Axiomatic approach to feature subset selection based on relevance

Hui Wang, David Bell, Fionn Murtagh
School of Information and Software Engineering
Faculty of Informatics
University of Ulster Magee College
Northland Road, Londonderry, BT48 7JL
Northern Ireland, UK
Email: {h.wang, da.bell, fd.murtagh}@ulst.ac.uk

Abstract

In this paper an axiomatic characterisation of feature subset selection is presented. Two axioms are presented: sufficiency axiom — preservation of learning information, and necessity axiom — minimising encoding length. The sufficiency axiom concerns the existing dataset and is derived based on the following understanding: any selected feature subset should be able to describe the training dataset without losing information, i.e., it is consistent with the training dataset. The necessity axiom concerns predictability and is derived from Occam's razor, which states that the simplest among different alternatives is preferred for prediction. The two axioms are then re-stated in terms of relevance in a concise form: maximising both the $r(X;Y)$ and $r(Y;X)$ relevance. Based on the relevance characterisation, a heuristic selection algorithm is presented and experimented with. The results support the axioms.

1 Introduction

The problem of feature subset selection (FSS hereafter) has long been an active research topic within statistics and pattern recognition (e.g., [9]), but most work in this area has dealt with linear regression. In the past few years, researchers in machine learning have realised (see for example, [18, 16]) that practical algorithms in supervised machine learning degrade in performance (prediction accuracy) when faced with many features that are not necessary for predicting the desired output. Therefore FSS has since received considerable attention from machine learning researchers interested in improving the performance of their algorithms.

Common machine learning algorithms, including top-down induction of decision trees, such as CART, ID3, and C4.5, and nearest-neighbour algorithms (such as instance-based learning), are known to suffer from irrelevant features [18, 19]. A good choice of features may not only help improve performance accuracy, but also aid in finding smaller models for the data, resulting in better understanding and interpretation of the data.

Broadly speaking, FSS is to select a subset of features from the feature space which is *good* enough regarding its ability to describe the training dataset and to predict for future cases. There is a wealth of algorithms for FSS (see for example, [2, 15, 1, 17, 14, 24]). With regard to how to evaluate the *goodness* of a subset of features, the FSS methods fall into two broad categories: *filter approach* and *wrapper approach*, which are illustrated in Figures 1 and 2. In the filter approach, a *good* feature set is selected as a result of pre-processing based on properties of the data itself and

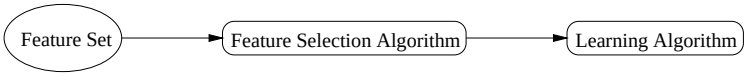

Figure 1: *Filter model.*

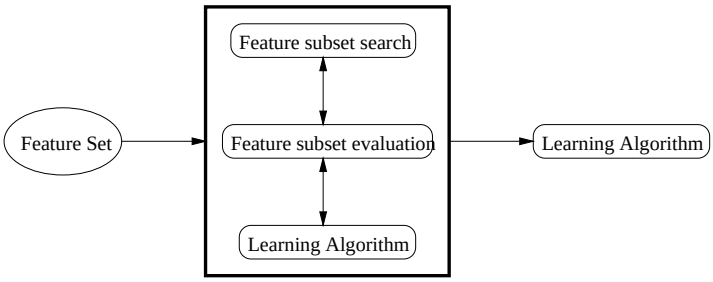

Figure 2: *Wrapper model.*

independent of the induction algorithm. Section 5.1 presents a review on the empirical use of the notion of *goodness* in this category.

There is a special type in this approach — *feature weighting* [15], which is slightly different from the mainstream filter approach in the way the search for good feature set is conducted. Basically the mainstream approach evaluates each subset of features and finds the "optimal", while the weighting approach weighs each individual feature and selects a "quasi-optimal" set of features, typically those whose weights exceed a given threshold [15, 17].

In the wrapper approach, feature selection is done with the help of induction algorithms. The feature selection algorithm conducts a search for a *good* feature set using the induction algorithm itself as part of the evaluation function. Typically, the feature subset which performs best for the induction algorithm will be selected.

Both types of approach to FSS are closely related to the notion of relevance. For example, FOCUS [2], RELIEF [15] and Schlimmer's model [22] use "relevance" to estimate the *goodness* of feature subset in one way or another. Section 5.2 presents a review in this respect. Although the wrapper approach does not use the relevance measure directly, it is shown [16] that the "optimal" feature subset obtained this way must be from the relevant feature set (strongly relevant and weakly relevant features).

However, the mathematical foundation for FSS is still lacking [26]. In [25], a unified framework for relevance was proposed. In this framework relevance is quantified and related to mutual information, and furthermore, it was shown that this quantification satisfies the axiomatic characterisations of relevance laid down by leading researchers in this area. This renders the notion of relevance having a solid mathematical foundation.

