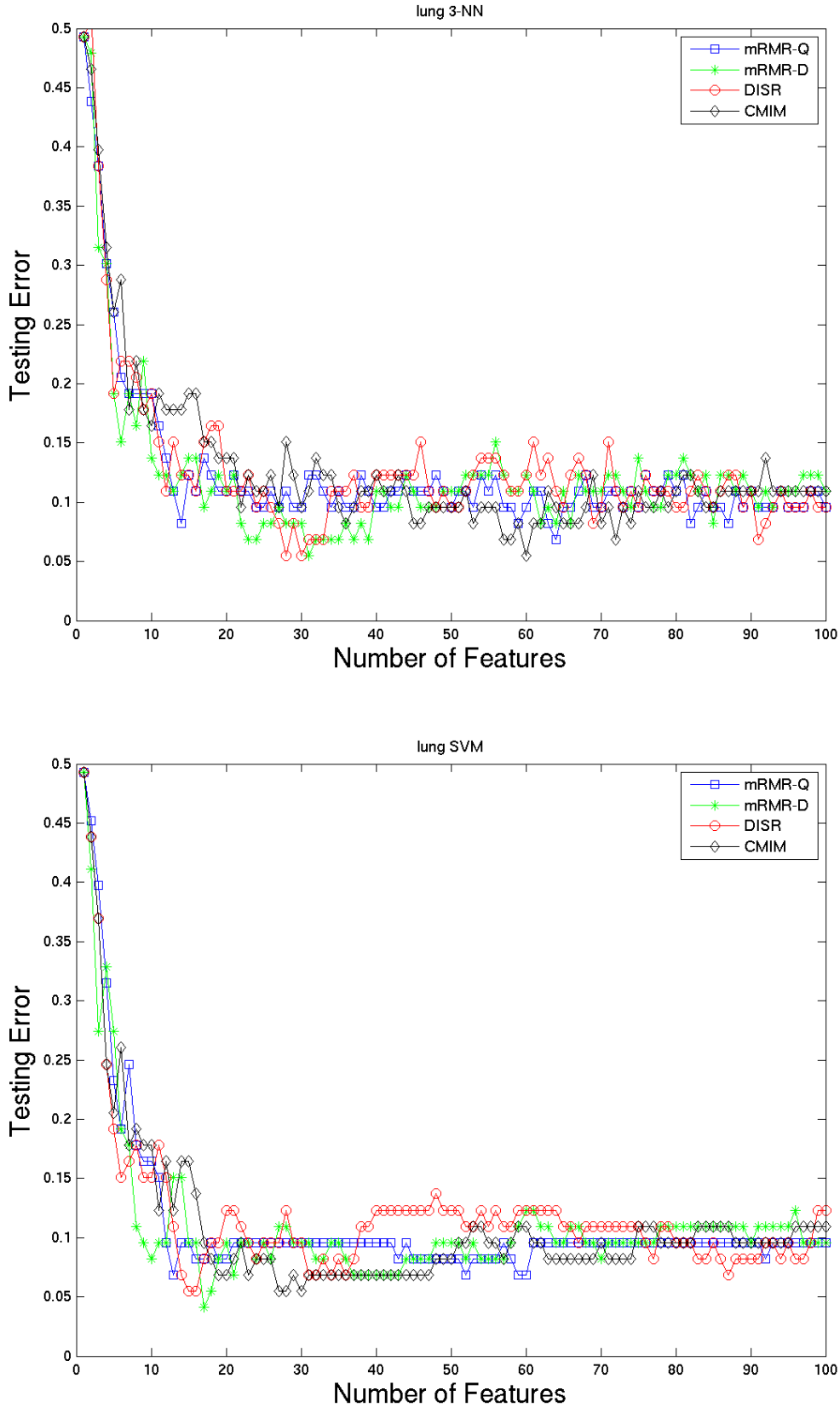

Figure 4.5: Lung Cancer Dataset, 3-NN and SVM Classifiers

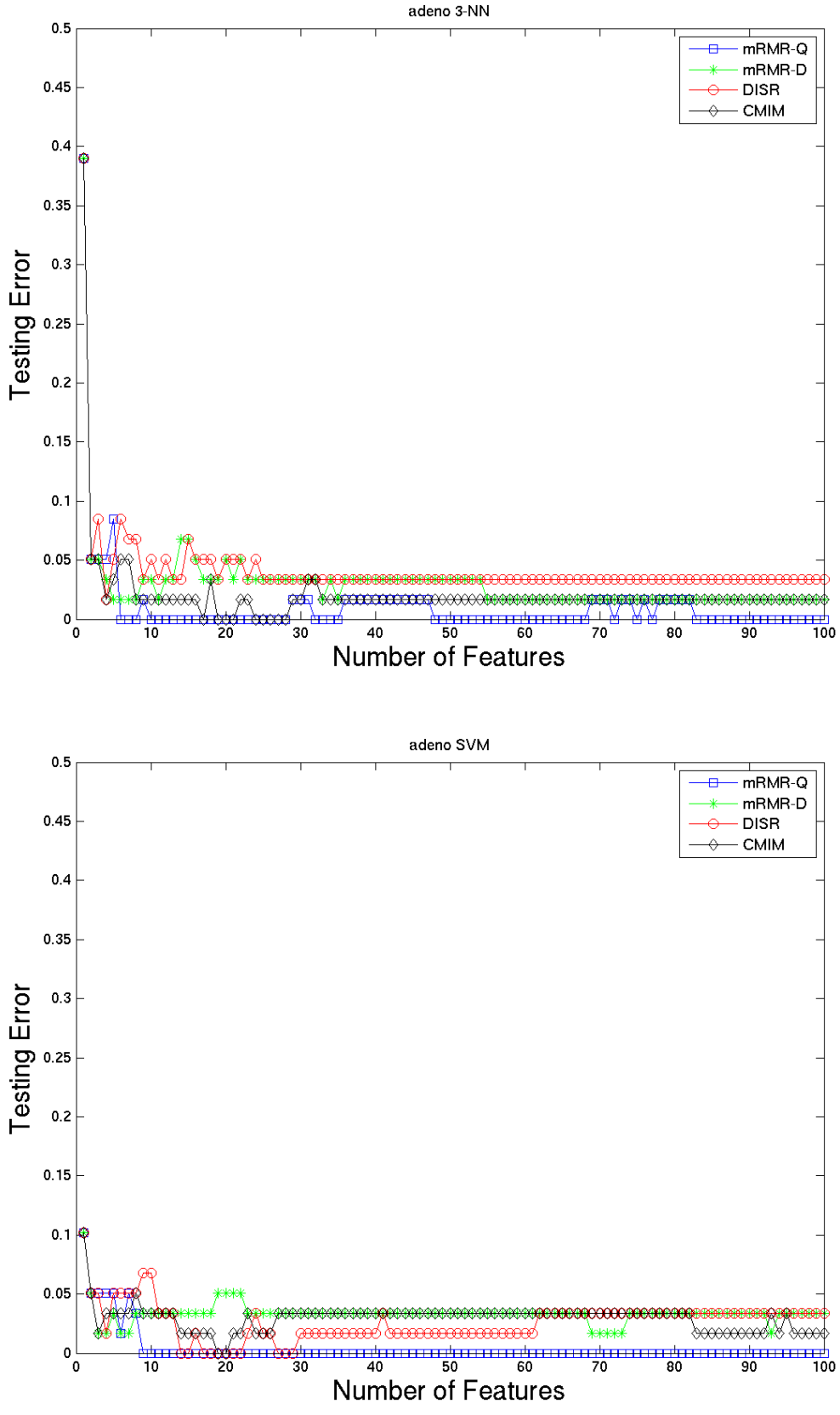

Figure 4.6: Adenocarcinoma Dataset, 3-NN and SVM Classifiers

provides the best performance with the 3-NN classifier, and in this case the result is deterministic as the colon dataset only contains two classes. The mRMR-D algorithm also reaches the best performance level (with an error rate of 6.4%), with only 5 features, but fails to converge to that level as more features are added. Using the SVM classifier the DISR algorithm converges to the lowest error rate, and the DISR algorithm and mRMR-Q reach the lowest error rate of 8.1%, using 2 and 10 features respectively.

4.3.3 Leukaemia Dataset

Using both classifiers the CMIM and mRMR-Q algorithms achieve the best performance with this dataset, as they converge to a perfect test error rate. With the 3-NN classifier the perfect classification rate is achieved with 4 features by the mRMR-D algorithm, though as it adds more features to the set its performance decreases. With the SVM classifier the perfect classification rate is achieved with 8 features by the DISR algorithm, though again as it adds more features the performance decreases.

4.3.4 Lung Cancer Dataset

Using the 3-NN classifier the performance of all the feature selection algorithms is approximately equivalent. There is large variation around the mean with no algorithm consistently outperforming the others. The lowest classification error with the 3-NN classifier is achieved by DISR, mRMR-D and CMIM of 5.5%. Using the SVM classifier mRMR-D achieves the lowest classification error, with 17 features, and 4.1% error.

4.3.5 Lymphoma Dataset

Using the 3-NN classifier, mRMR-Q performs consistently better than the other algorithms, converging to a lower error rate, and achieving the lowest classification error with 37 features and an error rate of 2.1%. Using the SVM classifier the performance of all the techniques is approximately equivalent, with the mRMR-D algorithm achieving the lowest error rate of 2.1% with the fewest features.

4.3.6 NCI 9 Dataset

Using the 3-NN classifier this dataset is extremely variable, with a wide range in classification performance by adding or removing a single feature. This could be in part due to the construction of the dataset, with the varying numbers of examples per class, and the problem

that the nearest neighbour classifier has with multiclass datasets. However the mRMR-Q algorithm consistently outperforms the other algorithms after selecting the first 18 features, achieving the lowest classification error of 10% using the first 44 features. Using the SVM classifier, the dataset is more stable with more separation of the different feature selection algorithms. Again the mRMR-Q algorithm outperforms the other algorithms, achieving the lowest classification error of 10% first when using 51 features.

4.3.7 Execution Speed

Table 4.1 lists the execution time for the MATLAB/C++ implementations of the algorithms across the various datasets, when executed on an Athlon XP 3000+, with 1GB of RAM, and selecting the 100 features used in the above test results.

Dataset	CMIM	DISR	mRMR-D	mRMR-Q
Adenocarcinoma (41672 features)	0.76	76.48	28.13	28.41
Colon (2000 features)	0.24	4.35	1.66	1.68
Leukaemia (7070 features)	0.45	16.57	6.69	6.72
Lung (325 features)	0.21	0.97	0.34	0.32
Lymphoma (4026 features)	0.54	14.63	4.39	4.43
NCI9 (9712 features)	0.37	29.56	8.12	8.14

Table 4.1: Algorithm execution time (s)

The results for mRMR-D and mRMR-Q are separated for completeness, as the algorithms only differ by an arithmetic operator, and any differences in performance are due to the task scheduling in the CPU. There is a small indication that in large datasets the costly floating point division operation slightly increases the execution time of the mRMR-Q algorithm, but the increase is negligible. The optimised CMIM implementation given in [4] provides extremely fast results, due to the minimisation step in the algorithm specification, causing it to not evaluate the mutual information when it would not affect the current feature choice. This decouples the algorithm's time complexity from the number of features, and thus there is little correlation between the number of features and the execution time of the algorithm. In contrast the DISR algorithm evaluates the Symmetric Relevance, which is a costly operation, and requires far more execution time than the mRMR's simple mutual information evaluation.

4.4 Conclusions

The mRMR set of algorithms consistently perform the best on these selected datasets, though the CMIM algorithm provides a good solution with a vastly reduced execution time, in comparison with the other algorithms.

4.4.1 Summary

This chapter has provided a brief comparison of the 3 different feature selection algorithms. The test results were obtained using the testing framework developed for the rest of the work, and the algorithm implementations used therein. This gives a baseline performance level and a comparison of how the different feature selection algorithms perform when used with the chosen classifiers and the selected datasets.

Chapter 5

Investigating the first feature

5.1 Introduction

In this chapter the selection of the first feature in common feature selection algorithms is investigated, and numerous alternatives to the standard method are developed and tested.

There have been many different feature selection algorithms [6] created due to the growth in processing power and the availability of ever more complex datasets. These algorithms all use variations on information theoretic values to construct a measure for the usefulness of the feature, and so to determine an optimum feature set. All of these algorithms also start from a basic assumption, that the feature with the highest mutual information is the best feature, and should be chosen to start the algorithm. It can be shown that the feature with the highest mutual information is not automatically a member of the ideal feature set. This was shown in [22], but can also be seen in a simple XOR problem by adding a noise feature (see Table 5.1) as shown in [6]. Strict selection of the highest mutual information feature selects the Noise feature and another feature, whereas the best feature pair is XY and does not contain the Noise feature.

X	Y	Noise	Class
1	1	0	0
1	0	1	1
0	1	1	1
0	0	1	0

I(X;Class)	0
I(Y;Class)	0
I(Noise;Class)	0.3113
I(X & Y;Class)	1
I(X & Noise;Class)	0.5
I(Y & Noise;Class)	0.5

Table 5.1: XOR Problem

The reason that the highest mutual information feature is selected in the first stage of the algorithm is due to the greedy assumption in the forward search. Each of the criteria for

selecting features in CMIM, DISR and mRMR reduce to the selection of the highest mutual information when the selected feature set is empty, but as has been demonstrated this might not be the best solution to the problem of selecting the first feature.

5.2 Creating New Criteria

To investigate the effect modifying the first feature has on the performance of a feature selection algorithm requires a consideration of what aspects are useful in a given feature that make it a good choice for forming the initial condition of the algorithm.

There are two possibilities investigated in this research. The first is to identify a feature that contains the most independent information from the dataset (i.e. the feature with the most information that is not contained in the other features). The second possibility is to identify a feature that combines with other features to give the greatest possible amount of information. These two strategies can be identified as the two main ideas behind the feature selection algorithms considered in this research. CMIM and mRMR aim to find features that are strongly independent of each other, to capture all the information in the dataset, in as few features as possible. DISR in contrast tries to find the features that combine best with the current set of features it has selected, to provide a set of features that work well together to capture the information in the dataset. These two different approaches will be referred to as the *independence* and *combination* approaches.

By analysing the different approaches of the algorithms several different methods for determining the initial feature were created and tested.

5.2.1 CMIM Criterion

The CMIM selection criterion is created by applying the CMIM feature selection criterion, and taking the whole feature space as the selected set. It selects the feature that is the most independent of the whole feature set, using individual conditional mutual information calculations. It uses the pessimistic assumption in the original CMIM feature selection criterion, by taking the minimum value for the conditional mutual information over all other features as the score for that particular feature, then selecting the feature that produces the maximum value. This criterion takes the independence approach to selecting the first feature.

$$X_{\text{CMIMS}} = \arg\max_{X}(\min_{Z} I(X;Y|Z)) \tag{5.1}$$

A modification was proposed that took the sum over the conditional mutual informations, removing the pessimistic assumption as all features are being used in the calculation, so there is no need for approximating the feature independence. This technique removes the benefit of the minimisation, which is to prioritise features which are strongly independent of the whole dataset. The summation changes the criterion so it prioritises the feature with the highest information when all other features are known. This leads to a bias as one high score can distort the selection process, favouring a feature that combines well with a subset of features, over a feature which has strong general performance. The change to a summation makes this criterion take the combination approach to selecting the first feature.

$$X_{\text{CMIMS Sum}} = \arg\max_X (\sum_Z I(X;Y|Z)) \tag{5.2}$$

Both of these criteria have complexity of $O(N^2)$ with N being the number of features in the dataset.

5.2.2 DISR Criterion

The DISR selection criterion is created by applying the DISR feature selection criterion, and finding the two features that have the maximal information jointly on the class. This performs differently to the original criterion as there is no guarantee that the feature with the highest mutual information will combine well with any other feature in the dataset, whereas the new criterion finds the two features that jointly contain the most information about the class. The DISR criterion thus takes the combination approach to selecting the first feature.

$$X_{\text{DISR}} = \arg\max_{X,Z} (\frac{I(XZ,Y)}{H(XZ,Y)}) \tag{5.3}$$

The DISR selection criterion is symmetrical, and thus has a lower complexity than the other criteria developed, because the algorithm need only search for $Z > X$ as all previous values have been calculated at an earlier stage.

5.2.3 mRMR Criterion

There were two versions of the mRMR selection criterion created, to parallel the two different versions of the mRMR feature selection algorithm.

The mRMRD selection criterion is created by applying the mRMRD feature selection criterion, using the whole feature space as the selected set. It is similar to the CMIM selection criterion in that it selects the feature that is most independent of the feature set,

but by a different measure. It penalises the candidate features by their redundancy and selects the least redundant feature, over all the features. The pessimistic assumption was used as this aids in selecting a feature which is not skewed by a low redundancy score with only a small number of features. This takes the independence approach to selecting the first feature.

$$X_{\text{mRMR}} = \arg\max_{X\in\Omega}(\min_{Z}(I(X;Y) - I(X;Z))) \tag{5.4}$$

The modification replacing the minimisation with a summation was also tested. This selects for the feature which gives the smallest redundancy score. A similar analysis can be applied to the mRMR sum criterion as applied to the CMIM sum criterion, and as there this uses the combination approach to selecting the first feature.

$$X_{\text{mRMR}} = \arg\max_{X\in\Omega}(\sum_{Z}(I(X;Y) - I(X;Z))) \tag{5.5}$$

The mRMRQ selection criterion is the same as the mRMRD criterion, except using a quotient instead of a subtraction to penalise redundancy. As the quotient is inherently more unstable than the subtraction this gives different results to the mRMRD version, but with similar properties.

$$X_{\text{mRMR}} = \arg\max_{X\in\Omega}(\min_{Z}(\frac{I(X;Y)}{I(X;Z)})) \tag{5.6}$$

$$X_{\text{mRMR}} = \arg\max_{X\in\Omega}(\sum_{Z}(\frac{I(X;Y)}{I(X;Z)})) \tag{5.7}$$

5.2.4 Computational complexity

Each of these proposed new first feature selectors gives an increase in the computational complexity of the first stage of the algorithms. In general these algorithms have complexity order n^2 where n is the number of features, compared with the standard highest mutual information method which has complexity n. This is due to the double loop in each criteria where the whole feature set is compared with the currently selected feature, compared to the single loop for ranking the features by mutual information. In addition, the implementations of the mRMR and CMIM algorithms also require a list of the mutual informations between the features and the class to save recalculating this value at each stage, so the complexity n loop must stay in the algorithm.

5.2.5 Summary

The different criteria for selecting the first feature in a feature selection algorithm have been derived through applications of the criteria used for the selection of the remaining features. Each one selects a feature that has the properties that the algorithm is selecting for, with the aim of improving the performance of the algorithm as a whole. As the feature selection algorithms gain more complex information theoretic criteria to select features, the first step remains an area where there is little attention paid. The first step in all of these algorithms has not progressed in a similar manner, as it is still taken from the simple ranking procedure that was first developed to select features.

5.3 Results

Each of the first feature selectors was paired with each of the feature selection algorithms, and tested on the variety of datasets listed in chapter 3. The results are presented from the multiclass datasets, as these provide the most variation.

5.3.1 Lung cancer dataset

This dataset has 325 features discretised into 3 states. There are 7 classes and 73 examples. The results are split by classifier and feature selector. Figure 5.1 contains the results for CMIM and DISR using a 3-NN classifier, figure 5.2 contains the results for mRMR using the same classifier. Figures 5.3 & 5.4 contain the results when using a linear SVM.

5.3.2 NCI 9 Dataset

This dataset has 9712 features, discretised into 3 states. There are 9 classes and 60 examples. Figure 5.5 contains the results for CMIM and DISR using a 3-NN classifier, figure 5.6 contains the results for mRMR using the same classifier. Figures 5.7 & 5.8 contain the results when using a linear SVM.

5.3.3 Lymphoma Dataset

This dataset has 4026 features, discretised into 3 states. There are 9 classes, and 96 examples. Figure 5.9 contains the results for CMIM and DISR using a 3-NN classifier, figure 5.10 contains the results for mRMR using the same classifier. Figures 5.11 & 5.12 contain the results when using a linear SVM.

