Feature Selection for Heterogeneous Ensembles of Nearest-neighbour Classifiers Using Hybrid Tabu Search

Muhammad A. Tahir and James E. Smith

School of Computer Science, University of the West of England, UK
Muhammad.Tahir@uwe.ac.uk James.Smith@uwe.ac.uk

Abstract

The nearest-neighbour (NN) 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 this chapter, a new ensemble technique is proposed to improve the performance of NN classifiers. The proposed approach combines multiple NN classifiers, where each classifier uses a different distance function and potentially a different set of features (feature vector). These feature vectors are determined for each distance metric using a Simple Voting Scheme incorporated in Tabu Search (TS). The proposed ensemble classifier with different distance metrics and different feature vectors (TS–DF/NN) is evaluated using various benchmark data sets from the UCI Machine Learning Repository. Results have indicated a significant increase in the performance when compared with various well-known classifiers. The proposed ensemble method is also compared with an ensemble classifier using different distance metrics but with the same feature vector (with or without Feature Selection (FS)).

Key words: Nearest Neighbour, Tabu Search, Ensemble Classifier, Feature Selection

1 Introduction

The nearest-neighbour (NN) 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 NN 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 [1–3, 27].

The idea of using multiple classifiers instead of a single best classifier has aroused significant interest during the 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 the correct predictions of good classifiers [37]. 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 is often done in an iterative fashion, by changing the probability distribution from which the training set is resampled. Well-known examples are bagging [24] and boosting [25].
- Training multiple classifiers with different inductive biases to create diverse classifiers, e.g. "stacking" approach [38].
- 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.

Specific examples of these three different approach can be found in the literature relating to NN techniques. Bao et al. [10] 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, it uses 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 [15] 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, with the aim of taking some initial steps towards the automatic creation and adaptation of classifiers tuned to a specific data set. Building on [10,15], we explore the hypothesis that the overall ensemble accuracy can be improved if the choices of subsets arise from

- iterative heuristics such as tabu search [17] 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 that consists of multiple NN classifiers, each using a different distance metric and a feature subset derived using tabu search. To increase the diversity, a simple voting scheme is introduced in the cost function of Tabu Search. The proposed ensemble NN classifier (DF–TS–1NN) is then compared with various well-known classifiers.

The rest of this chapter is organized as follows. Section 2 provides review on Feature Selection Algorithms. Section 3 describes a proposed multiple distance function ensemble classifier, followed by experiments in Section 4. Sect. 5 concludes the paper.

2 Feature Selection Algorithms (a Review)

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 NN classifiers [7, 8]. It produces savings in the measuring features (since some of the features are discarded) and the selected features retain their original physical interpretation [9]. Feature selection is used in the design of pattern classifiers with three goals [9, 11]:

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 $\boldsymbol{X}$ is an original feature vector with cardinality n and $\bar{\boldsymbol{X}}$ is the new feature vector with cardinality $\bar{n}$, $\bar{\boldsymbol{X}} \subseteq X$, $J(\bar{\boldsymbol{X}})$ is the selection criterion function for the new feature vector $\bar{\boldsymbol{X}}$. The goal is to optimize $J()$. The problem is NP-hard [29, 30]. Therefore, the optimal solution can only be achieved by performing an exhaustive search in the solution space [1]. However, an exhaustive search is feasible only for small n. A number of heuristic algorithms have been proposed for feature selection to obtain near-optimal solutions [9, 11, 12, 31–34].

The choice of an algorithm for selecting the features from an initial set depends on n. The feature selection problem is said to be of small scale, medium scale, or large scale for n belonging to the intervals [0,19], [20,49], or [50,∞], respectively [11, 12]. Sequential Forward Selection (SFS) [35] 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 [32]. However, these algorithms suffer from the deficiency of converging to local optimal solutions for large scale problems when $n > 100$ [11, 12]. Recent iterative heuristics such as tabu search and genetic algorithms have proved to be effective in tackling this category of problems, which are characterized by having an exponential and noisy search space with numerous local optima [12, 17, 33, 36].

Tabu search (TS) has been applied to the problem of feature selection by Zhang and Sun [12]. In their work, TS performs the feature selection in combination with an objective function based on the Mahalanobis distance. This objective function is used to evaluate the classification performance of each subset of the features selected by the TS. The feature selection vector in TS is represented by a binary string where a 1 or 0 in the position for a given feature indicates the presence or absence of that feature in the solution. Their experimental results on *synthetic data* have shown that TS not only has a high probability of obtaining an optimal or near-optimal solution, but also requires less computational effort than other suboptimal and genetic algorithm based methods. TS has also been successfully applied in other feature selection problems [8, 13, 14].