In light of these, we attempt to characterise FSS in terms of the relevance framework, in order to give FSS a solid foundation for further theoretical study. We then present an algorithm for FSS based on the relevance characterisation. We also present some experimental results applying this algorithm to some real world datasets.

2 Characterisation of feature subset selection

In this section we are to characterise FSS in the realm of machine learning, which is confined to the following sense.

The input to a (supervised) learning algorithm is a training set D of m labelled instances of a target (concept) Y [1]. Typically D is assumed drawn independently and identically distributed (i.i.d.) from an unknown distribution over the labelled instance space. An unlabelled instance $\mathbf{x}$ is an element of the n dimensional space $X_1 \times X_2 \times \cdots \times X_n$, where X_i is the ith feature (or variable) in the feature space $X = \{X_1, X_2, \cdots, X_n\}$ [2]. Labelled instances are tuples $< \mathbf{x}, y >$ where y is the label, or output. Let $\mathcal{L}$ be a learning algorithm having a hypothesis space $\mathcal{H}$. $\mathcal{L}$ maps D to $h \in \mathcal{H}$ and h maps an unlabelled instance to a label. The task of the learning algorithm is to choose a hypothesis that best explains the given data D.

In this paper, the training set D will be represented by a relation table $r[X \cup Y]$ [3], where X is the set of features and Y is the *output* or *target* variable. In what follows we will use $r[X \cup Y]$ to denote both the *learning task* and the training set.

The problem of feature selection is then to search for a subset Π of X that not only performs well on the training dataset, but also predicts well on unseen new cases — it is *good* enough. Our objective in this section is to characterise what the best feature subset should be from first principles as well as some known principles.

2.1 The preservation of learning information

Given a dataset $r[X \cup Y]$, the learning task is to characterise the relationship between X and Y so that this relationship can be used to predict on future cases (either one in the dataset or a new case). Therefore any selected feature subset, if it is expected to work well on the given dataset, should preserve the existing relationship between X and Y hidden in the dataset. A natural measure of this relationship is the mutual information [7]. We call this relationship *learning information*.

Specifically, given a learning task $r[X \cup Y]$, the ***learning information*** is the mutual information $I(X;Y)$. Furthermore, suppose Σ and Π are two subsets of X. If $I(\Sigma;Y) = I(\Pi;Y)$, then we say that Σ and Π have the ***same contribution*** to the learning task. A ***sufficient feature set*** or simply ***SFS*** of a learning task is a subset, Σ, of X such that $I(\Sigma;Y) = I(X;Y)$. Clearly, all SFS's contribute the same to the learning task. This is re-stated as the following axiom:

Axiom 2.1 (Preservation of learning information) *For a given learning task $r[X \cup Y]$, the best feature subset, Π, should preserve the learning information contained in the training dataset. That is, $I(\Pi;Y) = I(X;Y)$.*

The following two lemmas follow directly from the chain rule for mutual information and the non-negativity of mutual information.

Lemma 2.1 *Given $r[X \cup Y]$. For any $\Pi \subseteq X$, $I(\Pi;Y) \leq I(X;Y)$.*

From this lemma and the additivity of mutual information [7] we know that given a SFS Π, removing all the remaining features Σ will not lose learning information contained in the original dataset. In other words, Y is conditionally independent of Σ given Π.

[1] *Target* or *target concept* is usually defined as a subset of an instance space [2], which can be interpreted as a bi-partition of the instance space. Here we use it in the more general sense: a target concept is an arbitrary partition of the instance space. It is regarded as a variable here.

[2] In this paper we use X_i to refer to both a variable and the domain of the variable, when this can be identified from the context.

[3] [6, 12]. We use the notation in [12]. A relation scheme R is a set of variables (features). A relation (table) over R is an indicator function for a set of tuples, written $r[R] : r[R](t) = 1$ if the tuple t is in the relation; $r[R](t) = 0$ otherwise. For the purpose of this paper, we extend the indicator function such that $r[R](t) = n$, where n is the frequency of tuple t appearing in the relation. With this extension, we can talk about the distribution of the tuples, which can be easily obtained.

Lemma 2.2 *If* Π *is a SFS for a learning task* $r[X \cup Y]$*, then any superset,* Σ*, of* Π *is also a SFS.*

This lemma helps in determining SFSs without having to calculate the learning information. This property is exploited in the design of an FSS algorithm later.

2.2 The simplest description: Occam's razor

Given a learning task, there may be a number of SFSs. However they may not perform the same on prediction. The best feature subset should perform best in this respect. However it is not easy to determine which subset of features predicts better since there is no full knowledge about the future. Although the dataset is assumed to be drawn i.i.d. from the labelled instance space according to an unknown distribution, this assumption doesn't help in individual cases. What we can do is to focus on the training dataset itself and then apply some empirical principles. There are a number of empirical principles. Occam's razor is one of them.