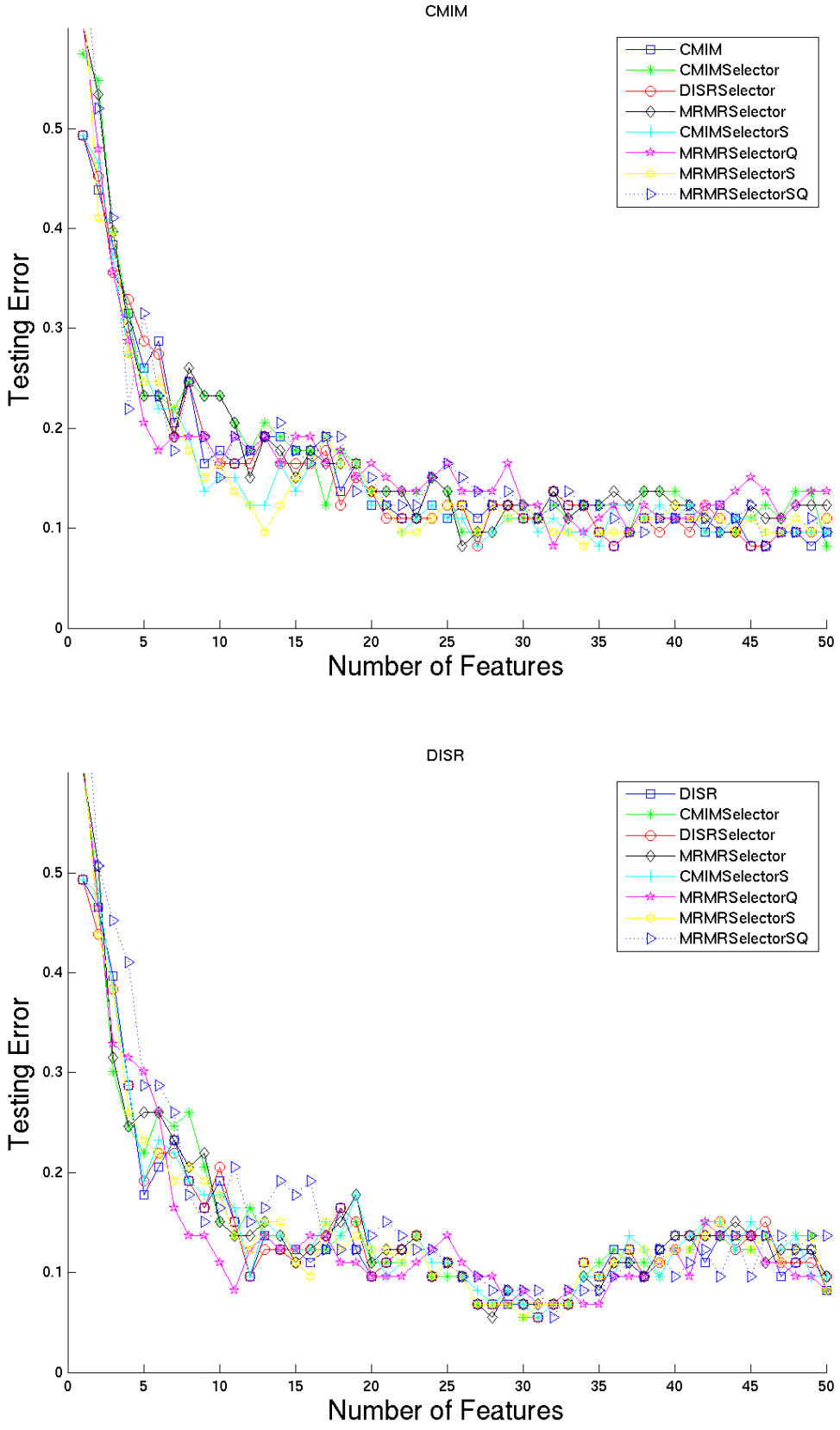

Figure 5.1: Lung Cancer Dataset, 3-NN Classifier, CMIM and DISR

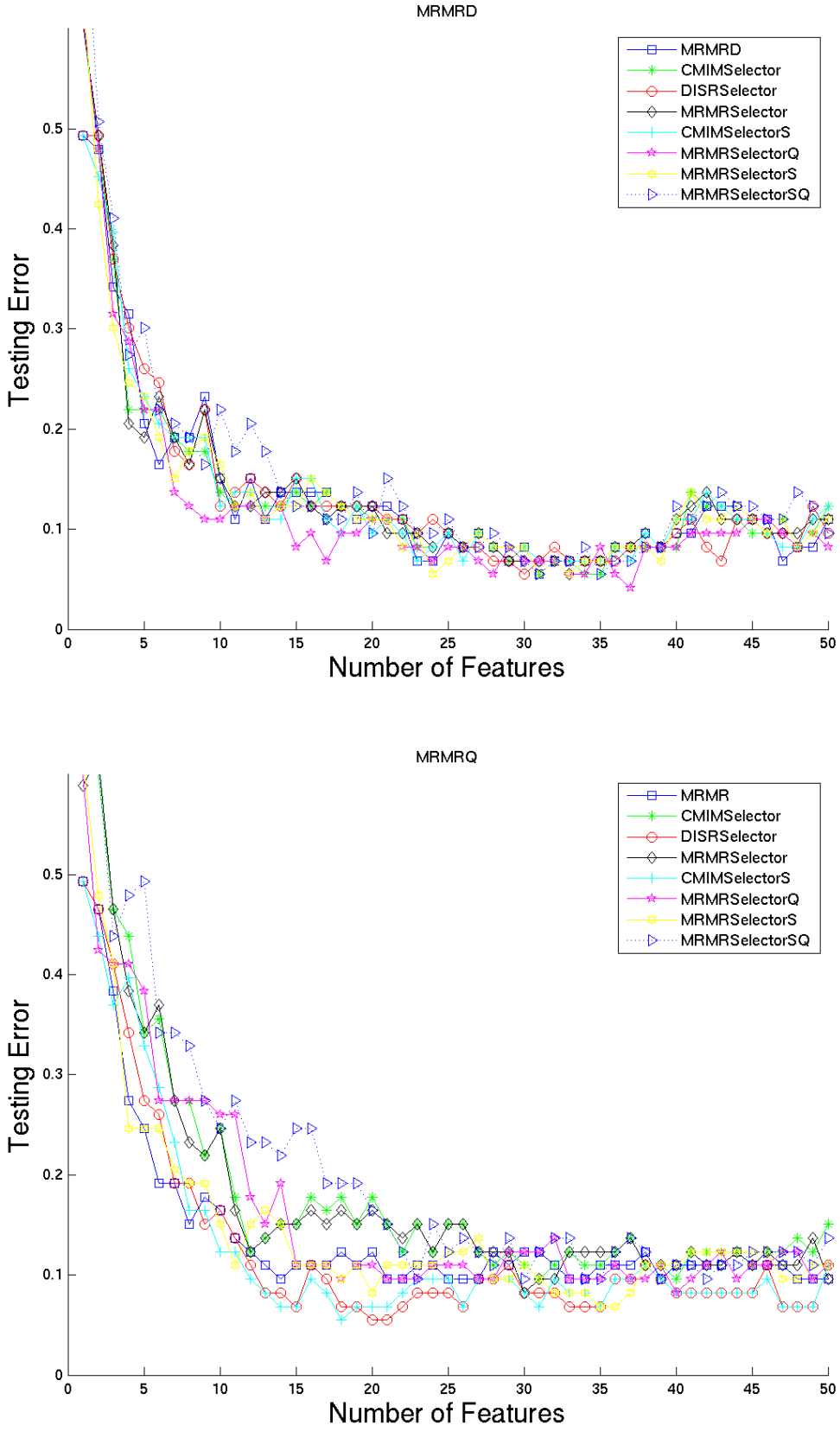

Figure 5.2: Lung Cancer Dataset, 3-NN Classifier, mRMR-D and mRMR-Q

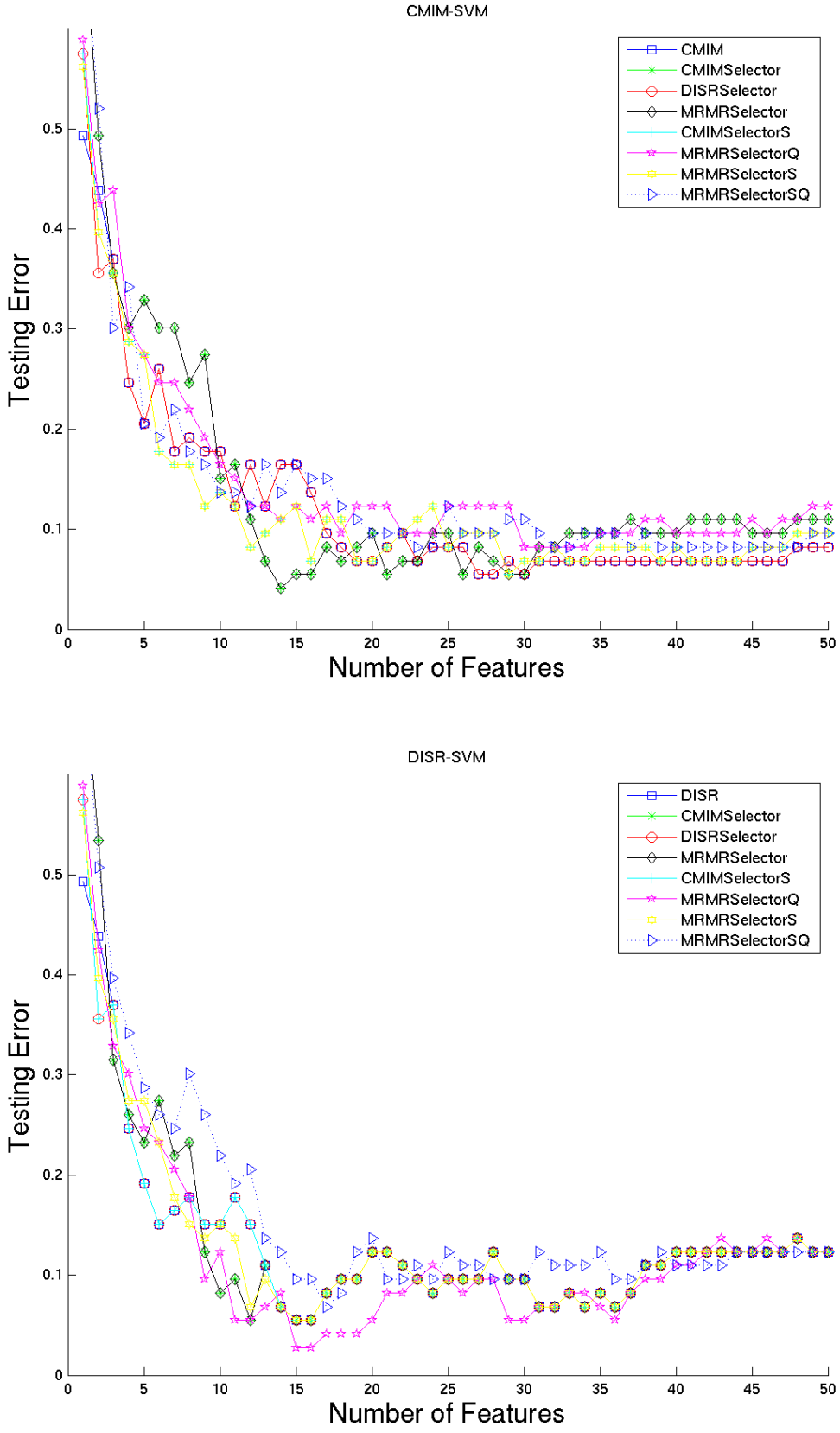

Figure 5.3: Lung Cancer Dataset, Linear SVM Classifier, CMIM and DISR

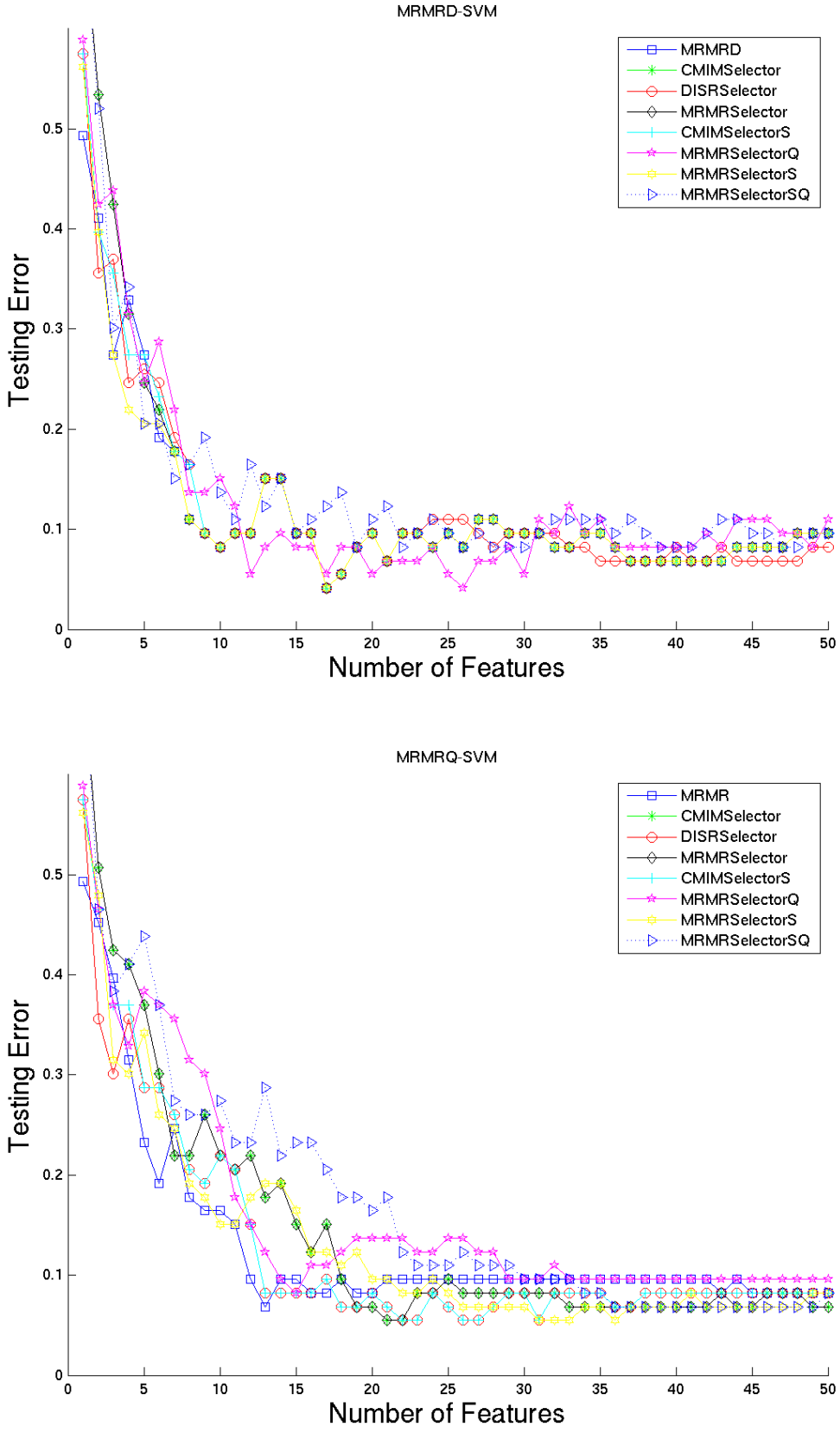

Figure 5.4: Lung Cancer Dataset, Linear SVM Classifier, mRMR-D and mRMR-Q

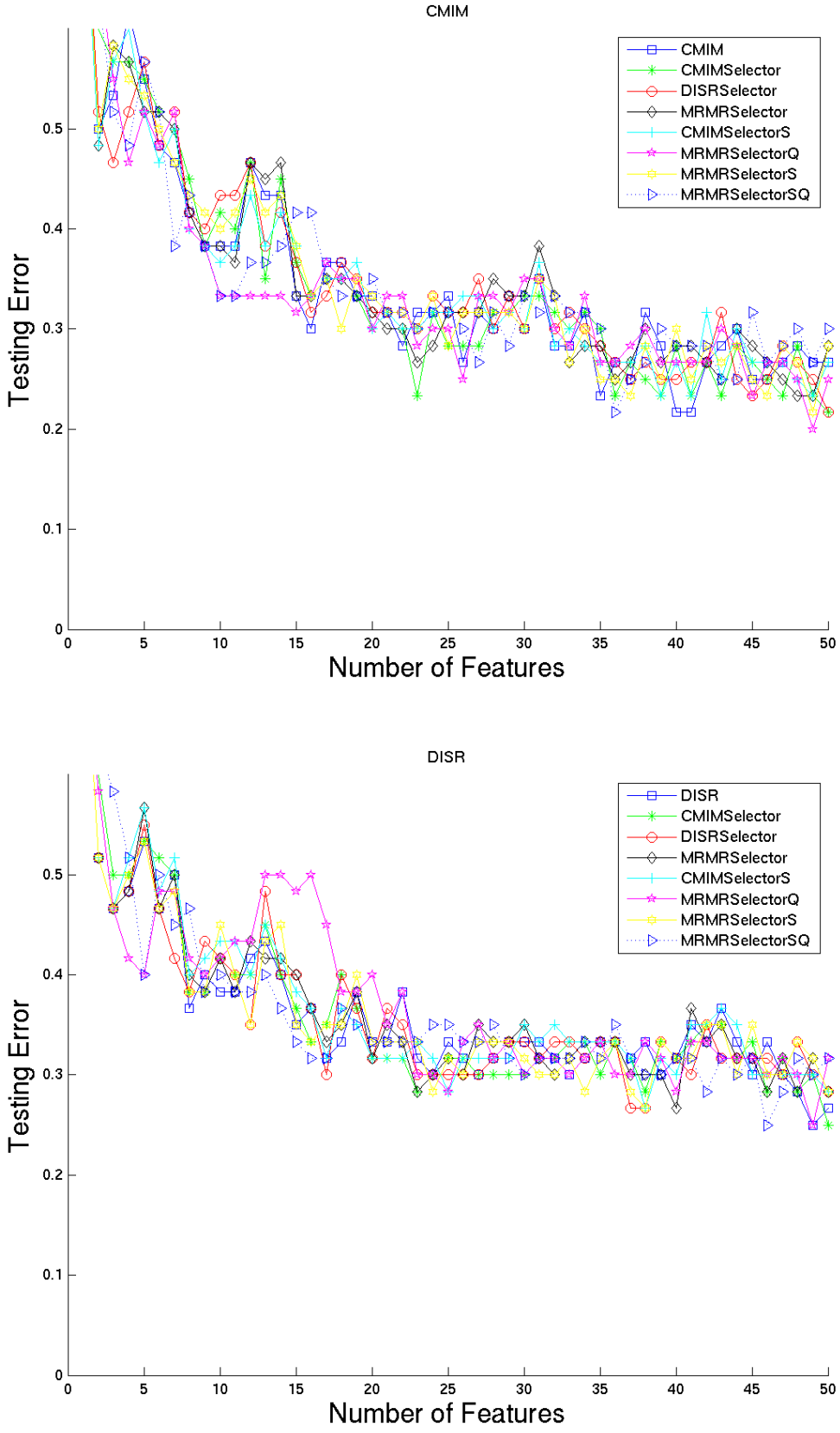

Figure 5.5: NCI9 Dataset, 3-NN Classifier, CMIM and DISR

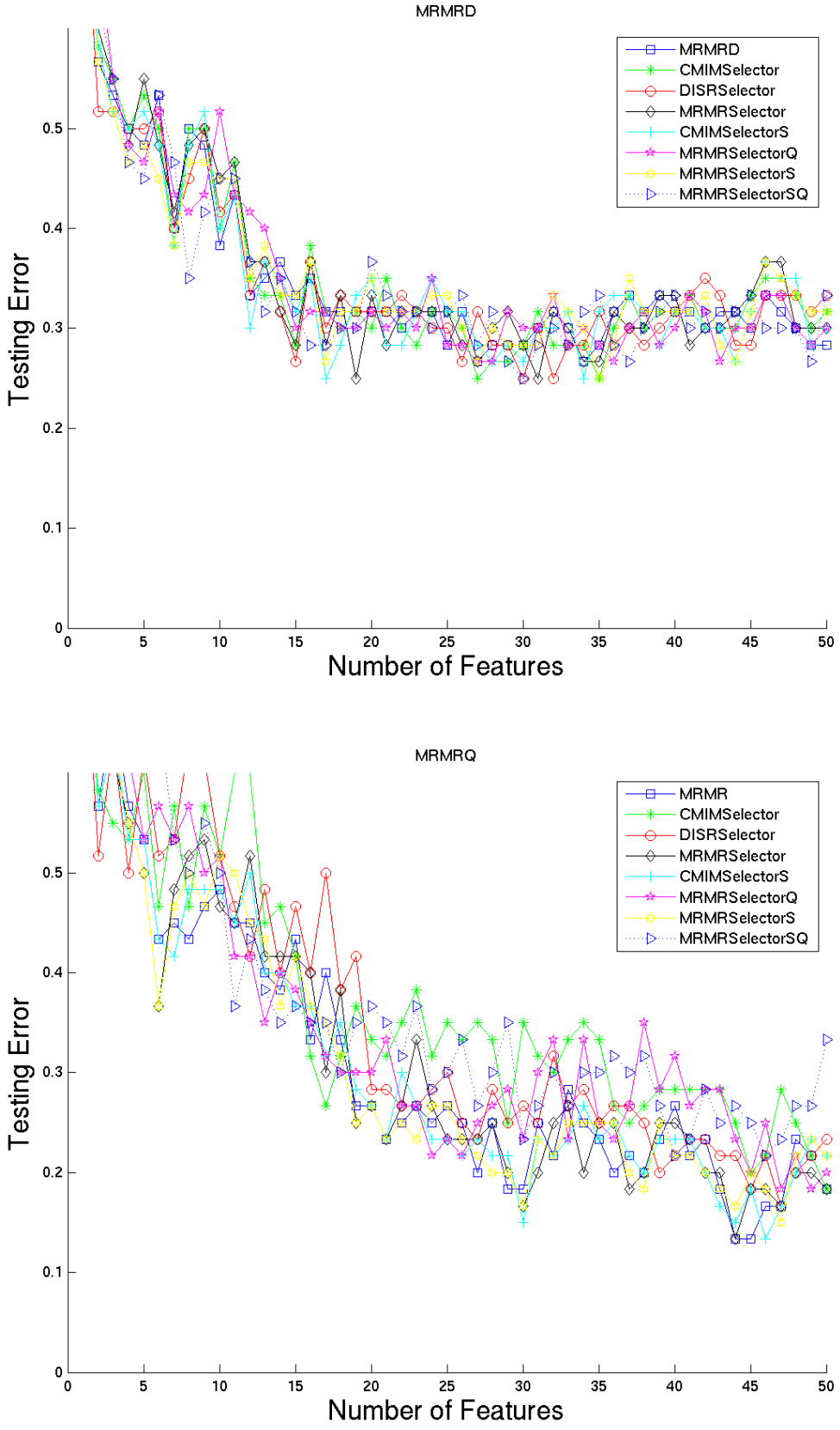

Figure 5.6: NCI9 Dataset, 3-NN Classifier, mRMR-D and mRMR-Q

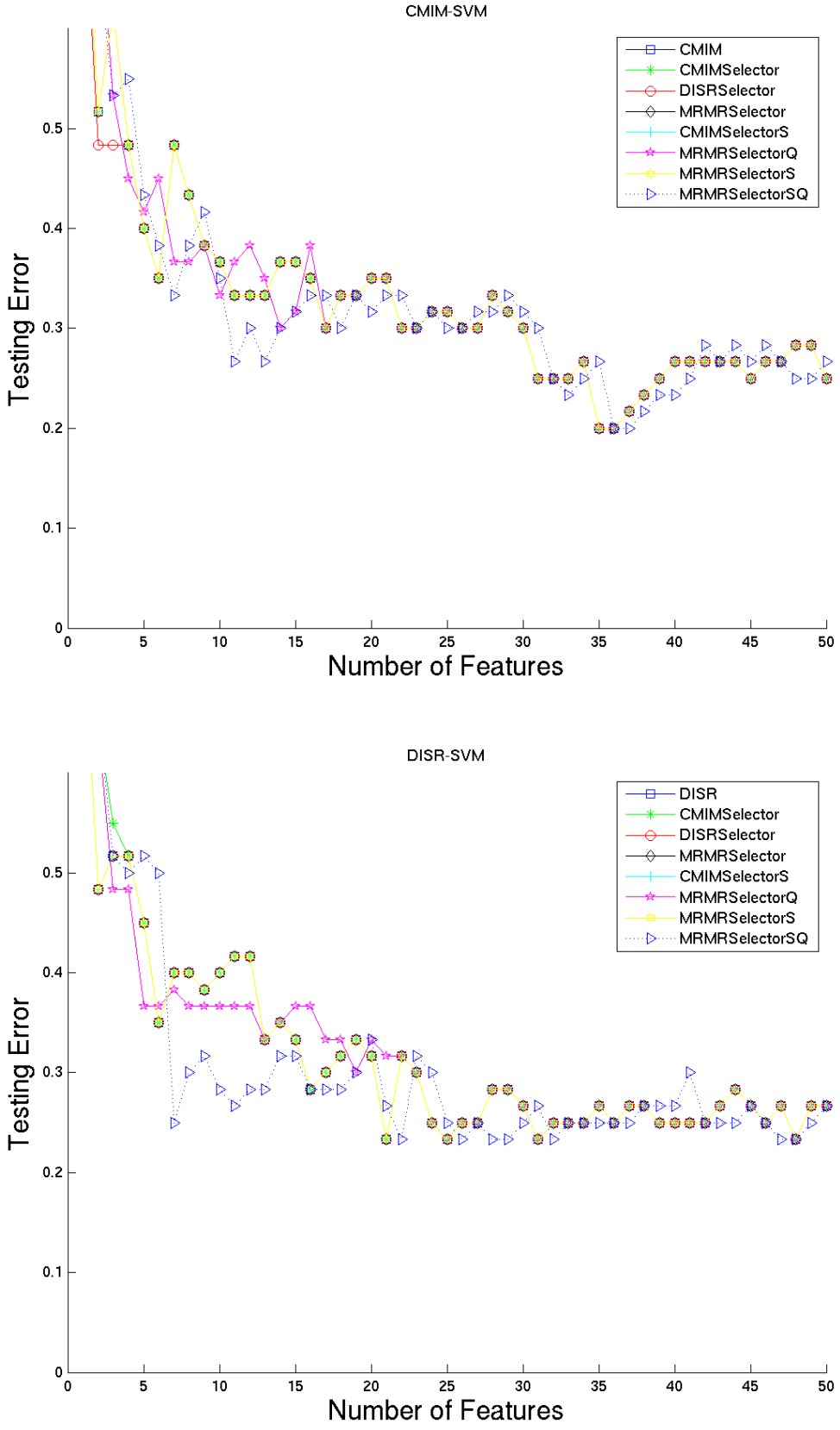

Figure 5.7: NCI9 Dataset, Linear SVM Classifier, CMIM and DISR

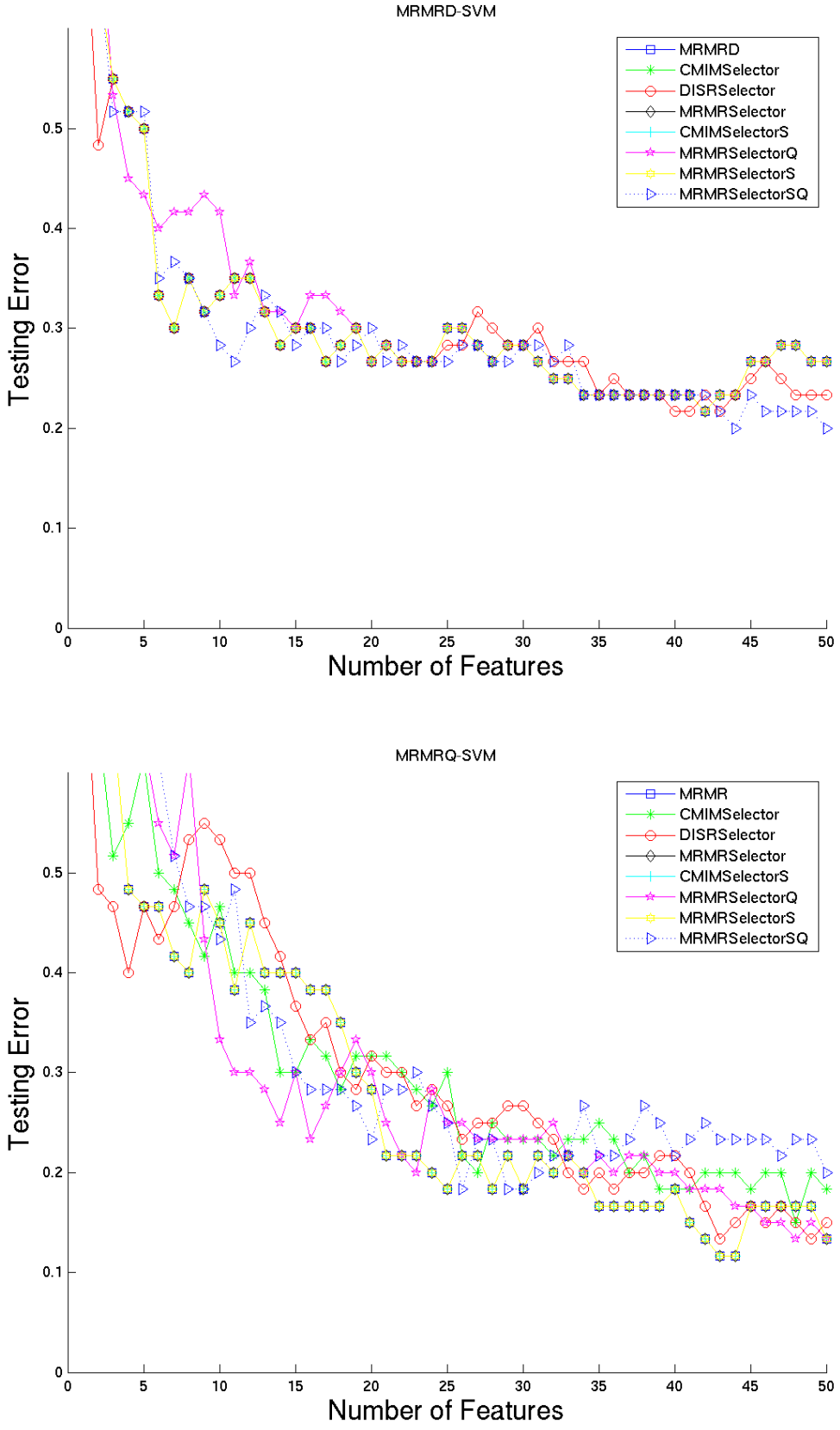

Figure 5.8: NCI9 Dataset, Linear SVM Classifier, mRMR-D and mRMR-Q

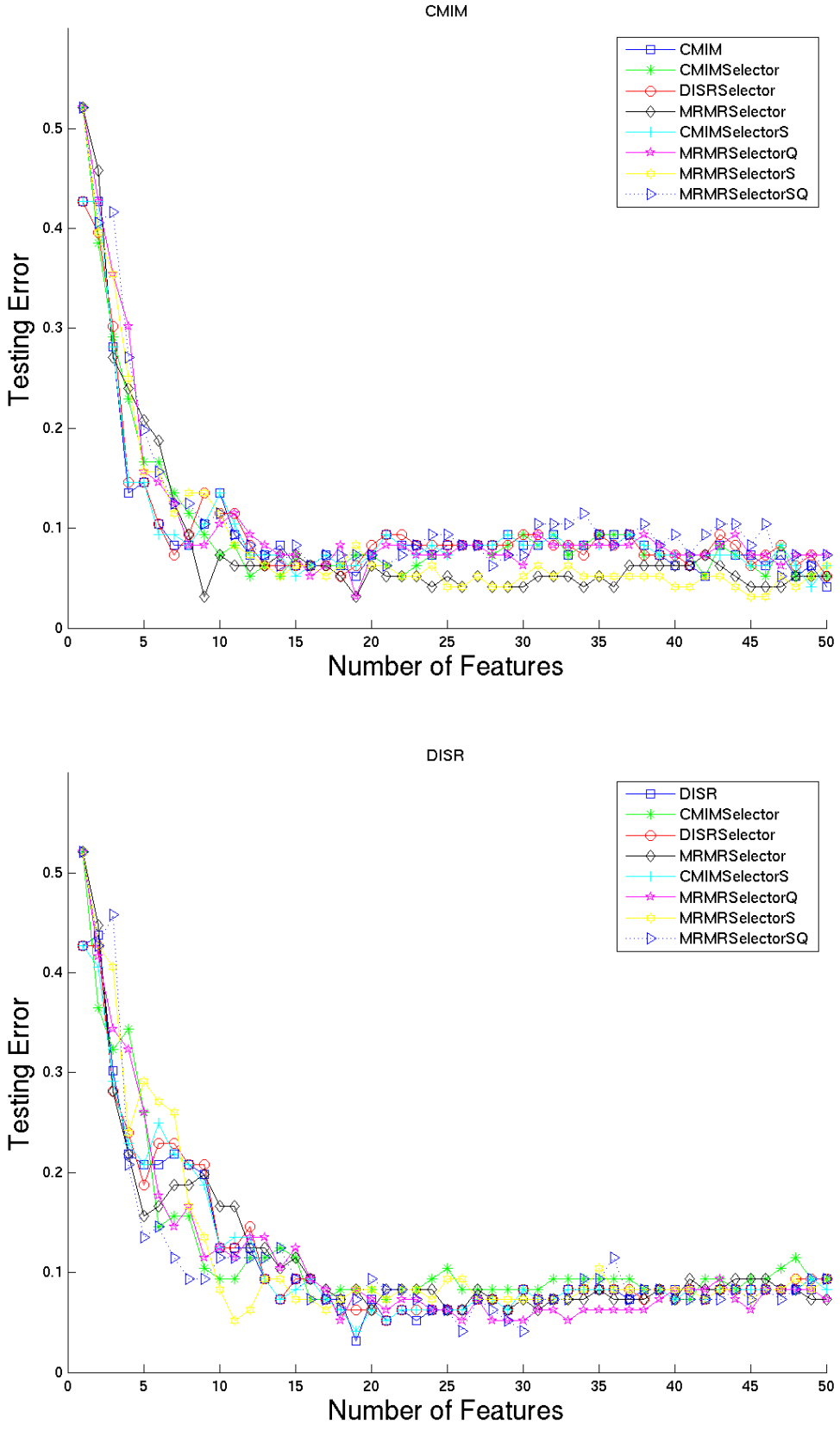

Figure 5.9: Lymphoma Dataset, 3-NN Classifier, CMIM and DISR

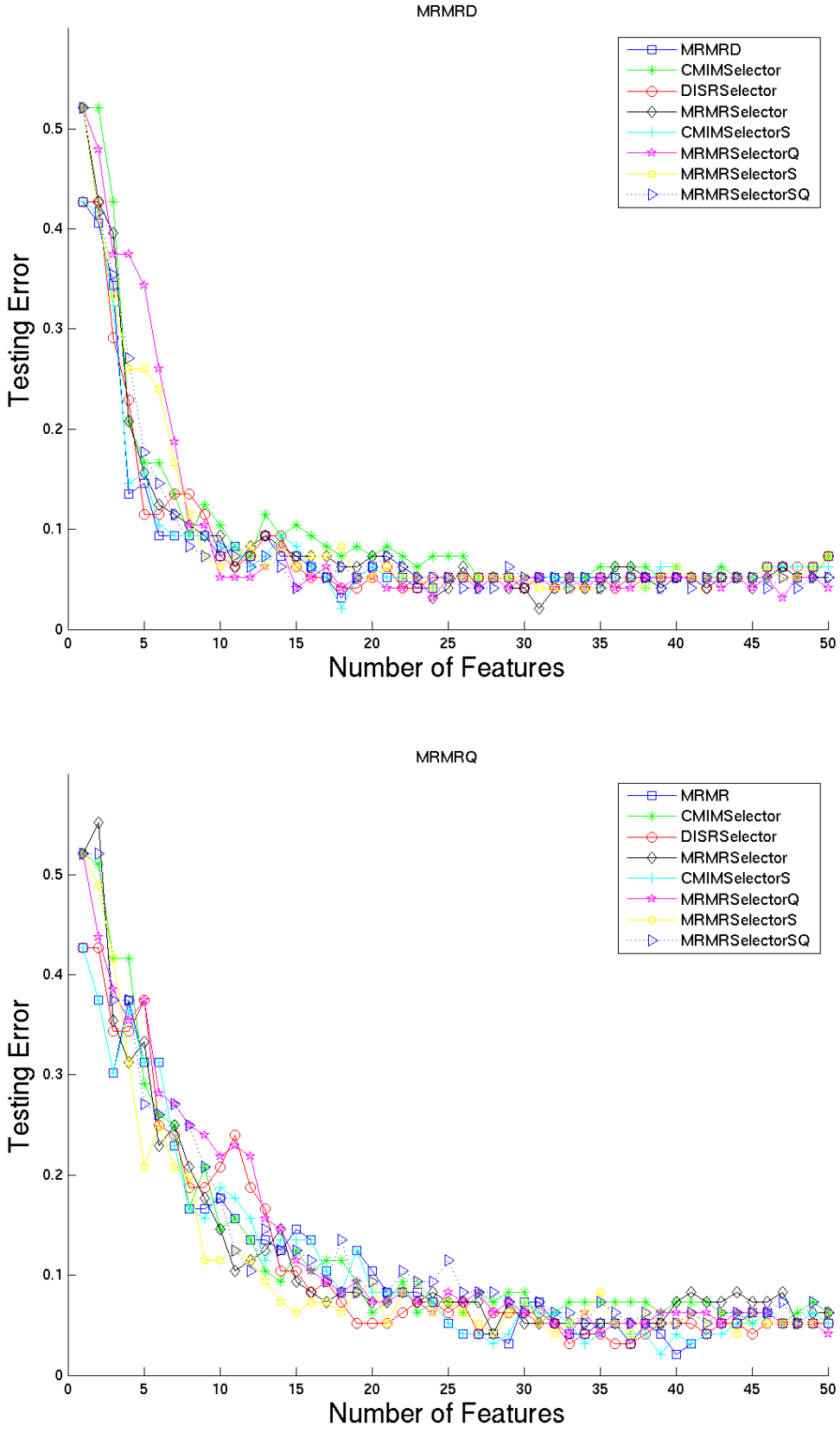

Figure 5.10: Lymphoma Dataset, 3-NN Classifier, mRMR-D and mRMR-Q

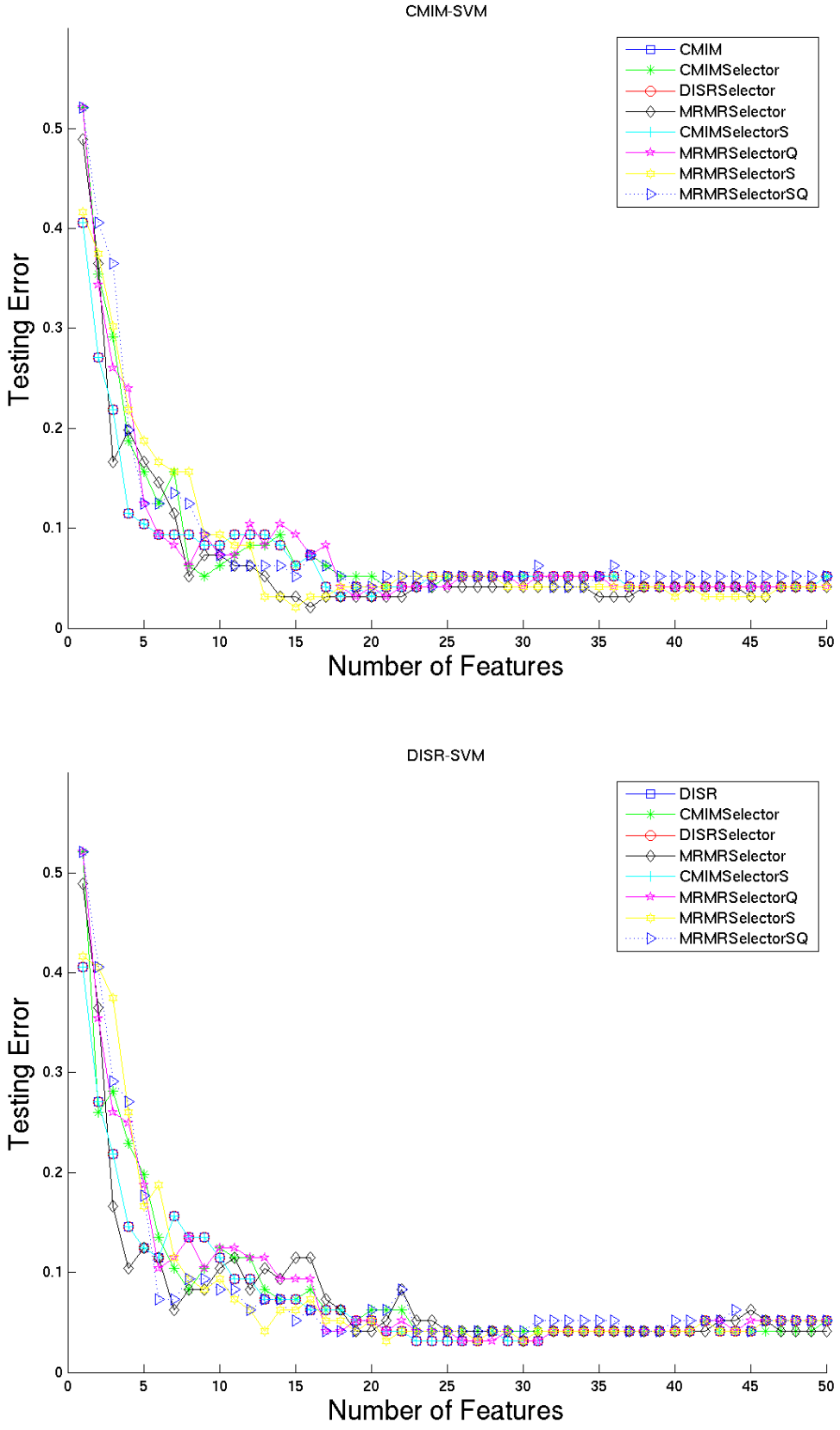

Figure 5.11: Lymphoma Dataset, Linear SVM Classifier, CMIM and DISR

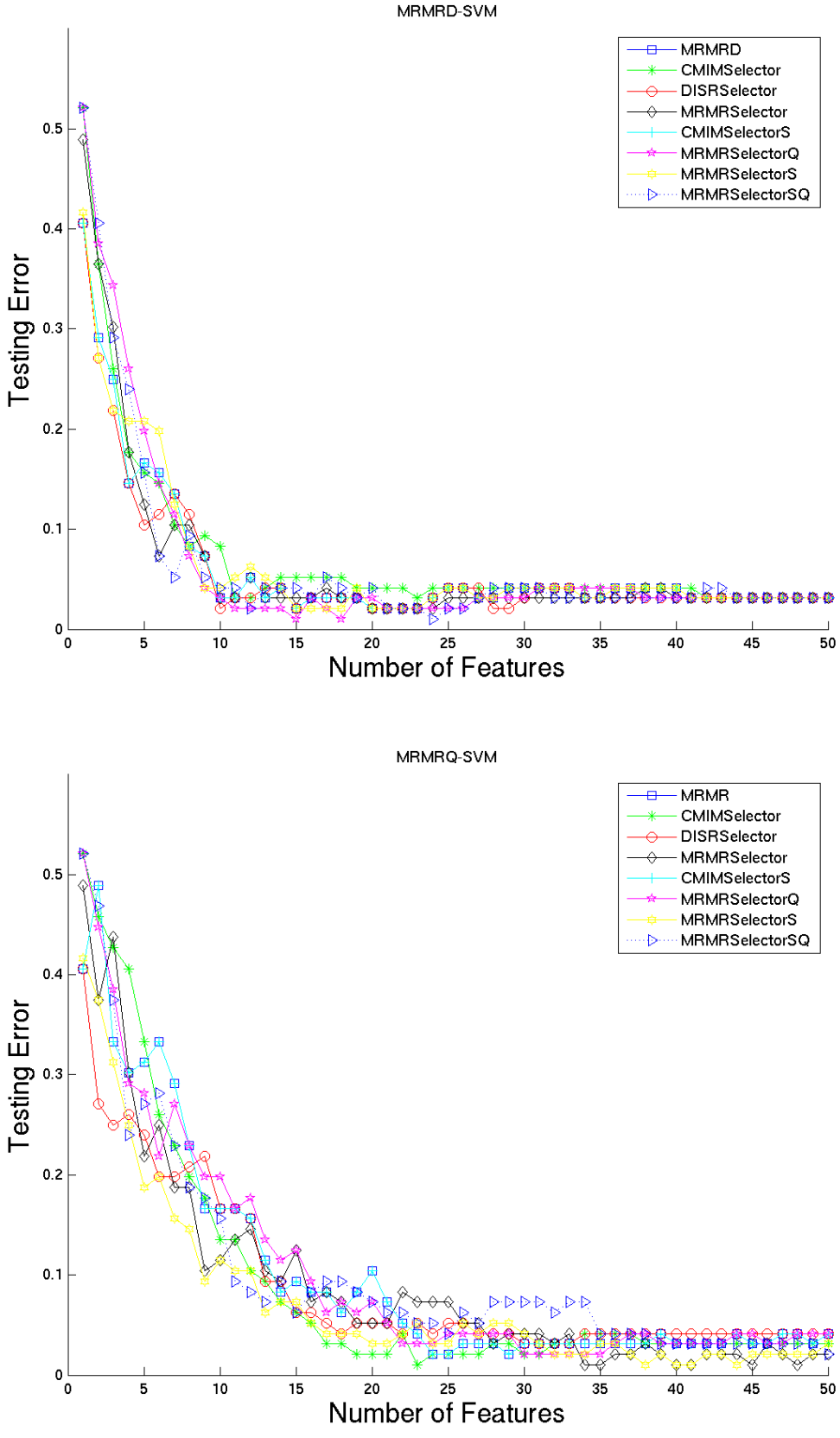

Figure 5.12: Lymphoma Dataset, Linear SVM Classifier, mRMR-D and mRMR-Q

5.4 Analysis

As can be seen from the results above, the mRMR-Q feature selection algorithm is highly sensitive to the choice of the first feature, and produces a wide variation in classification performance when the first feature is changed. Additionally the feature selection algorithms tend towards a similar classification performance, and select a similar set of features.

5.4.1 Lung Cancer Dataset

The best performing new first feature selector over this dataset is the DISR selector, as it has equivalent or better performance to the standard method of selecting the feature with the highest mutual information. The lowest classification error when using the SVM is found by combining the mRMR-Q selector with the DISR feature selection algorithm, which results in a 2% classification error, when using the first 15 features selected by the algorithm. The lowest classification error when using the 3-NN classifier is found by combining the mRMR-Q selector with the mRMR-D feature selection algorithm, which results in a 4% classification error, when using the first 37 features.

5.4.2 NCI 9 Dataset

This dataset is a good example of the differences between classifiers, with the various different feature sets giving the same SVM performance, whilst the 3-NN classifier has a large variation in classification performance between the different feature sets. Part of this variability could however be due to the nature of the dataset, as it has 9 different classes, with a wide range in the number of examples of each class. This confuses the 3-NN classifier as it will often reach a situation where it has 3 nearest neighbours, each with a different class, at which point the classification is determined by a random number generation.

With the SVM only the mRMR-Q sum selector generates a sufficiently different feature set to give a different classification performance, but only with the mRMR-D feature selector does it provide a feature set with a significantly better feature set. The lowest classification error with the SVM is found by the CMIM sum selector, the mRMR-D selector, the mRMR-D sum selector and the standard method simultaneously using the mRMR-Q feature selector, where it achieves a classification error of 12%. The results for the 3-NN classifier are too variable to draw any conclusions about the performance of the different first feature selectors.

5.4.3 Lymphoma Dataset

The best performing first feature selectors over this dataset are the mRMR-D, and the mRMR-D sum selectors, as they perform consistently better than the standard method when using feature selection methods other than DISR. When using the DISR feature selection algorithm their performance is approximately equivalent to the standard method. The lowest classification error with the SVM classifier is found by the mRMR-D, and mRMR-D sum selectors with the mRMR-Q feature selection algorithm, and with the mRMR-Q and mRMR-Q sum selectors when using the mRMR-D feature selection algorithm, with a classification error of 1%. It must be noted that the CMIM selector provided this lowest level of classification error with the mRMR-Q feature selection algorithm when using 23 features, before converging with the standard selector as more features were added to the selected set.

5.4.4 Summary

From these results it can be seen that changing the initial choice of feature may provide a worse basis for classification when used alone, if a feature other than the one with the highest mutual information is chosen, however it generally improves the resulting feature set. This is expected because of the way the different first features are selected. The different methods are selecting features which combine well with the rest of the dataset, but do not necessarily provide good classification performance when taken on their own (as the mutual information is a measure of the shared data between the feature and the class, the highest mutual information is the feature with the best classification performance when only using one feature). However when the selected feature sets are tested it can be seen that modifying the first feature so it is the highest ranked by another criteria than the highest mutual information improves or maintains classification performance.

5.5 Conclusions

This investigation into the selection of the first feature in common feature selection algorithms has created several new ways of selecting the first feature. The empirical testing of these different selectors has shown that several of them improve classification performance, though the amount of improvement is dataset dependent, and in general only becomes apparent after 10 or more features have been selected. The best performing selectors were those based upon the mRMR feature selection algorithm, though this may in part be due to the datasets involved, as the mRMR algorithm generated the best performing feature sets

suggesting that the datasets have a low amount of intra feature interactions.

In general the results indicate that *selecting an independent feature is more important than selecting a feature that combines well with the rest of the dataset*, though the sum selectors also show an improvement over the standard method.

5.5.1 Summary of the work

This chapter has detailed an investigation into the selection of the first feature in common feature selection algorithms. It has:

- Stated why the standard method for selecting the first feature is not optimal
- Constructed a set of different methods for selecting the first feature
- Empirically tested these methods against the standard method, using a variety of datasets, classifiers and feature selection algorithms
- Analysed the test results, and concluded that selecting an independent feature on average improves the classification performance

Chapter 6

Investigating the Rényi measures of information

6.1 Introduction

