Creating Diverse Nearest Neighbour Ensembles using Simultaneous Metaheuristic Feature Selection

Muhammad Atif Tahir and Jim Smith

Department of Computer Science
University of the West of England
Bristol BS16 1QY, UK
{*muhammad.tahir,james.smith*}*@uwe.ac.uk*

Abstract

The nearest-neighbour (1NN) classifier has long been used in pattern recognition, exploratory data analysis, and data mining problems. A vital consideration in obtaining good results with this technique is the choice of distance function, and correspondingly which features to consider when computing distances between samples. In recent years there has been an increasing interest in creating ensembles of classifiers in order to improve classification accuracy. This paper proposes a new ensemble technique which combines multiple 1NN classifiers, each using a different distance function, and potentially a different set of features (feature vector).

These feature vectors are determined for each distance metric simultaneously using Tabu Search to minimise the ensemble error rate. We show that this approach implicitly selects for a diverse set of classifiers, and by doing so achieves greater performance improvements than can be achieved by treating the classifies independently, or using a single feature set. Naturally, optimising a the level of ensembles necessitates a much larger solution space, to make this approach tractable, we show how Tabu Search at the ensemble level can be hybridised with local search at the level of individual classifiers. The proposed ensemble classifier with different distance metrics and different feature vectors is evaluated using various benchmark data sets from UCI Machine Learning Repository and a real-world machine-vision application. Results have indicated a significant increase in the performance when compared with various well-known classifiers. Furthermore, the proposed ensemble method is also compared with ensemble classifier using different distance metrics but with same feature vector (with or without Feature Selection (FS)).

Key words: Tabu Search, 1NN classifier, Feature Selection, Ensemble Classifiers.

1 Introduction

The nearest-neighbour (1NN) classifier has long been used in pattern recognition, exploratory data analysis, and data mining problems. Typically, the k nearest neighbours of an unknown sample in the training set are computed using a predefined distance metric to measure the similarity between two samples. The class label of the unknown sample is then predicted to be the most frequent one occurring in the k nearest-neighbours. The 1NN classifier is well explored in the literature and has been proved to have good classification performance on a wide range of real-world data sets [1–4].

The idea of using multiple classifiers instead of a single best classifier gained significant interest during last few years. In general, it is well known that an ensemble of classifiers can provide higher accuracy than a single best classifier if the member classifiers are diverse and accurate. If the classifiers make identical errors, these errors will propagate and hence no accuracy gain can be achieved in combining classifiers.

In addition to diversity, accuracy of individual classifiers is also important, since too many poor classifiers can overwhelm correct predictions of good classifiers [5]. In order to make individual classifiers diverse, three principle approaches can be identified:

- Each member of the ensemble is the same type of classifier, but has a different training set. This often done in an iterative fashion, by changing the probability distribution from which the training set is resampled. Well known examples are bagging [6] and boosting [7].
- Training multiple classifiers with different inductive biasses to create diverse

classifiers, e.g. "stacking" approach [8].

- Using the same training data set and base classifiers, but employing feature selection so that each classifier works with a specific feature set and therefore sees a different snapshot of the data. The premise is that different feature subsets lead to diverse individual classifiers, with uncorrelated errors [9].

Specific examples of these three different approach can be found in the literature relating to Nearest-Neighbour techniques. Bao et al. [10] have followed the second route, and proposed an ensemble technique where each classifier used a different distance function. However, although this approach does use different distance metrics, they use the same set of features, so it is possible that some errors will be common, arising from features containing noise which have high values in certain samples. An alternative approach is proposed by Bay [11] following the third route: each member of the ensemble uses the same distance metric but sees a different randomly selected subset of the features.

Here we propose and evaluate a method which combines features of the second and third approaches. Building on [10,11], we explore the hypothesis that the overall ensemble accuracy can be improved if those choices of subsets arise from

- iterative heuristics such as tabu search [12] rather than random sampling
- different distance metrics rather than single distance metric

Furthermore we hypothesise that these choices are best co-adapted, rather than learnt separately, as co-adaptation may permit implicit tackling of the problem of achieving ensemble diversity. In order to do this, and to distinguish the effects of different sources of benefits, a novel ensemble classifier is proposed in this paper that consists of multiple 1NN classifiers each using a

different distance metric and a feature subset derived using tabu search. The proposed ensemble 1NN classifier (DF-TS-1NN) is then compared with various well-known ensemble classifiers. Two diversity measures namely "Plain Disagreement Measure" and "Entropy" [13] are also used to evaluate whether ensemble diversity can be achieved by using proposed ensemble 1NN classifier.

The rest of this paper is organized as follows. Section 2 provides a review on Feature Selection Algorithms. Section 3 describes the propose multiple distance function ensemble classifier followed by experiments in section 4. In section 5, a case study is discussed. Section 6 concludes the paper.

2 Brief Review of Feature Selection Algorithms

The term feature selection refers to the use of algorithms that attempt to select the best subset of the input feature set. It has been shown to be a useful technique for improving the classification accuracy of 1NN classifiers [14,15]. It produces savings in calculating the features (since some of the features are discarded) and the selected features retain their original physical interpretation [16]. Feature Selection is used in the design of pattern classifiers with three goals [16,17]:

(1) to reduce the cost of extracting features
(2) to improve the classification accuracy
(3) to improve the reliability of the estimation of performance

The feature selection problem can be viewed as a multiobjective optimization problem since it involves minimizing the feature subset and maximizing

classification accuracy. Mathematically, the feature selection problem can be formulated as follows. Suppose X is an original feature vector with cardinality F and $\bar{X}$ is the new feature vector with cardinality $\bar{F}$, $\bar{X} \subseteq X$, $J(\bar{X})$ is the selection criterion function for the new feature vector $\bar{X}$. The goal is to optimize J(). The problem is NP-hard [18,19]. Therefore, the optimal solution can only be achieved by performing an exhaustive search in the solution space [1]. However, exhaustive search is feasible only for small F. A number of heuristic algorithms have been proposed for feature selection to obtain near-optimal solutions [16,17,20–24].

The choice of an algorithm for selecting the features from an initial set depends on F. The feature selection problem is said to be of small scale, medium scale, or large scale according as F belongs to the intervals [0,19], [20,49], or [50,∞], respectively [17,22]. Sequential Forward Selection (SFS) [25] is the simplest greedy sequential search algorithm. Other sequential algorithms such as Sequential Forward Floating Search (SFFS) and Sequential Backward Floating Search (SBFS) are more efficient than SFS and usually find fairly good solutions for small and medium scale problems [21]. However, these algorithms suffer from the deficiency of converging to local optimal solutions for large scale problems when $F > 100$ [17,22]. Recent iterative heuristics such as tabu search and genetic algorithms have proved to be effective in tackling this category of problems which are characterised by having an exponential and noisy search space with numerous local optima [12,22,23,26].

Tabu search (TS) has been applied to the problem of feature selection by Zhang and Sun [22]. In their work, the tabu search performs the feature selection in combination with an objective function based on Mahalanobis distance. This objective function is used to evaluate the classification performance of

each subset of the features selected by the TS. Feature selection vector in TS is represented by a binary string where a 1 or 0 in the position for a given feature indicates that the presence or absence of that feature in the solution. Their experimental results on *synthetic data* have shown that the tabu search not only has a high possibility to obtain the optimal or near-optimal solution, but also requires less computational effort than the other suboptimal and genetic algorithm based methods. Later, Tabu Search has been successfully applied in other feature selection problems [15,27–29].

3 Proposed Ensemble Multiple Distance Function Classifier

In this section, we describe the proposed algorithm for constructing an ensemble of classifiers using multiple distance functions. For the purposes of this paper, each of the base classifiers is 1NN, we use different functions, and Tabu Search to optimise the feature set for each classifier, and so we denote this approach DF-TS-1NN. The use of n classifiers, each with a different distance function and potentially different set of features is intended to increase the likelihood that the errors of the individual classifiers are not correlated. In order to achieve this it is necessary to find appropriate feature sets *within the context of the ensemble as a whole*. However with F features and n different classifiers, the search space for the Tabu Search acting at the ensemble level is of size 2^{F*n}. Initial experiments showed that in order to make the search more tractable it is advantageous to hybridise the Tabu Search by incorporating into each iteration independent phases of local search. These act only within the sub-space of features for each classifier, and use the individual classifier's fitness. Figure 1 shows the hybrid training phase of the proposed classifier.