Occam's razor, known as *the principle of parsimony*, is a tool that has application in many areas of science, and it has been incorporated into the methodology of experimental science. This principle is becoming influential in machine learning, where this principle can be formulated as: given two hypotheses that both are consistent with a training set of examples of a given task, the simpler one will guess better on future examples of this task [4, 27, 3]. It has been shown (see for example, [4]) that, under very general assumptions, Occam's razor produces hypotheses that with high probability will be predictive of future cases.

One basic question is concerned with the meaning of "simplicity", namely *Occam simplicity*. Typically Occam simplicity is associated with the difficulty of implementing a given task, namely *complexity of implementation*. For example, the number of hidden neurons in neural networks [3]; the number of leaf nodes of a decision tree [10, 11]; the minimum description length (MDL) [21, 20]; and the *encoding length* [23]. However, Wolpert [27] noticed that the complexity of implementation is not directly related to the issue of prediction or generalisation, therefore there is no direct reason to believe that minimisation of such a complexity measure will result in improvement of generalisation. Wolpert [27] then derived the *uniform simplicity measure*, which is concerned exclusively with how learning generalises. Wolpert showed [27] that when expressed in terms of the uniform simplicity measure Occam's razor is indeed a way to set up a good generaliser.

The main disadvantage of uniform simplicity measure is that the calculation of it needs "all learning sets and all questions", as well as guessing distribution and simplicity distribution [27]. This is impossible in practice. It seems that uniform simplicity measures have only theoretical significance. Fortunately many of the conventional simplicity measures are shown to be rough approximations to the uniform simplicity measure [27]. In practice we can only rely on approximations, like those mentioned above.

Back to our problem: Most of the practical simplicity measures (approximations to uniform simplicity measure) are model-dependent. However we are looking at FSS independently of any learning model, so a model-independent simplicity measure is required. Entropy seems an ideal candidate, as it measures the average number of bits (encoding length) to describe a source (e.g., a random variable).

Using the entropy as the Occam simplicity measure in our context, we have: given a learning task $r[X \cup Y]$, the Occam's razor dictates the selection of a SFS Π which minimises $H(\Pi, Y)$, where H is Shannon's entropy function. To make this formal, we re-state it, in conjunction with the information preservation axiom, as the following axiom:

Axiom 2.2 (Minimum encoding length) *Given a learning task* $r[X \cup Y]$ *and a set of sufficient feature subsets. The one* Π *which minimises the joint entropy* $H(\Pi, Y)$ *should be favoured with*

respect to its predictive ability.

Now we set out to characterise the Π which minimises the joint entropy.

Lemma 2.3 *Given a learning task $r[X \cup Y]$, consider two SFSs $\Pi, \Sigma \subseteq X$. $H(\Pi, Y) \leq H(\Sigma, Y) \iff H(\Pi) \leq H(\Sigma)$.*

Proof. Since both Π and Σ are SFSs, by definition we have $I(\Pi; Y) = I(\Sigma; Y) = I(X; Y)$. Therefore we have $H(Y) - H(Y|\Pi) = H(Y) - H(Y|\Sigma) \iff H(Y|\Pi) = H(Y|\Sigma)$. Furthermore we have $H(\Pi) \leq H(\Sigma) \iff H(\Pi) + H(Y|\Pi) \leq H(\Sigma) + H(Y|\Sigma) \iff H(\Pi, Y) \leq H(\Sigma, Y)$. □

According to this lemma, the most favourable feature subset would be the sufficient one which has the least marginal entropy.

2.3 Characterisation of feature subset selection in terms of relevance

In the previous two sections we have derived two axiomatic characterisations of FSS: the preservation of learning information, and minimum encoding length. In this section we are going to show the above two axioms can all be re-stated in terms of relevance, in an even more concise form.

Given two variables X and Y, by definition (see appendix), the relevance of X to Y is $I(X; Y)/H(Y)$, wrt. $r(X; Y)$. Therefore for a SFS $\Pi \subseteq X$, i.e., $I(\Pi; Y) = I(X; Y)$, it is clearly $r(\Pi; Y) = r(X; Y)$. So preserving learning information amounts to preserving the relevance relationship. Since $r(\Pi; Y) \leq r(X; Y)$ in general (due to the fact that $I(\Pi; Y) \leq I(X; Y)$), the Π which preserves learning information in fact maximises the relevance $r(X; Y)$.

Consider two SFSs Π and Σ. Since, by definition, $I(\Pi; Y) = I(\Sigma; Y) = I(X; Y)$, we have $H(\Pi) \leq H(\Sigma) \iff I(\Pi; Y)/H(\Pi) \geq I(\Sigma; Y)/H(\Sigma) \iff r(Y; \Pi) \geq r(Y; \Sigma)$. Therefore, in conjunction with the previous requirement, the most favourable feature subset would be the sufficient one which maximises the relevance $r(Y; X)$.