3 Proposed Ensemble Multiple Distance Function Classifier (DF–TS–1NN)

In this section, we discuss the proposed ensemble multiple distance function TS/1NN classifier (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 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 the search space is of size $2^{F \cdot n}$. Initial experiments showed that in order to make the search more tractable it is advantageous to hybridize the global nature of TS in the whole search space, with local search acting only within the sub-space of the features of each classifier. Figure 1 shows the training phase of the proposed classifier.

During each iteration, 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 NN error rate for the appropriate distance metric Di. From the set of N neighbours, the M best are selected for each classifier.[1] All M^n possible combinations

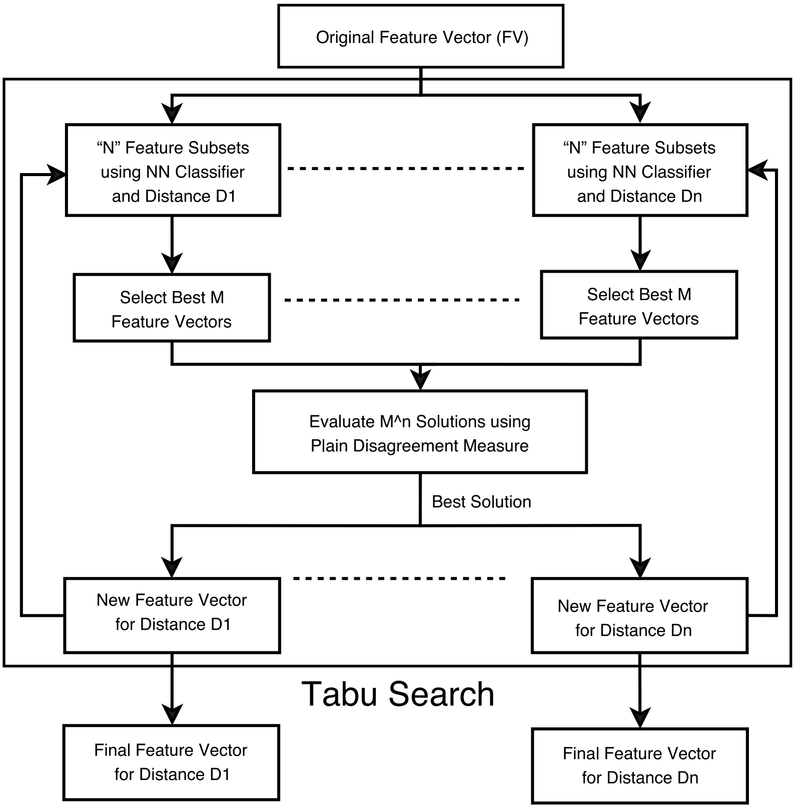

Original Feature Vector (FV)
"N" Feature Subsets using NN Classifier and Distance D1
"N" Feature Subsets using NN Classifier and Distance Dn
Select Best M Feature Vectors
Select Best M Feature Vectors
Evaluate M^n Solutions using Plain Disagreement Measure
Best Solution
New Feature Vector for Distance D1
New Feature Vector for Distance Dn
Tabu Search
Final Feature Vector for Distance D1
Final Feature Vector for Distance Dn

Fig. 1 Training phase of proposed DF–TS–1NN classifier

[1] In this study $M = 2$, $n = 5$, and $N = \sqrt{F}$, where F = Total Number of Features

are then evaluated using a simple voting scheme (SVS) and the best is selected to go forward to the next iteration. Thus, the feedback from the SVS allows TS to search iteratively for combinations of feature vectors that improve the classification accuracy. Implicitly it seeks feature vectors for the different distance measures whereby the errors are not correlated, and so provides diversity. By using *n* distance functions, *n* feature vectors are obtained using TS in the training phase. In the testing phase, the *n* NN classifiers with their different feature vectors are combined as shown in Fig. 2.

In the following subsections, feature selection using TS and the various distance metrics described in this paper are discussed as they are at the heart of the proposed algorithm.

3.1 Distance Metrics

The following five distance metrics are used for NN classifiers. All metrics are widely used in the literature.

- 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: $C = \sum_{i=1}^{m}(x_i - y_i)/(x_i + y_i)$
- Squared chord distance: $S_c = \sum_{i=1}^{m}(\sqrt{x_i} - \sqrt{y_i})^2$
- Squared Chi-squared distance: $C_s = \sum_{i=1}^{m}(x_i - y_i)^2/(x_i + y_i)$

where x and y are the two input vectors and m is the number of features.