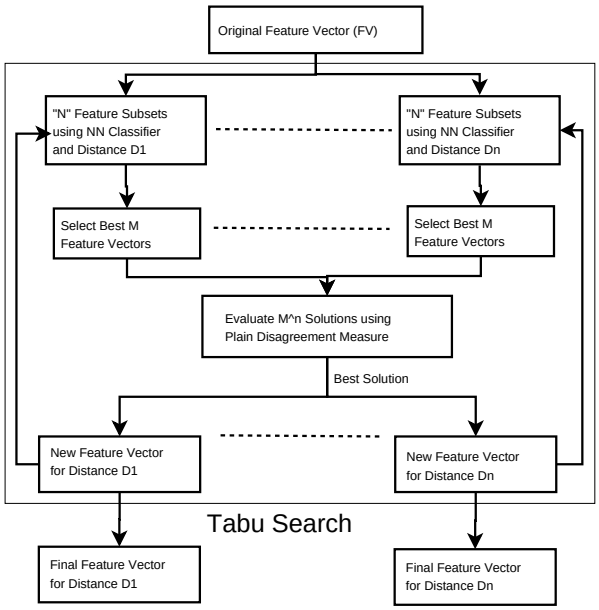

Fig. 1. Training Phase of proposed DF-TS-1NN classifier.

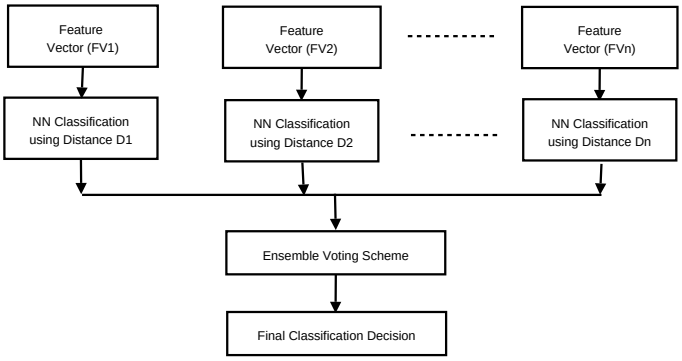

Fig. 2. Testing Phase.

The hybridisation with local search works as follows. During each iteration of Tabu Search, N random neighbours with *Hamming Distance* 1 from the current feature set FV_i are generated for each classifier $i \in \{1, \ldots, n\}$ and evaluated using the 1NN error rate for the appropriate distance metric D_i. From the set of N neighbours, the M best solutions are selected for each classifier. All M^n possible combinations are then evaluated using a simple voting scheme (SVS) and the best is selected to go forward to the next iteration. Considering $M > 1$ neighbours at the individual classifier level means that the feedback

from the SVS allows Tabu Search to iteratively search for *combinations* of feature vectors that improve the classification accuracy. Implicitly, this approach seeks feature vectors for the different distance measures whereby the errors are not correlated and so provides diversity - so it is possible that the selected combination might include a feature vector for one or more classifiers which do not have the best individual classifier accuracy.

The result of this Tabu Search training phase for the ensemble is a set of n feature vectors. These define the n classifiers in the ensemble which are combined for testing as shown in Figure 2.

3.1 Distance Metrics

The following five distance metrics, all widely used in the literature, are used within the 1NN classifiers to compute a distance between two m-dimensional vectors x and y.

- Squared Euclidean Distance: $E = \sum_{i=1}^{m}(x_i - y_i)^2$
- Manhattan Distance: $M = \sum_{i=1}^{m}(x_i - y_i)$
- Canberra Distance Distance: $C = \sum_{i=1}^{m}(x_i - y_i)/(x_i + y_i)$
- Squared Chord Distance: $S_c = \sum_{i=1}^{m}(\sqrt{x_i} - \sqrt{y_i})^2$
- Squared Chi-squared Distance: $C_s = \sum_{i=1}^{m}(x_i - y_i)^2/(x_i + y_i)$

3.2 Diversity Measure

Diversity is an important measure to evaluate the performance of an ensemble classifier [30]. In this paper, we have used two diversity measures namely "Plain

Disagreement Measure" and "Entropy" [13] to evaluate the impact of diversity in improving the classification accuracy using proposed ensemble classifier.

The plain disagreement measure is most commonly used pairwise measure for diversity in the ensemble of classifiers [13,31]. For two classifiers a and b, the plain disagreement is the fraction of the samples on which the classifiers make different predictions:

$$\text{Plain Disagreement} = \frac{1}{N_s}\sum_{k=1}^{N_s} Diff(C_a(s_k), C_b(s_k)) \tag{1}$$

where N_s is the number of samples in the data set, $C_i(s_k)$ is the class assigned by classifier i to sample k, and $Diff(x, y) = 0$, if $x = y$, otherwise $Diff(x, y) = 1$. This measure varies from 0 to 1. The measure is equal to 0, when the classifiers return the same classes for each instance, and it is equal to 1 when the predictions are always different [13].

Entropy is non pairwise measure for diversity in the ensemble of classifiers [13,32]. If S is the number of base classifiers, then the entropy is defined as:

$$\text{Entropy} = \frac{1}{N_s}\sum_{a=1}^{N_s}\sum_{b=1}^{C} -\frac{N_b^a}{S} * log(\frac{N_b^a}{S}) \tag{2}$$

where N_s is the number of samples in the data set, C is the number of classes and N_b^a is the number of base classifiers that assign sample a to class b. In order to keep this measure of diversity within the range [0,1] the logarithm should be taken to the base C.

3.3 Feature Selection and Diversity using Tabu Search

Tabu Search (TS) was introduced by Glover [33,34]. Starting from an initial solution, TS examines a set of feasible neighbouring solutions and moves to the best admissible neighbour, even if this causes the objective function to deteriorate. This process may permit escape from local optima, and provide a global search character. To avoid cycling, solutions that were recently explored are declared forbidden or tabu for a number of iterations. The tabu list stores characterization of the moves with lead to those solutions. The tabu status of a solution is overridden when certain *aspiration* criteria are satisfied [12].

For a data set with F Features, Tabu Search is run for T_s iterations with a Tabu list of size $T = ceil(\sqrt{F})$. As described above, local search is used to bias the sampling of the neighbourhood of the current solution. Thus in practice we examine a set of M^n neighbourhood solutions drawn from a set of size N^n where $N = ceil(\sqrt{F})$ [1]. The Aspiration criteria is deemed met if a solution has the lowest error rate yet seen. For a single classifier, each solution is represented by a binary vector of length F indicating the incorporation(1) or not(0) of the corresponding feature into the distance measurement for that classifier. For an ensemble of n classifiers, the solution vector therefore has length nF. All features are included in the initial solution. We use a 0/1 cost function, $C_{ij} = 1$ if datum i is misclassified, by classifier j otherwise zero. In order to estimate the 1NN error rate rates for each classifier in the local search phase, we apply $B-$-fold cross-validation.

[1] in the case that we evolve each classifier's feature subset independently, we consider N neighbours

Preliminary results using different subsets of the distance measures (i.e. $n < 5$) showed that showed that different combinations were better for different features sets. Therefore we believe that our approach of using all 5 is more generic. We set $M = 2$ and $T_s = 200$ for all data sets to reduce the computational burden - again preliminary results do not show benefits from increasing these.

Figure 3 shows an example showing neighbourhood solutions during one iteration. Let us assume that the cost of the three different feature subsets in the solution are 50, 48, and 47 using Distance Metrics 1, 2, and 3 respectively. $N = 4$ neighbours are then randomly generated for each distance metric using $HD1$. $M = 2$ best solutions are selected and $M^n = 2^3 = 8$ solutions are evaluated using ensemble cost function. The best solution is then selected for the next iteration.

