Ensemble Algorithms for Feature Selection

Jeremy D. Rogers and Steve R. Gunn

Image, Speech and Intelligent Systems Research Group
School of Electronics and Computer Science
University of Southampton, U.K.

Abstract. Many feature selection algorithms are limited in that they attempt to identify relevant feature subsets by examining the features individually. This paper introduces a technique for determining feature relevance using the average information gain achieved during the construction of decision tree ensembles. The technique introduces a node complexity measure and a statistical method for updating the feature sampling distribution based upon confidence intervals to control the rate of convergence. A feature selection threshold is also derived, using the expected performance of an irrelevant feature. Experiments demonstrate the potential of these methods and illustrate the need for both feature weighting and selection.

1 Introduction

Ensemble algorithms have achieved success in machine learning by combining multiple weak learners to form one strong learner. The Adaboost algorithm, [1] and the Bagging algorithm [2] are two examples of this. Much research has been conducted into understanding the mechanics of these methods and of finding ways to improve them. The explanations centre around the idea of diversity in the base learners, which enable good exploration of possible hypotheses. A good generalisation ability of an ensemble can be obtained by constructing accurate base learners that make their mistakes in different parts of the training data. Many improvements to ensemble algorithms exploit this idea by attempting to increase the diversity. The Random Forest technique, [3], is one such algorithm, which adopts the randomisation principle of [4] to achieve an increase in diversity. The base learners in this algorithm are CART based trees [5]. These trees usually perform a search through a large number of possible binary splits for every feature in order to find the optimal split for each node. The criterion for each split is the measure of information gain, which is the reduction in entropy that results from the split. The Random Forest algorithm uses Bagging to generate a training set for each tree. Diversity is injected into the ensemble by choosing a feature randomly at each node in the tree construction and optimising the split over a set of possible split values along that feature. Due to the random exploration of features, Random Forest lends itself to feature selection well.

Traditional approaches to feature selection have typically taken two forms, the Filter method which attempts to select the optimal feature subset by

J. Winkler, N. Lawrence, M. Niranjan (Eds.): Machine Learning Workshop, LNAI 3635, pp. 180–198, 2005.

analysing the structure of the data and the Wrapper method which performs a search through possible feature subsets and uses the learning algorithm to test the suitability of each. Both of these methods attempt to eliminate irrelevant features and reduce the probability of discovering false relationships in the data. Also, by reducing the dimensionality of the data, the computational requirement imposed upon the learning algorithm is reduced. The ability of each feature subset is partially dependent upon the learning algorithm used. The Wrapper method uses the learning algorithm to evaluate each feature subset and consequently, has the advantage of incorporating this bias. However, when using high dimensional data, the process of searching through possible feature subsets can be computationally expensive.

The selection of features is not the only application that is available once the relevance of each feature is known. Feature weighting algorithms are also used, where all of the features are included in the learning process. In this case each feature is relied upon to a different extent, which is determined by its level of importance. Although this approach can improve generalisation, it does not create a reduction in dimensionality. In random forest the feature weighting concept can be realised by applying these levels of feature importance to the feature sampling distribution from which the features are randomly chosen.

When feature selection is applied to ensemble learning, the criterion for selection is somewhat different. The identification of relevant features is still an important issue, but the selected features also need to promote diversity in the constructed base learners. [6] proposed an algorithm which employed the random subspace method, [7] to generate the learners which were evaluated in terms of their accuracy and diversity.

The measure of feature importance adopted here is the average information gain achieved during tree construction and a node complexity measure is introduced to improve the accuracy of this measure. These levels of importance are applied to the feature sampling distribution in a parallel scheme where the rate at which the feature sampling distribution is updated is controlled using a confidence interval method. This is also compared to a fast two stage method where the feature sampling distribution is set before forest construction.

Feature weighting is shown to be successful here, but if the data contains a significant number of irrelevant features, a selection scheme will improve performance. An approximate threshold for feature selection is derived here, which attempts to predict the expected average information gain achieved by an irrelevant feature. This is compared to a correlation based feature selection algorithm, CFS [8] and it is shown that both feature weighting and selection should be exploited to optimise the generalisation.

Section Two introduces the concept of feature relevance and the measure adopted in this paper. The node complexity measure is also introduced here. Section Three examines the methods of updating the feature sampling distribution and introduces the parallel algorithm. The feature selection threshold is described in section Four and the experimental results are shown in section Five. Section Six discusses the results and gives some directions for further work.

2 Identifying Feature Relevance

2.1 Defining Feature Relevance

In order to identify relevant features it is important to first consider a suitable definition for feature relevance. The goal is to identify the features that carry as much information concerning the target as possible and eliminate information that is repeated in multiple features. As discussed by [9], an obvious definition is that a feature X_i is relevant if there exist values x_i and y assigned to X_i and the target Y respectively, such that,

$$P\left(Y = y | X_i = x_i\right) \neq P\left(Y = y\right) \tag{1}$$

Intuitively this makes sense, as knowledge about the value of a relevant feature should affect the prediction of the target. However, it is not always the case that a relevant feature taken by itself provides valuable information about the target. This is illustrated using the XOR example.

Example 1. If the target, Y is given by the exclusive OR of the binary features, X_1 and X_2, then Y is fully described by the features and they are both relevant.

$$Y = X_1 \oplus X_2$$

However, if each feature assumes the values of 1 or 0 with equal probability, then the above definition shows them both to be irrelevant. This is because knowing the value of one of the features gives no information about the target without knowing the value of the other feature.

