Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition Letters 28 (2007) 438–446
Pattern Recognition Letters
www.elsevier.com/locate/patrec

Simultaneous feature selection and feature weighting using Hybrid Tabu Search/*K*-nearest neighbor classifier

Muhammad Atif Tahir [a,*], Ahmed Bouridane [b], Fatih Kurugollu [b]

[a] *School of Computer Science, University of the West of England, Bristol, BS16 1QY, United Kingdom*
[b] *School of Computer Science, The Queen's University of Belfast, Belfast BT7 1NN, Northern Ireland, United Kingdom*

Received 25 August 2005; received in revised form 14 August 2006
Available online 25 October 2006

Communicated by R.P.W. Duin

Abstract

Feature selection and feature weighting are useful techniques for improving the classification accuracy of *K*-nearest-neighbor (*K*-NN) rule. The term feature selection refers to algorithms that select the best subset of the input feature set. In feature weighting, each feature is multiplied by a weight value proportional to the ability of the feature to distinguish pattern classes. In this paper, a novel hybrid approach is proposed for simultaneous feature selection and feature weighting of *K*-NN rule based on Tabu Search (TS) heuristic. The proposed TS heuristic in combination with *K*-NN classifier is compared with several classifiers on various available data sets. The results have indicated a significant improvement in the performance in classification accuracy. The proposed TS heuristic is also compared with various feature selection algorithms. Experiments performed revealed that the proposed hybrid TS heuristic is superior to both simple TS and sequential search algorithms. We also present results for the classification of prostate cancer using multispectral images, an important problem in biomedicine.

Keywords: Tabu Search; *K*-NN classifier; Feature selection; Feature weighting; Prostate cancer diagnosis

1. Introduction

The *K*-nearest-neighbor (*K*-NN) classifier has long been used in pattern recognition, exploratory data analysis, and data mining problems. Typically, the *K*-nearest neighbors of an unknown sample are selected from the training set in order to predict the class label as the most frequent one occurring in the *K*-neighbors. The *K*-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 (Cover and Hart, 1967; Domeniconi et al., 2002; Michie et al., 1994).

Both feature selection and feature weighting techniques are useful for improving the classification accuracy of the *K*-NN rule (Raymer et al., 2000; Paredes and Vidal, 2000; Wettschereck et al., 1997). The term feature selection refers to algorithms that select the best subset of the input feature set. These algorithms are used in the design of pattern classifiers that have three goals (Jain et al., 2000; Kudo and Sklansky, 2000):

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

Feature selection leads to savings in measuring features (since some of the features are discarded) and the selected features retain their original physical interpretation (Jain et al., 2000). However, the feature selection is NP-hard problem (Cover and Hart, 1967). Feature weighting is a

* Corresponding author. Tel.: +44 117 3283357; fax: +44 117 3283182.
E-mail address: muhammad.tahir@uwe.ac.uk (M.A. Tahir).

0167-8655/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2006.08.016

more general method in which the original set of features is multiplied by a weight value proportional to the ability of the feature to distinguish pattern classes (Raymer et al., 2000; Paredes and Vidal, 2000). A good review of feature weighting algorithms was carried out by Wettschereck et al. (1997). These algorithms can be divided into two groups: one which searches a set of weights through an iterative algorithm and uses the performance of the classifier as a feedback to select a new set of weights (Raymer et al., 2000; Puch et al., 1993; Lowe, 1995); and the other computes the weights using pre-existing model's bias, e.g. conditional probabilities, class projection, and mutual information (Domeniconi et al., 2002; Paredes and Vidal, 2000; Guverenir and Akkus, 1997). Feature weighting is more appropriate for problems where the features vary in their relevance. Feature selection algorithms perform best when the features used to describe instances are either highly correlated with the class label or completely irrelevant (Wettschereck et al., 1997).