The Rényi extension to entropy, and the associated extension to the Kullback-Liebler divergence provides a framework for a parametrised measure of information. However various properties of the Shannon entropy do not hold under Rényi's extension, including the creation of conditional entropies and the relationship between the mutual information and entropy.

To create a measure of information that is valid over the whole space of the Rényi entropy and generalised divergences means the measure must be parametrised by α, it must measure the amount of shared information between two variables, with a high value when the variables are strongly correlated, and a low value when they are strongly independent, and it must be symmetric with respect to the placement of X & Y. Also the measure should only use the Rényi entropy and generalised divergences, instead of the Shannon entropy and the Kullback-Liebler divergence.

This leads to three competing formulations of the mutual information in the Rényi space, which will be compared in detail in this chapter.

6.2 Different Formulations of Rényi Information

The Shannon mutual information can be described in 3 different ways. It is the sum of the individual entropies minus the joint entropy, the entropy minus the conditional entropy, and the Kullback-Liebler divergence between the joint probability density and the sum of the marginal (individual) probability densities. All these definitions are equivalent when using

the Shannon entropy, and given in equations (6.1) & (6.2). However, when using the Rényi entropy & divergence all these formulations give different results (equation (6.3)).

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ I(X;Y) &= H(Y) - H(Y|X) \\ I(X;Y) &= H(X) + H(Y) - H(X,Y) \end{aligned} \tag{6.1}$$

$$I(X;Y) = D_{KL}(p(x,y)||p(x)p(y)) \tag{6.2}$$

$$H_\alpha(X) - H_\alpha(X|Y) \neq H_\alpha(Y) - H_\alpha(Y|X) \neq H_\alpha(X) + H_\alpha(Y) - H_\alpha(X,Y) \tag{6.3}$$

The standard formulation of the Rényi mutual information (derived from the information gain measure in [19]) is to use the Rényi generalised divergence between the joint probability density and the sum of the marginal probability densities, as in the Kullback-Liebler definition of Shannon's mutual information. This gives a measure which tends towards the Shannon mutual information as $\alpha \to 1$ from both sides. Additionally the measure meets the three requirements set out in the introduction, namely that it is parametrised by α, measures the shared amount of information between two variables, and is constructed solely from Rényi's measures.

$$I_{\alpha\text{Div}}(X;Y) = D_\alpha(p(x,y)||p(x)p(y)) \tag{6.4}$$

There is no reason why the mutual information cannot be constructed from the sum of the entropies minus the joint entropy, as this formulation simply returns the uncertainty which is jointly contained in both of the variables, or the information gain when about a variable when another is known. It can be shown that this value also tends towards the Shannon mutual information as $\alpha \to 1$, as the individual Rényi entropies tend to the Shannon entropy at this value, so the overall value must approach the Shannon mutual information. Intuitively then this value must be equal to the value derived from the divergence measure, as this is also measuring the information gain about one variable when another is known. However this is not the case with the joint entropy based measure, and the divergence measure both converging to the Shannon mutual information as $\alpha \to 1$, but diverging when $\alpha \neq 1$.

$$I_{\alpha\text{Joint}}(X;Y) = H_\alpha(X) + H_\alpha(Y) - H_\alpha(XY) \tag{6.5}$$

Creating the conditional formulation of the Rényi entropy is more difficult. As has been shown (in chapter 2), the value derived from simple conditioning of the Rényi entropy by

another variable diverges from the value derived by taking the joint entropy minus the entropy of the conditioning variable ($H_\alpha(XY) - H_\alpha(Y) \neq H_\alpha(X|Y)$). As the definition of the conditional entropy is the uncertainty remaining in one variable when the conditioning one is known, and this is the value the equation provides, then the conditional entropy has an unknown meaning in Rényi's information theory.

$$I_{\alpha\text{Cond}}(X;Y) = H_\alpha(X) - H_\alpha(X|Y) \tag{6.6}$$

$$I_{\alpha\text{Cond}}(Y;X) = H_\alpha(Y) - H_\alpha(Y|X) \tag{6.7}$$

$$I_{\alpha\text{Cond}}(X;Y) \neq I_{\alpha\text{Cond}}(Y;X) \tag{6.8}$$

As these values are asymmetric it is not suitable for use as a mutual information. This can be seen in figure 6.1, which uses data from the Lung dataset, with X = feature 40, and Y = class, graphed for $0.9 \leq \alpha \leq 1.11$.

Both of these equations converge to the Shannon mutual information as $\alpha \to 1$ and they also diverge from the other formulations of the Rényi mutual information discussed above. It is clear that the conditional equations and the joint formulation must diverge as the conditional entropy is no longer equal to the joint entropy minus the entropy of the conditioning variable, and it is the substitution of this relationship into either entropy formulation for the Shannon mutual information that relates the two. However as these equations are asymmetric with respect to the positions of X & Y they cannot be used as a valid measure of the mutual information.

Of the three different formulations possible for the mutual information the standard divergence method is the accepted one, as it directly takes a measure of the similarity of two probability distributions as the mutual information, and the set of divergences was constructed as an information gain measure by Rényi in [19]. However, as the joint entropy is a valid construct in Rényi's information theory, the use of the joint formulation as a mutual information should be equally valid, as it is still a measure of the information gain when one variable of a pair is known.

The conditional formulation is less well founded. The conditioning of a variable should be a property that is independent of the entropy measure used, as it effectively splits the variable into several separate sub-variables depending upon the value of the conditioning variable. However due to the position of the logarithm in the Rényi entropy it is not possible to separate the variables and apply Bayes rule to gain the joint entropy and the entropy of the

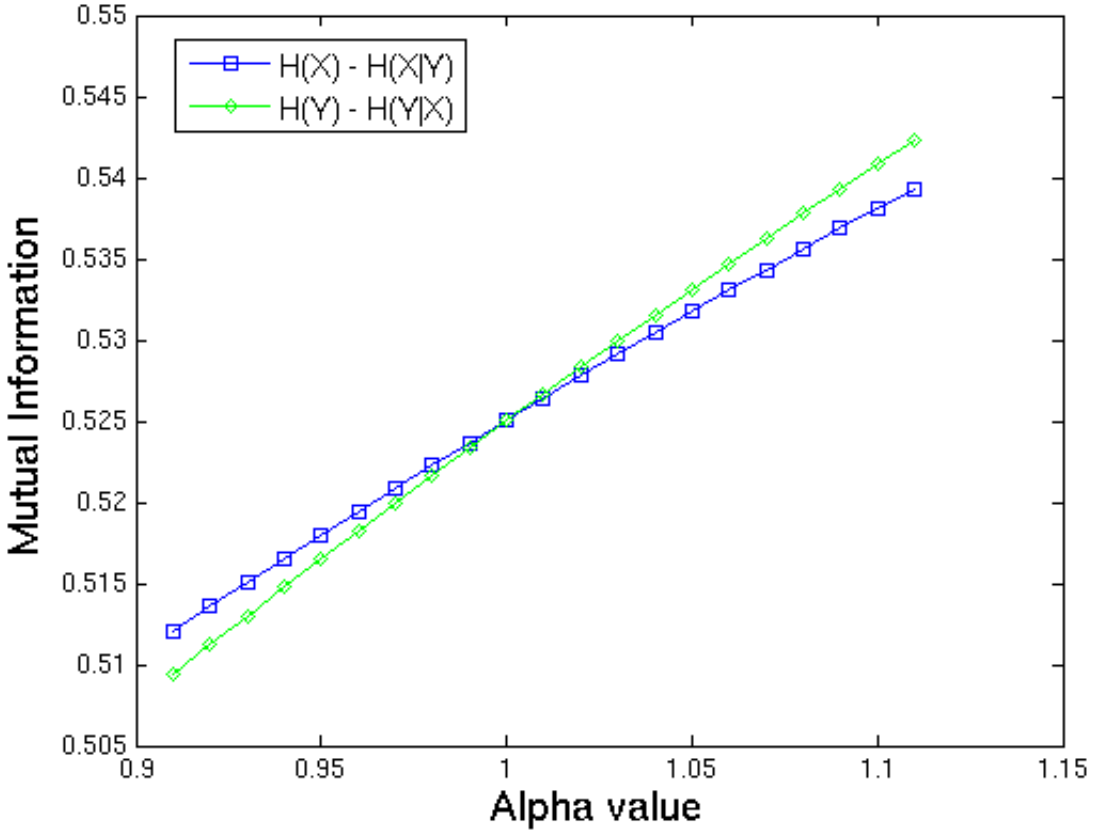

Figure 6.1: The asymmetry of the conditional formulation

conditional variable. Whilst the conditional equations do appear similar to a mutual information it is due to the lack of symmetry, and the unknown properties of the equation derived for the conditional entropy, which prevents it from being used as a mutual information.

Figures 6.2, 6.3 & 6.4 show the varying ways the different formulations of the mutual information diverge. The datapoint at $\alpha = 1$ is generated with the Shannon mutual information to give continuity to the graph, with the lines of best fit passing through this point when graphed in the region around $\alpha = 1$ incrementing α in steps of 0.01. The various different formulations vary in data dependent ways with no formulation having constant properties in relation to the others.