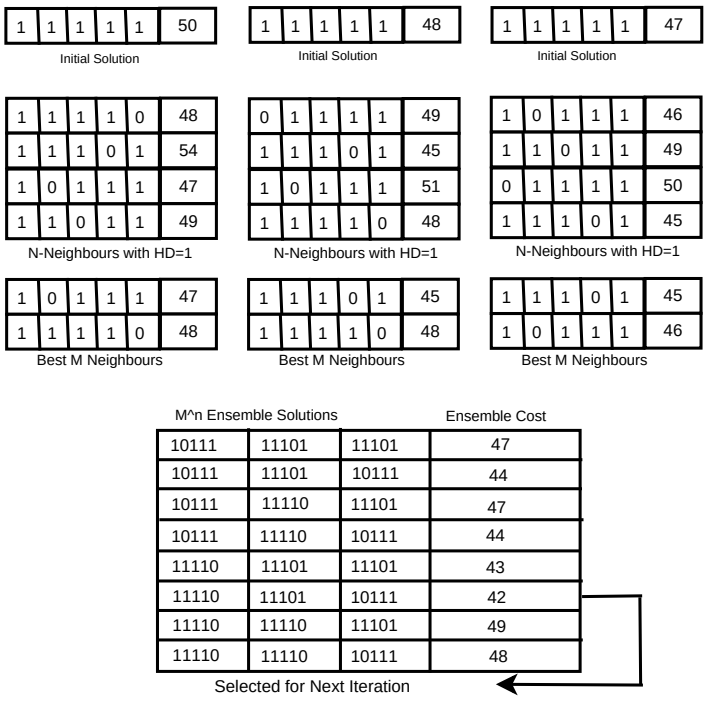

Fig. 3. An example showing neighbourhood solutions during one iteration in proposed tabu search method. $n = 3$, $N = 4$, and $M = 2$.

3.4 Algorithmic Cost

In each of the T_s iterations, $N \cdot n$ 1NN classifiers are created and used to classify the N_s samples. Since we apply $B-$ fold cross validation, obtaining a prediction for each data item requires $(B-1)/B \cdot N_s \cdot F$ calculations. The cost of the "local search" phase is thus $O\left(N \cdot n \cdot N_s^2 \cdot F\right)$. Since the predictions for each classifier-data item pair can be stored and can be accessed in linear time, the cost of evaluating the M^n classifier combinations and selecting the best to become the new incumbent solution is $O(M^n \cdot N_s)$. The total cost of creating the algorithm via Tabu Search is thus: $O\left(T_s \cdot (N \cdot n \cdot N_s^2 \cdot F + M^n \cdot N_s)\right)$. Typically in our work $N, M, n, F << N_s$, as are the combined terms $N \cdot n$ and M^n, so the cost is approximately $O(Ts \cdot N_s^2)$.

It should be noted that using modern multi-core processors it is simple to speed up the computational time considerably since the evaluation of the $M * n$ 1NN classifiers can be done independently in parallel. The computational time is dominated by the square of the number of data samples. In previous work [48] using Genetic Algorithms to select features for a Self-Organising Map [47], we have shown how data-set sub-sampling can be applied to greatly reduce the computational effort. It remains for future work to evaluate whether the use of such more rapidly computed approximates of the error rates can successfully be exploited within the Tabu Search metaheuristic.

Finally, we should note that this is the cost of creating an algorithm for a new dataset. As detailed later, for the purposes of the performance comparisons, we used repeated $B-$ fold cross-validation with multiple runs of Tabu Search, and so the computational time and effort was significantly larger.

4 Experiments

To evaluate the effectiveness of our method, extensive experiments were carried out to determine the best training method, and to benchmark the approach against several well known methods for creating single classifiers or ensembles.

4.1 Benchmark Methods

For comparison we used the following methods as implemented in WEKA [36].

- Decision Tree Method (C4.5): A classifier in the form of a tree structure, where each node is either a leaf node or a decision node [3,37].
- Random Forest (RF): Ensemble using a forest of random trees [38].
- Naive Bayes Algorithm (NBayes): The Naive Bayes Classifier technique is based on Bayesian theorem. Despite its simplicity, Naive Bayes can often outperform numerous sophisticated classification methods [39].
- Bagging: A method for generating multiple versions of a predictor and using these to get an aggregated predictor (ensemble) [6]. For the sake of completeness we evaluated the use of both C4.5 and 1NN as the base classifier.
- AdaBoost1: A meta-algorithm for constructing ensembles which can be used in conjunction with many other learning algorithms to improve their performance [7]. Again we used both C4.5 and 1NN as the base classifiers.
- Random Sub Space (RSS): This method generates an ensemble of classifiers, each using a pseudo randomly selected subsets of the features, that is, classifiers constructed in randomly chosen subspaces [31]. 1NN is used as the base classifier, thus this is equivalent to our DT-TS3-1NN algorithm but with the meta-heuristic learning component removed.

We considered the following variations of the proposed ensemble algorithms

(1) DF-1NN: Ensemble Classifier using NN classifiers with each classifier having different distance metrics (DF) but without feature selection.

(2) DF-TS1 -1NN: As above (1) but using Tabu Search (TS) to perform feature selection independently for each classifier.

(3) DF-TS2-1NN: As above (1) but with a single common feature set selected by Tabu Search based on the ensemble accuracy.

(4) DF-TS3-1NN: Proposed Ensemble classifier. As above (1) but with different feature subsets derived simultaneously for each classifier using TS.

4.2 Data sets Descriptions and Experimental Setup:

We used a range of datasets of different characteristics from the UCI [40]. These, along with the Tabu Search parameters, are described in Table 1.

Table 1
Data sets Description. N = Number of neighbourhood solutions sampled.

Name	N_s (size)	Features	Classes	T (Tabu List)	N
Australian	690	14	2	4	4
Breast Cancer	569	32	2	6	6
CMC	1473	9	3	3	3
Dermatology	358	34	6	6	6
Diabetes	768	8	2	3	3
German	1000	20	2	5	5
Heart	270	13	2	4	4
Ionosphere	351	34	2	6	6
Musk	476	166	2	13	13
SatImage	4435	36	6	6	6
Segment	2310	18	7	5	5
Sonar	208	60	2	8	8
Spectf	269	44	2	7	7
Vehicle	846	18	4	5	5

To estimate the predictive accuracy of classifiers it is necessary to split the available data into disjoint test and training sets, and it is well known that

the results obtained will depend on the particular choice of test/train split. Therefore in all data sets, repeated $B-$fold stratified cross validation has been used to estimate error rates [41,42].

For B-fold CV, each data set is divided into B blocks and B classifiers are trained, each using *(B-1)* blocks as a training set and the remaining block as a test set. Therefore, each block is used exactly once as a test set. The average accuracy of the B classifiers is used as the prediction of the accuracy. Evaluating the proposed approach requires both that the test set should never be seen by TS, and also an estimate of the value of a particular ensemble feature vector. Therefore during the search process the solution quality was estimated using $(B-1)-$fold CV with the current training set. As an example; if there are 100 samples, and $B = 10$; the data set is first divided into 10 different sets with each set consists of 90 training and 10 test samples. For each of the ten test sets, our proposed algorithm is trained, using as its objective function the 9-Fold CV accuracy on the remaining 90 training samples. The accuracy of the evolved feature subsets is then evaluated by classifying the 10 items in the the test set (which has never seen by TS) against the 90 training samples. This whole process is then repeated for each of the ten splits.

For greater statistical rigour, each experiment was run 5 times using different random 10-CV partitions [43]. The mean and standard deviations of these are presented below, and we also apply statistical hypothesis tests. The comparison results reported in section 4.4, also used five replicates of ten-fold cross validation with the same data splits.

In every case, since we are establishing whether there is a difference between a group of algorithms, we have begun by applying a two-way ANOVA with

the data set and algorithm as independent factors. If it is confirmed that with greater than 95% confidence the results from the different algorithms do not come from the same underlying distribution, we then apply post-hoc testing using Tamhane's T2 test (which does not assume equal variances) to establish whether the observed pair-wise differences are statistically significant.

In order to offset any bias due to the different range of values for the original features in 1NN classifier, they are normalized over the range [1,10] using Equation 3 [14], where $x_{i,j}$ is the j^{th} feature of the i^{th} pattern, $x'_{i,j}$ is the corresponding normalized feature, and as before N_s is the size of the data set..

$$x'_{i,j} = \left(\frac{x_{i,j} - min_{k=1...N_s} x_{(k,j)}}{max_{k=1...N_s} x_{(k,j)} - min_{k=1...N_s} x_{(k,j)}} * 10\right) \tag{3}$$

4.3 Comparison of Different ways of Creating Feature Sets

Table 2 shows the classification accuracy using various distance functions within single classifiers, and for the ensemble technique, all without feature selection. As can be seen, on some data sets there is a wide discrepancy between the accuracy obtained with different distance metrics. With the simple voting scheme the votes of the less accurate classifiers can dominate, so that the ensemble performs worse than the best single classifier on those datasets.