It is claimed by [9] that two different types of relevance are required for successful feature selection, strong relevance and weak relevance. Strong relevance is used to describe a feature, which carries information about the target that is not repeated in any other feature and is defined in the following manner, If S_i is the subset of all of the features apart from X_i and s_i is a value assignment to those features, then X_i is strongly relevant if there exists some x_i, y and s_i such that,

$$P\left(Y = y | X_i = x_i, S_i = s_i\right) \neq P\left(Y = y | S_i = s_i\right) \tag{2}$$

Removing a strongly relevant feature from the set will result in a loss of information about the target. Weak relevance is used to describe features that carry information about the target, but which is repeated in other features. Unlike strongly relevant features, if a weakly relevant feature is removed, no information about the target is lost. X_i is weakly relevant if it is not strongly relevant and there exists some subset, S_i' of S_i and values y, x_i and s_i' such that,

$$P\left(Y = y | X_i = x_i, S_i' = s_i'\right) \neq P\left(Y = y | S_i' = s_i'\right) \tag{3}$$

Some feature selection schemes examine the correlation between features in an attempt to discover this weak relevance such as [10], [11] and the CFS algorithm

of [8]. This type of method highlights information that is shared between features but does not discriminate between information that is useful for describing the target and random correlations in the data.

The definition of weak relevance can also be viewed as a definition of conditional independence, where the target Y is conditionally independent of the feature X_i given a subset S'_i, if there exists no value assignments to these variables which satisfy the inequality. This idea can be extended to the concept of identifying Markov Blankets [12]. If S'_i is some subset of the features, such that $X_i \notin S'_i$, and S_i is the subset of remaining features, that excludes S'_i and X_i, then S'_i is a Markov Blanket for X_i if there are no value assignments to the variables such that,

$$P\left(S_i = s_i, Y = y | X_i = x_i, S'_i = s'_i\right) \neq P\left(S_i = s_i, Y = y | S'_i = s'_i\right) \tag{4}$$

Therefore, S'_i is a Markov Blanket for X_i if it subsumes all of the predictive information about the target and the remaining features that is contained within X_i. One of the important properties of Markov Blankets is that features can be removed recursively. Koller and Sahami [12], showed that if features are only removed when a corresponding Markov Blanket is discovered, a feature that has been eliminated will not become relevant again as more features are removed.

The estimation of Markov Blankets can be difficult and algorithms, such as [12] and [11] use measures of correlation to find approximations of Markov Blankets. The measure of correlation varies between the standard linear correlation, which is limited to only identifying linear correlations in the data, and measures from information theory such as information gain, conditional entropy and symmetrical uncertainty. Roobaert et al. [10], use information gain as a measure of feature importance by calculating the reduction in entropy of the target caused by separating the data with the given feature. The information gain on the target Y, caused by the feature X_i is given by.

$$IG\left(Y|X_i\right) = H\left(Y\right) - H\left(Y|X_i\right), \tag{5}$$

where $H\left(Y\right)$ is the entropy of the class and $H\left(Y|X\right)$ is the conditional entropy. Care must be taken with this approach as information gain will favour features with more partitions .

This method evaluates the importance of a feature solely by examining the correlation to the target and consequently is equivalent to the definition of feature relevance given by Equation 1, which has already been shown to give incorrect results in certain situations.

The definitions of strongly and weakly relevant features are sufficient to describe the usefulness of the features concerned because they are based on the analysis of feature subsets, rather than individual features or pairs of features. There is always a possibility that redundancy and interaction can occur amongst larger feature subsets. This is one of the reasons for the accuracy of Wrapper methods but is also implemented by methods using random feature subset combination such as the Parcel algorithm of [13]. This algorithm regards classifiers

utilising diverse feature subsets as useful if they extend the convex hull over the ROC space.

When employing decision trees such as CART or Random Forest, estimates for the correlations between the features and the target are generated as part of the construction process in the form of information gain values. [14] uses these measures of feature importance to increase performance of the learning algorithm. Although these measures appear to be simpler forms of information gain, there are some benefits to using this method over standard information gain. Each terminal node in a decision tree can be viewed as a learner that has been trained on the features that were used in the path from the root. Consequently, the information gain values are not simply measures of the individual feature performance but measures of the ability of the feature in a variety of possible feature subsets. It is clear from the XOR example that if the data was split using one of the features and then split again using the other feature, that the second split would reveal the relevance of the feature. Therefore, there is some allowance for relationships between the features with this method. The advantage of this, as a technique for feature selection, over the random subspace method [7], is that multiple feature subsets can be evaluated within an individual learner, thus yielding a more efficient subset exploration. Also, because the decision trees are used for both the feature selection and learning processes, the bias of the learning algorithm is incorporated.

The problem with using the average information gain achieved by each feature, is that some nodes in the tree are easier to split than others. The number of ways a node can be split is determined by its composition and when smaller nodes are split, the measure of information gain is clearly more unreliable. The following introduces a measure to weight each value of information gain according to its reliability.

2.2 A Node Complexity Measure

This measure attempts to assign a value of reliability to a node by examining its composition and calculating the information associated with the splitting of such a node. For every split, the data is projected along a single feature. The assumptions made here are that once the data is projected into the one dimensional space, no data points lie on top of one another and that during the split optimisation procedure, all possible splits are found. Figure 1 shows the possible split positions for one node once the data is projected into the one dimensional space.