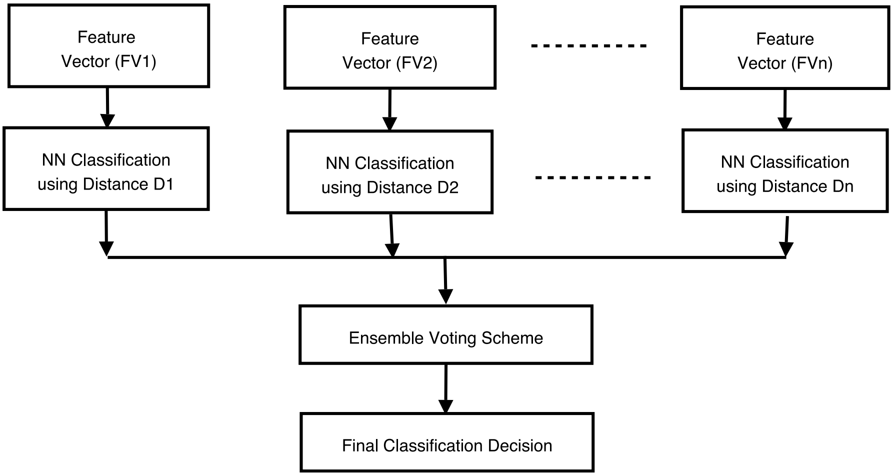

Feature Vector (FV1)
Feature Vector (FV2)
Feature Vector (FVn)
NN Classification using Distance D1
NN Classification using Distance D2
NN Classification using Distance Dn
Ensemble Voting Scheme
Final Classification Decision

Fig. 2 Testing Phase

3.2 Feature Selection and Diversity using Tabu Search

TS was introduced by Glover [5, 6] as a general iterative metaheuristic for solving combinatorial optimization problems. TS is conceptually simple and elegant. It is a form of local neighbourhood search which starts from an initial solution, and then examines feasible neighbouring solutions. It moves from a solution to its best admissible neighbour, even if this causes the objective function to deteriorate. To avoid cycling, solutions that were recently explored are declared forbidden or tabu for a number of iterations. The tabu list stores a characterization of the moves that led to those solutions. The tabu status of a solution is overridden when certain criteria (aspiration criteria) are satisfied. Sometimes intensification and diversification strategies are used to improve the search. In the first case, the search is accentuated in promising regions of the feasible domain. In the second case, an attempt is made to consider solutions over a broader area of the search space and so provide it with a global nature. The flow chart of the TS algorithm is given in Table 1.

Table 1 Algorithm Tabu Search (TS)

Ω	: **Set of feasible solutions**
S	: Current Solution
S^*	: Best admissible solution
$Cost$	: Objective function
$N(S)$	: Neighbourhood of solution S
V^*	: Sample of neighbourhood solutions
T	: Tabu list
AL	: Aspiration Level

```
Begin
Start with an initial feasible solution $S \in \Omega$.
Initialize tabu list and aspiration level.
For fixed number of iterations Do
        Generate neighbour solutions $V^* \subset N(S)$.
        Find best $S^* \in V^*$.
        If move $S$ to $S^*$ is not in T Then
                Accept move and update best solution.
                Update tabu list and aspiration level.
                Increment iteration number.
        Else
                If $Cost(S^*) < AL$ Then
                        Accept move and update best solution.
                        Update tabu list and aspiration level.
                        Increment iteration number.
                End If
        End If
End For
End
```

The size of the tabu list can be determined by experimental runs, watching for the occurrence of cycling when the size is too small, and the deterioration of solution quality when the size is too large [16]. Suggested values of tabu list include $Y, \sqrt{Y}$ (where Y is related to problem size, e.g. number of modules to be assigned in the quadratic assignment problem (QAP), or the number of cities to be visited in the travelling salesman problem (TSP), and so on) [17].

Objective Function

A simple voting scheme is used in each instance of n classifiers. The objective function is the number of instances incorrectly classified using a simple voting scheme. The objective is to minimize

$$Cost = \sum_{i=1}^{S} C_i \tag{1}$$

where S is the number of samples, $C_i = 1$ if instance is classified incorrectly after simple voting in n classifiers, else $C_i = 0$.