Table 3 shows the accuracy achieved when Tabu Search is used to perform feature selection of the individual classifiers. Comparing the results for individual classifiers with feature selection (TS-E,TS-M,TS-C,TS-C_s,TS-S_c) to those without (Table 2) it can be seen that feature selection always increases the accuracy.

Table 2
Mean and standard deviation of classification accuracy (%) for individual classifiers and ensemble, all without feature selection. M=Manhattan, E=Euclidean, C=Canberra, C_s = Chi-Squared, S_c = Squared-Chord.

Data Set	E	M	C	C_s	S_c	DF-1NN
Australian	79.7 (4.96)	79.8 (4.63)	**83.7** (3.5)	80.3 (4.46)	80.2 (4.15)	82.3 (4.15)
Breast Cancer	95.2 (2.42)	95.0 (3.05)	95.2 (2.76)	95.5 (2.45)	**95.5** (2.45)	**95.5** (2.48)
CMC	43.0 (2.82)	43.3 (3.36)	**45.6** (3.14)	44.2 (2.85)	44.9 (2.79)	44.3 (2.83)
Dermatology	95.4 (4.24)	96.0 (3.41)	96.3 (3.08)	**97.5** (2.80)	96.9 (2.99)	97.4 (2.84)
Diabetes	**70.3** (4.38)	69.6 (5.16)	65.6 (5.18)	69.3 (4.51)	69.4 (4.53)	69.8 (4.63)
German	70.5 (3.56)	71.1 (3.77)	70.2 (4.14)	70.5 (3.64)	70.0 (3.75)	**72.2** (3.54)
Heart	76.1 (7.71)	77.9 (7.00)	**79.0** (6.75)	76.7 (7.37)	76.1 (7.68)	76.9 (7.39)
Ionosphere	86.9 (5.22)	90.6 (4.69)	**92.2** (4.53)	89.0 (4.95)	88.6 (5.06)	90.2 (4.61)
Musk	85.8 (3.96)	83.9 (5.17)	84.7 (5.46)	**86.2** (3.91)	86.1 (3.84)	**86.2** (3.84)
SatImage	90.0 (1.09)	**90.5** (1.37)	90.3 (1.27)	90.2 (1.38)	90.1 (1.43)	90.4 (1.22)
Segment	97.2 (1.15)	**97.6** (1.02)	95.2 (1.64)	96.7 (1.16)	96.6 (1.24)	97.1 (1.04)
Sonar	83.0 (7.62)	85.0 (6.89)	**87.1** (6.53)	86.1 (6.28)	86.5 (6.35)	85.6 (6.69)
Spectf	70.1 (8.99)	**70.7** (7.71)	69.8 (8.80)	69.7 (8.88)	69.9 (9.15)	70.6 (9.54)
Vehicle	69.5 (4.06)	69.5 (4.01)	69.6 (3.92)	70.6 (3.50)	70.5 (3.74)	**70.7** (3.60)

Table 3
Mean and Standard Deviation of Classification Accuracy (%) using individual classifiers and with FS using TS. M=Manhattan, E=Euclidean, C=Canberra, C_s = Chi-Squared, S_c = Squared-Chord.

Data Set	TS-E	TS-M	TS-C	TS-C_s	TS-S_c
Australian	86.2 (3.35)	86.7 (3.35)	86.1 (2.94)	85.9 (3.49)	**86.9** (2.70)
Breast Cancer	96.8 (1.82)	**97.5** (1.82)	**97.5** (2.03)	97.0 (1.64)	97.1 (1.83)
CMC	48.5 (3.62)	48.2 (3.26)	**49.0** (3.46)	48.9 (3.57)	48.9 (3.61)
Dermatology	96.0 (3.85)	96.5 (2.89)	96.8 (3.1)	96.8 (2.80)	**97.0** (3.00)
Diabetes	71.1 (4.21)	71.0 (3.91)	**71.9** (3.61)	71.0(4.31)	71.3 (3.90)
German	73.3 (3.32)	74.0 (3.38)	73.8 (3.38)	**74.5** (3.20)	73.1 (3.32)
Heart	80.3 (5.82)	81.6 (5.67)	82.4 (5.58)	**82.5** (5.89)	82.0 (5.08)
Ionosphere	93.6 (3.94)	95.4 (3.50)	**95.9** (3.65)	92.9 (4.18)	93.9 (4.35)
Musk	89.8 (3.26)	89.0 (4.02)	89.6 (4.22)	**90.1** (3.78)	89.2 (3.28)
Satimage	**91.2** (0.96)	91.4 (1.00)	91.1 (1.01)	91.0 (1.00)	91.1 (1.13)
Segment	97.8 (0.94)	**98.0** (0.87)	97.7 (0.92)	97.9 (0.86)	97.9 (0.90)
Sonar	85.2 (6.51)	87.2 (7.55)	**90.7** (6.04)	90.1 (5.43)	87.6 (6.89)
Spectf	82.0 (6.40)	82.2 (5.66)	**82.7** (6.66)	81.0 (6.75)	**82.7** (6.65)
Vehicle	74.0 (3.77)	**74.7**(4.29)	74.0 (3.16)	74.1 (3.21)	73.5 (3.60)

Table 4 shows the classification accuracy obtained using different variations on the way that feature selection is performed for the ensemble. This shows that the use of feature selection to derive a common subset for all classifiers (DF-TS2-1NN) results in improved performance compared to the same algorithm

without feature selection (DF-1NN in Table 2), but now the mean accuracy is higher than the best individual classifier on most data sets. This is a good example that indicates that in order for ensembles to work well - the member classifiers should be accurate.

The other condition for ensembles to work well is diversity, and the performance improves further when feature selection is done independently for each classifier (DF-TS1-1NN), as they can now use potentially different feature sets. However, this approach only implicitly (at best) tackles the diversity issue, and the performance is further increased when different feature subsets co-adapt, so that each feature set is optimized in the context of the ensemble as whole (DF-TS3-1NN).

Comment: text to follow in seperate email from Jim once he has analysed the data

Table 4
Mean and Standard Deviation of Classification Accuracy (%) using various variations of the proposed classifier.

Data Set	DF-TS1-1NN	DF-TS2-1NN	DF-TS3-1NN
Australian	88.4 (3.53)		**89.1 (3.34)**
Breast Cancer	97.3 (1.83)		**97.5 (1.71)**
CMC	49.8 (3.72)		**52.8 (3.20)**
Dermatology	**97.5 (2.60)**		97.3 (2.79)
Diabetes	73.9 (4.21)		**76.5 (2.43)**
German	76.6 (3.73)		**77.7 (2.90)**
Heart	84.0 (5.42)		**86.1 (4.65)**
Ionosphere	94.9 (3.93)		**95.0 (3.75)**
Musk	90.7 (3.23)		**91.7 (2.78)**
Satimage	**92.5(0.85)**		92.3 (0.98)
Segment	98.2 (0.80)		**98.8 (0.60)**
Sonar	90.0 (5.50)		**90.6 (5.41)**
Spectf	**84.7 (6.50)**		83.5 (7.35)
Vehicle	75.8 (4.01)		**76.2 (3.60)**

Table 5 shows the number of features used by proposed classifier on a typical (randomly chosen) run from each dataset. Different features have been used

by the individual classifiers that are part of the whole ensemble classifier, thus increasing diversity and producing an overall increase in the classification accuracy. F_{Common} represents those features that are used by every classifier in the ensemble while $F_{Ensemble}$ is the total number of features used in the ensemble. As can be seen on most data sets there are few, if any, features that are used by every classifier. This is a cause of diversity amongst the decision of the different classifiers. The fact that these feature sets are learned rather than simply assigned at random is responsible for the different classifiers all remaining accurate - the other pre-requisite for successful formation of an ensemble. This issue is explored further in the next section.

Tables 6 and 7 show the diversity using "Plain Disagreement" and "Entropy" for the four variations of the proposed ensemble methods, averaged over all 50 runs (5 repeats of ten-fold CV). It is clear from these tables that both pairwise and non-pairwise diversity measures are high in all datasets except

Table 5
Total Number of Features used by proposed classifier. F_T = Total Available Features, F_M= Feature using Manhattan Distance, F_E=Features using Euclidean Distance, F_C=Features using Canberra Distance, F_{C_s} = Features using Chi-Squared Distance, F_{S_c} = Feature using Squared-Chord Distance.