The problem is now a matter of considering how many possible arrangements of the data are possible. For binary classification problems, n is the number of examples contained in the node and i is the number of positive examples. The number of possible arrangements is given by the combinatorial function,

$$C_i^n = \frac{n!}{i!\,(n-i)!} \tag{6}$$

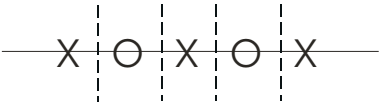

Fig. 1. Illustration of possible splits in one dimensional space of a node consisting of 3 data points from one class and 2 from the other

However, some of these arrangements are merely reflections of each other and will, therefore, result in the same optimised information gain. For example, a node containing only two examples, one of each class, will have two possible arrangements. The optimal split value is the same in both cases and this node can yield only one information gain value. Using the assumptions stated above, this example would be split perfectly by all features and would result in a maximum information gain value. Therefore, this illustrates the need for effective weighting of the nodes.

Not all of the arrangements have a reflected twin because some arrangements are symmetrical about their centre. These symmetrical arrangements shall be referred to as unique and their frequency designated by A_{u}. Their counterparts shall be referred to as non-unique and their frequency can be written $C_i^n - A_{\mathrm{u}}$. A_{u} can be calculated by considering the arrangements of half of the data and then taking the reflection to form the other half. This technique is dependant upon whether the values of n and i are odd or even and the corresponding functions are given in Table 1.

Table 1. Number of unique arrangements for node

n	i	A_{u}
EVEN	**EVEN**	$C_{\frac{i}{2}}^{\frac{n}{2}}$
ODD	**ODD**	$C_{\frac{i-1}{2}}^{\frac{n-1}{2}}$
ODD	**EVEN**	$C_{\frac{i}{2}}^{\frac{n-1}{2}}$
EVEN	**ODD**	0

The probability of a random occurrence of a particular unique arrangement, U_{x} is simply the probability of any particular arrangement,

$$P\left(U_{\mathrm{x}}\right) = \frac{1}{C_i^n} \tag{7}$$

As non-unique arrangements have two possible configurations, their corresponding probability, N_{x} is,

$$P\left(N_{\mathrm{x}}\right) = \frac{2}{C_i^n} \tag{8}$$

Assuming that the arrangements are random, the node complexity measure, NC, which is the information associated with the split of node l is,

$$NC\left(l\right) = -\frac{A_{\mathrm{u}}}{C_i^n}\log_2 P\left(U_{\mathrm{x}}\right) - \frac{C_i^n - A_{\mathrm{u}}}{C_i^n}\log_2 P\left(N_{\mathrm{x}}\right), \tag{9}$$

which can be simplified to,

$$NC\left(l\right) = \log_2 C_i^n - \left(1 - \frac{A_{\mathrm{u}}}{C_i^n}\right) \tag{10}$$

This is a suitable weight for calculating the average information gain because it represents the node complexity and therefore, how useful it is in identifying the predictive power of the feature.

3 The Feature Sampling Distribution

The measures of feature importance can be used to select the most relevant features but another application is to include all of the features in the learning process but weight their relevance to the problem according to their measure of importance. Random Forest can be adapted quite easily to achieve this. The standard Random Forest method chooses a feature at each split randomly from the set of all possible features. This feature sampling distribution is typically uniform but can be altered to incorporate the learned feature importance. By applying this technique, features that are deemed to be more important are chosen with a greater probability. CART trees consider all features at each stage of construction and choose the feature that provides the highest information gain. Altering the feature sampling distribution can be viewed as increasing the similarity of the randomly created trees to the ideal CART tree. If a feature selection algorithm is applied to Random Forest, then the class of possible trees that can be built is restricted and the diversity is reduced. By altering the feature sampling distribution, a trade off is introduced between increasing the strength of the base learners and maintaining the diversity of the ensemble. The goal is then to maximise the generalisation performance by optimising the feature sampling distribution in terms of these factors.

The alteration of the feature sampling distribution can be achieved in two ways. A two-stage method can be adopted, where an evaluation of the feature importance is conducted first and then applied to the construction of a Random Forest. Another approach is to combine the evaluation stage and the construction stage in a parallel scheme. As each tree in the forest is constructed, it can be used to evaluate the features and update the feature sampling distribution accordingly. The two-stage approach has the advantage of developing a reliable and accurate estimate of the ideal feature sampling distribution from which to build the forest. The parallel approach would be faster but has the problem of instability during the initial stages of the algorithm. When the forest is still small, there is very little information about the features from which to update

the sampling distribution. Initial overweighting of some features may create a sampling distribution that is far from ideal and the algorithm may not be able to recover from this as more trees are added. An implementation of the parallel method by [14] uses the measure of information gain as the feature importance metric. The weights of the features are updated according to,

$$w(X_i, m) = C \cdot I(X_i, 0) + \sum_{j=1}^{m} w(X_i, j), \tag{11}$$

where $w(X_i, m)$ is the weight assigned to feature i after construction of the m^{th} tree. $I(X_i, 0)$ is taken as the impurity of the whole data and C is a parameter which is used to control the rate at which the feature sampling distribution changes. By increasing the value of C the rate is decreased and the problem of initial overweighting is overcome. However, if C is too high, the sampling distribution will not change significantly and a forest very close to a standard Random Forest will be produced. Therefore, there is a need for tuning of the C parameter, which can typically be achieved using cross validation on the training data but the advantage of the small computational requirement is lost.