Initial Solution

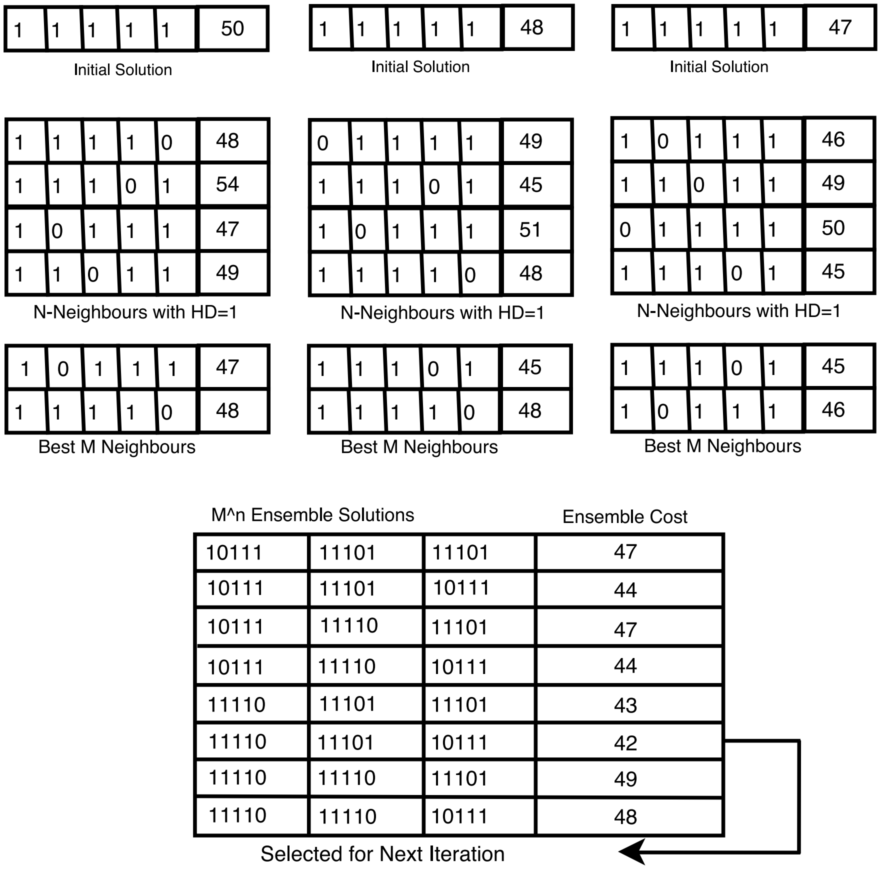
The feature selection vector is represented by a 0/1 bit string where 0 indicates that the feature is not included in the solution while 1 indicates that it is. All features are included in the initial solution.

Neighbourhood Solutions

During each iteration, N random neighbours with *Hamming Distance* 1 (HD1) are generated for the feature set for each classifier and evaluated using the NN error rate with the appropriate distance metric as the cost function. Neighbours are generated by randomly adding or deleting a feature from the feature vector of size F. Among the neighbours, M best solutions are selected, yielding M possible classifiers for each of the n distance metrics. The M^n resulting ensembles are then evaluated using Equation (1) and the one with the best cost (i.e. the solution which results in the minimum value of Equation (1)) is selected and considered as a new current solution for the next iteration. Note that these ensembles may be quickly evaluated since we pre-computed the decision of each of the $M \times n$ classifiers during the local search phase. 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 the ensemble cost function. The best solution is then selected for the next iteration.

Tabu Moves

A tabu list is maintained to avoid returning to previously visited solutions. In our approach, if an ensemble solution (move) is selected at iteration i, then selecting the same ensemble solution (move) for T subsequent iterations (tabu list size) is tabu.



1	1	1	1	1	50

Initial Solution

1	1	1	1	1	48

Initial Solution

1	1	1	1	1	47

Initial Solution

1	1	1	1	0	48
1	1	1	0	1	54
1	0	1	1	1	47
1	1	0	1	1	49

N-Neighbours with HD=1

0	1	1	1	1	49
1	1	1	0	1	45
1	0	1	1	1	51
1	1	1	1	0	48

N-Neighbours with HD=1

1	0	1	1	1	46
1	1	0	1	1	49
0	1	1	1	1	50
1	1	1	0	1	45

N-Neighbours with HD=1

1	0	1	1	1	47
1	1	1	1	0	48

Best M Neighbours

1	1	1	0	1	45
1	1	1	1	0	48

Best M Neighbours

1	1	1	0	1	45
1	0	1	1	1	46

Best M Neighbours

M^n Ensemble Solutions			Ensemble Cost
10111	11101	11101	47
10111	11101	10111	44
10111	11110	11101	47
10111	11110	10111	44
11110	11101	11101	43
11110	11101	10111	42
11110	11110	11101	49
11110	11110	10111	48

Fig. 3 An example showing neighbourhood solutions during one iteration of the proposed TS method. $n = 3$, $N = 4$, and $M = 2$

Aspiration Criterion

The aspiration criterion is a mechanism used to override the tabu status of moves. It temporarily overrides the tabu status if the move is sufficiently good. In our approach, if an ensemble solution is selected at iteration i and this move results in a best cost for all previous iterations, then that solution is selected even if that feature is in the tabu list.

Termination Rule

The most commonly used stopping criteria in TS are

- after a fixed number of iterations
- after some number of iterations when there has been no increase in the objective function value
- when the objective function reaches a pre-specified value.