The feature selection problem is NP-hard problem. Therefore, the optimal solution cannot be guaranteed to be acquired except when performing an exhaustive search in the solution space (Cover and Van Campenhout, 1997). However, exhaustive search is feasible only for small number of features n. Different algorithms have been proposed for feature selection to obtain near-optimal solutions (Jain et al., 2000; Kudo and Sklansky, 2000; Zhang and Sun, 2002; Siedlecki and Sklansy, 1989; Pudil et al., 1994). The choice of an algorithm for selecting the features from an initial set depends on n. The feature selection problem is of small scale, medium scale, or large scale if n belongs to [0,19], [20,49], or [50,∞], respectively (Kudo and Sklansky, 2000; Zhang and Sun, 2002). Sequential algorithms such as Sequential Forward Floating Search (SFFS) and Sequential Backward Floating Search (SBFS) are efficient and usually find fairly good solutions for small and medium scale problems (Pudil et al., 1994). But they suffer from the problem of trapping into local optimal solutions for large scale problems (Kudo and Sklansky, 2000; Zhang and Sun, 2002). Modern iterative heuristics such as Tabu Search and genetic algorithms have been found effective in tackling this category of problems which have an exponential and noisy search space with numerous local optima (Zhang and Sun, 2002; Siedlecki and Sklansy, 1989; Sait and Youssef, 1999).

Tabu Search (TS) has been applied to the problem of feature selection by Zhang and Sun (2002). In their work, the Tabu Search performs the feature selection in combination with Mahalanobis distance as an objective function. 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 0/1 bit string where 0 indicates the feature is not included in the solution while 1 indicates the feature is included. 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 (Tahir et al., 2004a; Tahir et al., 2004b; Korycinski et al., 2003).

In this paper, a Hybrid Tabu Search/K-NN algorithm is proposed to perform both feature selection and feature weighting simultaneously with the objective of improving the classification accuracy. This approach uses both a feature weight vector and a feature binary vector on the encoding solution of Tabu Search. The feature weight vector consists of real values while feature binary vector consisting of either 0 or 1. A K-NN classifier is used to evaluate each weight set evolved by TS. In addition to feature weight and binary vectors, the value of K used in K-NN classifier is also stored in the encoding solution of TS. Neighbors are calculated using an squared Euclidean distance defined as:

$$D(x,y) = \sum_{i=1}^{m} (x_i - y_i)^2 \tag{1}$$

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

In the weighted K-NN classifier, the feature values of the training patterns and the unknown pattern are multiplied by the corresponding weight values prior to classification. In the proposed approach, the weight value can be 0 for some features. Thus, the feature space is expanded in the dimensions associated with highly weighted features, and compressed in the dimensions associated with less highly weighted features. This allows the K-NN classifier to distinguish more accurately among patterns along the dimensions

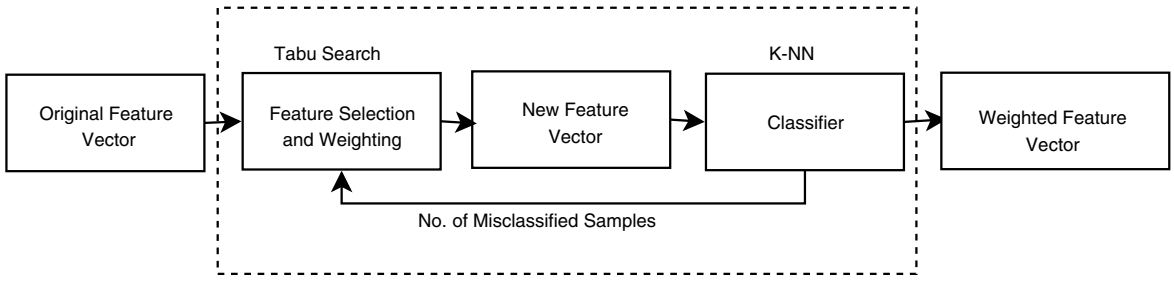

Fig. 1. Training phase of proposed hybrid TS/K-NN classifier with simultaneous feature selection and weighting.

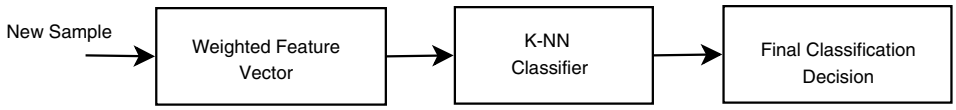

Fig. 2. Testing phase.

associated with highly-weighted features (Raymer et al., 2000).

The classification accuracy obtained from TS/*K*-NN classifier is then compared and assessed with published results of several commonly-employed pattern classification algorithms. Fig. 1 shows the training phase of proposed hybrid model of TS/*K*-NN classifier. The feedback from the *K*-NN classifier allows the Tabu Search to iteratively search for a feature vector that improves the classification accuracy. In the testing phase, only *K*-NN classifier is used as shown in Fig. 2.

This paper is organized as follows. Section 2 gives an overview about Tabu Search followed by the proposed algorithm for simultaneous feature selection and weighting using hybrid TS/*K*-NN classifier in Section 3. Section 4 discusses experiments carried out while Section 5 concludes the paper.