Summarising the above discussion we have the following theorem:

Theorem 2.1 *Given a learning task $r[X \cup Y]$, the most favourable feature subset is the Π which is sufficient (preserving learning information, $I(\Pi; Y) = I(X; Y)$) and minimises the joint entropy $H(\Pi, Y)$ among all other SFSs. Putting it concisely, this is the one which has maximum $r(\Pi; Y)$ and maximum $r(Y; \Pi)$.*

This theorem formalises the more or less intuitively justified connection between relevance and FSS.

3 A relevance-based algorithm for feature selection

In this section we present a heuristic FSS algorithm which is based on the characterisation in the previous section. A straightforward algorithm is to systematically examine all feature subsets and find one which satisfies the above two axioms. Unfortunately, as shown in [8], this class of algorithms turns out to be NP-complete. Branch and bound based on the characteristics of relevance was attempted [25], but it was shown to be also exponential in general. So we attempted heuristic approaches. Here we are to present our preferred heuristic FSS algorithm.

Our objective is to find a sufficient subset of features, which is close to optimal in the above axiomatic sense. The heuristic used here is: if a feature or attribute is highly relevant on its own, it it very likely that this feature is in the optimal feature set. Since features are examined individually, we need to take into account the correlation among individual features. Consider, for example, two features x_1, x_2, let Y be the target. Suppose $r(x_1; x_2) = 1$, $r(x_1; Y) = 0.9$, and $r(x_2; Y) = 0.95$. If

x_1 is selected, then x_2 is not needed any more since $r(x_2; Y|x_1) = 0$ according to Lemma 6.1. In other words, x_2 becomes irrelevant given x_1. Our algorithm should not select them both. To this end, we design our algorithm, which takes advantage of conditional relevance.

Algorithm 3.1 (CR: feature selection based on conditional relevance) *Given a learning task $r[X \cup Y]$, where $|X| = N$,*

- *Calculate, for every $x \in X$, the relevance $r(x; Y)$, and find the feature x_0 with largest relevance value;*
- *Main procedure:*
  1. *$BSFS = \{x_0\}$;*
  2. *Repeat: Add x_i to $BSFS$ such that x_i is not in $BSFS$ and $r(x_i; Y|BSFS)$ is the largest among all possible relevance values.*
  3. *Until $r(BSFS; Y) = 1$;*
- *Return $BSFS$.*

Clearly the time complexity for calculating relevance and finding the largest is $O(N)$. We now analyse the complexity for the main procedure. At loop k where there are k features left for inspection, we need to compute conditional relevance $r(x_i; Y|BNAS)$ for all k features, hence a complexity of $O(k)$. To find the feature with largest conditional relevance value, we need $k - 1$ comparisons, hence a complexity of $O(k - 1)$. In the worst case we need to loop from $k = N - 1$ to $k = 1$, hence the complexity is $\sum_{k=N-1}^{1} 2k - 1 = O(N^2)$. Therefore the overall complexity for above algorithm is $O(N^2)$.

This algorithm is highly dependent on the choice of the initial set of features, which is the individual feature most relevant to Y. The BSFS selected by CR is guaranteed to be SFS, but not guaranteed to be necessary. It is conjectured that if x_0 is in the optimal SFS, then the BSFS found by CR is indeed optimal.

4 Experiment and evaluation

Here we are to evaluate the performance of the feature selection algorithm presented in the previous section using some real world datasets. We choose three datasets from the U. C. Irvine machine learning repository: Australian, Diabetes, and Heart. Some general information about these datasets is shown in Table 1.

To evaluate the performance of out feature selection algorithm, we chose to use the C4.5 module in the Clementine package in our experiment. We feed the selected feature subsets to C4.5 and compare the results with and without feature subset selections.

The test accuracies by C4.5 without and with feature selection are shown in Table 2. The evaluation method we used is cross validation implemented in Clementine. From these experiment results we see that applying our feature selection algorithm does indeed improve the test accuracies for all three datasets, and the corresponding decision trees have smaller sizes. However the success is limited in the sense that the accuracy improvements were not very great in this case. The reason is probably that C4.5 has a built-in feature selection facility based on mutual information. It is then reasonable to believe that if the feature selection algorithm described above is used with other learning algorithms without built-in feature selection facilities (e.g., nearest neighbour), the accuracy improvement could be higher than those reported here.