6.3 Constructing the Conditional Mutual Information

Due to the problems in finding a conditional entropy that has a valid meaning in Rényi's information theory the conditional mutual information is harder to form. The possible

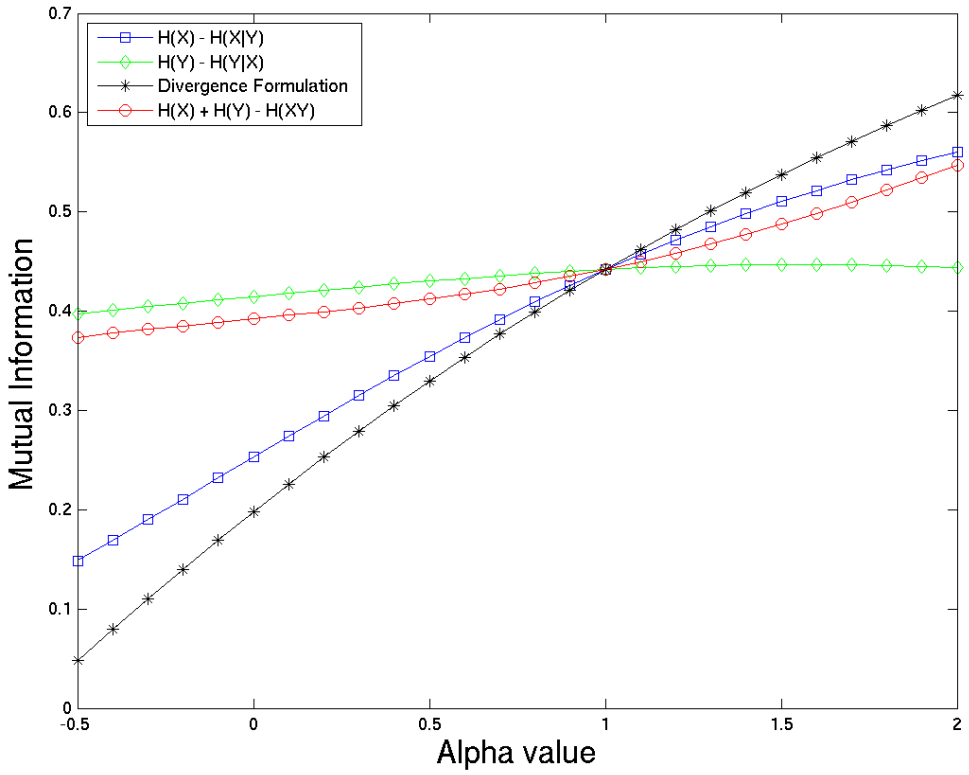

Figure 6.2: Rényi mutual information - lung dataset X = feature 14, Y = class

formulations for the conditional mutual information using Shannon entropy are given below:

$$I(X;Y|Z) = H(X|Z) - H(X|YZ) = H(X|Z) + H(Y|Z) - H(XY|Z) \tag{6.9}$$

$$I(X;Y|Z) = \sum_{z \in Z} p(z) D_{KL}(p(x,y|Z=z)||p(x|Z=z)p(y|Z=z)) \tag{6.10}$$

As the first two formulations use the conditional entropy, these are not valid, and have strange behaviour as shown in figure 6.5. The divergence formulation is more stable, and is thus used as the basis for the Rényi conditional mutual information.

Therefore the equation for the conditional mutual information using the Rényi generalised divergence is given in equation (6.11).

$$I_\alpha(X;Y|Z) = \sum_{z \in Z} p(z) D_\alpha(p(x,y|Z=z)||p(x|Z=z)p(y|Z=z)) \tag{6.11}$$

Now all the different measures have been constructed it is possible to extend the three feature selection algorithms chosen (CMIM, DISR and mRMR) to work with the Rényi

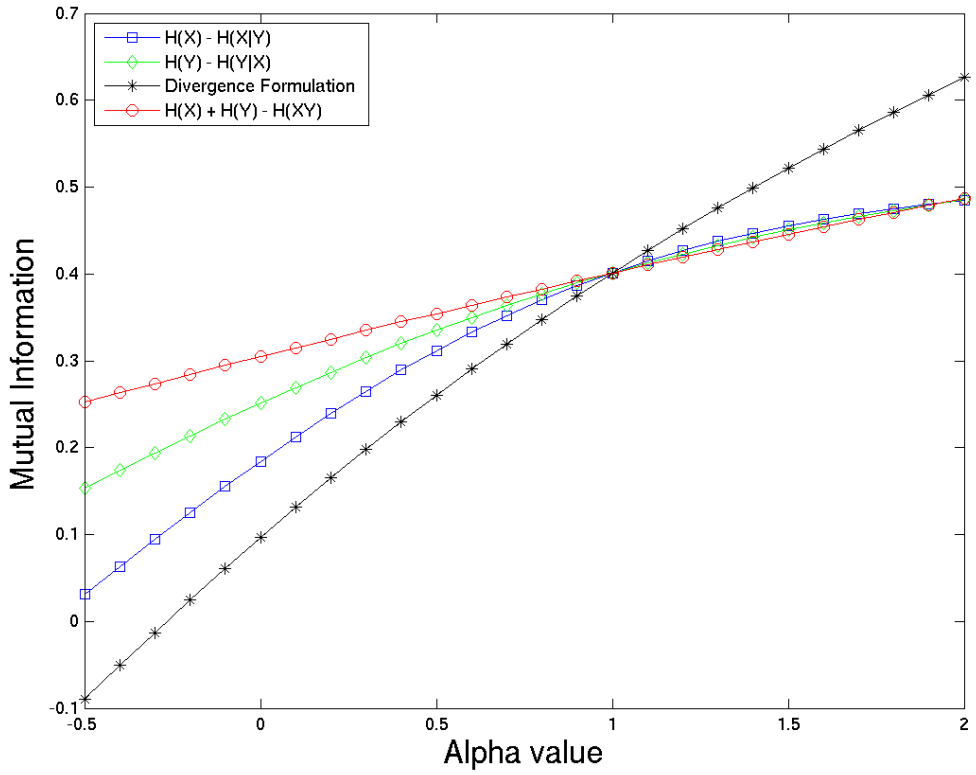

Figure 6.3: Rényi mutual information - lung dataset X = feature 29, Y = class

entropy and information measures.

6.4 Reconstructing the Algorithms

Due to the complexities of constructing the algorithms in the Rényi space, there were different ways the algorithms could be constructed. The algorithms based around simple mutual informations with Shannon entropy could have two possible formulations, either using the joint entropy form of mutual information, or the divergence form of mutual information. The CMIM algorithm which is based around the conditional mutual information has only one valid formulation, due to the inconsistencies in the constructed version of the Rényi conditional entropy.

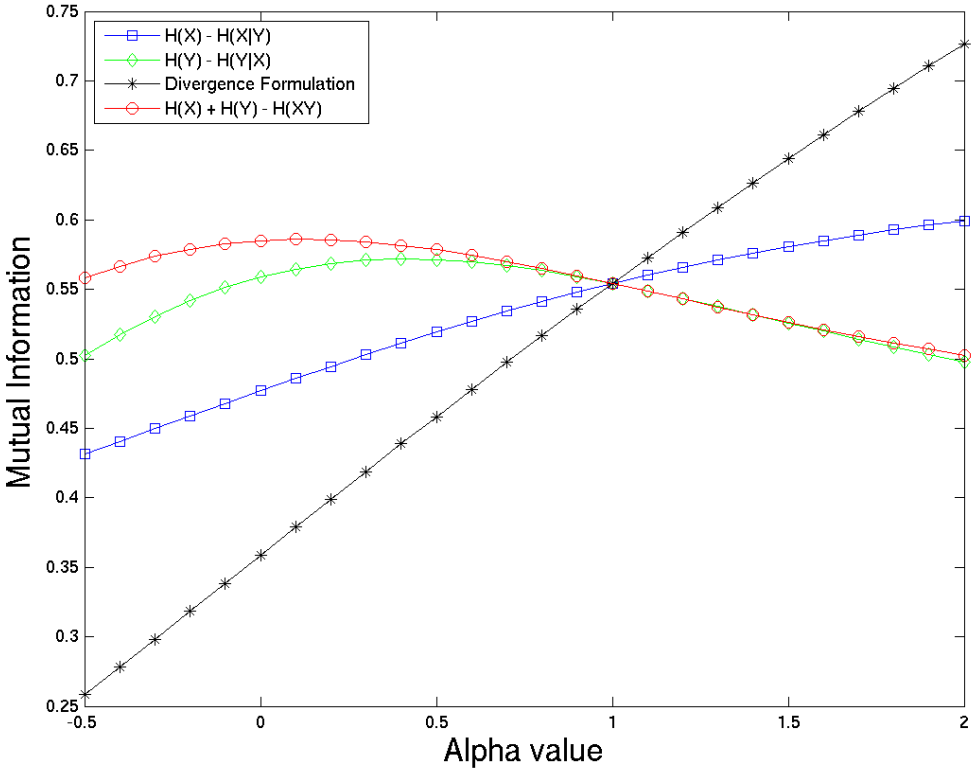

Figure 6.4: Rényi mutual information - lung dataset X = feature 49, Y = class

6.4.1 Constructing CMIM

As there is only one valid form of the conditional mutual information with the Rényi entropy, then the reconstruction of the CMIM algorithm is straightforward. The conditional mutual information function in the original algorithm can be simply replaced with the conditional mutual information divergence formulation, to lead to a valid algorithm. However the initial state of the CMIM algorithm involves populating the score vector with the mutual information of each feature on the class, and there are two valid forms for this mutual information (namely the divergence and the joint entropy). The divergence formulation was chosen as this is the valid form for the conditional mutual information, and the testing is designed to investigate different formulations of the Rényi mutual information.

6.4.2 Constructing DISR & mRMR

Both of these algorithms use standard mutual informations, which can be replaced with two different formulations when working with the Rényi entropy. Two versions of the algorithms

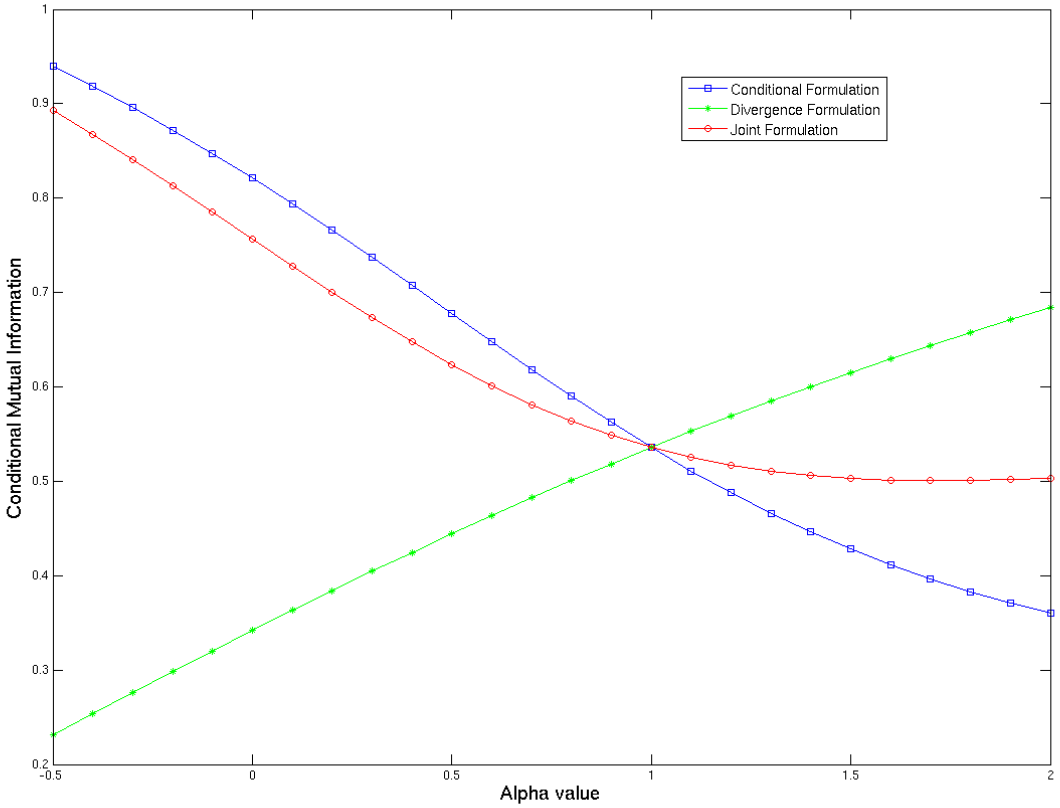

Figure 6.5: The different formulations of conditional mutual information

were constructed, with the first version using the divergence formulation for the Rényi mutual information, and the second using the joint formulation for the Rényi mutual information. Additionally the joint entropy used in the calculation of the Symmetric Relevance (equation (2.20)) in the DISR algorithm was replaced with the Rényi joint entropy. In both cases all the mutual informations, including the initial feature ranking step to determine the first feature, were replaced with Rényi mutual informations.

6.5 Results

The tests were ran according to the plan detailed in chapter 3. As the α parameter in the Rényi Entropy and the Rényi generalised divergence is a continuous parameter and the performance of the algorithms cannot be defined by an equation it is impossible to fully capture the behaviour of the algorithms when the alpha parameter is varied. An

approximation is gained by sampling the results at fixed values of α. It was decided to sample the α parameter between 0.1 and 2.0 in steps of 0.1, to provide an estimate of the selection power of different values of α. Each test run provides 19 selected feature sets, in addition to the standard Shannon algorithms, which are provided for comparison. This data is separated out over 4 different graphs, to preserve the readability of the results, with the Shannon algorithms provided on each graph for comparison, and the SVM and 3-NN classifier results provided on separate graphs. This leads to 8 graphs per valid formulation of each algorithm with the Rényi mutual information, per dataset. A subset of the graphs with interesting properties are given below.

6.5.1 CMIM Results

There is one valid form for the CMIM algorithm when working with the Rényi mutual information. The results are shown for the lung dataset, and the NCI9 dataset.

Lung Dataset

Two results are presented, $0.6 \leq \alpha \leq 1.1$ and $1.6 \leq \alpha \leq 2.0$, in figures 6.6 and 6.7.

In figure 6.6, the algorithm using $\alpha = 0.6$ outperforms the rest of the α values, by being consistently lower than the traditional Shannon version of CMIM, when using the 3-NN classifier, and after the first 15 features have been selected. When selecting the last 15 features a noticeable divide opens between $\alpha < 0.9$ and $0.9 \leq \alpha \leq 1.1$ including the Shannon algorithm. In general after selecting the 15th feature, the algorithms with $\alpha < 1$ provide an increase in classification performance over the standard Shannon algorithm. When using the SVM classifier, the results are less clear. The values either side of 1 follow the Shannon classification error quite closely until 50 features have been selected, where the differences in the features chosen cause divergence in the classification performance. The α values below 0.9, have an erratic classification performance until after the 40th feature has been selected, where they start to improve upon the classification performance of the other values.

In figure 6.7, the higher values of α are outperformed by the Shannon algorithm when using the SVM classifier, only at the end producing a set of features which is an improvement on the one generated by the Shannon algorithm. Using the 3-NN classifier the picture is less clear, as the classification errors are more variable, with the high α values appearing to provide a performance increase when using selecting up to 30 features, before the performance becomes similar to the standard Shannon algorithm.

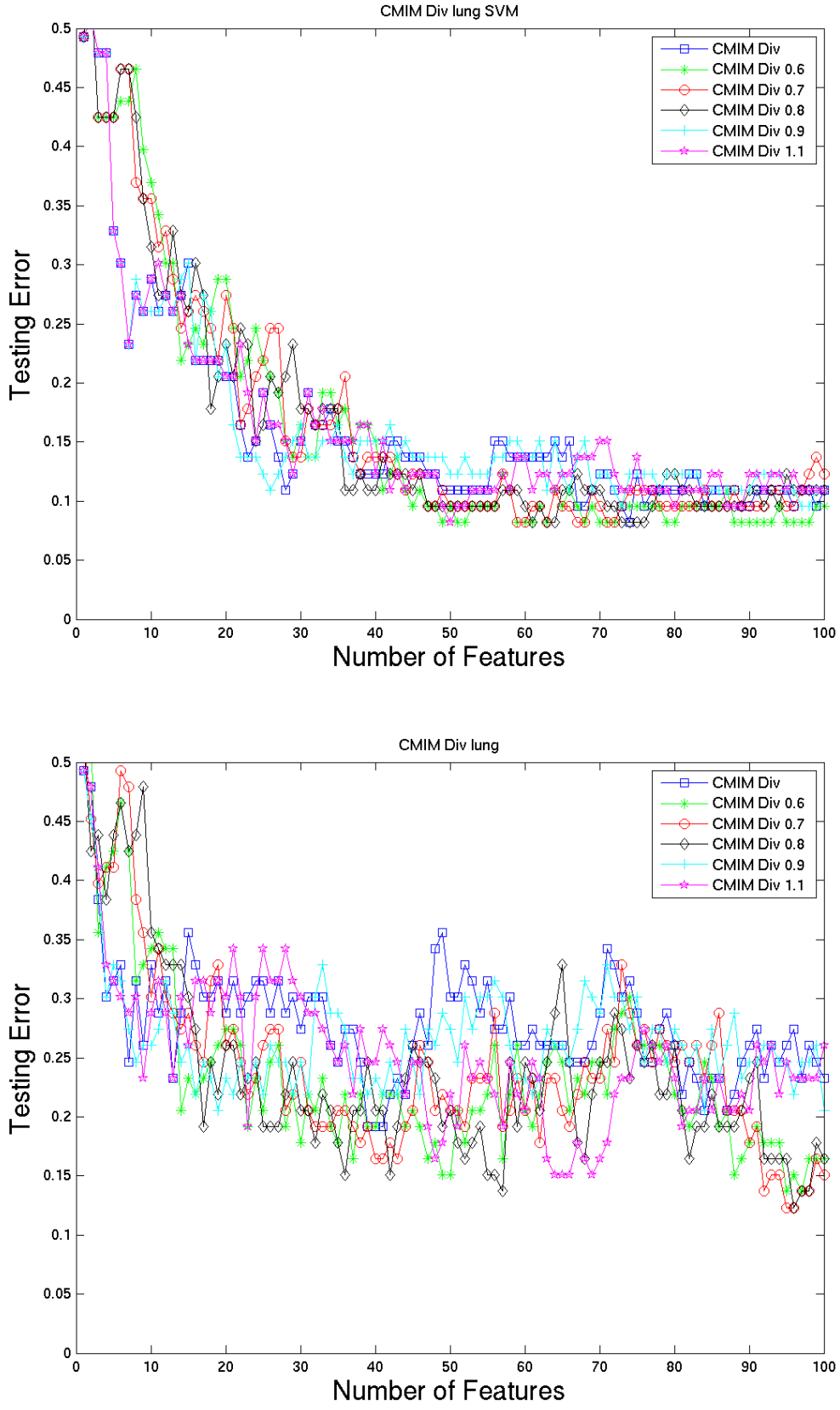

Figure 6.6: CMIM, Lung Cancer Dataset, $0.6 \leq \alpha \leq 1.1$

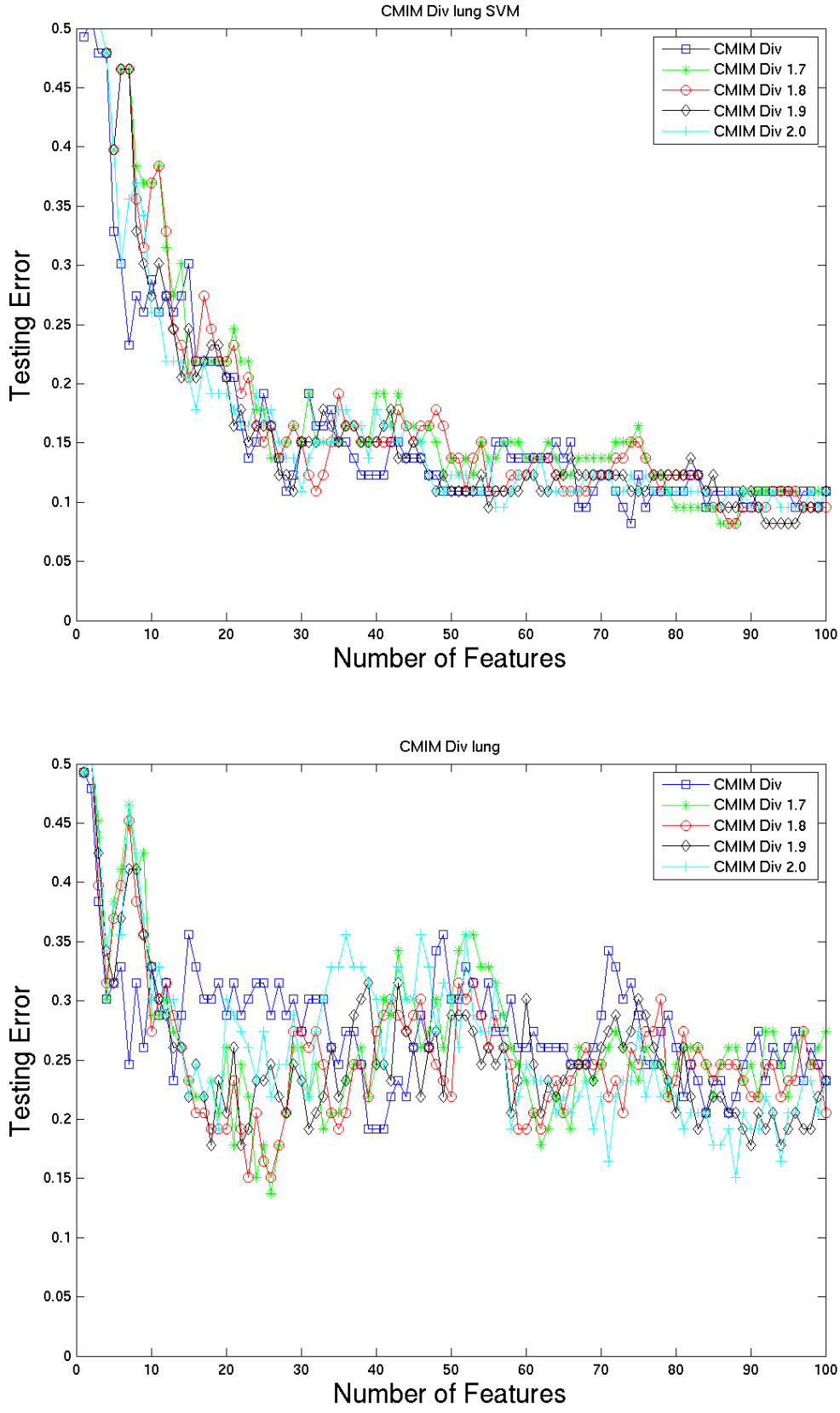

Figure 6.7: CMIM, Lung Cancer Dataset, $1.7 \leq \alpha \leq 2.0$

NCI9 Dataset

Four results are presented, but only using the SVM classifier, as due to the number of classes in the NCI9 dataset and the properties of the 3-NN classifier the results are extremely variable. The results cover the whole range of α tested, with $0.1 \leq \alpha \leq 1.1$ in figure 6.8 and $1.2 \leq \alpha \leq 2.0$ in figure 6.9.

In figure 6.8, $\alpha < 0.6$ reaches a low classification error faster than the standard Shannon algorithm, but plateaus at that level and doesn't provide a decrease in classification error as the number of features increases. For $0.6 \leq \alpha \leq 1.1$ the performance is approximately the same as the Shannon algorithm.

In figure 6.9, where $1.2 \leq \alpha \leq 1.6$ the performance is equivalent to the Shannon algorithm except when selecting the features between 40 and 50, where the performance improves over the standard Shannon algorithm. Additionally when $\alpha = 1.4$ in this region it has a classification error 17.5 percentage points lower than the Shannon algorithm.

6.5.2 DISR Results

There are two valid forms for the DISR algorithm when working with the Rényi mutual information, the divergence formulation, and the joint entropy formulation. The results presented form a comparison between the two different formulations, and what advantages, if any, they provide.

NCI 9 Dataset

Three results are presented, using the SVM classifier. These results show the contrast between the classification power of the divergence formulation and the joint entropy formulation.

In figure 6.10 the variation between the two formulations of the Rényi mutual information at lower values of α can be seen, with the joint entropy formulation generating classification errors greater than 50%, where the divergence formulation generates classification errors in the region of 25-30%. This shows that for low values of α the joint entropy formulation is unable to select features according to their information. For $0.4 \leq \alpha \leq 0.5$ the two formulations perform equivalently, though the joint formulation still varies more than the divergence formulation for the value $\alpha = 0.4$.

In figure 6.11 the variation between the two formulations is minimal, with the joint formulation being closer centred on the standard Shannon algorithm.