2. Overview of Tabu Search

Tabu Search (TS) was introduced by Fred Glover (Glover, 1989; Glover, 1990) as a general iterative metaheuristic for solving combinatorial optimization problems. Tabu Search is conceptually simple and elegant. It is a form of local neighborhood search. Each solution $S \in \Omega$ has an associated set of neighbors $N(S) \subseteq \Omega$ where Ω is the set of feasible solutions. A solution $S' \in N(S)$ can be reached from S by an operation called a *move to* S'. TS moves from a solution to its best admissible neighbor, 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 status of a solution is overridden when certain criteria (aspiration criteria) are satisfied. The Tabu Search algorithm is given in Fig. 3.

Algorithm Short-Term-TS

Ω : **Set of feasible solutions**
S : Current Solution
S^* : Best admissible solution
$Cost$: Objective function
$N(S)$: Neighborhood of solution S
V^* : Sample of neighborhood 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 neighbor 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

Fig. 3. Algorithmic description of Tabu Search (TS) (Sait and Youssef, 1999).

The size of 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 (Glover et al., 1993). 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 travel salesman problem (TSP), and so on) (Sait and Youssef, 1999).

3. Proposed Tabu Search technique for simultaneous feature selection and extraction

3.1. Encoding solution

Unlike existing encoding solutions which consist of using only 0/1 bit string, our proposed structure of the TS encoding solution consists of three parts and illustrated in Fig. 4. The first part, $W_1W_2\ldots\ldots W_n$ consists of a real-valued weight for each of the n features. The second part, $B_1B_2\ldots\ldots B_n$, consists of 0/1 bit string for each of the n features, and the third part, k, is the value of K of the K-NN classifier. Thus, the value of k is stored in the encoding solution and determined automatically along with feature weights.

3.2. Objective function

Our objective in this work is to improve the classification accuracy. Therefore, an objective function is to minimize the total number of misclassified samples as shown in Eq. (2).

$$\text{Cost} = \sum_{i=1}^{n} C_i \ * \text{MCS}_i \tag{2}$$

where n is the number of classes, C_i is the misclassification cost for each sample in class i [1], and MCS_i is the total number of misclassified samples for class i.

[1] In some data sets, classification-cost penalties are available.

W_1	W_2	.....	W_n	B_1	B_2	.....	B_n	k

Fig. 4. The structure of Encoding Scheme used in TS.

3.3. Initial solution

All features are included in the initial solution by assigning 1 to the corresponding features values as shown in Fig. 5. All weights are assigned to 1.0, and $k = 1$ is used for the initial solution.

3.4. Neighborhood solutions

For Tabu Search, it is important that there should be a move from one solution to different neighbours. In our approach, neighbours are generated from the first two parts of the encoding solution. $M \times N$ different neighbours are generated from the first part by assigning M random weights to N different features. P different neighbours are generated by randomly adding or deleting a feature from the second part. Thus, the total number of neighbourhood solutions (V^*) for each iteration is $M \times N + P$. Fig. 6 shows an example showing different neighbours from the initial solution. Each neighbour solution is a single move from the initial/previous solution. Among the neighbours, the one with the best cost (i.e. the solution which results in the minimum value of Eq. (2) is selected and considered as the new current solution for the next iteration. As an example, let us assume that neighbour 2 from Fig. 6 is considered as new current solution for the next iteration. Fig. 7

1.0	1.0	.....	1.0	1	1	.....	1	1

Fig. 5. Initial solution.

	W_1	W_2	.....	W_n	B_1	B_2	.....	B_n	k
M=2, N=2	**10.1**	1.0	.....	1.0	1	1	.....	1	**3**
	4.1	1.0	.....	1.0	1	1	.....	1	**5**
	1.0	**1.8**	.....	1.0	1	1	.....	1	**1**
	1.0	**5.8**	.....	1.0	1	1	.....	1	**5**
P=3	1.0	1.0	.....	1.0	1	**0**	.....	1	**7**
	1.0	1.0	.....	1.0	1	1	.....	**0**	**7**
	1.0	1.0	.....	1.0	**0**	1	.....	1	**1**

Fig. 6. An example showing some possible neighbours from the initial solution. $V^* = 7$, $M = 2$, $N = 2$ and $P = 3$. K is assigned randomly between 1 and 7 for different neighbours.