Dataset	features	no. of examples	no. classes	class distribution
Australian	14	690	2	44.5%(+)
Diabetes	8	768	2	65.1%(+)
Heart	13	270	2	55.56%(+)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Australian	D	C	C	D	D	D	C	D	D	C	D	D	C	C
Diabetes	C	C	C	C	C	C	C	C						
Heart	C	D	D	C	C	D	D	C	D	C	C	C	D	

Table 1: *General information about the datasets, where D refers to discrete (here categorical) and C refers to continuous.*

Dataset	C4.5		C4.5-CR		
	Size of trees	Test accuracy	Selected features	Size of trees	Test accuracy
Australian	32	85.2	2,3,8,9,10,13,14	22	85.7
Diabetes	54	72.9	2,5,6,7,8	42	74.2
Heart	16	77.1	5,9,12,13	12	80.8

Table 2: *Decision tree sizes, test accuracies on decision trees generated by C4.5 without and with feature selection, together with the selected feature sets. The evaluation method we used is cross validation implemented in Clementine. The datasets are from the U. C. Irvine machine learning repository: Australian credit, Diabetes, and Heart.*

We also carry out an experiment to inspect the change of accuracies through gradually adding features in the order of relevance values. We first rank all the features according to their individual relevance values ($r(X;Y)$ only) and start evaluation from the one with highest relevance value. The results are shown in Figure 3. From this figure we can see that as features are gradually added in the order, the accuracy will on average go up first and reach a peak and then go down. This diagram justifies to some extent our algorithm, although the algorithm may not always find the feature subsets corresponding exactly to the peak points.

Another observation from this experiment is that the performance of C4.5 for the three datasets is (in descending order): Australian, Heart and Diabetes (Table 2) in terms of the average (test) accuracy, while the percentage of continuous features is in the (descending) order: Diabetes (8/8), Heart (7/13), and Australian (6/14). It indicates that C4.5 doesn't work as well for continuous features as for discrete features. Our feature selection algorithm didn't change this situation. In C4.5, continuous features are treated as discrete features in such a way that their values are divided into two groups, each of which is a discrete cluster used in the classification. From the granularity point of view [13], the granularity of the continuous features are made simply too coarse. In our feature selection algorithm, continuous features are treated as discrete features in such a way that each continuous value is taken to be a discrete value and is used individually in the classification. Again the granularity here seems too fine. This points to a direction for future studies: what is the proper granularity for a continuous feature for use in classification?

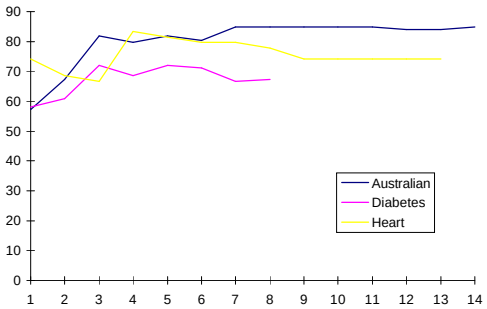

Figure 3: *Accuracy vs. first k features used in the relevance ranking, where k starts from 1. The $(r(X;Y))$ relevance-based rankings for the three datasets are as follows. Australian: 2,14,8,3,13,7,10,9,5,6,4,12,11,1; Diabetes: 7,6,2,5,8,4,1,3; Heart: 5,8,10,13,3,12,1,4,9,11,2,7,6.*

5 Comparison with related work

In this section we are to take a closer look at some related work from the relevance point of view and compare them with ours.

5.1 How is the best feature subset characterised in the literature?

In [15], the best feature subset is characterised as *sufficient and necessary* to describe the target. Ideally the sufficiency and necessity requirement is quantified by a measure $J(\Pi, Y, D)$ which evaluates the feature subset Π for the target concept Y and the given data D: the best feature subset should have the best value of $J(\Pi, Y, D)$. However the nature of the sufficiency and necessity requirement was not made clear in [15]. In the context of learning from examples, it seems reasonable that sufficiency concerns the ability of a feature subset to describe the given dataset (called *qualified* later on), while the necessity concerns the optimality among all the qualified feature subsets regarding predictive ability. From this we can say that our two axiomatic characterisations are possible interpretations of the sufficiency and necessity requirement proposed in [15].

In practice, the best feature subsets are measured in pragmatic ways. For example, in FOCUS [2] a *good* feature subset is a minimal subset which is consistent with the training dataset. Here the consistency can be understood as the sufficiency requirement, since only when the feature subset is consistent with the given dataset can it qualify to describe the dataset without losing learning information. The minimality of feature subset can be understood as the necessity requirement, as it was used as a bias of learning regarding which subset can predict better for future cases. In RELIEF [15], a *good* subset is one whose elements each has a relevance level greater than a given threshold. Here the relevancy and the threshold together determine whether a given feature subset is sufficient (or qualified) to describe the given dataset. But there is no direct justification as to why the feature subset determined in this way would perform better in predicting for future cases, i.e., necessary. In [22] a *good* subset is one of the minimal determinations, but nothing is mentioned as to which one is the *best*. Here all the minimal determinations are sufficient, but which of these is necessary is left open.