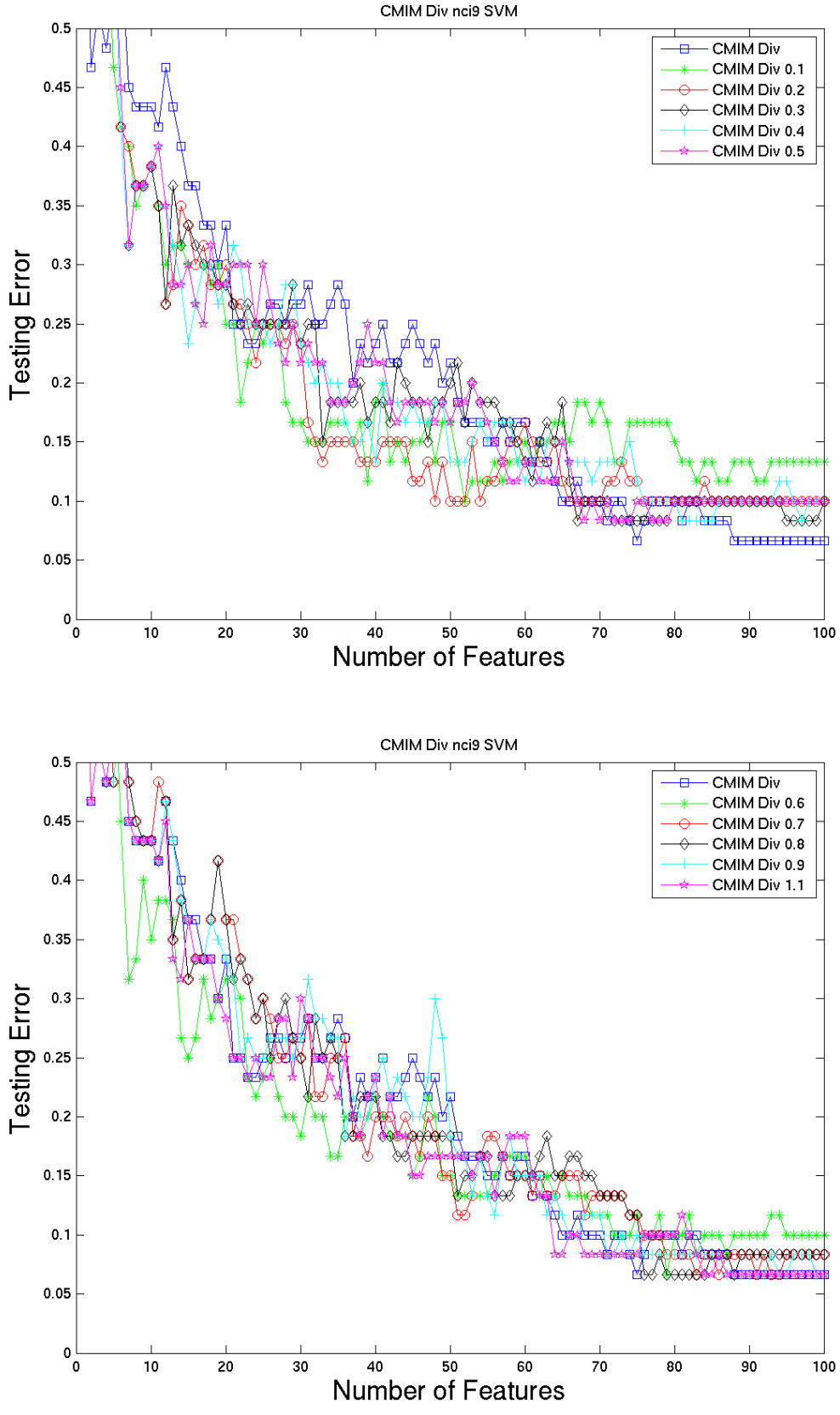

Figure 6.8: CMIM, NCI9 Dataset, SVM Classifier, $0.1 \leq \alpha \leq 0.5$ & $0.6 \leq \alpha \leq 1.1$

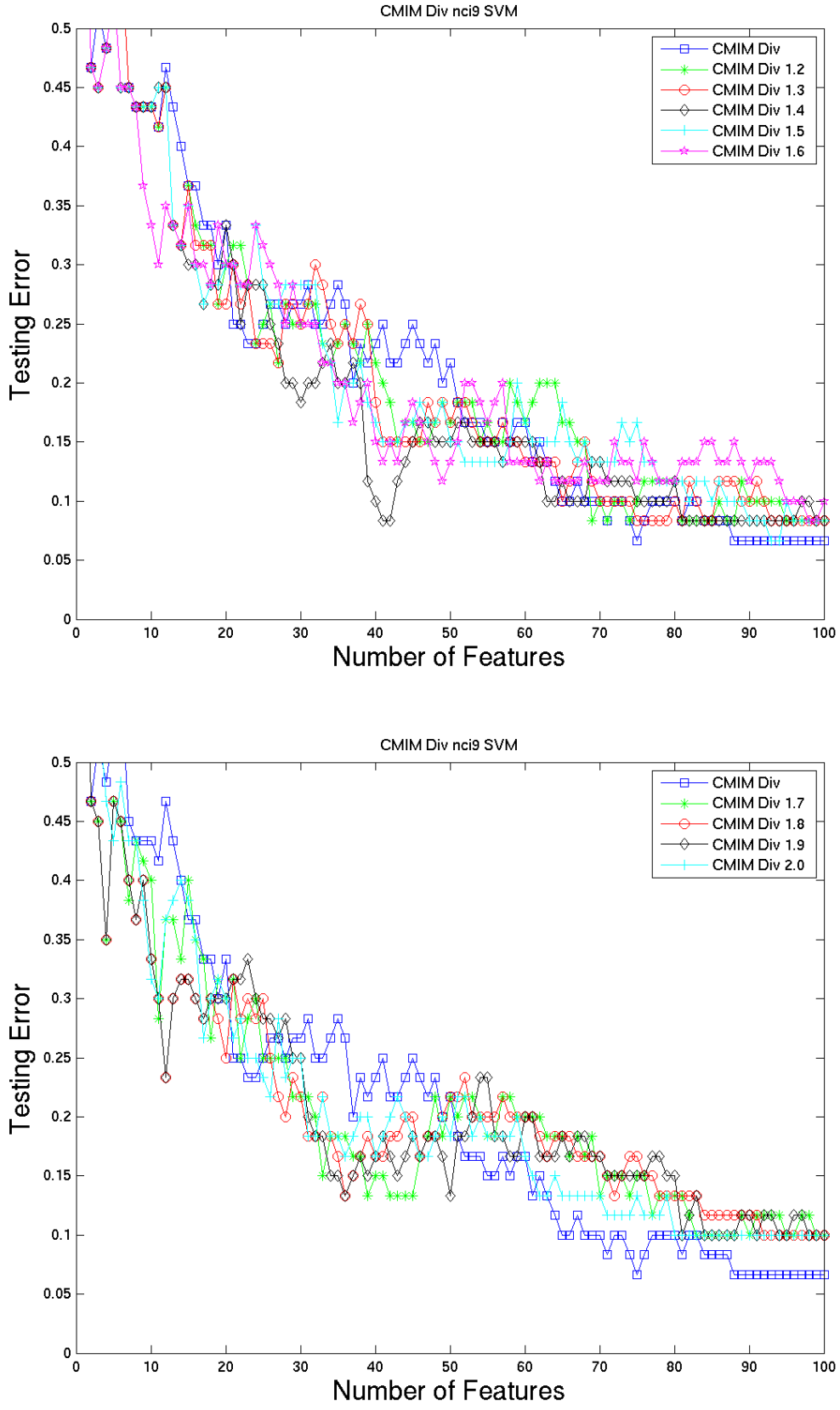

Figure 6.9: CMIM, NCI9 Dataset, SVM Classifier, $1.2 \leq \alpha \leq 1.6$ & $1.7 \leq \alpha \leq 2.0$

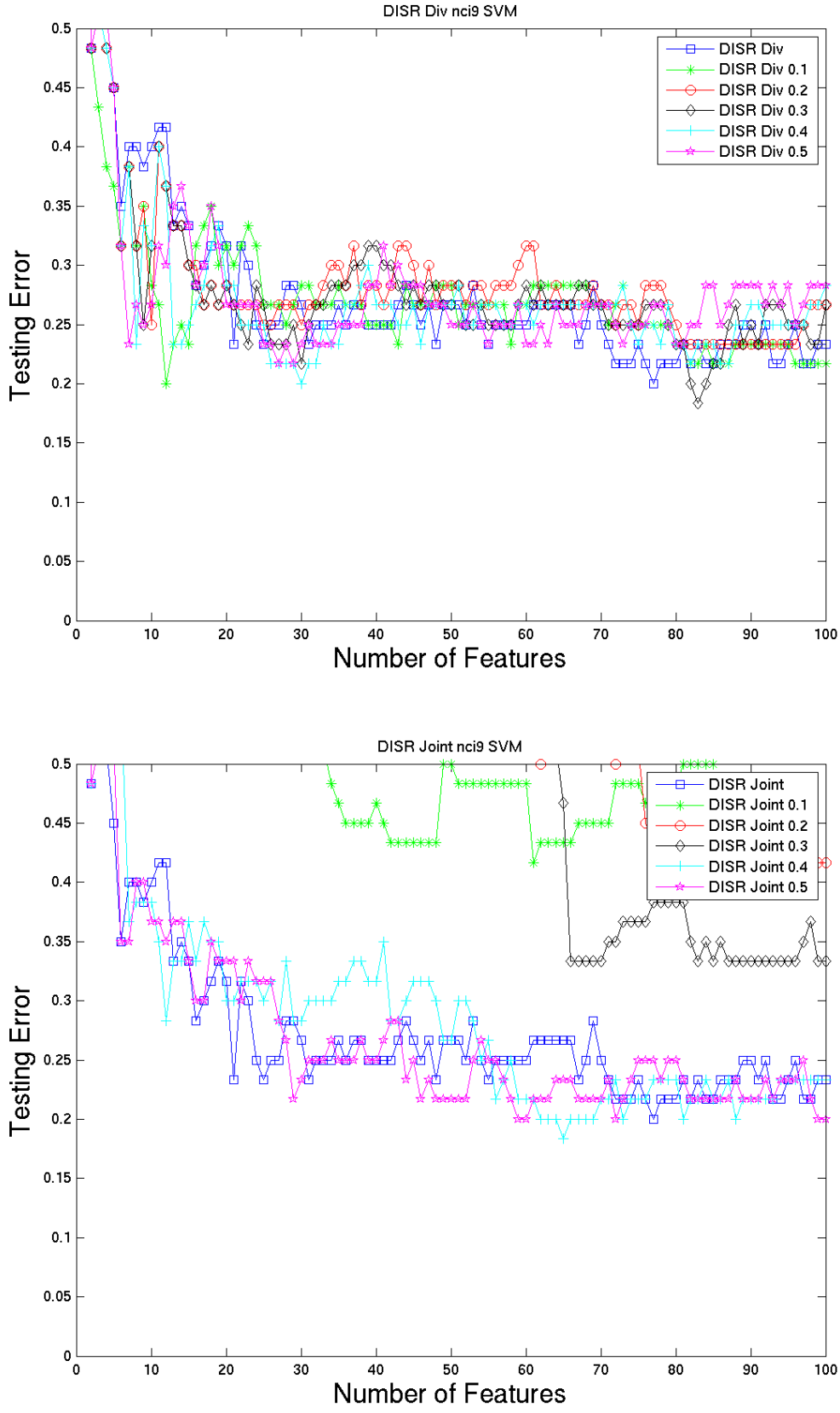

Figure 6.10: DISR Joint and Divergence, NCI9 Dataset, SVM Classifier, $0.1 \leq \alpha \leq 0.5$

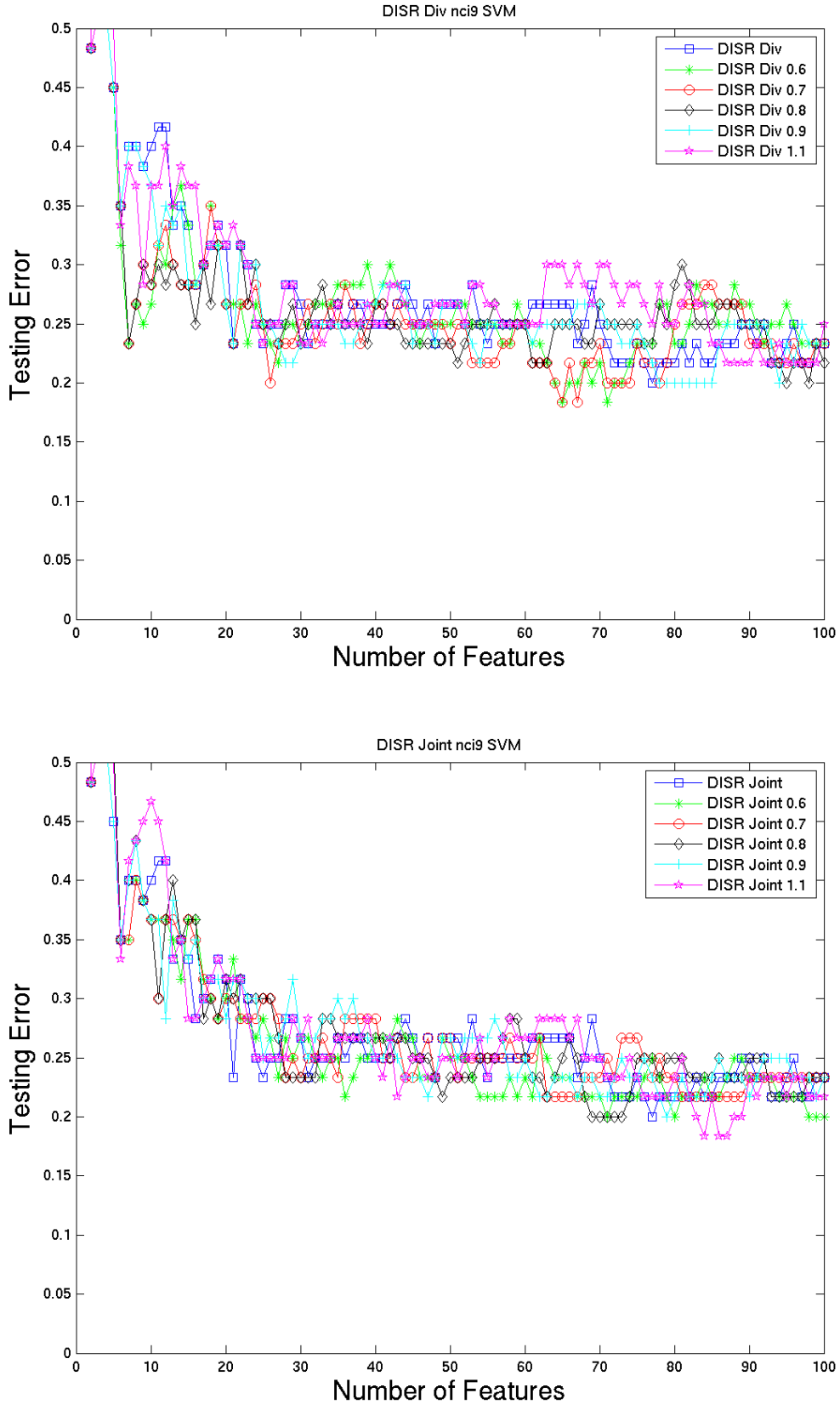

Figure 6.11: DISR Joint and Divergence, NCI9 Dataset, SVM Classifier, $0.6 \leq \alpha \leq 1.1$

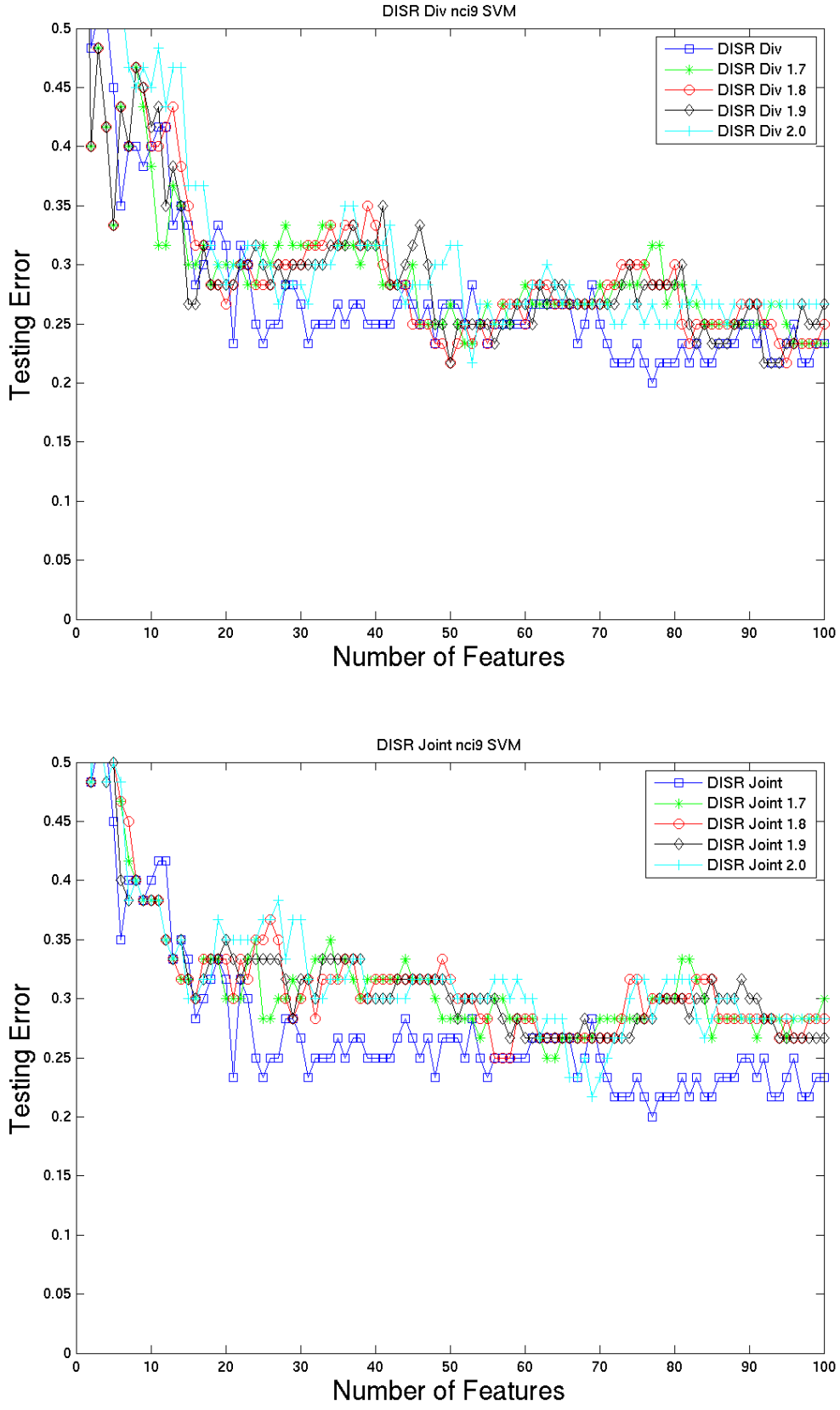

Figure 6.12: DISR Joint and Divergence, NCI9 Dataset, SVM Classifier, $1.7 \leq \alpha \leq 2.0$

In figure 6.12, the variation between the two formulations increases, with the joint formulation losing classification accuracy as the α value increases.

6.5.3 mRMR-D Results

There are two valid forms for the mRMR-D algorithm when working with the Rényi mutual information, the divergence formulation, and the joint entropy formulation. The results presented form a comparison between the two different formulations, and what advantages, if any, they provide.

Adenocarcinoma Dataset

The adenocarcinoma dataset has two classes, and thus the 3-NN classifier is not subject to the problems found with multiclass datasets. These results show the difference in classification performance between the joint formulation and the divergence formulation. Figure 6.13 shows the results from the SVM classifier for both formulations. The divergence formulation consistently performs better for $0.2 \leq \alpha \leq 0.5$, compared with both the joint formulation and the standard Shannon algorithm. The joint formulation outperforms the Shannon algorithm whilst selecting features 23 through 62, but converges to the Shannon result for higher numbers of features.

Figure 6.14 shows the results from the 3-NN classifier for both formulations. With this classifier both formulations are outperformed by the Shannon algorithm, with the joint formulation initially performing as well as the Shannon algorithm before strongly diverging to give a higher classification error.

Colon Dataset

The colon dataset also has two classes, but the results presented are from the SVM classifier. These results show how the joint formulation can improve upon the classification performance of the divergence formulation for given α values. Figure 6.15 shows the results from the SVM classifier with $1.2 \leq \alpha \leq 1.6$. The joint formulation consistently outperforms the divergence formulation for all values except when $\alpha = 1.6$ where the performance is mixed when compared with the Shannon version, but results in a much lower classification error. The minimum error achieved by the divergence formulation is 3.2%, selecting 61 to 65 features, with $\alpha = 1.5, 1.4$ and briefly $\alpha = 1.2$ (though this value gives an highly varying performance in the set of features selected). After this low point the classification error for the values of α shown rise until they exceed the error for the standard Shannon algorithm.

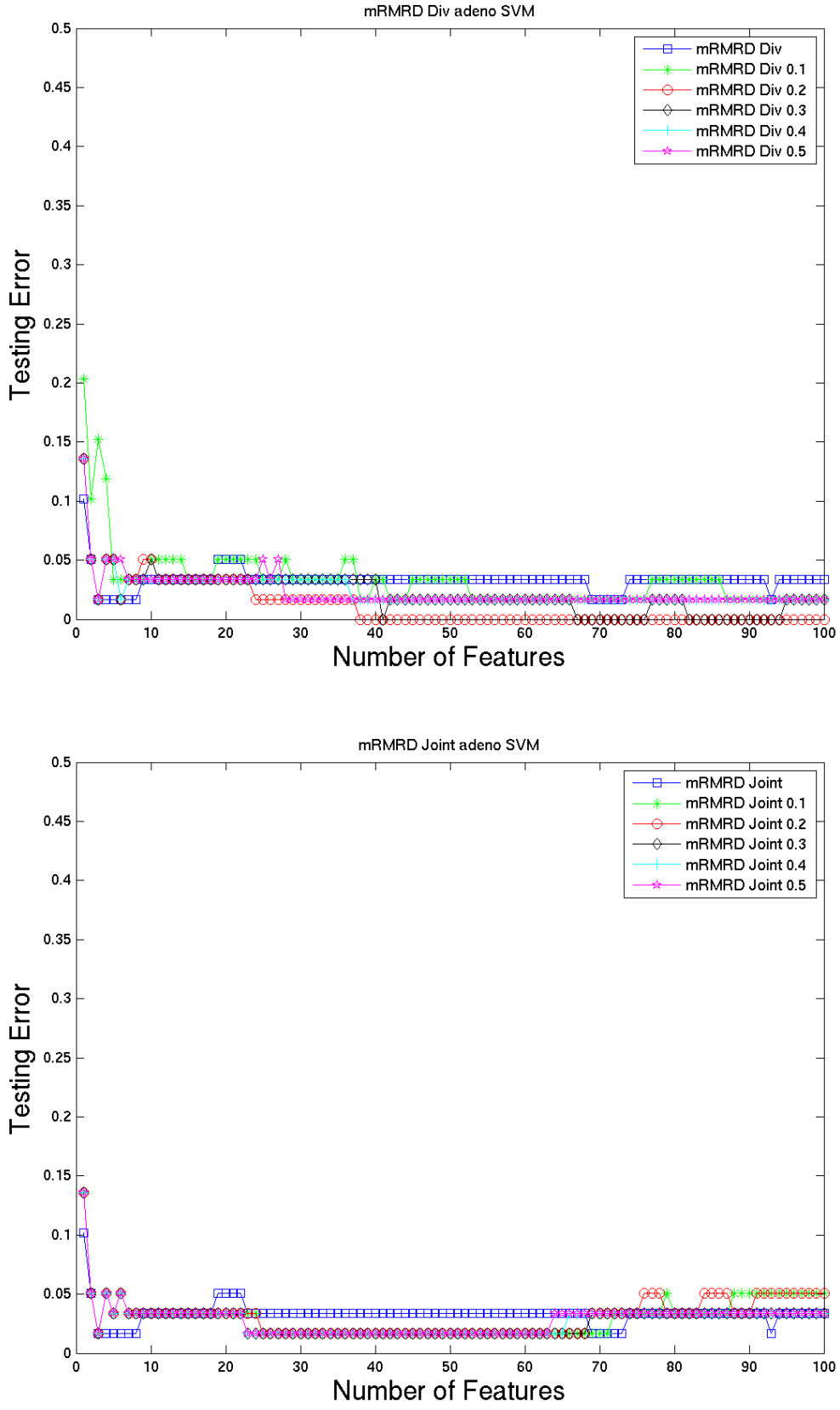

Figure 6.13: mRMR-D Joint and Divergence, Adenocarcinoma Dataset, SVM Classifier, $0.1 \leq \alpha \leq 0.5$

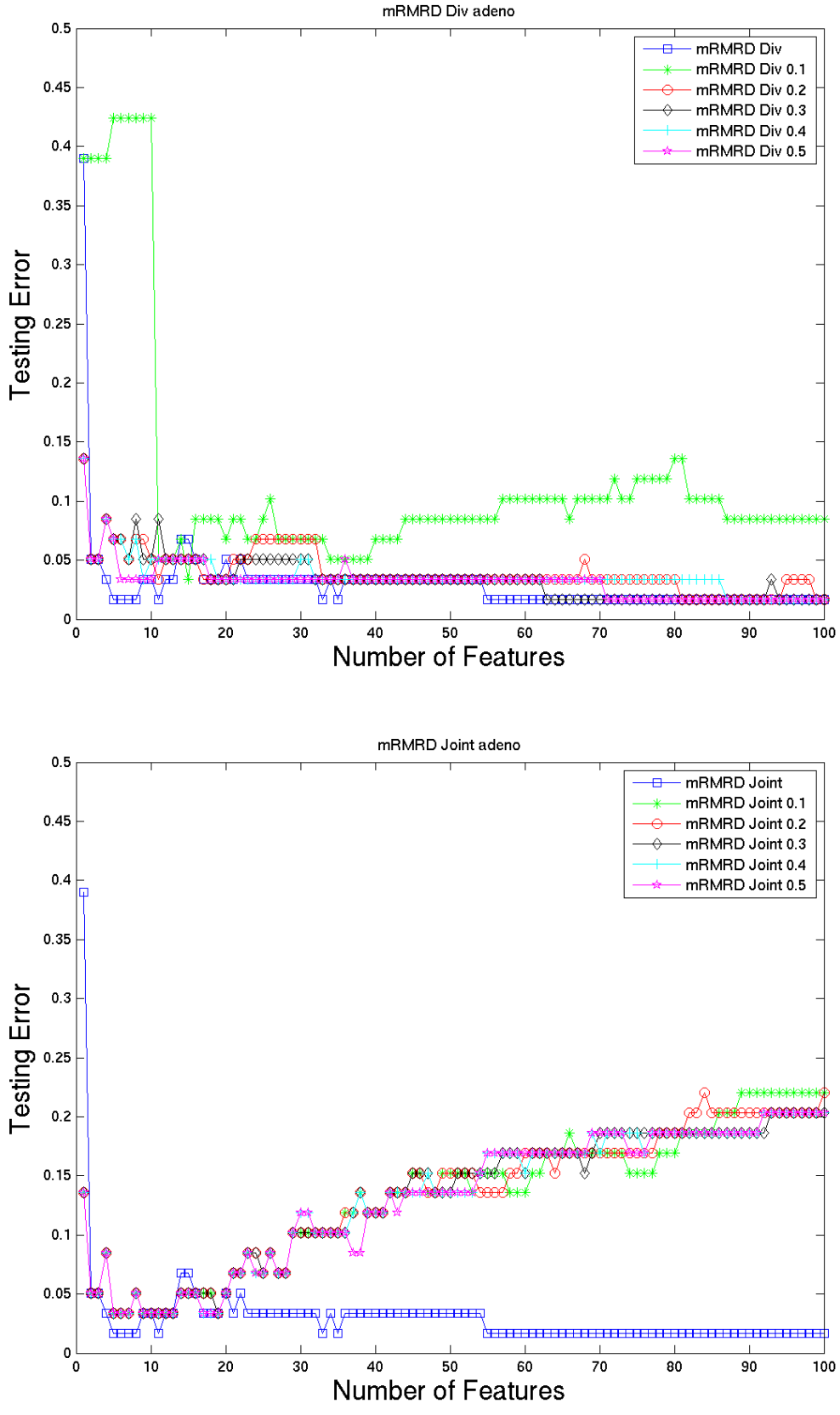

Figure 6.14: mRMR-D Joint and Divergence, Adenocarcinoma Dataset, 3-NN Classifier, $0.1 \leq \alpha \leq 0.5$

With the joint formulation this low point of classification error is reached consistently when $\alpha = 1.6$ selecting 89 features, though all other solutions remain below the error rate for the Shannon algorithm and do not rise in a similar manner to the divergence formulation.