3.1 A Stable Parallel Method Using Confidence Intervals

A method of avoiding the cross validation stage would certainly be beneficial to the performance of the algorithm but a way of estimating the optimal convergence rate of the sampling distribution is required. The method introduced here, is to calculate a confidence interval for the estimate of expected information gain for each feature. Effectively, by observing the information gain values one is sampling from a distribution, which is assumed here to be normal. What is then required is the ability to approximate the probable distance between the mean of this normal distribution and the observed average information gain. Although the mean and variance of the true distribution are unknown, this can be accomplished by using the pivotal quantity method.

Given the sample mean (observed average information gain), $\overline{IG}$, the sample variance, S^2, the sample size, m and the true mean of the distribution μ. The pivotal quantity is,

$$\frac{\overline{IG} - \mu}{S/\sqrt{m}}, \tag{12}$$

and has a Student's t distribution with $m - 1$ degrees of freedom. A confidence interval can then be constructed within the distribution of the pivotal quantity.

$$P\left[q_1 < \frac{\overline{IG} - \mu}{S/\sqrt{m}} < q_2\right] = \gamma, \tag{13}$$

which gives the bound,

$$\overline{IG} - \frac{q_2 S}{\sqrt{m}} < \mu < \overline{IG} - \frac{q_1 S}{\sqrt{m}} \tag{14}$$

The process then consists of taking the observed information gains for each feature, calculating the sample mean and sample variance and deciding what level of confidence to use. A value of 0.95 for γ is typical. As the Student's t distribution is symmetrical, the optimal boundary values will be when $q_1 = -q_2$. These can be calculated from the value of γ by using an inverse Student's t distribution and then used to give the confidence interval around the sample mean.

If the sample mean is calculated using the weighted method then the sample variance must be weighted accordingly. Also, the sample size m must be re-examined, as a definition is required for a unit observation. A sensible value should be close to the information associated with the split of a node, averaged over all nodes in the tree. However, this is not known before the construction of the forest and must remain constant throughout.

These confidence intervals can then be used to update the feature sampling distribution by choosing values for each feature that lie within each confidence interval that yield the most uniform distribution. Here, the average information gain for each feature is viewed as assuming a value within a range of possible values, which are determined by the corresponding confidence interval. These average information gains can then be normalised and applied directly to set the feature sampling distribution, but their values must first be chosen such that they remain similar to each other and within their respective ranges. One simple method for achieving this, is to find the midpoint between the maximum lower bound and minimum upper bound of all of the confidence intervals. The value for each feature is then chosen to be as close to this value as possible without falling outside of the corresponding confidence interval.

This method will only update the feature sampling distribution when it has a confidence equal to γ. As more trees are added to the forest, the confidence in each estimate increases and the confidence intervals become smaller. Consequently, the feature sampling distribution becomes less uniform and closer to the ideal.

The confidence interval construction requires the calculation of an inverse Student's t distribution and the mean and variance of information gain for each feature. The experiments in this paper update the confidence intervals after the construction of every tree, in order to utilise the information concerning the features as soon as it is available. However, the computational load can be reduced, if desired, by updating the confidence intervals after a larger number of trees have been constructed. The cost of this is that some of the trees will be constructed using a feature sampling distribution that has not been created from all of the information that is available.

4 A Feature Selection Threshold

It is conjectured that the average information gain during the construction of decision trees is a measure of feature relevance. As previously discussed, it tests the feature on different areas of the input space and consequently accounts for the

different relationships between features. If these measures of feature importance are applied to learning algorithms which are based on decision trees, it also contains the bias of the learning algorithm. However, the worst performance of any given feature for the splitting of any given node, is that no reduction in entropy is possible and an information gain of zero is achieved. This means that the average information gain for any feature is the mean of a non-negative sample. The problem that arises from this, is that a feature which is completely irrelevant will produce some reductions in entropy purely by chance. Therefore, a non-zero feature importance value will be produced. This problem is particularly detrimental to performance when there are a relatively large number of irrelevant features and these values are used to update the feature sampling distribution. This is because, the probability of sampling any of the irrelevant features is the sum of all of their individual probabilities and although these may be small, the total can easily become significant if there are many. To overcome this problem, a feature selection threshold is introduced here, which approximates the expected information gain that is achieved by an irrelevant feature, given the size of the node being split.

Assuming that the task is binary classification and the data is projected onto a single feature, a node of size n, containing i positive examples has C_i^n possible arrangements. If the feature is irrelevant then these arrangements occur with equal probability. By constructing all of the possible arrangements and finding the maximum information gain, the expected value is calculated for various node constitutions and the outcome is shown in Figure 2.

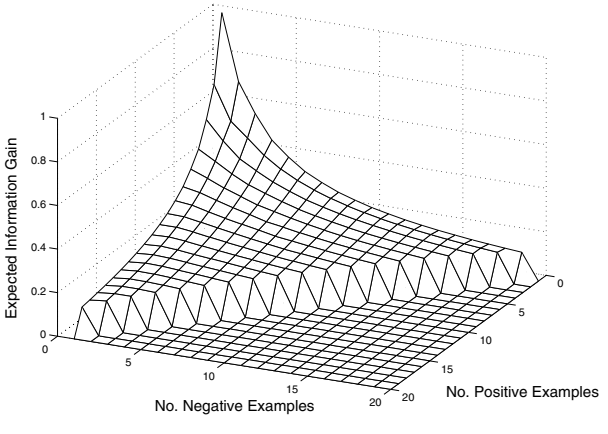

Fig. 2. Expected information gain for nodes containing various numbers of positive and negative examples

Due to the huge computational cost of evaluating the expected information gain in this manner, it is not a feasible method for feature selection, however, it can easily be approximated. For a fixed node size, the maximum expected

information gain appears to be when there are equal numbers of positive and negative examples and the minimum occurs when the ratio is most unbalanced. Therefore, the minimum expected information gain for a node of fixed size n, occurs when it contains only one example of one class, $i = 1$ or $i = n - 1$. This can be calculated in the following manner.