5.2 Re-modelling using the relevance framework

Many FSS algorithms use "relevance" to estimate feature usefulness in one way or another. The FOCUS [2] algorithm starts with an empty feature set and carries out breadth-first search until it finds a minimal combination Π of features which is consistent with the training dataset. The

features in Π are relevant to the target concept C. In terms of the relevance framework [25], this requirement amounts to $r(\Pi; C) = 1$ and $|\Pi|$ being minimum.

RELIEF is a feature relevance estimation algorithm, but the meaning of relevance is different from ours and has not been theoretically justified. It associates with each feature a weight indicating the relative relevance of that feature to the concept class (C) and returns a set of features whose weights exceed a threshold. This amounts to firstly calculate, for each feature X, $r(X; C)$, and then select a set of features such that for any X in this set, $r(X; C) \geq \tau$, where τ is the threshold. Compared to FOCUS, this method is computationally efficient. Furthermore, it allows features to be ranked by relevance.

Schlimmer [22] described a related approach that carries out a systematic search through the space of feature sets for all (not just the one with minimal cardinality) minimal determinations which are consistent with training dataset. The algorithm has an attractive polynomial complexity due to the space-for-time technique: caching the search path to avoid revisiting states. A determination is in fact a SFS, and a minimal determination is such a SFS that removing any element will render it not being a SFS anymore. Therefore this algorithm amounts to finding all SFSs within a given length such that for each of these, Π, $r(\Pi; C) = 1$ and for any $X \in \Pi$, $r(\Pi / \{X\}; C) < 1$.

Most recent research on feature selection differs from these early methods by relying on wrapper strategies rather than filtering schemes. The general argument for wrapper approaches is that the induction method that will use the feature subset should provide a better estimate of accuracy than a separate measure that may have an entirely different inductive bias. John, Kohavi, and Pfleger [14] were the first to present the wrapper idea as a general framework for feature selection. The generic wrapper technique must still use some measure to select among alternative features. One natural scheme involves running the induction algorithm over the entire training data using a given set of features, then measuring the accuracy of the learned structure on the training data. However, John et al argue that a cross-validation method, which they use in their implementation, provides a better measure of expected accuracy on novel test cases.

The major disadvantage of wrapper methods over filter methods is the former's computational cost, which results from calling the induction algorithm for each feature set considered. This cost has led some researchers to invent ingenious techniques for speeding the evaluation process.

The wrapper scheme in [16] does not use the relevance measure directly; rather, it uses the accuracy obtained by applying an induction algorithm as the measure for the goodness of feature sets. However, Kohavi and Sommerfield show that the "optimal" feature set X obtained this way must be from the relevant feature set (strongly relevant and weakly relevant features). As shown in [25] their strong relevance and weak relevance can be characterised by our relevance formalism, so the wrapper scheme can also be modelled by our relevance, $r(X; C) > 0$.

However, Caruana and Freitag [5] observe that not all features that are relevant are necessarily useful for induction. They tested FOCUS and RELIEF on the calendar scheduling problem, where they fed the feature sets obtained by those two algorithms to ID3/C4.5, and found that a more direct feature selection procedure, hill-climbing in feature space, finds superior feature sets. They didn't explain the reason for this. But a possible explanation based on relevance is as follows. For a given concept class C there are many SFS's, where for each SFS, X, $r(X; C) = 1$. One of the many SFS's, which satisfies some criteria, should be optimal in general. This optimal feature set may not be the minimal one in general. Starting from Occam's razor, we argue that the optimal one should be such that $r(C; X)$ is maximised.

In conclusion from the above discussion, RELIEF, Schlimmer's algorithm, and Wrapper take into account only the sufficiency condition, evidenced by their addressing only $r(X; C)$. FOCUS takes into account both sufficiency and necessity conditions. But the necessity is measured by the cardinality of the feature subset being minimal. The relationship of this measurement to the

Occam's razor characterisation above is not clear yet.

6 Conclusion

In this paper we have derived, from first principles and Occam's razor principle, two axiomatic requirements for any feature subset to qualify as "good": preservation of learning information and minimum encoding length. Since FSS has traditionally linked with relevance, we further showed that when identified with the variable relevance in the unified framework for relevance, relevance has a direct relationship with FSS: maximising relevance in both ways (i.e., $r(X;Y)$ and $r(Y;X)$) will result in the favourable feature subset.

Based on the axiomatic characterisation of FSS, one heuristic FSS algorithm was designed and presented. This algorithm weights (ranks) features using conditional relevance $r(X;Y|Z)$ in a step-wise way: it starts with the feature with the highest unconditional relevance value and then keeps selecting features with highest conditional relevance values with respect to the current selected subset. This algorithm can get rid of highly correlated features, and it is shown to have a complexity of $O(n^2)$.