In this work, the termination condition is a fixed number of iterations.

4 Experiments

To evaluate the effectiveness of our method, extensive experiments were carried out, and comparisons with several methods performed.

4.1 Methods

The proposed (DF–TS–1NN) algorithm is compared with the following methods. All methods are implemented using the WEKA library [26].

- 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, 20].
- Decision Table (DT): It uses a simple decision table majority classifier [21].
- Random Forest (RF): Ensemble Classifier using a forest of random trees [22].
- Naive Bayes Algorithm (NBayes): The Naive Bayes Classifier technique is based on Bayes' theorem. Despite its simplicity, Naive Bayes can often outperform numerous sophisticated classification methods [23].
- Bagging: A method for generating multiple versions of a predictor and using these to get an aggregated predictor (ensemble) [24]. C4.5 is used as base classifier.
- AdaBoost1: A meta-algorithm for constructing ensembles which can be used in conjunction with many other learning algorithms to improve their performance [25]. C4.5 is used as base classifier.

In addition, we compare the following variations of the proposed ensemble algorithms:

1. DF–1NN: Ensemble Classifier using NN classifiers with each classifier having different distance metrics (DF) and without FS.
2. DF–TS1–1NN: Ensemble Classifier using NN classifiers, each using a different distance metric. FS using TS is applied independently for each data set.
3. DF–TS2–1NN: Ensemble Classifier as above but with a single common feature set selected by TS. Subsets for various distance metrics are derived using TS.
4. DF–TS3–1NN: Proposed Ensemble Classifier. Different feature subsets for each classifier derived simultaneously using TS.

4.2 Data Sets Descriptions and Experimental Setup

We have performed a number of experiments and comparisons with several benchmarks from the UCI [4] in order to demonstrate the performance of the proposed classification system. A short description of the benchmarks used, along with TS runtime parameters are given in Table 2.

The tabu list size and number of neighbourhood solutions are determined using the following equation:

$$T = N = ceil\,(\sqrt{F}) \tag{2}$$

where T is the tabu list size, N is the number of neighbourhood solutions and F is the number of features.

Table 2 Data sets description. P = Prototypes, F = Features, C = Classes, T = Tabu list size, N = Number of neighbourhood solutions

Name	P	F	C	T	N
Statlog Diabetes	768	8	2	3	3
Statlog Heart	270	13	2	4	4
Statlog Australian	690	14	2	4	4
Statlog Vehicle	846	18	4	5	5
Statlog German	1000	20	2	5	5
Breast Cancer	569	32	2	6	6
Ionosphere	351	34	2	6	6
Sonar	208	60	2	8	8
Musk	476	166	2	13	13

In all data sets, B-fold cross-validation has been used to estimate error rates [18]. For B-fold CV, each data set is divided into B blocks 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. Each experiment was run 100 times using different random 10-CV partitions and the results were averaged over the 100 runs [19].

The number of iterations for FS using TS is 200 for all data sets, which was chosen after preliminary experimentation.

In order to offset any bias due to the different range of values for the original features in the NN classifier, the input feature values are normalized over the range [1,10] using Equation (3) [7]. Normalizing the data is important to ensure that the distance measure allocates equal weight to each variable. Without normalization, the variable with the largest scale will dominate the measure.

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

where $x_{i,j}$ is the jth feature of the i-th pattern, $x'_{i,j}$ is the corresponding normalized feature, and n is the total number of patterns.

4.3 Comparison of Different ways of Creating Feature Sets

Table 3 shows the classification accuracy using various distance functions within single classifiers, and for the ensemble technique 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 used here the votes of the less accurate classifiers can dominate, so that the ensemble performs worse than the best single classifier on those datasets.

Table 4 shows the classification accuracy using various distance functions and with FS and compared with the various variations of the proposed method. Comparing the results for individual classifiers with feature selection ($\{E, M, C, C_s, S_c\}$) to those without (Table 3) it can be seen that the accuracy is increased in every case – a nice example of the value of performing feature selection.

Table 3 Classification accuracy (%) using individual classifiers and various variations of the proposed classifier. 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	82.1	82.0	**85.7**	82.3	82.4	84.0
Breast Cancer	95.3	95.2	95.2	95.4	95.4	**95.6**
Diabetes	**70.5**	69.7	66.0	69.4	69.6	70.2
German	70.9	71.1	70.2	70.5	70.0	**71.8**
Heart	78.1	79.6	**80.8**	79.0	78.3	79.0
Ionosphere	87.0	90.7	**92.2**	89.1	89.0	90.3
Musk	85.4	83.3	84.0	**86.1**	86.0	86.0
Sonar	82.5	84.6	**86.6**	86.0	86.4	85.4
Vehicle	69.6	69.5	69.6	70.4	70.4	**70.7**