Lymphoma Dataset

The lymphoma dataset has 9 classes, but generates lower classification errors than expected as can be seen from the results in chapter 4. Figure 6.16 shows the results from the SVM classifier with $0.1 \leq \alpha \leq 0.5$ for both formulations. The joint formulation outperforms the divergence formulation consistently after selecting the first 10 features and its performance is consistently lower than the Shannon algorithm after selecting the first 40 features. In contrast the divergence formulation only begins to outperform the Shannon algorithm after 76 features have been selected.

Figure 6.17 shows the results from the SVM classifier with $1.7 \leq \alpha \leq 2.0$ for both formulations. In contrast to the previous results, here the divergence formulation performs equivalently to the Shannon algorithm, whilst the joint formulation is outperformed by more than 5 percentage points on average. This shows how the changing α parameter modifies the properties of the different formulations in different ways therefore results derived from one formulation cannot be transferred to another.

6.5.4 mRMR-Q Results

There are two valid forms for the mRMR-Q algorithm when working with the Rényi mutual information, the divergence formulation, and the joint entropy formulation. The results presented form a comparison between the two different formulations, and what advantages they provide.

Colon Dataset

The results presented for the colon dataset using the mRMR-Q algorithm are similar to those presented for the mRMR-D algorithm, with the joint formulation outperforming the divergence formulation with $1.2 \leq \alpha \leq 1.6$ (presented in figure 6.18. However the joint formulation has one more noticeable feature using the mRMR-Q algorithm, which is the unusual performance of the value $\alpha = 1.5$ when selecting the 28th and 29th features, where the classification error drops substantially to give only one incorrect classification under the leave-one-out cross validation testing. This result is not repeated in other tests, and is the lowest error estimate gained for this dataset using the SVM classifier.

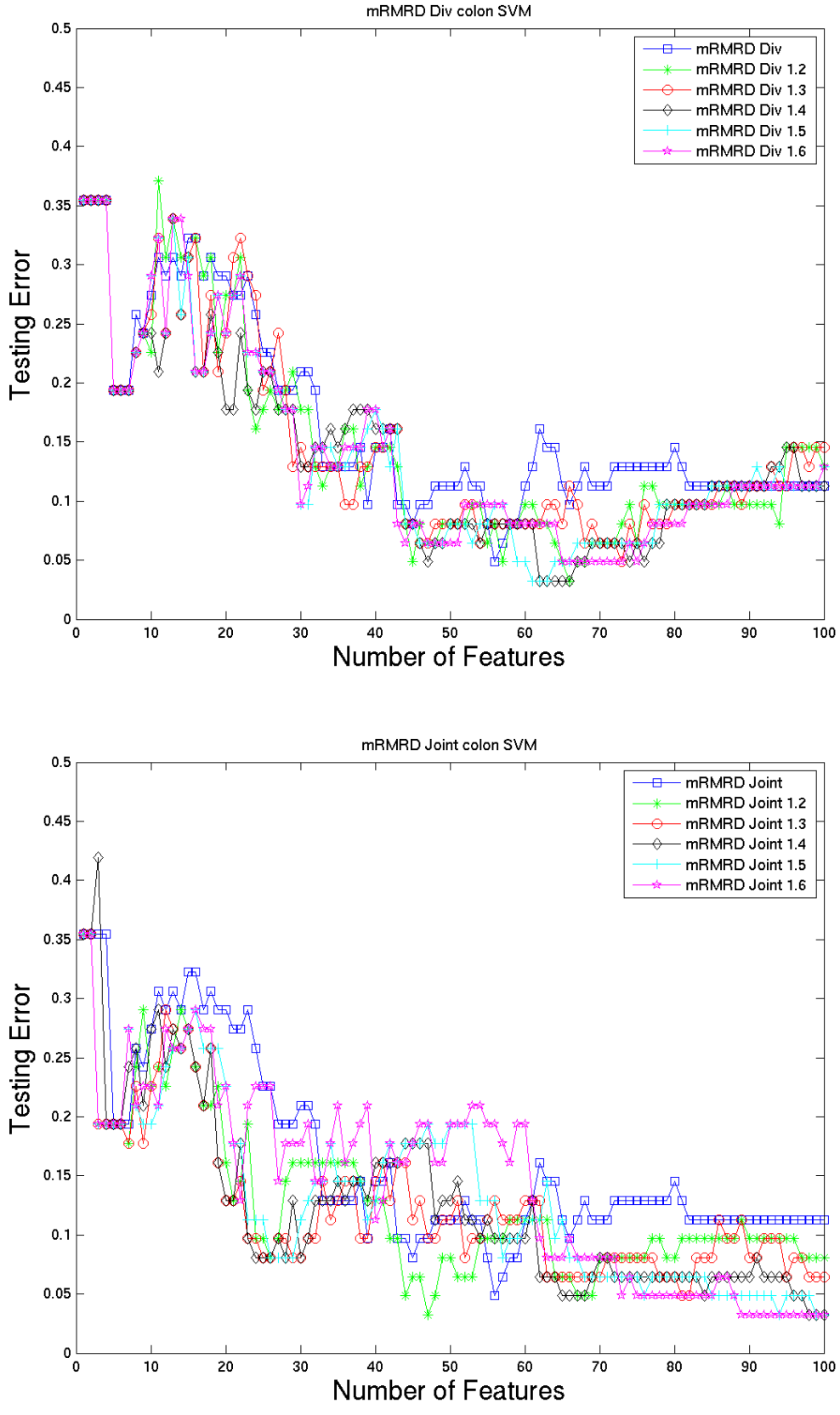

Figure 6.15: mRMR-D Joint and Divergence, Colon Dataset, SVM Classifier, $1.2 \leq \alpha \leq 1.6$

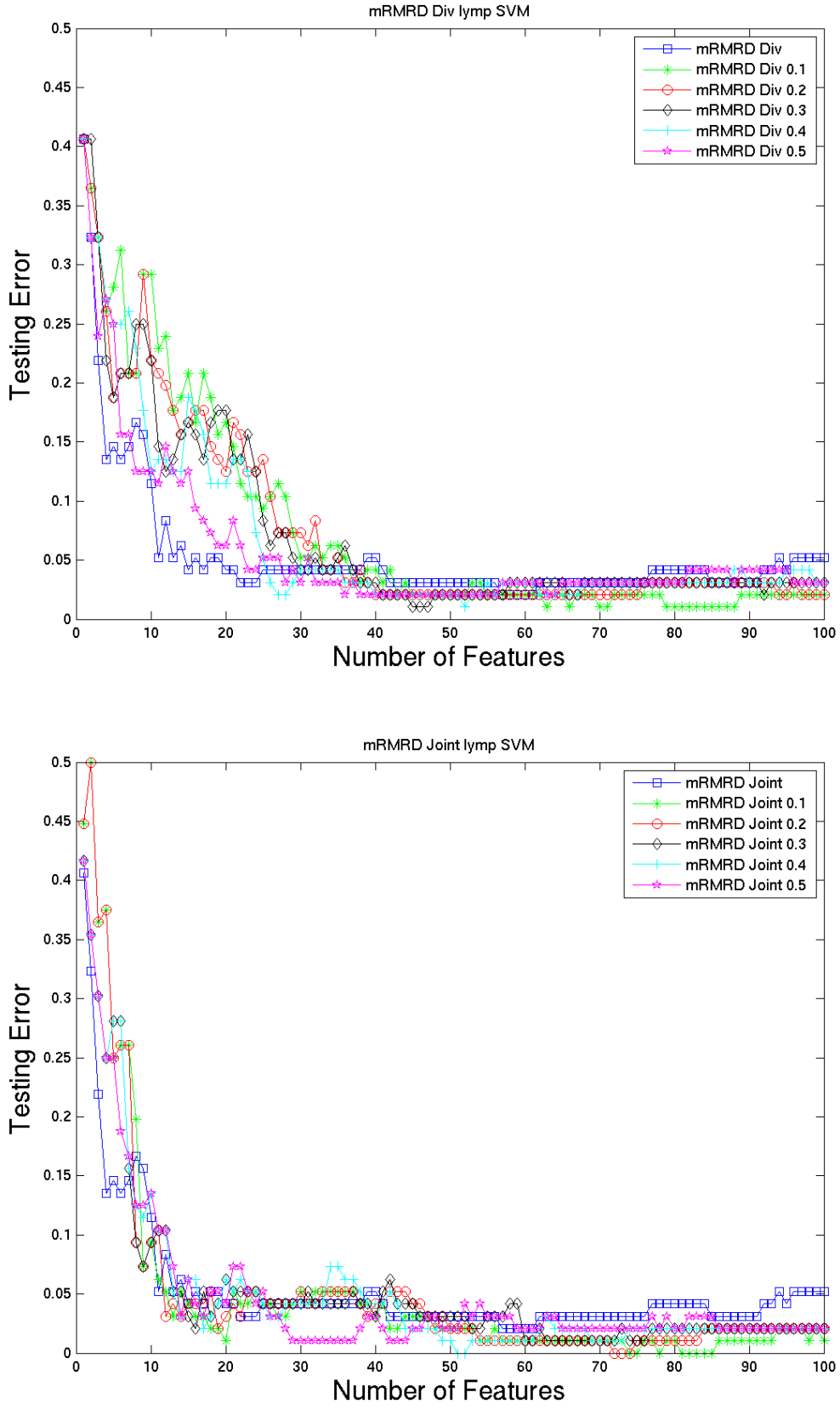

Figure 6.16: mRMR-D Joint and Divergence, Lymphoma Dataset, SVM Classifier, $0.1 \leq \alpha \leq 0.5$

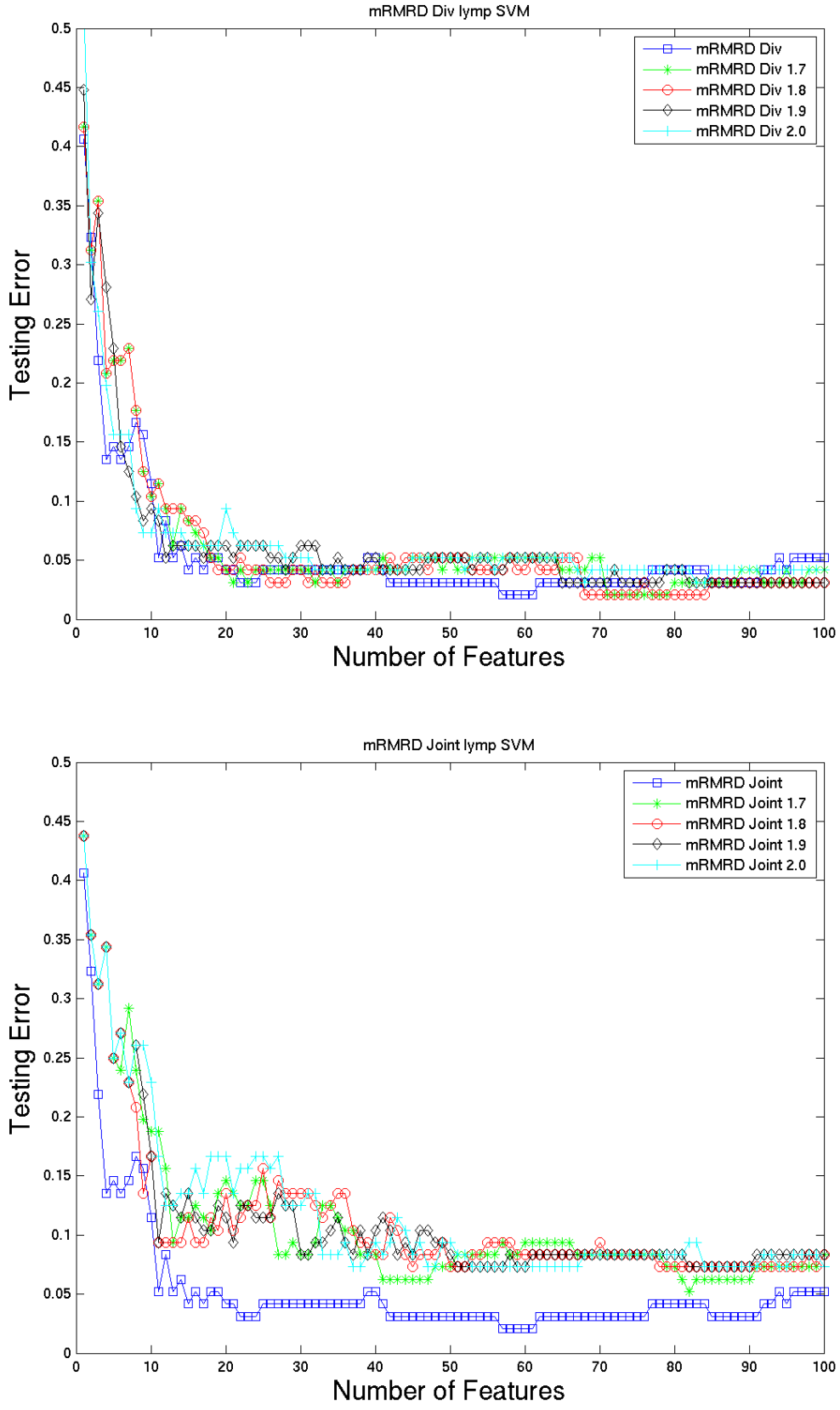

Figure 6.17: mRMR-D Joint and Divergence, Lymphoma Dataset, SVM Classifier, $1.7 \leq \alpha \leq 2.0$

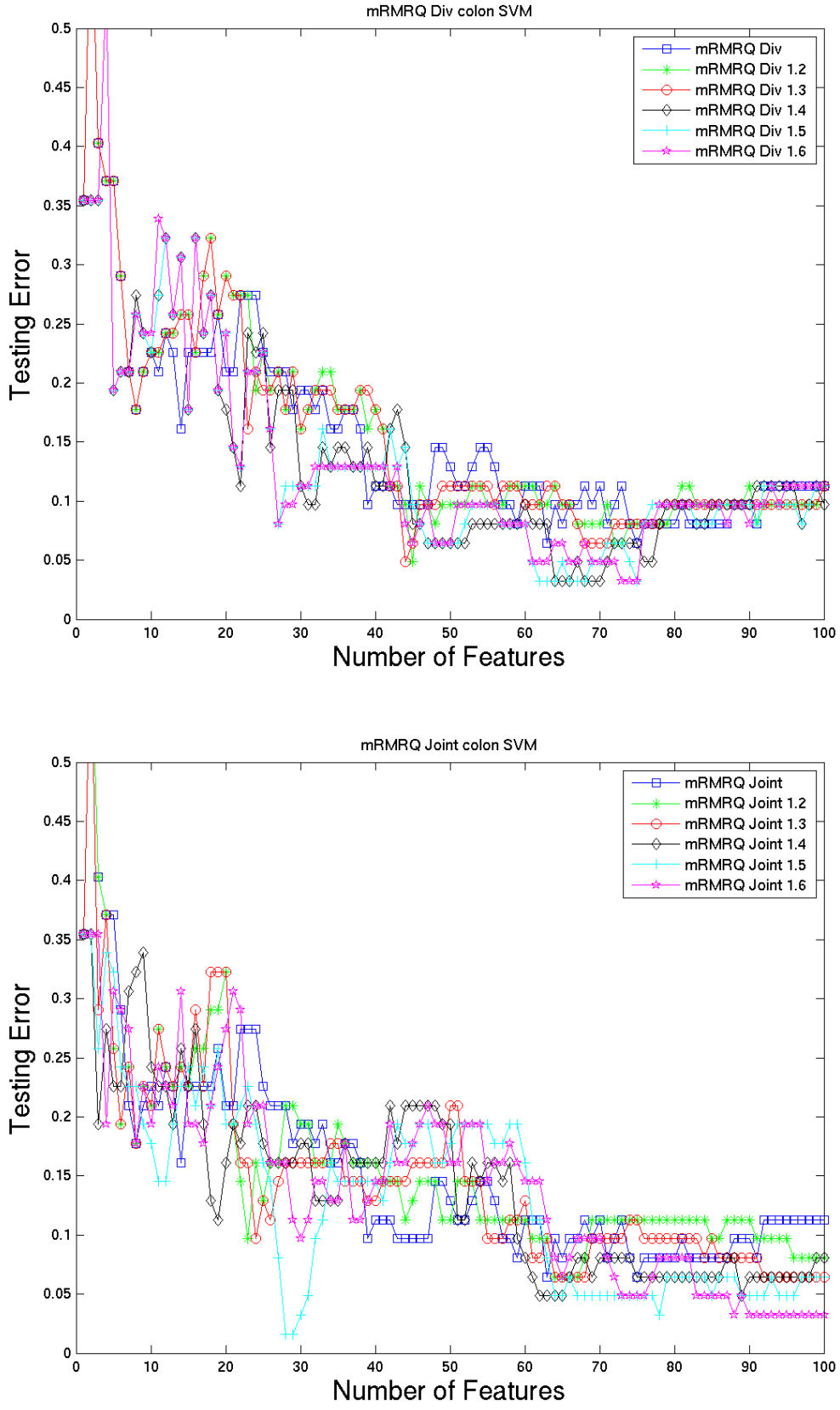

Figure 6.18: mRMR-Q Joint and Divergence, Colon Dataset, SVM Classifier, $1.2 \leq \alpha \leq 1.6$

Figure 6.19 presents the results using the SVM classifier with $1.7 \leq \alpha \leq 2.0$. In this result it can be seen how the joint formulation has greater variance depending upon the α value, compared to the divergence estimate, which is highly convergent over this range. Additionally the performance of the joint formulation can be seen to decrease in this range as the α increases, whilst it remains an improvement upon the Shannon algorithm for $\alpha = 1.7, 1.8$ the performance is not consistently better over the selection of 100 features, with the end of the testing range providing the most stable performance and the best result for the Rényi mutual information.

Figure 6.20 presents the results using the 3-NN classifier with $1.7 \leq \alpha \leq 2.0$. The performance of the Rényi mutual information in the algorithms is lower than the Shannon algorithm, only occasionally improving upon the classification error. This is representative of several results where the SVM performance is not replicated under the 3-NN classifier, when using different versions of the Rényi feature selection algorithms.

Lung Dataset

Figure 6.21 presents the results from the SVM classifier with $1.7 \leq \alpha \leq 2.0$. In the results the divergence formulation outperforms the joint formulation and the Shannon algorithm after 50 features have been selected, though this happens at the higher end of the α range, which is uncommon in the rest of the results. The joint formulation outperforms the Shannon algorithm when using 60 - 80 features, but only equals the best classification performance of the Shannon algorithm when it does so, in contrast to the divergence formulation outperforming the Shannon algorithm's best by 3 percentage points.

Lymphoma Dataset

Figure 6.22 presents the results from the SVM classifier with $0.1 \leq \alpha \leq 0.5$. In the results the divergence formulation has less variance around the Shannon algorithm, with performance approximately the same, and achieving the same best classification error. The joint formulation has much more variance in the results with $\alpha = 0.1, 0.3$ providing the best classification results, and achieving a minimal classification error.

Figure 6.23 presents the results from the SVM classifier with $0.6 \leq \alpha \leq 1.1$. In the results both formulations perform equivalently to the Shannon algorithm, with a slightly greater variance in the results of the joint formulation. This leads to the joint formulation gaining the optimal classification accuracy before converging with the Shannon algorithm.

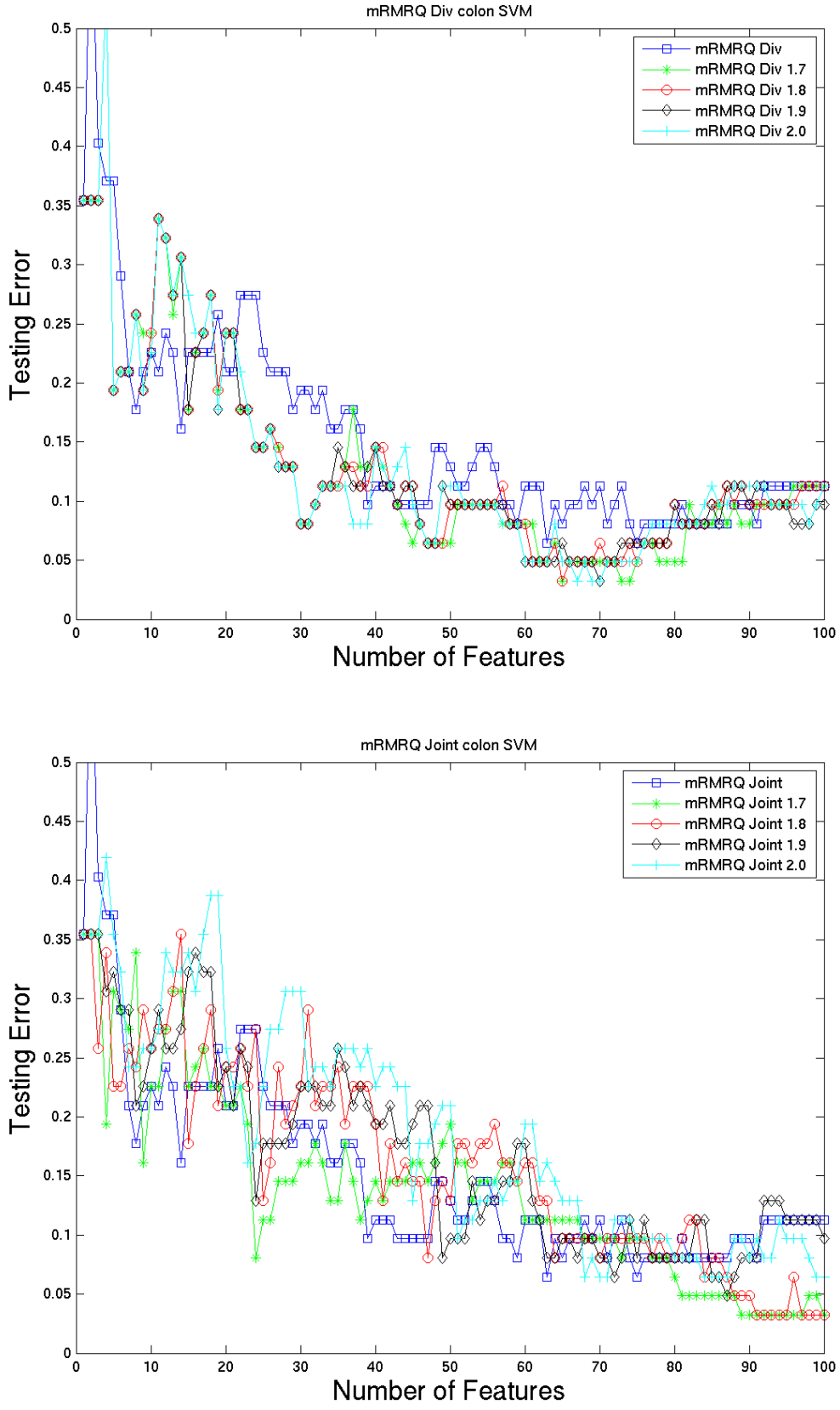

Figure 6.19: mRMR-Q Joint and Divergence, Colon Dataset, SVM Classifier, $1.7 \leq \alpha \leq 2.0$

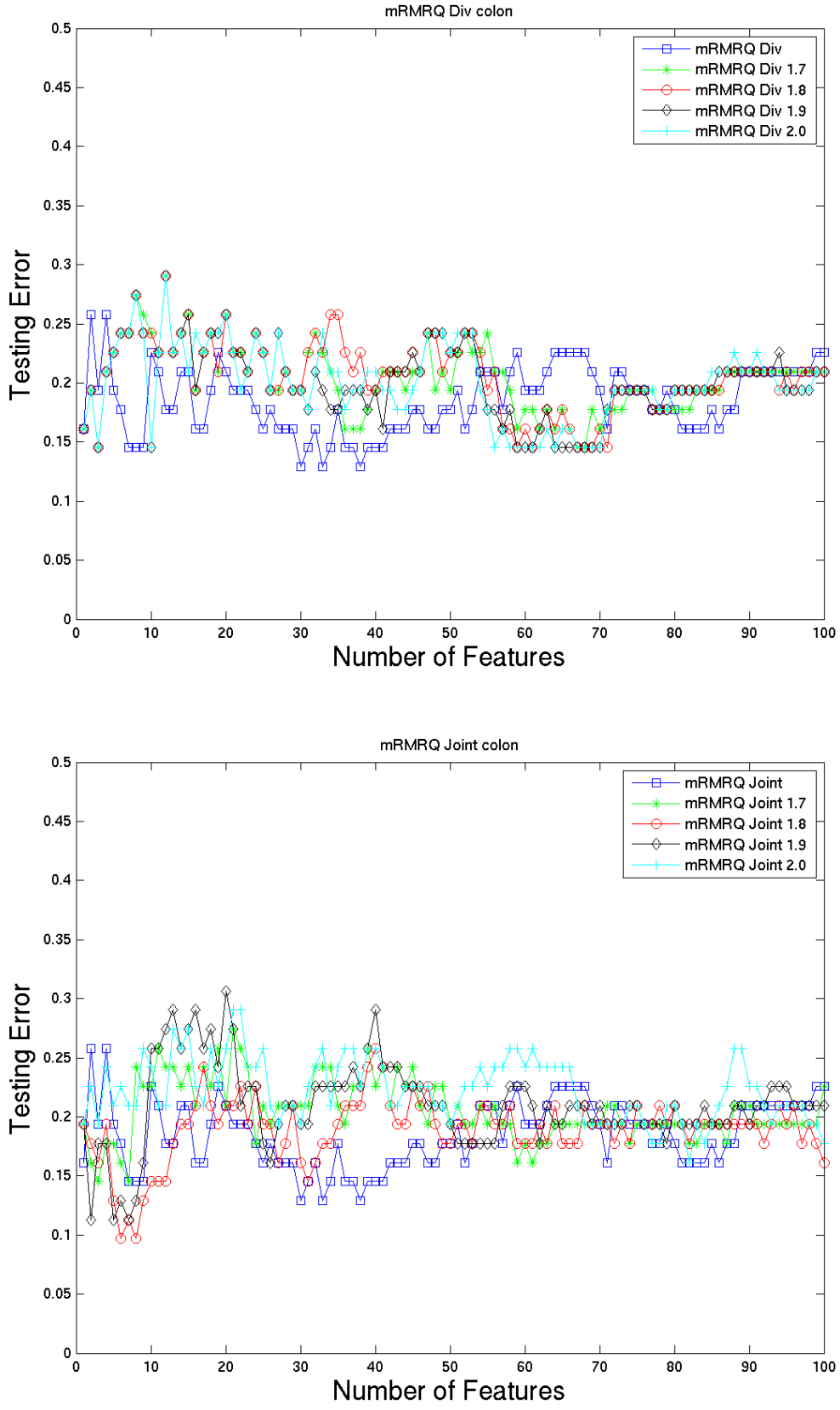

Figure 6.20: mRMR-Q Joint and Divergence, Colon Dataset, 3-NN Classifier, $1.7 \leq \alpha \leq 2.0$