We also presented evaluation results using three real world problems: Australian credit, Diabetes diagnosis, and Heart diagnosis, all from the UCI machine learning repository. The purpose of the evaluation is two fold. Firstly, we evaluated the performance of the algorithm. The results are quite encouraging: the average test accuracies on three datasets were all improved, and the resultant decision trees had smaller tree sizes. Since C4.5 has a built-in feature selection process, which is based on gain ratio defined by mutual information, we conjecture that if the algorithm is used with other learning algorithms without a built-in feature selection process (e.g., nearest neighbour), the accuracy improvement could be higher.

Secondly, we evaluated the relationship between relevance and learning accuracy. The results show a strong connection between relevance and learning accuracy. When all features are ranked according to their conditional relevance values, adding features one by one to the feature set would lead to a clear pattern of accuracy: first ascending to a peak and then descending gradually. Therefore we conclude that highly relevant features can improve learning accuracies and highly irrelevant features can degrade learning accuracies.

As an aside, we observed that C4.5 based learning accuracy (whether or not feature selection is used) is related to the proportion of continuous features: the higher the proportion of continuous features, the lower the accuracy. It is argued that one possible reason is that in C4.5 continuous features are bi-partitioned, which could be too coarse. Future studies in this direction will focus on developing algorithms to find proper granularities for continuous features.

References

[1] D. W. Aha and R. L. Bankert. Feature selection for case-based classification of cloud types. In *Working notes of the AAAI94 Workshop on Case-based Reasoning*, pages 106–112. AAAI Press, 1994.

[2] H. Almuallim and T. G. Dietterich. Learning with many irrelevant features. In *Proc. Ninth National Conference on Artificial Intelligence*, pages 547–552. MIT Press, 1991.

[3] B. Amirikian and H. Nishimura. What size network is good for generalization of a specific task of interest? *Neural Networks*, 7(2):321–329, 1994.

[4] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Occam's Razor. *Information Processing Letters*, 24:377–380, 1987.

[5] R. Caruana and D. Freitag. How useful is relevance? In *Proceedings of the 1994 AAAI Fall Symposium on Relevance*, pages 21–25. AAAI Press, 1994.

[6] E. F. Codd. A relational model of data for large shared data banks. *Commun. ACM*, 3(6):377–387, 1970.

[7] T. M. Cover and J. A. Thomas. *Elements of information theory.* John Wiley & Sons, Inc., 1991.

[8] S. Davies and S. Russell. NP-Completeness of Searches for Smallest Possible Feature Sets. In *Proceedings of the 1994 AAAI Fall Symposium on Relevance*, pages 37–39. AAAI Press, 1994.

[9] P. A. Devijver and J. Kittler. *Pattern recognition: A statistical approach.* New York: Prentice-Hall, 1982.

[10] U. Fayyad and K. Irani. What should be minimized in a decision tree? In *AAAI-90: Proceedings of 8th National Conference on Artificial Intelligence*, 1990.

[11] U. Fayyad and K. Irani. The attribute selection problem in decision tree generation. In *AAAI-92: Proceedings of 10th National Conference on Artificial Intelligence*, 1992.

[12] Joe R. Hill. Relational Databases: A Tutorial for Statisticians. In E. M. Keramidas and S. M. Kaufman, editors, *Computing Science and Statistics: Proc. of the 23rd Symposium on the Interface*, pages 86–93, 1991.

[13] Jerry R. Hobbs. Granularity. In *Proc. IJCAI85*, pages 432–435, 1985.

[14] G. H. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. In *Proceedings of the 11th international conference on machine learning*, pages 121–129. New Brunswick, NJ: Morgan Kaufmann, 1994.

[15] K. Kira and L. A. Rendell. The feature selection problem: traditional methods and a new algorithm. In *AAAI-92*, pages 129–134, 1992.

[16] R. Kohavi and D. Sommerfield. Feature Subset Selection Using the Wrapper Method: Overfitting and Dynamic Search Space Topology. In U. M. Fayyad and R. Uthurusamy, editors, *Proceedings of KDD'95*, pages 192–197, 1995.

[17] I. Kononenko. Estimating attributes: analysis and extensions of RELIEF. In *Proceedings of the 1994 European Conference on Machine Learning*, 1994.

[18] Pat Langley. Selection of relevant features in machine learning. In *Relevance: proc. 1994 AAAI Fall Symposium*, pages 127–131. AAAI Press, 1994.

[19] D. Michie, D. J. Spiegelhalter, and C. C. Taylor. *Machine learning, neural and statistical classification.* Ellis Horwood, 1994.

[20] J. Quinlan and R. Rivest. Inferring decision trees using the minimum description length principle. *Information and Computation*, 80:227–248, 1989.

[21] J. Rissanen. Stochastic complexity and modeling. *Ann. Statist.*, 14:1080–1100, 1986.

[22] J. C. Schlimmer. Efficiently inducing determinations: a complete and systematic search algorithm that uses optimal pruning. In *ML93*, pages 284–290, 1993.