Previous Best Solution

4.1	1.0	.....	1.0	1	1	.....	1	**5**

7 Neighbours from Previous Best Solution

	W_1	W_2	.....	W_n	B_1	B_2	.....	B_n	k
M=2, N=2	4.1	**6.5**	.....	1.0	1	1	.....	1	**3**
	4.1	**9.8**	.....	1.0	1	1	.....	1	**3**
	4.1	1.0	.....	**3.0**	1	1	.....	1	**5**
	4.1	1.0	.....	**6.7**	1	1	.....	1	**7**
P=3	4.1	1.0	.....	1.0	**0**	1	.....	1	**1**
	4.1	1.0	.....	1.0	1	**0**	.....	1	**5**
	4.1	1.0	.....	1.0	1	1	.....	**0**	**7**

Fig. 7. An example showing some possible neighbours obtained from the previous best solution. $V^* = 7$, $M = 2$, $N = 2$ and $P = 3$. K is assigned randomly between 1 and 7 for different neighbours.

shows some possible neighbours from this new solution. Among the new neighbours, the one with the best cost will be considered as the next solution.

3.5. Tabu moves

A Tabu list is maintained to prevent returning to previously visited solutions. This list contains information that, to some extent, forbids the search from returning to a previously visited solution (Sait and Youssef, 1999). In our implementation, if a feature is added/deleted or weighted at iteration i, then adding/deleting or weighting the same feature (move) for next T iterations (Tabu list size) is Tabu.

3.6. Aspiration criterion

Aspiration criterion is a method used to override the Tabu status of moves whenever appropriate. It temporarily overrides the Tabu status if the move is sufficiently good. In our approach, if a feature is added/deleted or weighted at iteration i and this move results in a best cost for all previous iterations, then this feature is allowed to add/delete or weighted even if it is in the Tabu list. The aspiration criterion selected here will avoid missing good solutions and thus will not lock the algorithm in the neighborhood of some local minimum.

3.7. Termination rule

The most commonly used stopping criteria in TS are

- after a fixed number of iterations,
- after some number of iterations without an improvement in the objective function value,

- when the objective reaches a pre-specified objective value.

In our algorithm, termination condition is a predefined number of iterations.

4. Experiments

To evaluate the effectiveness of our method, extensive experiments were carried out. Comparisons with several methods were also performed as will be shown in the following section.

4.1. Methods

In the following experiments we compare several classification approaches mentioned below:

- Locally Adaptive Metric Nearest-Neighbor (ADAMENN): This classifier estimates a flexible metric for producing neighborhoods that are elongated along less relevant feature dimensions and constricted along most influential ones (Domeniconi et al., 2002).
- Discriminant Adaptive Nearest Neighbor (DANN): This classifier uses local discrimination information to estimate a subspace for global dimension reduction. Local linear discriminant analysis is used to estimate an effective metric for computing neighborhoods (Hastie and Tibshirani, 1996).
- Class-Dependent Weighted Dissimilarity Measure for Nearest Neighbor Classification Problems (CDW): This method uses a weighted dissimilarity measure for NN classification. The weights are obtained using minimization of an criterion index through Fractional-Programming (Paredes and Vidal, 2000).
- *K*-Nearest Neighbor (*K*-NN): In this classifier, the *K* nearest neighbor of a unknown sample in the training set is computed in order to predict the class label as the most frequent one occurring in the *K*-neighbors (Cover and Hart, 1967; Michie et al., 1994; Duda et al., 2001).
- Decision Tree Method (C4.5): Decision tree is a classifier in the form of a tree structure, where each node is either a leaf node or a decision node (Quinlan, 1993; Michie et al., 1994).
- 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 (Michie et al., 1994).
- Linear Discriminant Analysis (LDisc): The linear discriminant analysis consists of searching some linear combinations of features, which provide the best separation between classes. These different combinations are called discriminant functions (Michie et al., 1994).
- Quadratic Discriminant Analysis (QDisc): The quadratic discriminant function is most simply defined as the logarithm of the appropriate probability density function, so that one quadratic discriminant is calculated for each class (Michie et al., 1994).

In addition, we compare with several feature selection algorithms mentioned below.

- Sequential Forward Search (SFS) (Whitney, 1971): SFS is the simplest greedy search algorithm. It starts with an empty feature subset and sequentially add features that results in the highest objective criteria. The main disadvantage of SFS is that it is unable to remove features that become irrelevant after the addition of other features.
- Sequential Forward Floating Selection (SFFS) (Pudil et al., 1994): SFFS improved the SFS method by introducing backward steps after each forward step as long as the objective criteria increases.