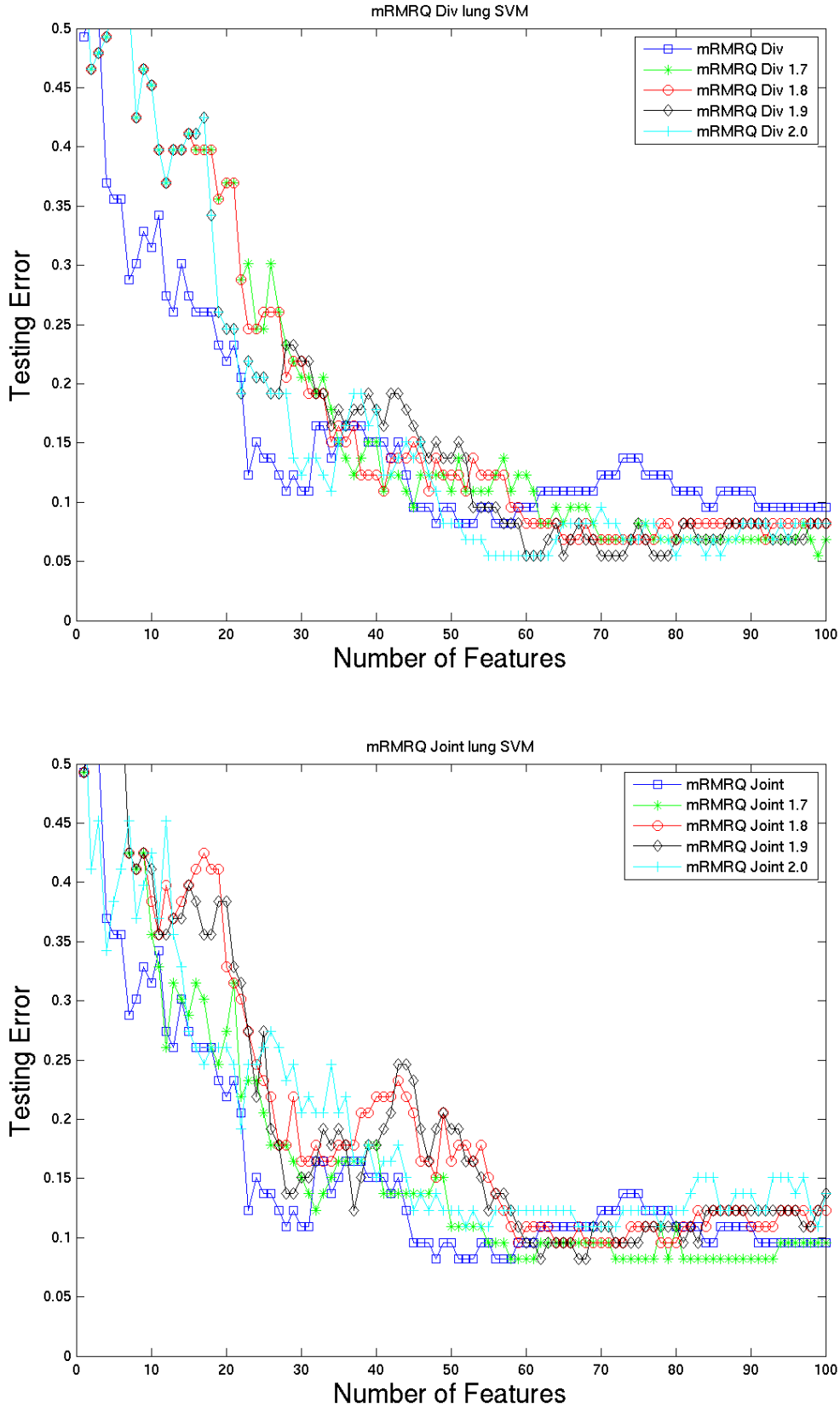

Figure 6.21: mRMR-Q Joint and Divergence, Lung Dataset, SVM Classifier, $1.7 \leq \alpha \leq 2.0$

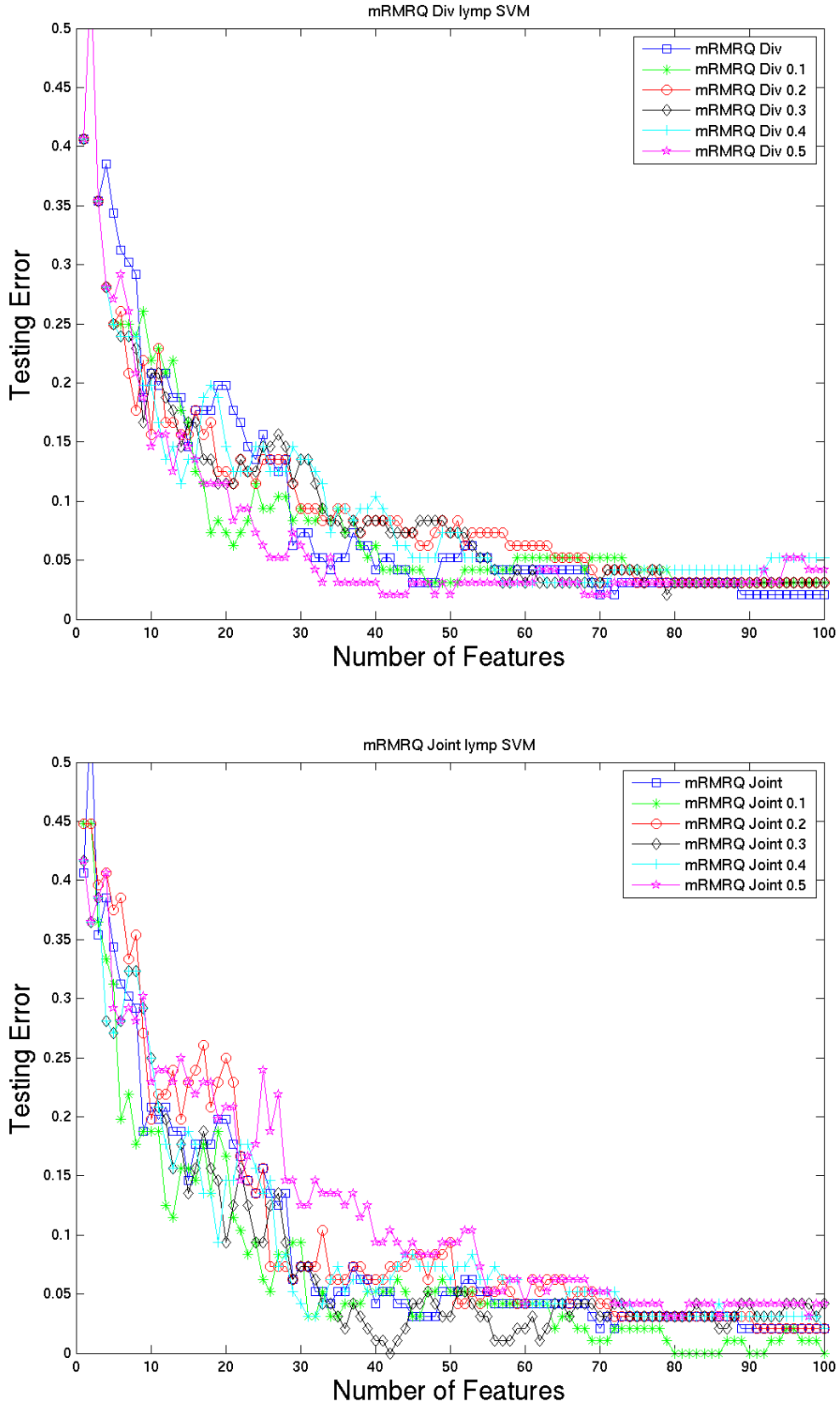

Figure 6.22: mRMR-Q Joint and Divergence, Lymphoma Dataset, SVM Classifier, $0.1 \leq \alpha \leq 0.5$

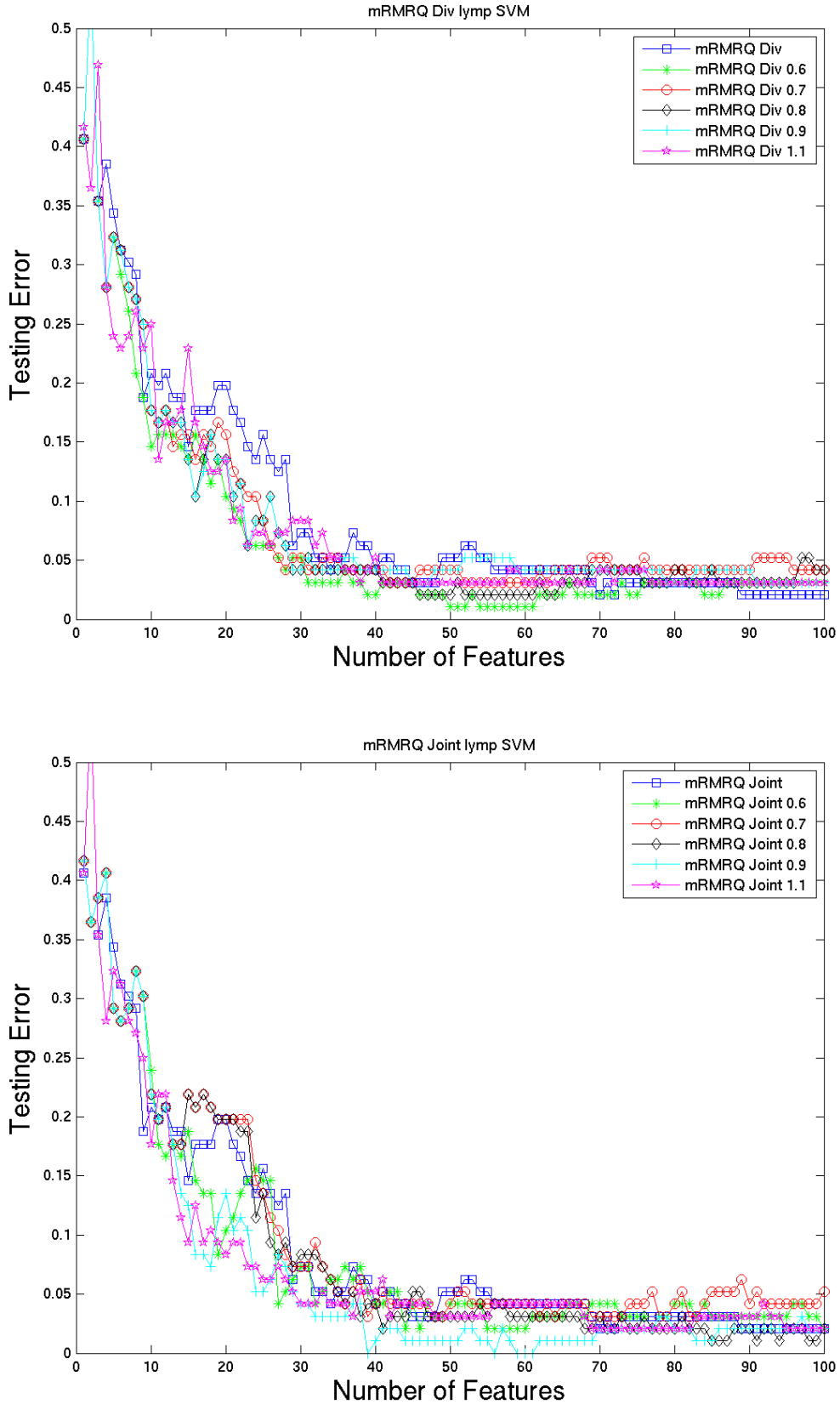

Figure 6.23: mRMR-Q Joint and Divergence, Lymphoma Dataset, SVM Classifier, $0.6 \leq \alpha \leq 1.1$

6.6 Analysis

6.6.1 Analysis of CMIM variations

The Rényi version of the CMIM algorithm can only use the divergence formulation as the others use the invalid Rényi conditional entropy. Therefore the only parameter varied in the CMIM testing was the α parameter. The varying of this parameter leads to some improvement over the standard Shannon method of measuring the mutual information, but it is dataset dependent and does not provide a measurable improvement over all of the tested datasets. Over all the datasets the extremes of the α parameter provided little improvement over the Shannon algorithm, though the region around $\alpha = 1$ provides similar performance to the Shannon algorithm with occasional improvements to the classification error.

6.6.2 Analysis of DISR variations

The Rényi version of the DISR algorithm can use both the joint and divergence formulations of the Rényi mutual information. Therefore there are two parameters to be varied in the DISR testing, both the α parameter and the formulation of the mutual information. The change to the Rényi mutual information does not result in any noticeable improvements to the classification performance of the feature sets produced by the DISR algorithm, though the region around $\alpha = 1$ still provides similar and occasionally better performance compared with the Shannon version of the algorithm. The joint formulation of the mutual information is more variable at either end of the tested range using this algorithm, with classification errors 25 percentage points greater than the Shannon algorithm for $\alpha < 0.3$, and up to 10 percentage points greater for $1.7 \leq \alpha \leq 2.0$.

6.6.3 Analysis of mRMR-D variations

The Rényi version of the mRMR-D algorithm can use both the joint and divergence formulations of the Rényi mutual information. Therefore there are two parameters varied in the testing of the Rényi mRMR-D algorithm. The results using this algorithm show the joint formulation has more variance than the divergence formulation, ranging above and below the classification errors of the Shannon algorithm, and thus can be an improvement upon the standard algorithm. For $\alpha < 0.5$ the joint formulation performs better than the Shannon algorithm on the datasets though this is only for the SVM classifier. The feature sets generated by the Rényi mRMR-D algorithm using the joint formulation do not perform well with the 3-NN classifier, and only have a performance increase when using the SVM.

The region where $1.2 \leq \alpha \leq 1.6$ gives a better performance from both the formulations, but $\alpha > 1.6$ degrades the performance of the joint formulation.

6.6.4 Analysis of mRMR-Q variations

The Rényi version of the mRMR-Q algorithm is fundamentally the same as the mRMR-D algorithm and thus the same testing constraints apply. Due to the similarity between the mRMR-D and mRMR-Q criteria the variation of the α parameter and changing the mutual information formulation provides similar results to the mRMR-D criterion.

6.7 Conclusion

The results from the empirical testing of the Rényi mutual information, and the algorithms constructed using it show that in general there is a value of α which can improve the classification result, with both formulations of the mutual information. The lack of a clear value or formulation that consistently provides the best result means that the value of α and the formulation must be added as a layer of model inference to the system, or an arbitrary value chosen for comparison of the work. The problem with adding a layer of inference (as must be done if a highly performing feature set is to be chosen by picking from among several different algorithms) is that it causes more complexity in the model of the data which leads to over-fitting of the problem, especially in low sample fields such as genomics. In general the performance of the divergence formulation mirrors or slightly exceeds the performance of the Shannon algorithm, and the joint formulation provides a more variable level of performance around the Shannon-based algorithms, dictated by the α value, the dataset and the classifier used.

A finer grained search of the region $0 < \alpha \leq 2.0$ could be performed to aid the estimation of which values of the α parameter improve upon the Shannon algorithms, but this would still be a dataset dependent problem, unless a wider ranging study of more data was performed.

6.7.1 Summary of the work

This chapter has detailed an investigation the properties of the Rényi entropy, and the various Rényi mutual informations that can be constructed. It has:

- Constructed 2 different methods for measuring the mutual information with the Rényi entropies and divergences.

- Investigated why the conditional formulation of the mutual information is invalid with the Rényi entropy.
- Modified the feature selection algorithms investigated previously to use the Rényi mutual informations.
- Empirically tested these algorithms against the standard method, using a variety of datasets and classifiers, and varying the α parameter.
- Analysed the test results, and concluded that the Rényi mutual informations can provide an improvement to classification accuracy when using selected values of the α parameter.
- Concluded that of the two different formulations of the Rényi mutual information, *the divergence formulation provides an improvement in classification accuracy over a wider range of α values than the joint formulation.*

Chapter 7

Graph-based Entropy Estimation

7.1 Introduction

7.2 Graph-based Entropy

7.2.1 Image Registration

Image registration is the process of identifying a transform that maps similar areas of two different images onto each other. It is used for creating a standard co-ordinate system for working with multiple images of the same area taken from similar non-identical viewpoints. The work by Neemuchwala et al. [14] is concerned with image registration problems, rather than strict feature selection. However it develops a series of interesting information theoretic constructs based around the Rényi entropy, which are used to determine the precise transformation for registering images.

The paper [14] provides a review of the previous work carried out by the authors in the image registration field, using several novel information theoretic techniques.

7.2.2 Graph-based entropy estimation

This paper presents a method for calculating the Rényi entropy of a dataset as a whole, in one calculation that scales with the number of samples in the dataset, rather than the dimensionality of the dataspace. It does this through work on entropic graphs, that is graphs which are related to the entropy level of the dataset. It shows that depending on the graph construction, an accurate estimation of the continuous Rényi entropy can be found, through calculation on the lengths of the edges of the graph. These graphs are mathematical

constructs, which are un-directed and exist as a set of points and edges which connect the points, and where the edges are weighted according to their length raised to a power γ.

Several different kinds of graph are proposed for this, but the type of graph used is the Minimal Spanning Tree (MST). The MST is a graph with a minimal summed edge weight which connects all the points in the space, and contains no closed cycles (so the graph can be represented as a tree with a root node), see figure 7.1, in which the black edges form the minimal spanning tree of the set of points, with the gray edges being unused edges from the set of available edges. In previous work the authors have proved that this forms an accurate estimate of the Rényi entropy [7].

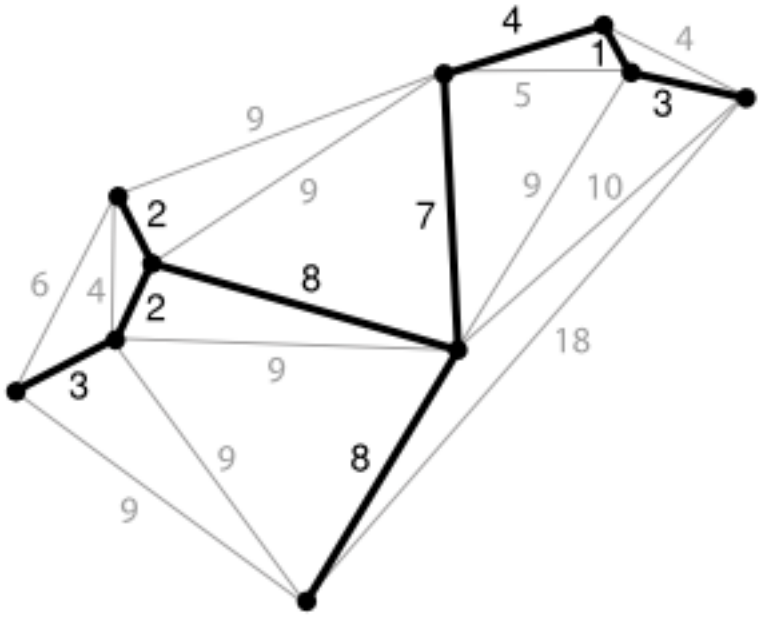

Figure 7.1: The Minimal Spanning Tree of a set of connected points, image sourced from http://en.wikipedia.org/wiki/Image:Minimum_spanning_tree.svg, retrieved on: 04-09-2008

In equation (7.1) Ω is the set of all graphs with the required properties (in this case all spanning trees), s is the euclidean distance between the datapoints in X, i is an iterator over all the edges in E, and γ is the edge exponent, ranging $0 < \gamma < d$ where d is the dimension of the feature space. This equation specifies the construction of the minimal graph, and provides an equation for its power-weighted sum $L_\gamma(X)$. $L_\gamma(X)$ is thus a function with value equal to the summation of the edges raised to the power γ of the minimal graph that fits the criterion, i.e. $L_\gamma(X)$ equals the summation of the MST's edge lengths (in figure 7.1 this is equivalent to the sum of the lengths of the bold black lines).

$$L_\gamma(X) = \min_{E \in \Omega} \sum_{i \in E} s_i^\gamma \tag{7.1}$$

In equation (7.2) $\alpha = 1 - \frac{\gamma}{d}$ and $\beta_{d,\gamma}$ is a constant independent of the data, depending only on the type of graph and n is the number of datapoints in X. This gives a value for α ranging $0 < \alpha < 1$.

$$\lim_{n \to \infty} \frac{L_\gamma(X)}{n^\alpha} = \beta_{d,\gamma} \int f^\alpha(x) dx \tag{7.2}$$

In equation (7.3) the equation for the Rényi entropy has been integrated with equation (7.2) to form an expression for the Rényi entropy in terms of the power-weighted sum of the edge lengths, with $\hat{H}_\alpha(f)$ denoting the estimation for the Rényi entropy. This expression is proved in [7] to be a strongly consistent estimator for the Rényi entropy of order α. This framework then provides a Rényi entropy estimator for $0 < \alpha < 1$ with the parameter γ being adjustable to generate different α values.

$$\hat{H}_\alpha(f) + \frac{1}{1-\alpha} \log(\beta_{d,\gamma}) = \frac{1}{1-\alpha} \log(\frac{L_\gamma(X)}{n^\alpha}) \tag{7.3}$$

This can be intuited by thinking of the Rényi continuous entropy, and what it measures. The Rényi continuous entropy is a measure of the variability of a variable, it takes high values for a wide variation in the data, and a low value for a low variation in the data. The graph linking all the points in the variable will have a similarly high value if there is a large variation in the data as there will be a number of long links between distant points, and will have a low value if there is a small variation in the data as most of the points will be close together. The equations derived give a formalisation of this link.

7.2.3 Conclusion

With the new method of estimating entropy, the paper goes on to detail a method for constructing and estimating the mutual information between two continuous spaces. This definition is not useful for the classification task as it uses an extremely large continuous data sample to provide the information and classification tasks are mainly mapping discrete data onto a discrete class. Then they apply the two derived constructs to the task of image registration. This formulation of the Rényi entropy estimation has a computational complexity which scales with the number of samples in the dataset, rather than the dimensionality of the dataset, and thus provides a framework for doing large scale analyses in a computationally tractable amount of time, similar to the Maximum Dependency feature selection concept in

[17].

7.2.4 A note on the continuous entropy

The equations give a value for the Rényi continuous entropy, however all the previous work in this paper has been using the discrete entropies. The continuous entropy can be used in a similar way to the discrete entropy, forming mutual informations and other values (in the Shannon space). However it does give much larger values when it is used to calculate a discrete variable. Also the continuous entropy is distorted when the discrete values have non-sequential values, e.g. if a variable takes the values $\{1, 2, 3\}$ then it will have a lower continuous entropy than if the variable took the values $\{1, 4, 9\}$ though this would give the same discrete entropy. This issue only causes a problem if the feature vectors have different possible values, however in all the test data they take the same range of values.

7.3 Graph-based Estimation for Feature Selection

Once above framework had been created, then it was applied to feature selection by Bonev et al. in [1]. It details an implementation of the graph-based Rényi entropy estimator, used to calculate the Maximum Dependency criterion from [17]. It goes on to state the comparative computational complexities between mRMR ($\mathrm{O}(n^3)$ where n is the dimension of the feature space) and the new algorithm ($\mathrm{O}(n^2)$) though it must be noted that the entropy estimation has its own computational complexity which scales as $\mathrm{O}(s \log(s))$ where s is the number of samples.

The algorithm reuses the Maximum Dependency criterion from [17] (equation (2.16)). It is made computationally tractable by the different method of estimating the Rényi entropy which is used to replace the Shannon entropy given in the original formulation. As the Rényi entropy tends towards the Shannon entropy as $\alpha \to 1$, a series of graph-based estimations are performed to capture the behaviour of the function in the region around 1. This is then used to construct an estimate for the Shannon entropy, which is then used to construct the mutual information.

$$X_{\mathrm{Graph}} = \arg \max_{X_n \in X_{/S}} I(X_S \bigcup X_n; Y) \tag{7.4}$$

The mutual information is constructed by using the conditional formulation, (i.e. $I(X;Y) = H(X) - H(X|Y)$), so an accurate method for constructing the conditional Rényi entropy is required. Simple conditioning of the Rényi entropy will provide a value that tends towards

the Shannon conditional entropy as $\alpha \to 1$, so this should give a valid estimate for the Shannon mutual information, though as been previously seen this value has little meaning when working solely with Rényi entropies.

7.4 Creating the Algorithm

For this work it was decided to investigate the properties of the Graph-based entropy estimator when using it solely with the Rényi entropy rather than using it to approximate the Shannon entropy.

This changes the formulation as it is based upon the Rényi entropy rather than the Shannon entropy, and the Rényi entropy has weaker additivity properties, though the previous research (along with [5]) indicates that the Rényi entropy is a better measure for quantifying useful information in some cases.

For forward selection this creates additional problems, as the estimate of the bias factor which constrains the graph value into an entropy is only valid when using in a high dimensional feature space. This means that the initial stages of forward selection are subject to a much greater bias, and are thus less reliable choices than the later stages. Other approximations can be found by generating graphs in a $[0, 1]^d$ space and calculating their lengths, then using this value to estimate the bias.

The bias estimator was chosen to be the large dimension approximation used by Neemuchwala et al. for reasons of simplicity, despite an approximation problem with the initial stages of forward selection. The equation is given in (7.5).

$$\beta_{\gamma,d} = \frac{\gamma}{2} \ln(\frac{d}{2\pi e}) \tag{7.5}$$

As the graph-based entropy estimator is only valid when the dimension of the space is greater than 1, a different function must be used to select the initial feature. In this work it was chosen to be the divergence formulation of the Rényi mutual information, with the same α parameter as the graph-based estimator. This was decided before the investigation into the different formulations of the Rényi mutual information, and the divergence method was chosen as it is the simplest derivation from the original work by Rényi in [18].

Due to the inability to accurately construct the divergence measure used by Neemuchwala et al. in a discrete space, the mutual information has to be constructed from the various entropies that can be calculated using the graph estimator. As the conditional formulation is ill-founded with the Rényi entropy, the joint formulation (equation (7.6)) was chosen.

$$I_\alpha(X;Y) = H_\alpha(X) + H_\alpha(Y) - H_\alpha(XY) \tag{7.6}$$

This raises further problems, as the entropy estimator used for the earlier Rényi work was a discrete estimator, and the graph-based estimator treats it as a continuous space thus providing a much higher estimate for the entropy of the tristate variables used in testing. This causes the $H_\alpha(X)$ and $H_\alpha(XY)$ terms which are estimated using the graph-based method to dominate the $H_\alpha(Y)$ term, which is a one dimensional vector of the classes, and is calculated using a discrete Rényi entropy estimator. This leads to a bad feature selection algorithm, as it doesn't prioritise information on the class. Instead it chooses the highest entropy features, leading to poor classification performance.