Turning to the use of feature selection to derive a common subset for all classifiers (DF–TS2–1NN), not only do we see improved performance compared to the same algorithm without feature selection (DF–1NN in Table 3), but now the mean accuracy is higher than the best individual classifier on most data sets. This is a good example, which 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). In all but two cases our proposed method (DF–TS3–1NN) outperforms the others and the means differ by more than the combined standard deviations, indicating a high probability that these are truly significantly different results. In the two cases where DF–TS1–1NN

Table 4 Mean and standard deviation of classification accuracy (%) using individual classifiers and variations of the proposed classifier. M = Manhattan, E = Euclidean, C = Canberra, C_s = Chi-squared, S_c = Squared-chord

Data Set	E	M	C	C_s	S_c	DF–TS1–1NN	DF–TS2–1NN	DF–TS3–1NN
Australian	86.5	88.1	86.4	85.9	86.8	89.0(0.61)	85.1(0.45)	**90.5**(0.48)
Breast Cancer	97.4	97.8	97.5	97.4	97.5	97.9(0.22)	97.6(0.32)	**98.0**(0.25)
Diabetes	71.7	70.8	71.1	70.1	70.3	**75.5**(0.71)	72.5(0.82)	74.5(0.85)
German	72.3	73.8	74.1	74.5	73.4	76.5(0.63)	74.2(0.71)	**79.8**(0.62)
Heart	83.2	82.6	82.2	84.0	83.0	85.0(1.11)	83.8(1.45)	**86.3**(0.90)
Ionosphere	93.3	95.4	96.2	91.1	94.3	95.3(0.41)	95.1(0.37)	**96.3**(0.52)
Musk	91.2	89.9	89.8	92.3	91.8	91.6(0.67)	92.3(0.82)	**94.5**(0.72)
Sonar	91.0	90.9	93.1	91.5	93.0	93.5(0.82)	93.4(1.00)	**94.7**(1.09)
Vehicle	73.9	75.1	74.2	74.9	74.2	**77.2**(0.60)	74.5(0.59)	76.9(0.61)

has a higher observed mean than DF–TS3–1NN, the differences are less than the standard deviation of either set of results, so they are almost certainly not significant.

Table 5 shows the number of features used by the proposed classifier for various data sets. 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 common for ensemble classifier, i.e. that are used by each classifier. 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 among the decisions of the different classifiers, and the fact that these feature sets are learnt 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.

4.4 Comparison with other Algorithms

Table 6 shows results of a comparison of classification accuracy (in %) between the proposed DF–TS–1NN classifier and others for different data sets. The proposed algorithm achieved higher accuracy on all data sets except Diabetes.

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
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
Sonar	60	31	33	27	35	33	0	58
Vehicle	18	9	11	13	7	13	0	17

Table 6 Average classification accuracy (%) using different classifiers. DT = Decision table. RF = Random forest

Data Set	C4.5	DT	RF	NBayes	Bagging	AdaBoost	1NN	DF–TS3–1NN
Australian	84.3	84.7	86.1	77.1	86.0	85.0	79.6	**90.5**
Breast Cancer	93.6	93.3	95.9	93.3	95.35	96.1	95.4	**98.0**
Diabetes	74.3	74.1	74.7	75.6	**76.0**	72.4	70.3	74.5
German	71.6	72.5	74.7	74.5	74.6	72.50	70.9	**79.8**
Heart	78.2	82.3	80.2	84.0	80.5	79.2	75.7	**86.3**
Ionosphere	89.8	94.2	95.4	92.8	92.2	90.3	87.5	**96.3**
Musk	82.7	80.8	87.8	73.9	88.2	90.0	85.6	**94.5**
Sonar	73.0	72.6	80.3	67.9	78.5	80.1	86.5	**94.7**
Vehicle	72.7	66.4	74.7	45.4	74.5	76.4	69.7	**76.9**

- For Australian, German and Ionosphere data sets there is improvement of 1.98%, 5.06% and 0.4% respectively when compared with the best of the other methods (Random Forest Classifier).
- For Heart, there is an improvement of 3.3% when compared with the best of the other methods (Naive Bayes Classifier).
- For Vehicle, Breast Cancer and Musk data sets, there is an improvement of 0.5%, 0.76%, and 4.55% respectively when compared with the best of the other methods (AdaBoost).
- For Sonar, there is an improvement of 7.8% when compared with the best of the other methods (1NN).
- Since Diabetes has only eight features, the proposed algorithm is unable to combine the benefits of feature selection and ensemble classifiers using different distance metrics.