The information gain is the difference between the parent entropy and the combined child entropy and as the parent entropy for any given composition is fixed, only the combined child entropy needs to be considered. The case used here is that there is only one positive example, $i = 1$, and the optimal split leaves this example in the left node of size n_{l}. The right node then contains only negative examples and will have an entropy of zero. The combined child entropy CE is then,

$$CE = -\frac{n_{\text{l}}}{n}\left[\frac{1}{n_{\text{l}}}\log_2\frac{1}{n_{\text{l}}} + \frac{n_{\text{l}}-1}{n_{\text{l}}}\log_2\frac{n_{\text{l}}-1}{n_{\text{l}}}\right] \tag{15}$$

$$= -\frac{1}{n}\left[(n_{\text{l}}-1)\log_2(n_{\text{l}}-1) - n_{\text{l}}\log_2 n_{\text{l}}\right] \tag{16}$$

Differentiating by n_{l} then gives,

$$\frac{\partial}{\partial n_{\text{l}}}[CE] = \frac{1}{n}\left[\log_2 n_{\text{l}} - \log_2(n_{\text{l}}-1)\right] \tag{17}$$

For the case when n_{l} is not equal to 1 and consequently, must be a positive value of a least 2, the entropy is always increasing with n_{l}. Therefore, the optimal split is obtained when n_{l} is minimal. For a parent node of size n, the single positive example can assume only one of the possible n positions. If n is taken to be even, then by symmetry only $\frac{n}{2}$ of the arrangements need to be considered. The expected child entropy can then be written,

$$E[CE] = -\frac{2}{n}\sum_{n_{\text{l}}=1}^{\frac{n}{2}}\frac{n_{\text{l}}}{n}\left[\frac{1}{n_{\text{l}}}\log_2\frac{1}{n_{\text{l}}} + \frac{n_{\text{l}}-1}{n_{\text{l}}}\log_2\frac{n_{\text{l}}-1}{n_{\text{l}}}\right] \tag{18}$$

$$= -\frac{2}{n^2}\log_2\left[\prod_{n_{\text{l}}=1}^{n/2}\frac{(n_{\text{l}}-1)^{n_{\text{l}}-1}}{n_{\text{l}}^{n_{\text{l}}}}\right] \tag{19}$$

$$= \frac{1}{n}\log_2 n - \frac{1}{n} \tag{20}$$

Re-introducing the parent entropy, the expected information gain for a node of size n with only one example of one class, $E(IG_{\text{L}})$ can be written,

$$E[IG_{\text{L}}] = \frac{1}{n} - \frac{n-1}{n}\log_2\frac{n-1}{n} \tag{21}$$

The expected information gain for the case when there are equal numbers of each class cannot be calculated as easily. By examining the data that was generated for the construction of Figure 2 and plotting the expected information

gain for the case when the classes are equal on a logarithmic scale, it is seen that this quantity can be approximated by the following expression,

$$E\left[IG_{\mathrm{U}}\right] = \left(\frac{n}{2}\right)^{-0.82} \tag{22}$$

These two quantities can be viewed as upper and lower bounds on the expected information gain that is achieved by splitting a node of size n and are shown in Figure 3. The mid-point between these two bounds represents a reasonable estimate of the expected information gain of an irrelevant feature and can be applied as a feature selection threshold.

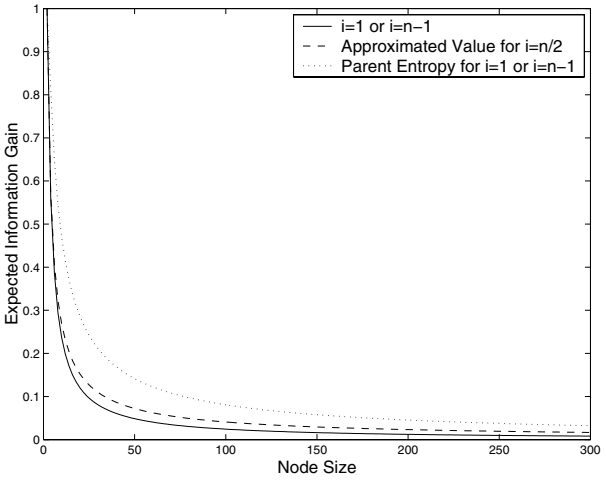

Fig. 3. Bounds on the expected information gain for varying node size. The parent entropy is also plotted for the case when $i = 1$ or $i = n - 1$ as this represents the maximum achievable information gain for this case.

5 Experiments

5.1 Datasets

The propeties of the data sets used in these experiments are shown in Table 2. The Wisconsin Breast Cancer (WBC), Pima Diabetes, Sonar, Ionosphere and Votes are available from the UCI Repository [15].

Simple is an artificial dataset consisting of 9 features and 300 examples. The output is generated according to the function,

$$Y = X_1^2 + 2X_2 \tag{23}$$

The remaining seven features are redundant and consequently this data set should benefit significantly from feature selection algorithms. It is important

Table 2. Data Set Properties

Data Set	No. Examples	No. Features	No. Relevant Features
WBC	683	9	?
Pima	768	8	?
Sonar	208	60	?
Ionosphere	351	34	?
Votes	435	16	?
Friedman	200	10	5
Simple	300	9	2

to note that as the input values to the function generator are randomly chosen between 0 and 1, feature 2 carries more predictive information than feature 1.