Data Set	F_T	F_E	F_M	F_C	F_{C_s}	F_{S_c}	F_{Common}	$F_{Ensemble}$
Australian	14	5	9	9	7	5	1	14
Breast Cancer	32	19	13	15	21	13	3	28
CMC	9	6	6	6	6	4	2	9
Dermatology	34	19	23	21	15	21	8	32
Diabetes	8	3	5	1	3	5	0	8
German	20	9	13	13	13	15	3	19
Heart	13	10	8	10	6	8	2	13
Ionosphere	34	11	13	15	11	11	2	26
Musk	166	84	74	76	86	90	0	124
Segment	18	6	12	12	10	8	2	17
SatImage	36	24	22	24	16	24	5	36
Sonar	60	31	33	27	35	33	0	58
Spectf	44	22	14	16	20	24	1	30
Vehicle	18	9	11	13	7	13	0	17

Table 6
Diversity using "Plain Disagreement Measure".

Data Set	D_{All} DF-1NN	$D_{Ensemble_1}$ DF-TS1-1NN	D_{Common} DF-TS2-1NN	$D_{Ensemble_2}$ DF-TS3-1NN
Australian	0.1088	0.1331		**0.1810**
Breast Cancer	0.0151	0.0265		**0.0432**
CMC	0.1424	0.2184		**0.4660**
Dermatology	0.0276	0.0360		**0.0450**
Diabetes	0.1147	0.2428		**0.3371**
German	0.1660	0.2552		**0.3027**
Heart	0.0748	0.1516		**0.2370**
Ionosphere	0.0526	0.0527		**0.08460**
Musk	0.0761	0.0866		**0.0978**
Satimage				
Segment	0.0311	0.0197		**0.0374**
Sonar	0.0883	0.1275		**0.1433**
Spectf	0.1184	0.2140		**0.2486**
Vehicle	0.1463	0.2022		**0.2739**

Table 7
Diversity using "Entropy".

Data Set	D_{All} DF-1NN	$D_{Ensemble_1}$ DF-TS1-1NN	D_{Common} DF-TS2-1NN	$D_{Ensemble_2}$ DF-TS3-1NN
Australian	0.1822	0.2106		**0.3104**
Breast Cancer	0.0278	0.0400		**0.0738**
CMC	0.1552	0.2534		**0.5269**
Dermatology	0.0174	0.0225		**0.0302**
Diabetes	0.1955	0.3965		**0.5708**
German	0.2825	0.4271		**0.5138**
Heart	0.1283	0.2370		**0.4034**
Ionosphere	0.0866	0.0790		**0.1450**
Musk	0.1279	0.1476		**0.1668**
Satimage				
Segment	0.0315	0.0118		**0.0232**
Sonar	0.1493	0.2085		**0.2433**
Spectf	0.1904	0.3476		**0.42146**
Vehicle	0.1268	0.1705		**0.2403**

Musk when selecting feature subsets for all of the classifiers simultaneously (DF-TS3-1NN). Thus, diversity plays an important role increasing the classification accuracy of various data sets using proposed ensemble technique. Since Musk has 166 features; we argue that feature selection alone plays an important role in improving the classification accuracy. Further, in many cases using

a common feature set (DF-TS2-1NN) actually reduces the diversity compared to not doing FS at all (DF-1NN), even though accuracy of DF-TS2-1NN is higher than accuracy of DF-1NN. In contrast; diversity does increase a little when FS done independently (DF-TS1-1NN) and thus justify our use of local search done independently in our hybrid TS algorithm.

4.4 Comparison with other algorithms

Table 8 shows the comparison of accuracy (in %) between the proposed DF3-TS-1NN classifier and others for different data sets. These results can be summarised as follows:

- On 12 of the 14 data sets DF-TS3-1NN produces the highest mean accuracy.
  - On 3 datasets (Australian, German, Segment) it is significantly better than all other algorithms.
  - On 4 datasets it is statistically better than all but one other method (RSS-1NN on Breast Cancer, Naive Bayes on Heart, Random forest on Ionosphere, AdaBoost on Musk).
  - Most of pairwise tests show DF-TS3-1NN is significantly better than the other method.
- On 2 datasets another method had a higher mean accuracy:
  - but this difference is not statistically significant,
  - and DF-TS3-1NN is significantly better than most other algorithms.
- With the exceptions of RSS on SatImage, AdaBoost/Bagging on Sonar, the DF-TS3-1NN significantly outperforms not just 1NN on every dataset, but also the alternative methods for producing ensembles of 1NN classifiers (AdaBoost, Bagging and Random Sub-Space).

- The significant performance advantages over Random SubSpace search indicate that learning is truly taking place during the Tabu Search phase.

Table 8
Average Classification Accuracy (%) using different classifiers. RF = Random Forest, RSS=Random Sub-Space Search. Bold type indicates algorithm with highest mean accuracy per dataset. Use of a * indicates that the difference in fitness between the best and all other clasifiers is significant with more than 95% confidence.

Dataset	C4.5	RF	NBayes	Bagging (C4.5)	AdaBoost (C4.5)	Bagging (1NN)	AdaBoost (1NN)	1NN	RSS -1NN	DF-TS3 -1NN
Aust.	84.52	85.43	77.18	85.93	84.97	79.86	77.34	82.33	82.67	**89.11***
	(3.88)	(4.30)	(3.79)	(3.57)	(4.23)	(4.59)	(4.53)	(4.15)	(4.41)	(3.34)
Breast	93.36	95.90	93.38	95.13	95.67	95.55	93.73	95.16	96.52	**97.54**
	(3.73)	(2.37)	(2.74)	(2.80)	(2.55)	(2.26)	(3.10)	(2.42)	(2.61)	(1.71)
CMC	50.35	50.56	49.02	**53.25**	51.61	43.21	43.02	42.95	46.38	52.75
	(4.05)	(3.48)	(3.93)	(3.73)	(2.81)	(2.77)	(3.26)	(2.82)	(2.25)	(3.24)
Derm.	95.77	96.83	97.28	96.83	96.87	95.42	92.66	95.44	96.66	**97.28**
	(2.57)	(2.86)	(3.06)	(2.72)	(3.24)	(4.10)	(4.53)	(4.24)	(3.41)	(2.79)
Diabet.	74.01	75.07	75.67	75.15	71.44	70.51	67.55	70.27	70.89	**76.45**
	(4.96)	(5.50)	(5.21)	(5.62)	(4.90)	(4.13)	(4.75)	(4.38)	5.32)	(3.43)
German	72.26	74.68	74.64	74.90	73.18	71.28	68.08	70.54	73.16	**77.66***
	(4.02)	(3.08)	(3.22	(3.25)	(3.57)	(3.76)	(3.49)	(3.56)	(3.54)	(2.90)
Heart	78.89	80.44	84.44	79.56	79.19	75.85	75.93	76.15	80.44	**86.07**
	6.87	(6.26)	(6.08)	(6.62)	(6.81)	(7.42)	(7.75)	(7.71)	(6.98)	(4.65)
Ionos.	89.52	93.04	82.44	91.86	92.33	86.73	87.78	86.92	89.80	**95.01**
	(3.21)	(4.37)	(5.79)	(4.31)	(4.02)	(4.98)	(5.38)	(5.22)	(4.87)	(3.75)
Musk	82.89	88.12	73.34	87.73	89.78	86.13	87.06	85.76	87.89	**91.65**
	(5.66)	(4.69)	(7.41)	(4.29)	(4.24)	(4.37)	(4.39)	(3.96)	(3.45)	(3.05)
SatImg	86.44	90.39	79.54	89.94	90.09	89.96	88.67	89.96	91.13	**92.33**
	(1.46)	(1.23)	(1.68)	(0.81)	(1.50)	(1.19)	(1.64)	(1.09)	(0.89)	(0.98)
Segmnt	96.93	97.83	80.04	97.60	98.35	97.06	96.73	97.18	97.20	**98.83***
	(1.04)	(0.98)	(1.71)	(0.93)	(0.79)	(1.20)	(1.29)	(1.15)	(1.20)	(0.60)
Sonar	72.40	80.91	67.73	77.74	81.16	86.90	86.36	83.01	88.64	**90.63**
	(9.11)	(8.10)	(9.43)	(7.64)	(8.19)	(6.79)	(7.05)	(7.62)	6.68	(5.41)
Spect	76.03	79.58	68.08	79.85	79.60	69.80	67.11	70.11	73.33	**83.51**
	(7.01)	(7.25)	(10.02)	(5.55)	(6.75)	(8.48)	(8.76)	(8.99)	(9.21)	(7.35)
Vehicle	73.37	74.77	45.09	74.76	**76.47**	69.85	68.29	69.51	71.25	76.24
	(4.08)	(4.09)	(4.20)	(3.63)	(3.87)	(4.31)	(3.72)	(4.06)	(4.19)	(3.60)

Tamhane's T2 test is a conservative post-hoc test. Use of the more common LSD test indicates that more of the increases in accuracy between DF-TS3-1NN and other methods are statistically significant. A two-way Analysis of Variance, with the algorithm and dataset as independent factors, and the ac-

curacy as the dependent variable confirmed that both of the factors the are statistically significant with over 95% confidence. Using Tamhane's T2 test was used to perform a post-hoc pairwise comparison between the different algorithms after the effects of the data set had been factored out confirmed that the performance of DF-TS3-1NN was better with more than 95% confidence.