As can be seen, the proposed method performs consistently well and outperforms other methods on all but one data set. Moreover, for the other methods there is considerable variation in performance according to how well the indicative bias of each method suits each data set. It is worth noting that the two methods of producing ensembles always improve the performance compared to the base C4.5 classifiers, apart from Ada-Boost on the Diabetes data set.

Figure 4 shows the standard deviation obtained over 100 runs of random 10-fold cross-validation 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

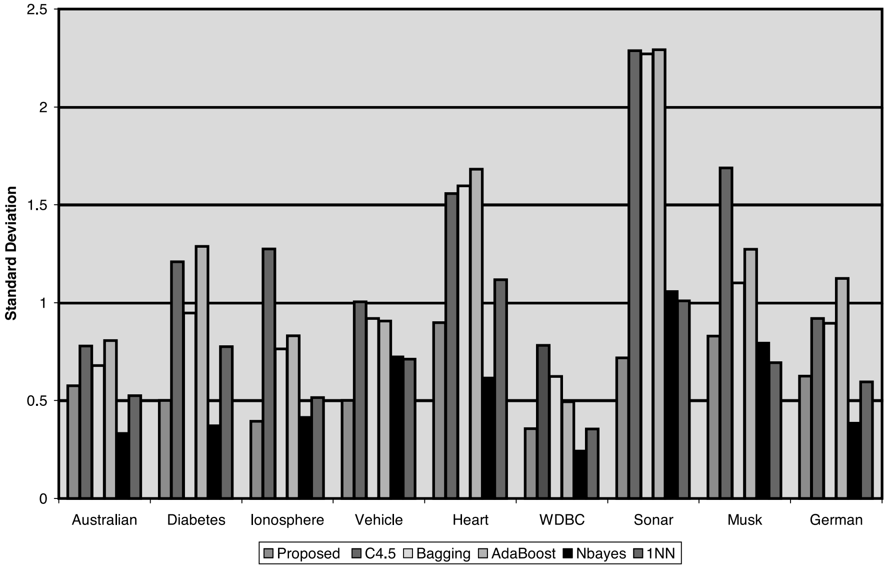


Fig. 4 Standard deviation for different algorithms on various data sets

algorithms, and is usually less than the observed difference in mean accuracies, suggesting that these are significant. In particular it is always less than that of the two other boosting algorithms. Thus if we think in terms of the Bias-Variance decomposition of classifier errors, it might initially appear that both the bias and the variance terms are reduced for this method, but this must be studied in more detail.

4.5 Analysis of Learning

Figures 5–7 show the classification accuracy (%) versus 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.