The Friedman dataset, [16] is another artificial dataset data set that is designed for testing feature selection algorithms. It is generated according to the following formula and has been thresholded in order to convert it into a binary classification problem. A threshold value of 14 was chosen to yield a reasonably balanced data set.

$$Y = 10\sin(\pi X_1 X_2) + 20\left(X_3 - \frac{1}{2}\right)^2 + 10X_4 + 5X_5 + N(0, 1.0) \tag{24}$$

5.2 The Node Complexity Measure

The Simple dataset is used to generate 5000 trees and the information gain values for all of the features are recorded. The values are discretised into intervals of size 0.01 so that each feature has a set of bins. As each node is split, the algorithm increments the bin corresponding to the feature being used and the information gain value obtained. As a comparison, one method increments the bins by a single unit, the other method increments by the measure of node complexity, $I(l)$. Incrementing by this value shows the effect of weighting the information gains in this manner. The results for three of the features are shown in Figure 4. The middle example is feature two, which carries the most information about the target. The left example is the next most important feature and the example on the right is a redundant feature.

It is particularly interesting to note the spikes that occur when unit weighting is used. At first glance, they appear to simply be noise but closer inspection reveals that they occur in the same places for all three features. The extreme right hand spike is the information gain that is achieved by the perfect split of a node made up from half of each class. This can occur when a node containing two examples, one from each class, is split. The spike immediately to the left of the maximum one can occur when a node containing two examples of one class and one of the other is split perfectly. These smaller nodes are much easier to split and can be split perfectly by features which carry very little information about the target. This is illustrated by the eradication of the spikes in the lower plots, where the sampling of the information gains is weighted according to

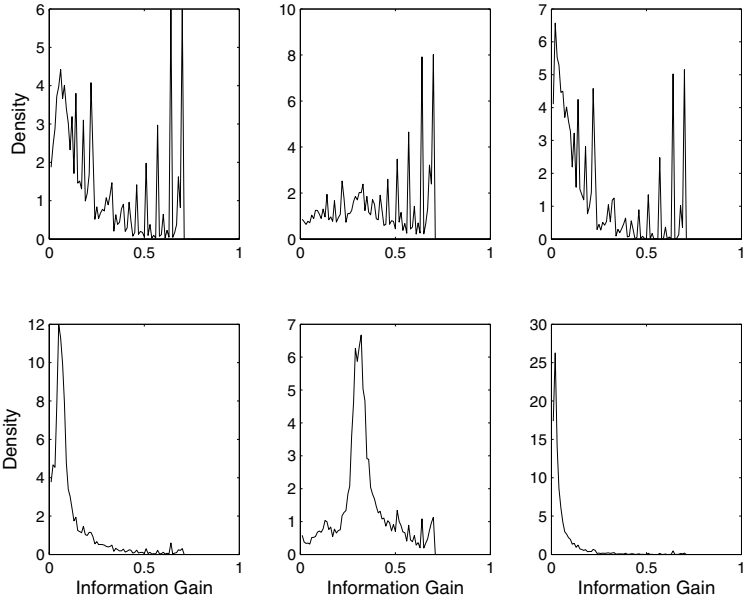

Fig. 4. Observed density functions of information gain for three features from the Simple dataset. Observed density using unit weighting (top) and observed density using node complexity weighting (bottom)

.

the node complexity. This result clearly demonstrates the ability of the node complexity measure to improve the estimate of feature importance by analysing the reliability of each sample.

5.3 Updating the Feature Sampling Distribution

The parallel method is employed to form confidence intervals on the estimates of feature importance and the feature sampling distribution is updated after every tree. The assumption that the information gain values are normally distributed is an approximation as the values are bounded on [0,1]. However, the shape of the distribution approximates normal if the node complexity weighting is used.

The initial feature sampling distribution is uniform and the algorithm keeps the most uniform distribution that is within the confidence interval of every feature. As the measures of information gain are weighted, a value for a unit of weight is required. This value should represent the information of the average split in a tree built on the dataset concerned. This value is approximated using a fraction of the node complexity of the entire data.

100 trials are conducted, using 90% of the data for training and 10% for testing. On each of the trials the rate of decrease of average confidence interval size is recorded and the result averaged over all of the trials. This represents

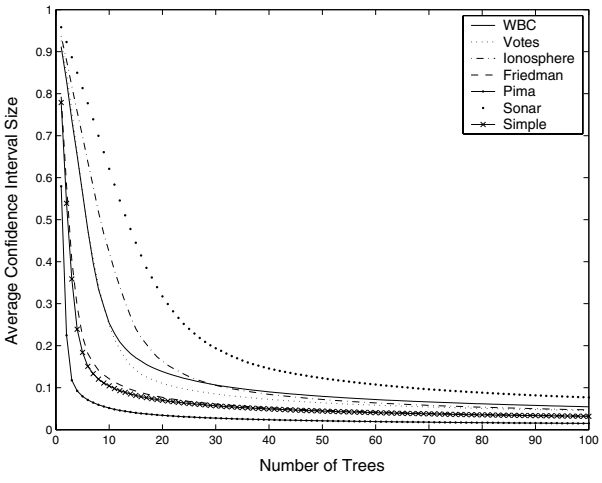

Fig. 5. Convergence rates for the feature sampling distribution. This shows how the average confidence interval size becomes smaller as more trees are added to the forest.

the rate of convergence towards the final feature sampling distribution and the results are shown in Figure 5.