[23] H. Schweitzer. Occam algorithms for computing visual motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(11):1033–1042, 1995.

[24] D. B. Skalak. Prototype and feature selection by sampling and random mutation hill-climbing algorithms. In *Proceedings of the 11th International Conference on Machine Learning*, pages 293–301. Seattle, WA: Morgan Kaufmann, 1994.

[25] Hui Wang. *Towards a unified framework of relevance* . PhD thesis, Faculty of Informatics, University of Ulster, N. Ireland, UK, October 1996. http://www.infm.ulst.ac.uk/~hwang/thesis.ps.

[26] Sholom M. Weiss and Casimir A. Kulikowski. *Computer systems that learn – classification and predication methods from statistics, neural networks, machine learning, and expert systems.* Morgan Kaufmann Publishers, San Mateo, California, 1991. ISBN 1-55860-065-5.

[27] D. H. Wolpert. The relationship between Occam's Razor and convergent guessing. *Complex Systems*, 4:319–368, 1990.

Appendix: A unified framework for relevance

Relevance is a common sense notion in our daily lives. It concerns the relationship among objects. Suppose we have two objects X and Y. We understand X is relevant to Y if knowing X happening would change the likelihood of Y. Based on this understanding, various formulations have been proposed with regard to different problem domains. A unified framework was proposed in [25] which unifies two basic types of relevance in a consistent way: variable relevance and instance relevance. Various guises of relevance can be modelled by this framework. For the purpose of this paper, we present a brief introduction to the variable relevance.

The relevance of one variable to another (target) variable is understood in information theoretic terms, as the *mutual information between the two variables relative to the entropy of the target variable*, or in other words, the relative reduction of entropy (uncertainty) of one variable due to the knowledge of another. The bigger the reduction, the higher the relevance. Formally we have:

Definition 6.1 *Given three random variables X, Y and Z with a joint probability distribution p, if $H(Y|Z) \neq 0$, then the* ***variable relevance*** *of X to Y given Z, denoted $r_{v,p}(X;Y|Z)$, is defined as*

$$r_{v,p}(X;Y|Z) = \frac{I(X;Y|Z)}{H(Y|Z)} = \frac{H(Y|Z) - H(Y|X,Z)}{H(Y|Z)}$$

If $H(Y|Z) = 0$, then $r_{v,p}(X;Y|Z) = 0$.

Where there is no ambiguity, p and v will be dropped for brevity.

This definition says that the relevance of X to Y given Z is indicated by the relative reduction of uncertainty of Y when X and Z are known. With this notion we can say that X is relevant to Y given Z with degree $r(X;Y|Z)$. Examples can be found in [25].

Theorem 6.1 (Dependency vs independency,[25]) *Suppose X, Y, and Z are three random variables with a joint distribution p. Then $r(X;Y|Z) = 1 \iff$ Y is conditionally fully dependent on X given Z; $r(X;Y|Z) = 0 \iff$ Y is conditionally independent of X given Z.*

This theorem shows that the definition of variable relevance agrees with two extreme cases of probabilistic dependence: full dependence and full independence. In other word, two extreme cases can be identified by our variable relevance measure: 0 for extreme irrelevance (conditional independence) and 1 for extreme relevance (full dependence).

There are some useful properties for variable relevance. Here we list some of them.

Lemma 6.1 ([25]) *The following properties hold for variable relevance:*

- *Continuity:* $r(X;Y|Z)$ *is continuous in the distribution* p.
- *Uniformity:* $0 \leq r(X;Y|Z) \leq 1$. *That is, relevance measures lie between two fixed extremes.*
- *Self-reflexiveness:* $r(X;X|Z) = 1$. *If* $Y \subseteq X$, *then* $r(X;Y) = 1$.
- *Symmetry:* $r(X;Y|Z) \geq 0 \iff r(Y;X|Z) \geq 0$. *But in general,* $r(X;Y|Z) \neq r(Y;X|Z)$.
- *Monotonicity: For two sets of variables* Σ *and* Ω, *if* $\Sigma \subseteq \Omega$, *then* $r(\Omega;Y) \geq r(\Sigma;Y)$.
- *Intransitivity: In general* $r(X;Y|Z) > 0 \,\&\, r(Y;W|Z) > 0 \not\Rightarrow r(X;W|Z) > 0$, *and* $r(X;Y|Z) = 0 \,\&\, r(Y;W|Z) = 0 \not\Rightarrow r(X;W|Z) = 0$.
- *Saturability: If* $r(X;Y|Z) = 1$, *then* $r(W;Y|Z,X) = 0$, *where* W *is any variable.*
- *Given two variables* X, Y, Z *with a joint distribution* p, $r(Z;Y) = 1 \iff r(X;Y|Z) = 0 \iff r(Y;X|Z) = 0$.

Other properties can be found in [25].