4.2. Data sets

We have performed a number of experiments and comparisons on several benchmarks from the Statlog project (Statlog Corpora. Dept) and UCI (Blake et al.) in order to demonstrate the performance of the proposed classification system. A short description of the used benchmarks is mentioned in Table 1.

Data sets with greater than 300 samples are randomly divided into the training and testing data. The training set consists of 70% of the patterns while the test set consists of 30% of patterns. We have run 5 trials for each data set and the final classification accuracy is the average of these trials. Five-fold cross validation is used for training. For the data sets with less than 300 samples, a leave-one-out cross validation technique has been used. For leave-one-out cross validation, a classifier is designed using $(n-1)$ samples and evaluated on the one remaining sample; this is repeated n times, with different training sets of size

Table 1
Data sets description

Name	Prototypes	Features	Classes
UCI Balance	625	4	3
UCI Iris	150	4	2
UCI Liver	345	6	2
Statlog Diabetes	768	8	2
UCI Glass	214	9	6
Statlog Heart	270	13	2
Statlog Australian	690	14	2
Statlog Vehicle	846	18	4
UCI Ionosphere	351	34	2
UCI Sonar	208	60	2

P = prototypes, F = features, C = classes.

Table 2
Average classification error rate (unit %)

	A	*C*4.5	*C*	*D*	*K*	*L*	*N*	*Q*	TS/*K*-NN (std)
Australian	–	15.6	15.2	–	16.7	14.1	20.8	20.7	**10.2**
Balance	–	22.6	**9.17**	–	15.6	–	9.5	–	10.9
Diabetes	–	26.3	24.8	–	29.7	22.5	23.5	26.2	**22.3**
Glass	24.8	31.8	–	26.6	28.0	40.7	–	–	**19.6**
Heart	–	78.1	19.4[a]	–	47.8	39.3	**37.4**	42.2	**37.4** **12.2**[a]
Ionosphere	7.1	11.3	–	–	14.3	13.1	19.1	–	**6.2**
Iris	3.0	8.0	–	6.0	5.33	**2.0**	–	–	3.33
Liver	30.7	36.3	–	–	37.1	38.6	42.7	–	**26.2**
Sonar	9.1	23.1	7.7	–	12.5	25.0	–	–	**5.80**
Vehicle	–	29.1	28.5	–	31.6	23.0	55.7	**15.0**	26.3

A = Adamenn, *C* = CDW, *D* = Dann, *K* = *K*-NN, *L* = LDics, *N* = NBayes, *Q* = Qdisc. std = Standard deviation.
[a] Means without considering cost matrix.

$(n-1)$. Furthermore, for data sets with more than two classes, Fuzzy *K*-NN classifier is used to avoid ties (Keller et al., 1985).

4.3. Results and discussion

Table 2 shows the comparison of classification error rate (in %) between TS and other classifiers for different data sets. The combination of feature selection (FS) and feature weighting (FW) technique using TS/*K*-NN has achieved higher accuracy to all data sets except Balance, Vehicle, and Iris. Even for Balance, Vehicle, and Iris data sets, TS/*K*-NN is better than many well-known classifiers. Thus, in 7 out of 10 data sets, TS/*K*-NN has achieved the best performance. For vehicle, TS/*K*-NN has best performance after discriminant classifiers. In Iris, and Balance data sets, TS/*K*-NN has not achieved the best performance. These two data sets unable to combine the benefits of FS and FW as only four features are available. Only FW using TS is used and FS technique is ineffective because of the limited number of features.

Fig. 8 shows the classification error rate with error bars denoting the standard deviation obtained over the five different splits of each data set for the different algorithms. From the graph, it is clear that the proposed classifier has superior results both in terms of error rate and standard deviation.