The convergence rates vary with the size of the trees constructed and with the dimensionality of the data. The Pima data set has few features and produces large trees so the features are picked more often within each tree. In contrast, the Sonar data set has 60 features and produces smaller trees. Therefore, the convergence rates can still vary significantly, despite the incorporation of the prior knowledge.

A two-stage method is also applied where a single CART tree is built on the training data before construction of the forest to produce estimates of the feature importance. The average information gain is calculated using the measure of node complexity as before. However, by using a CART tree, each split supplies an information gain value for every feature, regardless of whether or not it is used. The forest is then constructed using the resultant fixed feature sampling distribution.

These methods are also compared to the CFS algorithm of [8], which selects a subset of features that have high correlation with the class and low correlation with each other. The selected features are then used to construct a Random Forest using a uniform feature sampling distribution.

Table 3 shows the error rates for standard Random Forest (RF), the CFS algorithm (CFS), weighted sampling using confidence interval method (CI WS RF) and the two-stage method using CART for evaluation (Two-stage CART).

Both methods of updating the feature sampling distribution improve the accuracy for some data sets. This improvement is most noticeable for the artificial data sets, which contain a number of irrelevant features. The confidence interval method does not significantly reduce the accuracy for any data set tested here,

Table 3. Test errors showing the improvement that three feature relevance identification techniques give to Random Forest construction. The values in brackets are the corresponding variances of test error over the 100 trials.

Data Set	RF	CFS	CI WS RF	Two-stage CART
WBC	0.0226(0.0003)	0.0235(0.0002)	0.0259(0.0003)	0.0226(0.0003)
Sonar	0.1657(0.0079)	0.2271(0.0066)	0.1462(0.0061)	0.1710(0.0069)
Votes	0.0650(0.0014)	0.0398(0.0007)	0.0493(0.0014)	0.0432(0.0008)
Pima	0.2343(0.0019)	0.2523(0.0024)	0.2394(0.0020)	0.2474(0.0018)
Ionosphere	0.0725(0.0017)	0.0650(0.0014)	0.0681(0.0018)	0.0661(0.0011)
Friedman	0.1865(0.0060)	0.1685(0.0055)	0.1690(0.0051)	0.1490(0.0052)
Simple	0.0937(0.0028)	0.1653(0.0044)	0.0450(0.0011)	0.0270(0.0009)

suggesting that the problem of initial over weighting of the features has been avoided. The two-stage CART method is shown to work well here, although there is no control over the reliability of this method and may encounter problems with certain data sets. Both methods compare favourably to the CFS algorithm, as CFS eliminates relevant features for some of the data sets, and consequently degrades the accuracy significantly.

5.4 Feature Selection Thresholding

To view the suitability of the measures of expected information gain as feature selection thresholds, 100 trees are constructed on the Simple dataset and the average information gain for each feature was recorded. The measures of expected information gain for irrelevant features are also calculated. Figure 6 shows that the seven irrelevant features are within the bounds that an irrelevant feature is expected to be in and the two relevant features are shown to be more important.

To approximate the expected information gain for a node of size n with any given composition, the mid-point between the two measures is used. Again, the data is split into 90% for training and 10% for testing. 100 trees are constructed on the training data for feature evaluation. During this, the average information gain is recorded and the approximate expected information gain is calculated. A further 100 trees are then constructed for classification of the test data. This process is repeated for 100 trials and the results averaged. This experiment is performed three times, the first experiment simply applies the recorded average information gain values to the feature sampling distribution (WS) and the second uses the expected information gain to select the relevant features but leaves the sampling distribution uniform (FS). The third experiment combines both methods by selecting the relevant features and altering the sampling distribution of the remaining features (WS & FS). Again, these results are compared to the CFS algorithm. The error rates for these experiments are shown in Table 4.

The results for the artificial datasets show an improvement when using feature selection, which is not surprising as it is known that they contain a significant number of irrelevant features. The Votes dataset shows the benefits of

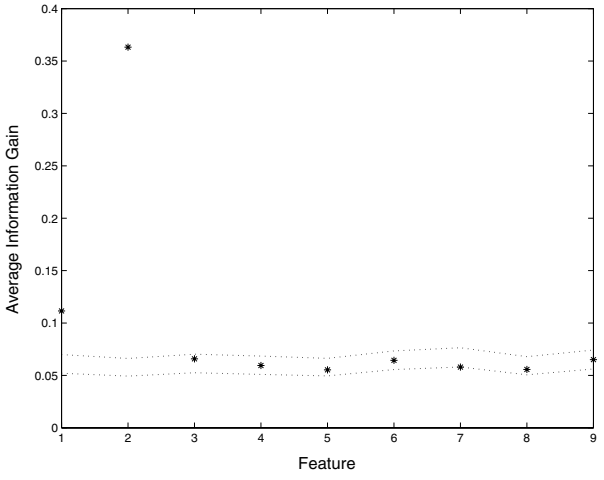

Fig. 6. The measures of feature importance for each feature in the Simple dataset and the approximate bounds for the expected values of irrelevant features. Features 1 and 2 are relevant, the remaining features are irrelevant.