Once the problems with the joint formulation were noticed, the conditional formulation (equation (7.7)) was used for the testing of the algorithm, as it provides an approximation of the mutual information, even though it is ill-founded, as both terms can be calculated using the graph-based estimator. The conditional entropy was calculated by simple conditioning on the graph-based estimator. This formulation was found to select features that were correlated with the class and increase classification performance over the joint formulation. The resultant feature selection algorithm is termed Graph-Based Forward Search (GBFS) for the remainder of this document.

$$I_\alpha(X;Y) = H_\alpha(X) - H_\alpha(X|Y) \tag{7.7}$$

7.4.1 Usefulness of this measure

There are two main problems with the use of the graph-based mutual information that has been constructed.

Firstly, the bias is a bad approximation for low dimensional spaces, so in the early stages of forward selection the mutual information estimate is wrong. As the bias is constant throughout a particular iteration it doesn't affect the performance of the algorithm, though it does cause negative mutual information values which have to be compensated for in the algorithm.

Secondly, the use of the conditional formulation of the mutual information is ill founded with the Rényi entropy. It varies with the correlation between two variables, but it is unknown what factors influence this value as it is not a true mutual information.

7.5 Rényi Entropy Genetic Algorithm

A further extension to the graph-based feature selector is proposed, as an alternative to a simple forward search.

The majority of feature selection techniques use variations on greedy search strategies, generally using a greedy forward search as the basis for the algorithm. The reason for this is twofold: firstly the algorithm is simple to construct, and secondly, there is no accurate way of estimating the total mutual information of a set of variables upon a class. The graph-based entropy estimator provides a solution to the second problem, as now the mutual information of a set of variables can be calculated in an efficient and accurate manner. As the aim of feature selection is to provide a set of features which contain the maximum information about a class, this function can be maximised over the whole feature space to provide an optimal feature set. The search space for such a maximisation is exponential with respect to the number of features, and thus an exhaustive search is not possible in most cases.

One such algorithm for searching such a wide search space is a genetic algorithm, which is based upon evolutionary principles applied to a population of competing solutions.

7.5.1 What is a Genetic Algorithm

Genetic algorithms consist of a population of solutions, with each solution described as a string of numbers or letters and an fitness function which can measure the performance of a particular string [2]. Additionally there are two operations that can be performed on the strings, mutation and crossover. Mutation replaces a single value from the string, with another possible value selected at random. Crossover takes two members of the population and produces a new member of the population with a percentage of its string copied from one string, and the remainder copied from the other string. The algorithm then iterates over a number of generations. At each generation each member of the population is evaluated, with a percentage of the low scoring members being removed. New members of the population are then generated using crossover from the existing high scoring members. Then a percentage of the population is mutated, to generate new (hopefully improved) solutions. This process is a form of stochastic hill climber, with each generation producing an improvement in the average performance of the population.

7.5.2 Why use a Genetic Algorithm

Greedy forward searches have a problem with highly interactive systems, such as feature selection. They assume that the addition of the current optimum to the solution only improves the solution, whereas in feature selection adding a feature to a set can lower the information contained in the set (see chapter 2). Also the three algorithms chosen for study proceed incrementally, adding one feature at a time to the solution set. This doesn't take into account complementarity of features in a useful way, as it can only cope with at best two way complementarity (i.e. two features combining to increase the information). There is no method for coping with adding together useful sets of features without using techniques such as boosting, or ensemble methods.

The genetic algorithm doesn't have this problem. By mutating and crossing over solutions it can generate solution sets where many features have changed in one generation, and can so test for interactions between large groups of features. As the process is optimising with respect to the evaluation function this process will slowly provide feature sets with a higher mutual information, which greedy forward search cannot find as they involve making non-optimal choices for a long term benefit. The levels of crossover and mutation have to be carefully chosen to ensure the system doesn't converge to a local optima whilst ignoring all other optimum solutions, as the populations converge. An increase in the level of mutation helps to increase the variability of solutions, to enable the algorithm to cover more of the search space.

7.5.3 Constructing the Genetic Algorithm

The version of the genetic algorithm used in the testing implements a subset of the features described above due to time constraints. It implements mutation, and the culling & repopulation parts of the algorithm, but does not contain a crossover operator. The algorithm also has a fixed mutation rate, where each feature in the solution has a 5% chance of being mutated into another feature. Additionally each solution has the same fixed length, there is no provision for growing or shrinking the number of features in a given solution.

Figure 7.2 contains the pseudocode algorithm for the Graph entropy genetic algorithm.

The implementation given for the genetic algorithm is relatively simple to test the validity of the graph-based mutual information as a measure of the performance of the set of selected features. This algorithm will be referred to as the Graph-Based Genetic Algorithm (GBGA) to differentiate it from the forward search method constructed above. It performs $g*p$ mutual information calculations, where g is the number of generations and p is the population size,

```
randomly generate population of solutions, of size popSize

test the initial fitness of the solutions

for i = 0 : numberOfGenerations
{
  copy out the top 5 solutions

  replace the population with a new population,
  taken from the top 5 solutions

  mutate each solution
  {
    for j = 0 : solutionLength
    {
      generate random number between 0 and 1
      if number < 0.05
      {
        swap that feature for an unused one
      }
    } - loop over solution length
  } - mutate function

  recheck the fitness of the solutions
} - loop over generations

return the top 5 solutions
```

Figure 7.2: Graph MI Genetic Algorithm Pseudocode

in comparison to the forward search, which performs $k * n$ mutual information calculations where k is the number of desired features and n is the total number of features. It enables a better coverage of the search space than the forward selector with a smaller number of mutual information calculations, as each calculation in the forward search differs by only 1 feature, whereas in the genetic algorithm each calculation could differ by the string length.

7.6 Results

The results for the two graph-based feature selection algorithms have been combined, along with comparison results taken from the Rényi mutual information based versions of the

4 feature selection algorithms used in the remainder of this research. The Rényi mutual information chosen was the divergence formulation as this is valid over all the different algorithms. All results were generated with $\alpha = 0.9$, for a more detailed investigation into the properties of the α parameter see chapter 6. The way of analysing the results used throughout the paper is not optimal for analysing the performance of the genetic algorithm, as it does not do a forward search of the space, and is thus not searching for incrementally better feature sets. Therefore testing the performance of progressively larger feature sets does not relate to the performance of the whole feature set, as the ordering of the features (in the selected set) is unrelated to their performance. The only valid testing result for the genetic algorithm is therefore the final result, i.e. the classification error when using all 100 selected features, as this is the value that is being optimised in the genetic algorithm.

7.6.1 Leukaemia Dataset

Figure 7.3 contains the results of the 6 algorithms compared when using the 3-NN and SVM classifiers.

7.6.2 Lung Dataset

Figure 7.4 contains the results of the 6 algorithms compared when using the 3-NN and SVM classifiers.

7.6.3 Execution Speed

Table 7.1 lists the execution time for the MATLAB/C++ implementations of the algorithms several datasets, when executed on an Athlon XP 3000+, with 1GB of RAM, and selecting the 100 features used in the above test results. The GBGA would not terminate on the Adenocarcinoma dataset, and so no time is recorded for that algorithm.

Dataset	CMIM	DISR	mRMR-D	GBFS	GBGA
Adenocarcinoma (41672 features, 59 samples)	1.0400	96.8500	46.0300	22050	x
Lung (325 features, 73 samples)	0.3500	1.3700	0.5300	224	3990.3
NCI9 (9712 features, 60 samples)	0.6000	43.0600	13.0900	3827.0	2195.4

Table 7.1: Rényi algorithm execution time (s)

The results given in the table combine both the mRMR variants into one score, as they provide almost the same execution time (see Table 4.1), and the time taken for the mRMR-Q algorithm was thus not recorded.

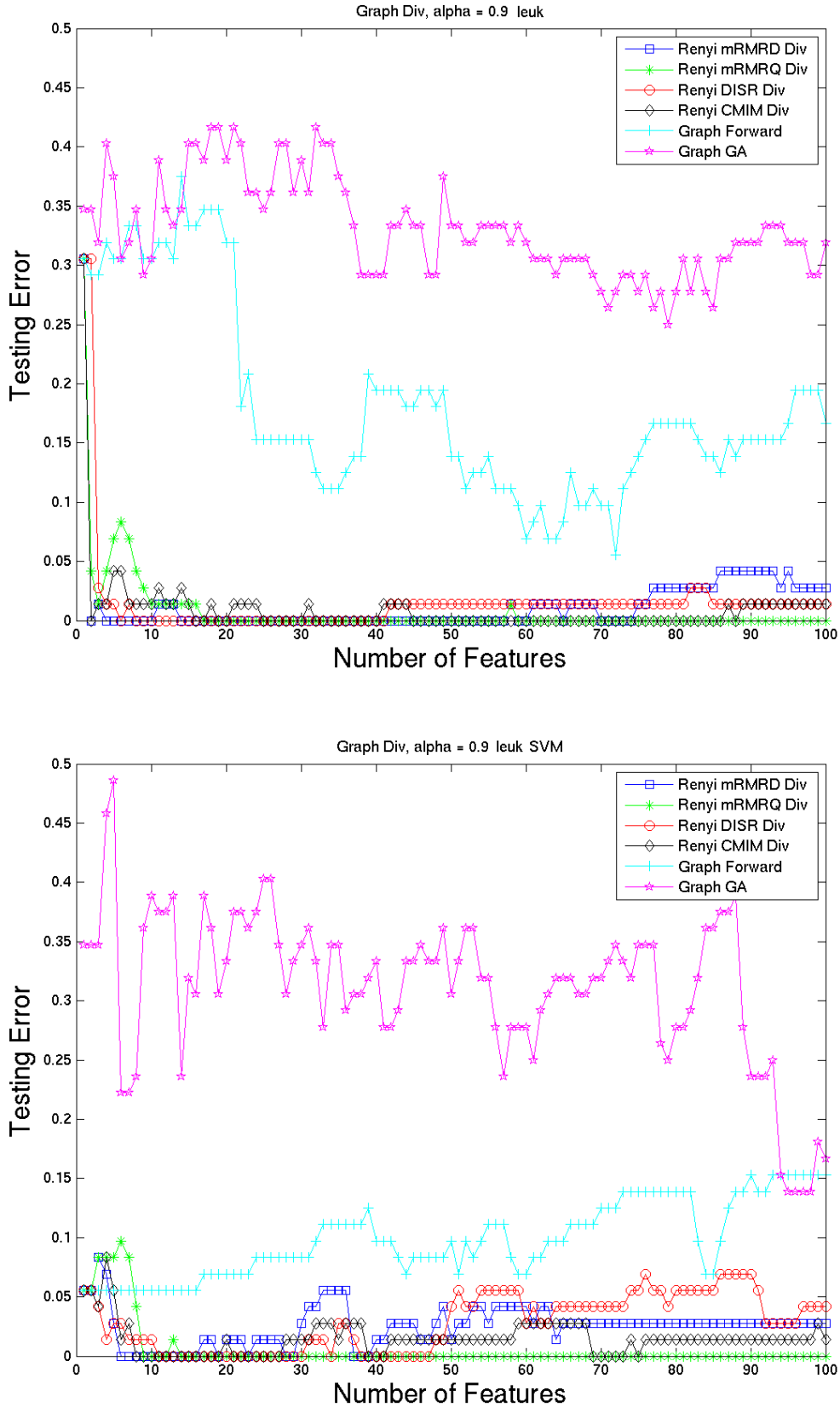

Figure 7.3: Leukaemia Dataset, 3-NN & SVM Classifier, $\alpha = 0.9$

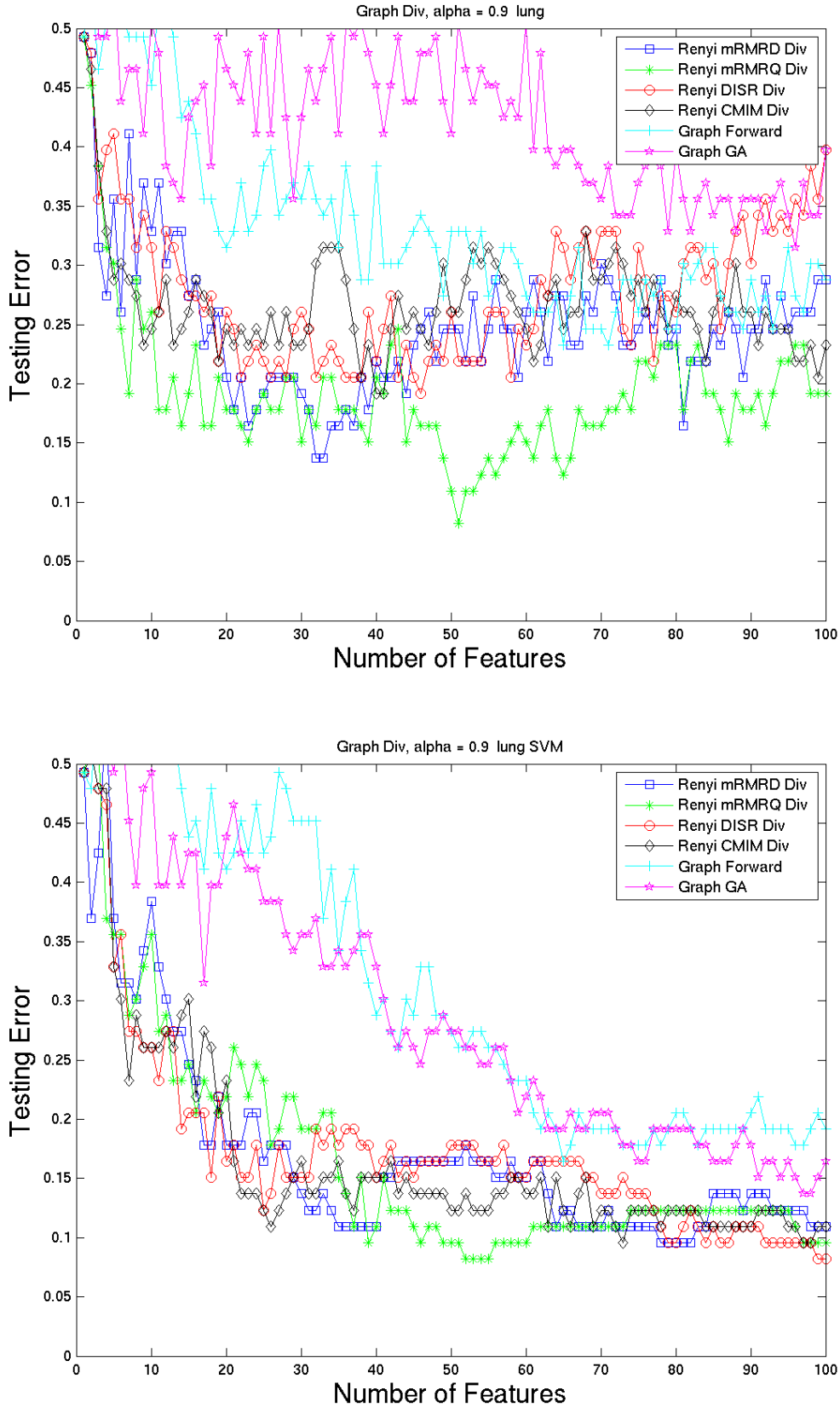

Figure 7.4: Lung Dataset, 3-NN & SVM Classifier, $\alpha = 0.9$

The graph-based methods result in a massive increase in execution time, over the standard methods recoded for the Rényi mutual information. This is due to a large factor in the calculation of the graph-based entropy, as the euclidean distance between each sample must be calculated, and this causes a large number of floating point operations per sample which increases as the dimensionality of the space increases. This factor is constant when using the GBGA as the dimensionality of the space is constant throughout execution, and the time difference within that algorithm is due to internal factors in the genetic algorithm, and due to differences in the number of samples.

7.7 Analysis

The results generated for both the graph-based feature selection algorithms are disappointing, with neither algorithm outperforming the standard techniques, and requiring vastly more computation and execution time to generate a result. The execution time of the genetic algorithm scales better with the number of samples than the forward selector, and its execution time can be constrained by the number of generations, the population size and the length of the returned solution.

7.7.1 Leukaemia Dataset

In the results both the graph-based feature selection algorithms have a much higher classification error than the standard techniques, with the genetic algorithm having the highest classification error. When using the SVM classifier the GBFS algorithm does provide performance approximately equivalent to the later stages of the Rényi DISR variant, but as the leukaemia dataset is a relatively simple system to provide a consistent perfect solution, both of the error rates are far above the optimum. It can also be seen that the final classification errors of the GBGA and the GBFS are very similar, differing by only 1.3 percentage points when using the SVM classifier.

When using the 3-NN classifier the difference between the graph-based methods and the standard methods is more stark, with the graph-based methods consistently outperformed by a greater margin than under the SVM classifier. This provides a good demonstration of the differences between the two classifiers, with the same feature set providing a classification accuracy of 31% in the 3-NN classifier, and 16% in the SVM classifier.

7.7.2 Lung Dataset

In this dataset the differences between the different methods of feature selection were less distinct. Using the 3-NN classifier the GBFS algorithm performs approximately equivalent to the Rényi algorithms, bar the mRMR-Q algorithm, after selecting 50 features. The final performance of the GBGA algorithm is approximately similar to the performance of the Rényi DISR algorithm, indicating that the genetic algorithm was beginning to converge towards the optimal solution.

Using the SVM classifier the results are more distinct than the 3-NN classifier. There are two bands of results, with the standard methods forming one lower band, and the two graph-based methods forming a band with a higher classification error. Additionally the GBGA finds a better performing feature set than the GBFS algorithm, when selecting 100 features.

7.7.3 Summary

The two graph-based methods do not currently provide a better method for searching the feature space, when compared with the standard algorithms changed to use the Rényi mutual information. Additionally the genetic algorithm variant is outperformed by the forward search with all datasets bar the Lung dataset, and only when using the SVM classifier. This indicates a problem with the construction of the graph-based entropy estimator as it should provide an improvement in the classification accuracy, as the forward search is an implementation of the Max Dependency criterion from the mRMR work in [17] and this is an optimal criterion for selecting informative features.

7.8 Conclusions

This investigation into graph-based methods of estimating the entropy has empirically tested the current method against a variety of other Rényi based algorithms, and created a new method of searching the feature space using the graph-based entropy estimator as the evaluation function for a genetic algorithm. The current implementation of the graph-based mutual information requires improvement, both to modify the bias function to ensure an accurate estimate, and to develop a continuous Rényi entropy estimator, so the mutual information can move to the joint formulation over the current ill-founded conditional formulation.

7.8.1 Summary of the work

This chapter has detailed an investigation into the graph-based entropy estimator. It has:

- Constructed a Max Dependency forward search using the estimator.
- Developed a new method for searching the feature space using the graph-based entropy estimator as a fitness function for a genetic algorithm.
- Empirically tested these algorithms against the standard feature selection algorithms (modified to work with the Rényi mutual information), using a variety of datasets and classifiers.
- Analysed the test results, and concluded that the current implementation of the genetic algorithm requires improvement before it can be used extensively.

Chapter 8

Conclusions

This document presents a wide ranging look into various different information theoretic feature selection techniques. Several different areas have been explored, encompassing different methods of searching the feature space, down to the optimisation of the selection of the first feature in the standard algorithms.

8.1 Summary of the research

8.1.1 First Feature Selection

The development of the first feature selectors shows an improvement in classification accuracy, and can be used to answer the question *"Is the feature with the highest mutual information the best choice?"*, to which the answer is no. The different methods of selecting the first feature can be tailored to the algorithm used, to improve the classification accuracy. Also the selectors provide a method for finding the most important feature or set of features, which carries the most information not contained in the remaining features.

8.1.2 Rényi Mutual Information

The investigation into the Rényi mutual information has shown there are various competing method for constructing the mutual information using the Rényi entropy, and that in general the best performing formulation is the divergence formulation, though this is data dependent. Additionally it has shown that the feature selection algorithms can be constructed with the Rényi mutual information, and they still provide a good framework for selecting informative features. The investigation into the properties of the α parameter shows that there are values of this parameter which substantially improve upon the use of the Shannon

mutual information, though the value is dependent upon the choice of mutual information formulation. The work on the Rényi mutual information raises further scope for investigating the dataset dependent properties of the α parameter, and whether this can be tailored to provide a better result for different kinds of dataset, with varying numbers of classes, features and samples.

8.1.3 Graph-based Entropy Estimation

The investigation into the Graph-based Entropy Estimator was inconclusive, with the constructed algorithm failing to reproduce the performance level described in [1]. In that work the estimator was extended to estimate the Shannon entropy through repeated runs to capture its behaviour as $\alpha \to 1$. In this work it can be seen that the estimator fails to perform as well when $\alpha \neq 1$ though this may in part be due to the properties of the implemented algorithm and bias function. Additionally it is recognised that the conditional formulation of the mutual information is invalid in the Rényi space, and this requires the creation of an estimator for the continuous Rényi entropy, to create the joint formulation for the mutual information.

Once the Graph-based mutual information is constructed it was realised that because it enabled the estimation of the mutual information of a multivariate feature set it could be used to form the basis of an evaluation function. This leads to the construction of a genetic algorithm to search the feature space, which provides numerous advantages over the standard greedy forward search employed in the rest of the information theoretic feature selection algorithms. The further development of this algorithm coupled with the refinement of the graph-based estimator should improve classification accuracy beyond the level attainable with a greedy forward search, due to the way the genetic algorithm searches the space.

8.2 Critique Of The Graph-based Entropy Estimation

The use of the flawed conditional form of the Rényi mutual information means the resultant calculations for the estimation of the mutual information are distorted in a non-parametric way. This makes it difficult to compare the results of the Rényi algorithms with the graph-based estimators. Additionally the choice of the bias function used in the work is ill-founded because it does not provide an accurate estimate for the bias in the graph length for small numbers of dimensions, leading to a distortion in the forward search algorithm. This problem is removed when using the genetic algorithm provided the string length is high, as it provides a progressively better estimate for higher dimensional spaces.

The implementation of the genetic algorithm used has a number of different flaws. Firstly the fixed relatively high rate of mutation means that the algorithm will move away from all optima, as there is no method for preserving an exact copy of a highly scoring solution for comparison with later generations. Secondly the lack of crossover forces a high rate of mutation to ensure that the algorithm doesn't find one local optima of the search space, and only generate solutions near to that optima.

8.3 Further Investigation

There are several areas for further investigation in each of the three topics discussed in this document, and a set of general observations to be applied to all of the topics covered.

8.3.1 General Observations

- All of the experiments were carried out using gene expression datasets, with high numbers of features and low numbers of examples, so an empirical study using datasets with lower numbers of features and higher numbers of examples would test if the conclusions drawn still hold.
- An investigation using different classifiers, including a 1-NN classifier (which removes the non-deterministic problem with the 3-NN), different kernels for the SVM, and the testing of a Naive Bayes classifier (as this is a common classifier and would enable the work to be compared with other research more effectively).

8.3.2 Investigation into First Feature Selection

- Further work on developing new first feature selectors, looking into selecting independent features using the mRMR framework.
- Investigating if the independence or combination route is the best for selecting the first feature, and explicitly constructing a framework for testing for the independence or combination of a particular selector.

8.3.3 Investigation into Rényi Mutual Information

- A more fine grained investigation into the region $0.1 \leq \alpha \leq 2.0$, to investigate if there is a continuous variation in performance between the α values.

- An investigation into higher values of α to see if the region above 2 provides any improvement in performance.
- An investigation into using different datasets to see if properties of the joint and divergence formulations can be derived.

8.3.4 Investigation into Graph-based Entropy Estimation

- The creation of a Parzen Window based estimator for the continuous Rényi entropy would enable the usage of the joint formulation of the Rényi mutual information instead of the ill-founded conditional formulation.
- The modification of the genetic algorithm to provide crossover, and a method for storing the previous optimal solutions.
- The creation of a better bias function which can provide an accurate estimate in low-dimensional spaces.
- An investigation into the effect of other values of the α parameter on the performance of the graph-based algorithms.
- An investigation into the performance of the continuous entropy, when applied to a discrete space.

Bibliography

[1] B. Bonev, F. Escolano, and M. Cazorla. A Novel Information Theory Method for Filter Feature Selection. *LECTURE NOTES IN COMPUTER SCIENCE*, 4827:431, 2007.

[2] H. Bremermann. Optimization through evolution and recombination. *Self-Organizing Systems*, pages 93–106, 1962.

[3] T. Cover and J. Thomas. *Elements of information theory*. Wiley New York, 1991.

[4] F. Fleuret. Fast Binary Feature Selection with Conditional Mutual Information. *The Journal of Machine Learning Research*, 5:1531–1555, 2004.

[5] A. Golan and J. Perloff. Comparison of maximum entropy and higher-order entropy estimators. *Journal of Econometrics*, 107(1-2):195–211, 2002.

[6] I. Guyon, A. Elisseefi, and L. Kaelbling. An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, 3(7-8):1157–1182, 2003.

[7] A. Hero III and O. Michel. Asymptotic theory of greedy approximations to minimal k-pointrandom graphs. *Information Theory, IEEE Transactions on*, 45(6):1921–1938, 1999.

[8] R. Kohavi and G. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

[9] D. Koller and M. Sahami. Toward optimal feature selection. *International Conference on Machine Learning*, 1996, 1996.

[10] J. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7(1):48–50, 1956.

[11] S. Kullback and R. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[12] P. Meyer and G. Bontempi. On the Use of Variable Complementarity for Feature Selection in Cancer Classification. *Applications of Evolutionary Computing*, pages 91–102, 2006.

[13] H. Neemuchwala. *Entropic Graphs for Image Registration*. PhD thesis, The University of Michigan, 2005.

[14] H. Neemuchwala, A. Hero, S. Zabuawala, and P. Carson. Image Registration Methods in High-Dimensional Space. *International Journal of Imaging Systems and Technology*, 16:130–145, 2006.

[15] E. Parzen. On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.

[16] H. Peng and F. Long. An efficient max-dependency algorithm for gene selection. *36th Symposium on the Interface: Computational Biology and Bioinformatics*, 2004.

[17] H. Peng, F. Long, and C. Ding. Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27, 2005.

[18] A. Rényi. On Measures of Information and Entropy. *Proceedings of the 4th Berkeley Symposium on Mathematics, Statistics and Probability*, pages 547–561, 1960.

[19] A. Renyi. *Probability theory*. Akademiai Kiado Budapest, 1970.

[20] C. Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27:379–423,623–656, 1948.

[21] K. Torkkola and W. Campbell. Mutual information in learning feature transformations. *Proc. 17th International Conf. on Machine Learning*, pages 1015–1022, 2000.

[22] G. Toussaint. Note on optimal selection of independent binary-valued features for pattern recognition (Corresp.). *Information Theory, IEEE Transactions on*, 17:618–618, 1971.

[23] L. Yu and H. Liu. Efficient Feature Selection via Analysis of Relevance and Redundancy. *The Journal of Machine Learning Research*, 5:1205–1224, 2004.