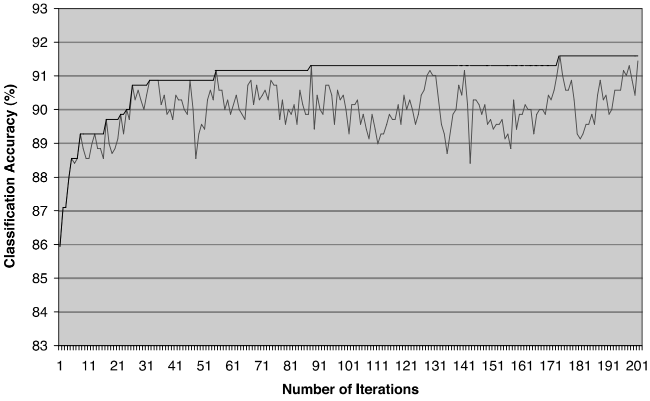


Fig. 5 Error rate vs number of iterations for Australian data set

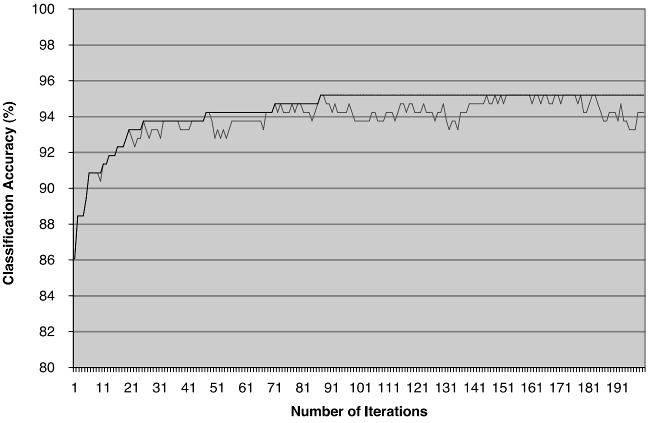


Fig. 6 Error rate vs number of iterations for Ionosphere data set

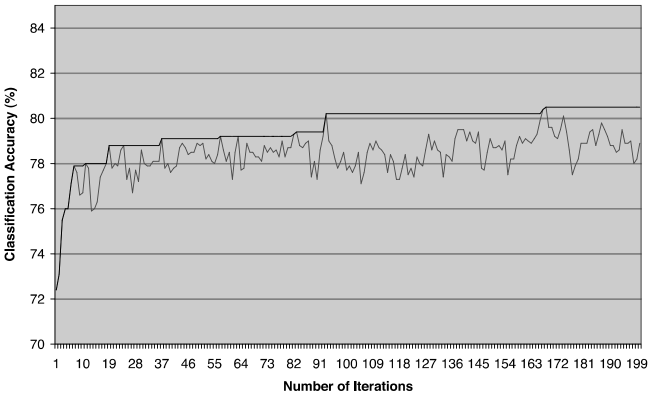


Fig. 7 Error rate vs number of iterations for German data set

5 Conclusions

A new ensemble technique is proposed in this paper to improve the performance of NN classifiers. The proposed approach combines multiple NN 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 optimizing the feature set independently for each distance metric, it is preferable to co-adapt them, so that each feature set is optimized 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 for the diversity of an ensemble so that it can be optimized. Our solution is to simply do this explicitly by letting TS 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 the UCI Machine Learning Repository. Results indicate a significant increase in performance compared with other different well-known classifiers.

This work is intended as a step towards the automatic creation of classifiers tuned to specific data sets. Having done our initial "proof of concept", the next stages of this research programme 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 NN 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. C. Blake, E. Keogh, and C. J. Merz. UCI Repository of machine learning databases, University of California, Irvine
5. F. Glover (1989). *Tabu Search I.* ORSA Journal on Computing, *1(3)*, 190–206
6. F. Glover (1990). *Tabu Search II.* ORSA Journal on Computing, *2(1)*, 4–32
7. M.L. Raymer et al (2000). *Dimensionality Reduction using Genetic Algorithms.* IEEE Transactions on Evolutionary Computation, *4(2)*, 164–171
8. 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
9. 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
10. Y. Bao and N. Ishii and X. Du (2004). *Combining 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. M. Kudo and J. Sklansky (2000). *Comparison of Algorithms that Select Features for Pattern Classifiers.* Pattern Recognition, *33*, 25–41
12. H. Zhang and G. Sun (2002). *Feature Selection using Tabu Search Method.* Pattern Recognition,, *35*, 701–711
13. 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, 2007
14. 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
15. S.D. Bay (1998). *Combining Nearest Neighbor Classifiers Through Multiple Feature Subsets* Proceedings of the Fifteenth International Conference on Machine Learning, 37–45
16. F. Glover, E. Taillard, and D. de Werra (1993). *A User's Guide to Tabu Search.* Annals of Operations Research, *41*, 3–28
17. S.M. Sait and H. Youssef (1999). *General Iterative Algorithms for Combinatorial Optimization.* IEEE Computer Society
18. 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
19. 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
20. J.R. Quinlan (1993). *C4.5: Programs for Machine Learning.* Morgan Kaufmann
21. R. Kohavi (1995). *The Power of Decision Tables.* Proceedings of the 8th European Conference on Machine Learning

22. L. Breiman (2001). *Random Forests.* Machine Learning, *45(1)*, 5–32
23. R. Duda and P. Hart (1973). *Pattern Classification and Scene Analysis* Wiley, New York.
24. L. Breiman (1996). *Bagging Predictors.* Machine Learning, *24(2)*, 123–140
25. Y. Freund and R.E. Schapire (1996). *Experiments with a New Boosting Algorithm.* Proceedings of International Conference on Machine Learning, 148–156
26. I.H. Witten and E. Frank (2005). *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd edn, Morgan Kaufmann, San Francisco
27. J. Wang, P. Neskovic, and L. Cooper (2007). *Improving Nearest Neighbor Rule with a Simple Adaptive Distance Measure* Pattern Recognition Letters, *28*, 207–213
28. 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)
29. 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
30. 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, 37–39
31. 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)*, 153–158
32. P. Pudil, J. Novovicova, and J. Kittler (1994). *Floating Search Methods in Feature Selection.* Pattern Recognition Letters, *15*, 1119–1125
33. W. Siedlecki and J. Sklansy (1989). *A Note on Genetic Algorithms for Large-scale Feature Selection.* Pattern Recognition Letters, *10(11)*, 335–347
34. 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
35. A.W. Whitney (1971). *A Direct Method of Nonparametric Measurement Selection.* IEEE Transactions on Computers, *20(9)*, 1100–1103
36. 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*, 183–190
37. O. Okun and H. Proosalut (2005). *Multiple Views in Ensembles of Nearest Neighbor Classifiers.* In Proceedings of the ICML Workshop on Learning with Multiple Views, Bonn, Germany, 51–58
38. D.H. Wolpert (1992). *Stacked Generalization*, Neural Networks, *5*, 241–259