Table 4. Comparison of CFS to three methods of applying the observed feature importances

Data Set	CFS	WS	FS	WS & FS
WBC	0.0235(0.0002)	0.0249(0.0003)	0.0245(0.0003)	0.0249(0.0003)
Sonar	0.2271(0.0066)	0.1757(0.0091)	0.1629(0.0060)	0.1643(0.0060)
Votes	0.0398(0.0007)	0.0464(0.0007)	0.0650(0.0013)	0.0439(0.0007)
Pima	0.2523(0.0024)	0.2312(0.0026)	0.2492(0.0021)	0.2486(0.0021)
Ionosphere	0.0650(0.0014)	0.0683(0.0017)	0.0747(0.0018)	0.0653(0.0016)
Friedman	0.1685(0.0055)	0.1555(0.0050)	0.1420(0.0060)	0.1370(0.0050)
Simple	0.1653(0.0044)	0.0393(0.0014)	0.0283(0.0009)	0.0303(0.0011)

feature weighting over the FS algorithm, as the error that is obtained from using feature selection only, is noticeably greater than when weighting is used. A problem with our feature selection approach is clearly encountered with the Pima dataset. This is most probably a consequence of the inaccuracy in the expected information gain value and the possible variance in the recorded information gain. The Pima dataset contains many features which are relevant, but whose relevance is very small. As a result of this, the recorded information gains for these relevant features are very close to the threshold and can easily fall below it. CFS also removes relevant features with this data set and consequently performs poorly. Using this threshold for average information gain as a feature selection algorithm, performs significantly better than CFS for some data.

6 Discussion

Random Forest is an ensemble algorithm, which improves the generalisation ability of weak learners by aggregation. The algorithm works well if the base learners from which the ensemble is comprised have a good generalisation ability and are diverse. The performance can be improved using ensemble feature selection, which chooses features that are not only predictive of the target and uncorrelated to each other but also promote diversity between the constructed hypotheses.

The average information gain achieved by each feature is shown to be a very reliable measure of feature importance if treated correctly. It is more than simply a measure of correlation as it tests the feature within different feature subsets and to an extent, accounts for interactions between the features. It also has a very small computational requirement, as the calculation is a product of forest construction. A method of weighting this average using a measure of node complexity is introduced and is shown to improve the accuracy considerably. By examining the distribution of the information gain, it appears reasonably normal.

Including all of the features and altering their sampling probabilities allows exploration of the trade-off between improving the accuracy of the base learners and maintaining the diversity within the ensemble. This can be performed by either using a parallel method or a two stage method. Parallel methods can suffer if the sampling distribution is updated too quickly. It is shown that by constructing confidence intervals on the estimates of feature importance, the rate of convergence can be controlled and the stability maintained. However, the rate of convergence is still dependent upon the dimensionality of the data and the resultant tree sizes.

A fast two stage method was introduced using a single CART tree to set the feature sampling distribution prior to forest construction. This method achieved surprisingly good results, although there are no bounds on the reliability of such a method. A logical next step would be to combine this method with a parallel one, where the CART tree could be used to initialise the feature sampling distribution. It could also provide prior knowledge concerning the average tree size, node complexity and number of features used. This would be useful for constructing the confidence intervals and controlling the convergence rates.

The convergence rate could also be controlled by altering the level of confidence used. Further work is required to find ways of identifying the optimal convergence rate in terms of the size of the constructed forest.

A threshold for feature selection was introduced here that approximates the performance of an irrelevant feature and performed well. It is clear that there are benefits to both feature selection and feature weighting algorithms. Irrelevant features degrade performance and need to be completely removed. The accuracy can be improved further by weighting the relevant features in order to reflect their relative importance. It has been shown here that both methods can be exploited to improve generalisation. A way of improving the measure of feature importance by identifying redundancies in the data should be investigated.

References

1. Freund, Y., Schapire, R.: A short introduction to boosting. Journal of Japanese Society for Artificial Intelligence **14** (1999) 771–780
2. Breiman, L.: Bagging predictors. Machine Learning **24** (1996) 123–140
3. Breiman, L.: Random forests. Machine Learning **45** (2001) 5–32
4. Dietterich, T.: An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. Machine Learning **40** (2000) 139–157
5. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification And Regression Trees. Wadsworth (1984)
6. Opitz, D.: Feature selection for ensembles. In: 16th National Conference on Artificial Intelligence, AAAI (1999) 379–384
7. Ho, T.: Nearest neighbours in random subspaces. In: Advances in Pattern Recognition. Volume 1451 of Lecture Notes in Computer Science., Springer (1998) 640–648
8. Hall, M.: Correlation-based feature selection for discrete and numeric class machine learning. In: 17th International Conference on Machine Learning. (2000) 359–366
9. John, G., Kohavi, R., Pfleger, K.: Irrelevant features and the subset selection problem. In Cohen, W., Hirsh, H., eds.: Machine Learning, Morgan Kaufmann (1994) 121–129
10. Roobaert, D., Karakoulas, G., Chawla, N.: Information gain, correlation and support vector machines. In Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L., eds.: Feature Extraction, Foundations and Applications, Springer (2005) In Press.
11. Yu, L., Liu, H.: Feature selection for high-dimensional data: A fast correlation-based filter solution. In: Machine Learning, AAAI (2003) 856–863
12. Koller, D., Sahami, M.: Toward optimal feature selection. In: International Conference on Machine Learning. (1996) 284–292
13. Scott, M., Niranjan, M., Prager, R.: Parcel: feature subset selection in variable cost domains. Technical report, Cambridge University Engineering Department (1998)
14. Borisov, A., Eruhimov, V., Tuv, E.: Tree-based ensembles with dynamic soft feature selection. In Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L., eds.: Feature Extraction, Foundations and Applications, Springer (2005) In Press.
15. Blake, C., Merz, C.: UCI repository of machine learning databases (1998)
16. Friedman, J.: Multivariate adaptive regression splines. The Annals of Statistics **19** (1991) 1–141