Figure 4 shows the standard deviation of each data set for different algorithms. From the graph, it is clear that the standard deviation of the proposed classifier compares favorably with other algorithms. In particular; the standard deviation is almost same for all algorithms in which 1NN is used as base classifier.

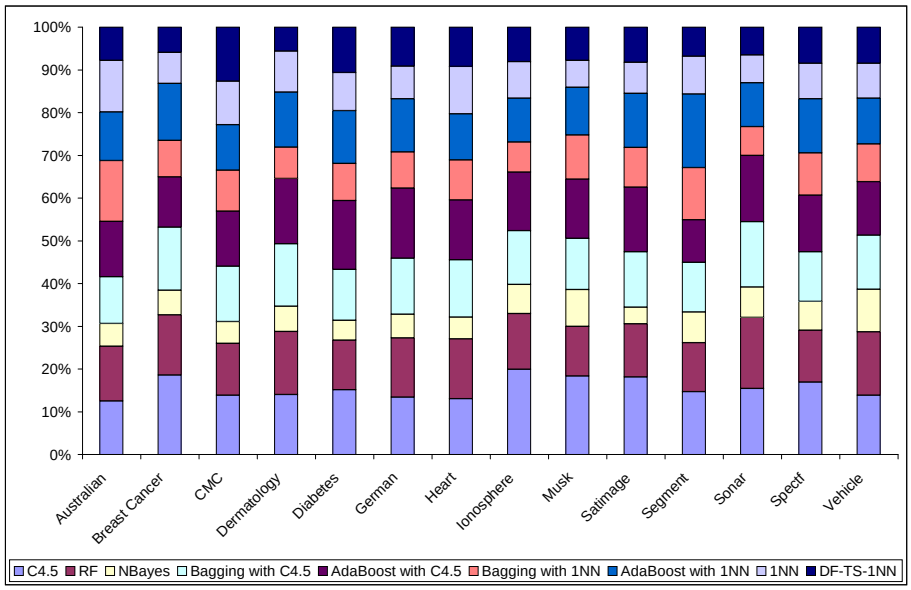

Fig. 4. Standard deviation as 100% stacked column for different algorithms on various data sets .

Comparisons with, or even definitions of "state of the art" are always difficult in a rapidly changing field, where it is not always possible to replicate algorithms. In order to indicate the relative merit of the approaches tested, Table 9 shows a comparison between the results obtained with our approach, and the best results found on on-line comparison site maintained by the Nicolaus Copernicus University [44]. This website provides a comprehensive comparison of many different algorithms on a range of data sets using various methods of error estimation. Wherever possible we have quoted the best given results from repeated n-fold cross validation and reported their standard deviations. As one would expect from the No Free Lunch theorem, this best result is not always obtained with the same "state of the art" classifier, and we have reproduced published results here, so statistical hypothesis testing was not performed. In some cases, results are taken from the statlog project, which used a single n-fold cross validation and does not report the variation between runs. In one case the only results available were for a test/train methodology. These last two groups of results should therefore be treated with increasing caution.

As can be seen our approach gives a higher mean n-fold c.v. accuracy on 6 of the 9 data sets for which we have results. We are unable to apply rigorous hypothesis testing, but on the basis of the published standard deviations, it would appear that only for the case of the Sonar and possible the vehicle datasets are the best published results likely to be significantly better.

Table 9
Comparison of mean results from multiple runs of n-fold cross validation between DF-TS3-1NN and the best results from literature found on [44]. Standard deviations between runs of n-fold cross validation are given where available. If not *sl* indicates the source is the statlog project, which only used 1 run, or *test* indicates results were only available for a holdout method. Final Column indicated algorithm used with standard acronyms

Data Set	DF-TS3-1NN		Literature		
	mean	std. dev.	mean	std. dev	Algorithm
Australian	89.11	3.34	86.9	*statlog*	Cal5
Breast Cancer	97.54	1.71	97.5	1.8	Naive Bayes
Diabetes	76.45	3.43	77.7	*statlog*	Log. Disr.
Heart	86.07	4.65	84.9	0.7	SVM
Ionosphere	95.01	3.75	94.9	2.6	kNN-Simplex
SatImage	92.33	0.98	91.0	*test*	MLP
Segment	98.83	0.60	97.2	*statlog*	kNN-Manhattan
Sonar	90.63	5.41	99.8	0.1	MLP-BP
Vehicle	76.24	3.60	85.0	*statlog*	Quad. Discr.

4.6 Analysis of Learning

Figures 5- 7 show the classification accuracy (%) vs number of iterations for Australian, Ionosphere and German data sets using one run of the solution search space using TS. The figure clearly indicates that TS focuses on a good solution space. The proposed TS algorithm progressively zooms towards a better solution subspace as time elapses; a desirable characteristics of approximation iterative heuristics.

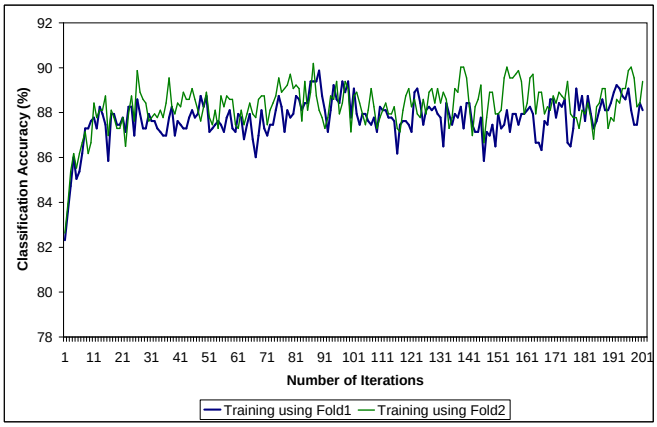

Fig. 5. Error Rate vs Iterations for Australian Data set using 2 different Folds.

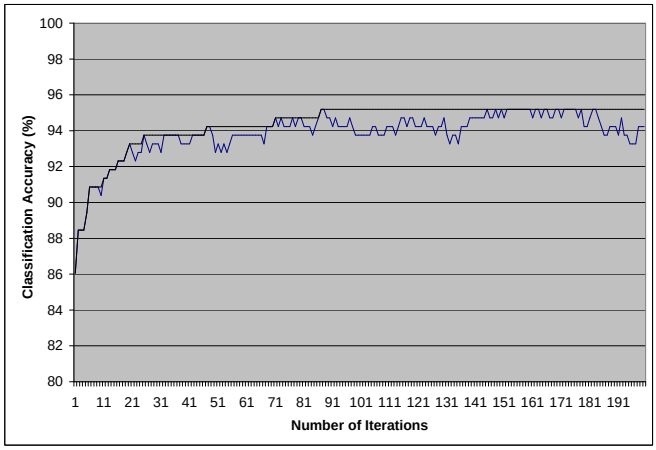

Fig. 6. Error Rate vs Iterations for Ionosphere Data set.

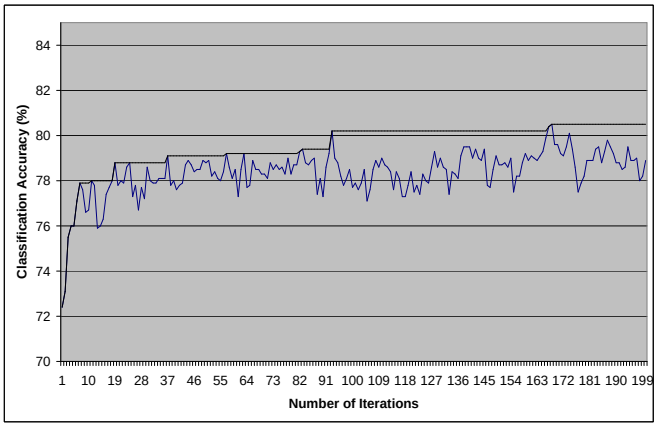

Fig. 7. Error Rate vs Iterations for German Data set.

5 Case Study: Industrial Application (CD Print Data)

The most extensive application of our proposed technique is as part of the "DynaVis" [45] automatically self-reconfigurable and adaptive fault detection framework for manufacturing quality control shown in Figure 8. This framework classifies each image as good or bad, and adapts the classifier on-line in response to the operator's feedback. The particular example we will demonstrate here concerns inspecting the printing of images and text of CDs and DVDs, the objective being to detect faults due to weak colours, incorrect palettes etc. For this print application, a "master" image is available and the approach taken is to subtract this from the image of each produced part so as to generate "contrast" images which are then characterised according to the structure and characteristics of the deviation pixels.

One aspect of the DynaVis system is the recognition that not only will factors such as fatigue cause inconsistencies in the labels applied by individual operators, but there will also be systematic differences in the decisions made by different operators arising from factors such as inexperience and different roles in the organisation. In order to cope with this the system builds a model of each operator and uses an weighted voting technique to combine these - a

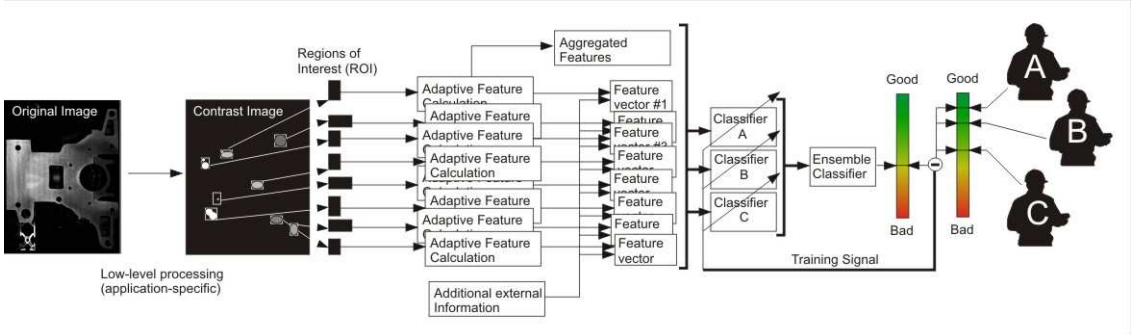

Fig. 8. Classification Framework for classifying images into good or bad.

true "mixture of experts". Therefore it is necessary to create classifiers which model the decisions made by each operator as closely as possible. It is also necessary for the technique used to combine these classifiers to take account of the fact that, since each operators may display different levels of inconsistency, the classifier(s) modelling them will have different levels of predicted accuracy. It is important to clarify that the feature-selection and training of the proposed ensemble classifier is performed off-line, and is done independently for each individual operator. During online processing, each image is classified by the set of n 1NN classifiers and a decision made for that particular operator. This whole process is done in parallel for each operator before the final results are combined.

The data set consists of 1534 images, each labeled by 4 different operators. For each image, 74 aggregated features are extracted, describing the distribution, density, shape etc. of the pixel fragments in the deviation images. Table 5 shows the classification accuracy obtained when using the proposed algorithm to train a classifier modelling the decisions of each different operator. The results clearly indicate a significant increase in classification accuracy compared with a range of other well-known techniques, and also illustrate how the different levels of consistency lead to differences in accuracies between operators. Figure 9 shows the standard deviation obtained over the 100 runs of random 10-fold cross validation for different operators. Again from the graph, it is clear that the standard deviation is almost same for all algorithms in which 1NN is used as base classifier.

Comment from jim: do we have the 100 classifications accuracies for each operator with each algorithm? If so I may as well run a proper stat analysis on them. This part will be finished on Thursday.

Table 10
Average Classification Accuracy (%) using different classifiers for real recorded CD images. RF = Random Forest. Bag = Bagging. Ada = AdaBoost

	Good/Bad	C4.5	RF	NBayes	Bag C4.5	Ada C4.5	Bag 1NN	Ada 1NN	1NN	DF-TS3 -1NN
Op1	1164/370	92.5	94.1	87.1	93.8	93.7	93.8	93.3	92.7	**95.6**
Op2	1262/272	95.5	96.6	92.4	96.5	96.8	96.4	95.8	95.8	**98.1**
Op3	1230/304	94.0	95.2	89.0	95.1	95.2	95.3	94.6	93.4	**96.0**
Op4	1223/311	95.0	95.8	90.4	95.9	95.7	96.2	95.8	94.8	**97.7**

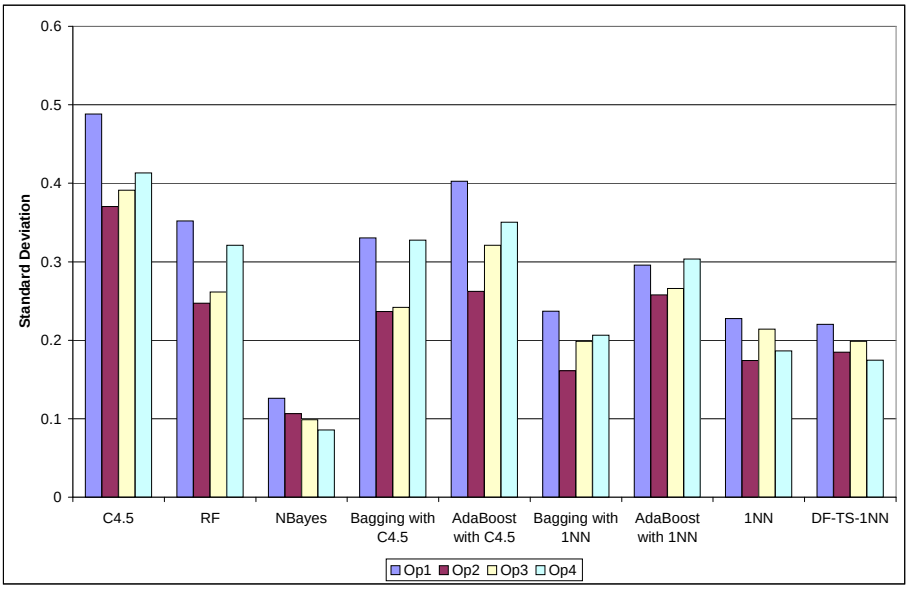

Fig. 9. Standard deviation for various algorithms on CD data set labelled by 4 different operators.

6 Conclusion

A new ensemble technique is proposed in this paper to improve the performance of nearest neighbour (1NN) classifier. The proposed approach combines multiple 1NN classifiers, where each classifier uses a different distance function and potentially a different set of features (feature vector). These feature vectors are determined using a combination of Tabu Search (at the level of the ensemble) and simple local neighbourhood search (at the level of the individual classifiers).

We show that rather than optimising the feature set independently for each

distance metric , it is preferable to co-adapt them, so that each feature set is optimised in the context of the ensemble as whole. This approach also implicitly deals with the problem tackled by many authors, namely of how to find an appropriate measure the diversity of an ensemble so that it can be optimised. Our solution is to simply do this explicitly by letting Tabu Search operate using the ensemble error rate as its cost function.

The proposed ensemble DF-TS-1NN classifier is evaluated using various benchmark data sets from UCI Machine Learning Repository and a real-world application. Results indicate a significant increase in the performance when compared with different well-known classifiers.

Our hypothesis is that the benefits that accrue from this approach are not limited to the use of kNN classifiers. It is relatively straightforward to see how the approach could be adapted to other distance-measure base classifiers such as SOM, LVQ [47]. Other authors have shown improvements from using ensembles with randomly chosen feature subsets (the RSS method [31]), and ee have published results elsewhere [46] showing that feature selection can bring improvements for single classifiers of various different types that are not based on distance metrics: even those such as C4.5 which implicitly perform their own feature selection. This is because many of these methods apply incremental greedy search to select features on which to "split", so the use of feature selection can aid the avoidance of local optima. It is our conjecture that these results could be further improved by using an ensemble of such classifiers as long as the feature selection was simultaneously.

This work is intended as a step towards the automatic creation of classifiers tuned to specific data sets. Future research will be concerned with automating

the choice of distance metric and k for each of our $k-NN$ classifiers. We will also consider ways of automatically selecting subsets of the training examples to use for classification, as a way of tackling the well-known scalability problems of 1NN as the number of training examples increases.

Acknowledgement

This work is supported by the European Commission (Project No. STRP016429, acronym DynaVis). This publication reflects only the authors' views.

References

[1] T. M. Cover, and P. E. Hart (1967). *Nearest Neighbor Pattern Classification.* IEEE Transactions on Information Theory. *13(1)*, 21–27.

[2] C. Domeniconi, J. Peng, and D. Gunopulos (2002). *Locally Adaptive Metric Nearest-Neighbor Classification.* IEEE Transactions on Pattern Analysis and Machine Intelligence. *24(9)*, 1281–1285.

[3] D. Michie, D. J. Spiegelhalter and C. C. Taylor (1994). *Machine Learning, Neural and Statistical Classification.* Ellis Horwood..

[4] J. Wang, P. Neskovic, and L. Cooper (2007). *Improving Nearest Neighbor rule with a simple adaptive distance measure* Pattern Recognition Letters, *28*, 207–213.

[5] O. Okun and H. Proosalut (2005). Multiple views in ensembles of nearest

neihbor classifiers. *In Proceedings of the ICML Workshop on Learning with Multiple Views.* Bonn, Germany, pp. 51–58.

[6] L. Breiman (1996). *Bagging predictors.* Machine Learning, 24(2), 123–140.

[7] Y. Freund and R. E. Schapire (1996). Experiments with a new boosting algorithm. Proceedings of International Conference on Machine Learning, 148–156.

[8] D. H. Wolpert (1992). Stacked generalization, *Neural Networks* 5, pp. 241–259.

[9] A. Tsymbal and S. Puuronen and D. W. Patterson (2003). *Ensemble feature selection with the simple Bayesian classification* Information fusion, *4(2)*, 87–100.

[10] Y. Bao and N. Ishii and X. Du (2004). *Cobmining Multiple k-Nearest Neighbor Classifiers Using Different Distance Functions* Lecture Notes in Computer Science (LNCS 3177), 5th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL 2004), Exeter, UK.

[11] S. D. Bay (1998). *Combining Nearest Neighbor Classifiers Through Multiple Feature Subsets* Proceedings of the Fifteenth International Conference on Machine Learning, 37–45.

[12] S. M. Sait and H. Youssef (1999). *General Iterative Algorithms for Combinatorial Optimization.* IEEE Computer Society.

[13] A. Tsymbal, M. Pechenizkiy and P. Cunningham (2004). *Diversity in Random Subspacing Ensembles.* Lecture Notes in Computer Science (LNCS 3181),

6th International Conference on Data Warehousing and Knowledge Discovery, Zaragoza, Spain.

[14] M. L. Raymer et al (2000). *Dimensionality Reduction using Genetic Algorithms.* IEEE Transactions on Evolutionary Computation, *4(2)*, 164–171.

[15] M. A. Tahir et al (2006). *Novel Round-Robin Tabu Search Algorithm for Prostate Cancer Classification and Diagnosis using Multispectral Imagery.* IEEE Transactions on Information Technology in Biomedicine, *10(4)*, 782–793.

[16] A. K. Jain, and R. P. W. Duin, and J. Mao (2000). *Statistical Pattern Recognition: A Review.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(1), 4–37.

[17] M. Kudo and J. Sklansky (2000). *Comparison of algorithms that select features for pattern classifiers.* Pattern Recognition. *33*, 25–41.

[18] E. Amaldi, and V. Kann (1998). On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science*, 209, 237–260..

[19] S. Davies, and S. Russell (1994). NP-completeness of searches for smallest possible feature sets. *In Proceedings of the AAAI Fall Symposium on Relevance*, AAAI Press, pp. 37-39.

[20] A. K. Jain and D. Zongker (1997). Feature selection: Evaluation, application, and small sample performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2), pp. 153-158.

[21] P. Pudil, J. Novovicova, and J. Kittler (1994). Floating search methods in feature selection. *Pattern Recognition Letters*, 15, pp. 1119-1125.

[22] H. Zhang and G. Sun (2002). *Feature selection using tabu search method.* Pattern Recognition., *35*, 701–711.

[23] W. Siedlecki and J. Sklansy (1989). A note on genetic algorithms for large-scale feature selection. *Pattern Recognition Letters*, 10(11), 335–347.

[24] S. B. Serpico, and L. Bruzzone (2001). A new search algorithm for feature selection in Hyperspectral Remote Sensing Images. *IEEE Transactions on Geoscience and Remote Sensing*, 39(7), 1360–1367.

[25] A. W. Whitney (1971). A direct method of nonparametric measurement selection. *IEEE Transactions on Computers*, 20(9), 1100–1103.

[26] S. Yu, S. D. Backer, and P. Scheunders (2002). Genetic feature selection combined with composite fuzzy nearest neighbor classifiers for hyperspectral satellite imagery. *Pattern Recognition Letters.* 23, pp. 183–190.

[27] M. A. Tahir, A. Bouridane, and F. Kurugollu (2007). *Simultaneous feature selection and feature weighting using Hybrid Tabu Search/K-nearest neighbor classifier.* Pattern Recognition Letters, 28.

[28] D. Korycinski, M. Crawford, J. W Barnes, and J.Ghosh (2003). *Adaptive feature selection for hyperspectral data analysis using a binary hierarchical classifier and tabu search* Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, IGARSS.

[29] M. A. Tahir and J. Smith (2006). *Improving Nearest Neighbor Classifier*

using Tabu Search and Ensemble Distance Metrics Proceedings of the IEEE International Conference on Data Mining (ICDM).

[30] L. I. Kuncheva (2004), "Diversity in Multiple Classifier Systems (editorial), Information Fusion, *6(1)*, 3–4.

[31] T. K. Ho (1998). The random subspace method for constructing decision forests, IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(8), 832–844.

[32] P. Cunningham and J. Carney (2000). Diversity versus quality in classification ensembles based on feature selection In Proc. of the 11th European Conf. On Machine Learning, Barcelona, Spain, LNCS 1810, Springer, 109–116.

[33] F. Glover (1989). *Tabu search I.* ORSA Journal on Computing, 1(3), 190–206.

[34] F. Glover (1990). *Tabu search II.* ORSA Journal on Computing, 2(1), 4–32.

[35] F. Glover, E. Taillard, and D. de Werra (1993). *A user's guide to tabu search.* Annals of Operations Research., *41*, 3–28.

[36] I. H. Witten and E. Frank (2005). Data Mining: Practical machine learning tools and techniques, 2nd Edition, Morgan Kaufmann, San Francisco.

[37] J. R. Quinlan (1993). *C4.5: Programs for Machine Learning.* Morgan-Kaufmann.

[38] L. Breiman (2001). *Random Forests.* Machine Learning, *45(1)*, 5–32.

[39] R. Duda and P. Hart (1973). *Pattern Classification and Scene Analysis.* Wiley, New York.

[40] C. Blake, E. Keogh, and C. J. Merz. UCI Repository of machine learning databases, University of California, Irvine.

[41] R. Kohavi (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence* 2 (12): 1137-1143, Morgan Kaufmann, San Mateo.

[42] S. Raudys and A. Jain (1991). *Small Sample Effects in Statistical Pattern Recognition: Recommendations for Practitioners* IEEE Transactions on Pattern Analysis and Machine Intelligence, *13(3)*, 252–264.

[43] R. Paredes and E. Vidal (2006). *Learning Weighted Metrics to Minimize Nearest-Neighbor Classification Error* IEEE Transactions on Pattern Analysis and Machine Intelligence, *28(7)*, 1100–1110.

[44] URL: http://www.is.umk.pl/projects/datasets.html

[45] URL: www.dynavis.org

[46] Sannen, D., Nuttin, M., Caleb-Solly, P., Smith, J., Tahir, M.A., Lughofer, E. and Eitzinger, C. (2008). An On-Line Interactive Self-adaptive Image Classification Framework *Proceedings of 6th International Conference on Computer Vision Systems* LNCS 5008/2008: 171-180, Springer.

[47] Kohonen, T. (1990). *c.* Proceedings of the IEEE, *78*: 1464-1480.

[48] Smith, J.E., Fogarty, T.C. and Johnson, I.R. (1994) Genetic Feature Selection for Clustering and Classification *proc. IEE Colloquium on Genetic Algorithms in Image Processing and Vision* IEE Digest 1994/193.