Table 3 shows a comparison of feature selection algorithms (SFS and SFFS) with TS. TS algorithm is tested with feature selection only (i.e. weights are disabled), TS with feature weighting only (features are disabled) and TS with both feature selection and feature weighting. From the table, it is clear that feature selection using TS has identical error rate when compared with SFS and SFFS in all data sets except Ionosphere and Sonar. TS has outperformed both SFS and SFFS with feature vector of size >30 i.e. Ionosphere and Sonar. In Ionosphere, with 9 features out of 34, the minimum classification error rate is 6.6% using TS as compared to the error rate of 8.5% and 7.5% using SFS and SFFS, respectively. Similarly, in Sonar, with only 24 features out of 60, the minimum classification error rate is 3.4% using TS as compared to the

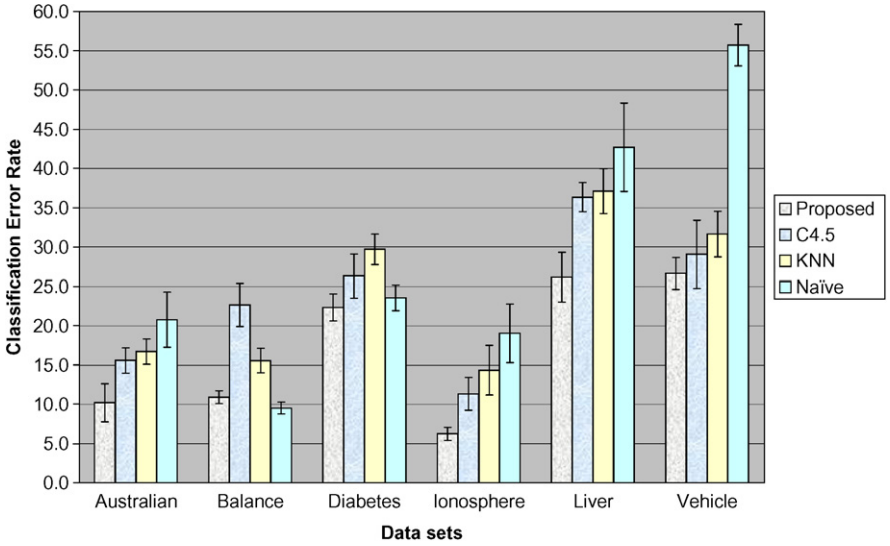

Fig. 8. Classification error rate and standard deviation (error bars) for four different algorithms on various data sets.

Table 3
Comparison of feature selection algorithms with TS

Method	Dataset	E	F	K	Dataset	E	F	K
SFS	Balance	15.6	4	9	Iris	4.6	4	5
SFFS		15.6	4	9		4.6	4	5
TS (FS)		15.6	4	9		4.6	4	5
TS (FW)		10.6	4	9		3.3	4	9
TS (FS + FW)		**10.6**	4	9		**3.3**	4	9
SFS	Liver	29.6	4	1	Diabetes	23.7	4	7
SFFS		29.6	4	1		23.7	4	7
TS (FS)		29.6	4	1		23.7	4	7
TS (FW)		25.0	6	3		22.3	8	7
TS (FS + FW)		**24.0**	4	3		**20.1**	4	5
SFS	Glass	25.7	7	1	Heart	43.0	5	9
SFFS		25.7	7	1		43.0	5	9
TS (FS)		25.7	7	1		43.0	5	9
TS (FW)		20.1	9	1		45.9	13	7
TS (FS + FW)		**19.6**	6	1		**37.4**	8	7
SFS	Australian	11.6	6	7	Vehicle	28.1	15	7
SFFS		11.3	5	9		28.1	15	7
TS (FS)		11.3	5	9		26.6	10	1
TS (FW)		9.1	14	7		28.9	18	3
TS (FS + FW)		**7.7**	9	3		**25.4**	13	3
SFS	Ionosphere	8.5	11	3	Sonar	7.2	35	1
SFFS		7.5	9	3		4.8	33	1
TS (FS)		6.6	9	3		**3.4**	24	1
TS (FW)		8.5	34	1		6.7	60	1
TS (FS + FW)		**5.7**	15	3		5.8	17	1

E = error rate, F = number of features, K = value of K in K-NN classifier, FS = feature selection, FW = feature weighting.

error rate of 7.2% and 4.8% using SFS and SFFS respectively.

Furthermore, it is also clear from Table 3 that simultaneous feature selection (FS) and feature weighting (FW) using hybrid TS/K-NN classifier has superior classification accuracy in all data sets except Sonar. In sonar data set, TS with feature selection only has the highest classification accuracy. It should be noted that TS using FS is the special case of our proposed algorithm in which weights are disabled. Also, in 4 data sets i.e. Liver, Diabetes, Glass, and Sonar, the number of features is less than feature selection algorithms i.e. SFS, SFFS, and TS with FS. Thus, the TS with simultaneous feature selection and weighting not only has the ability to find weights for K-NN classifier that result in higher classification accuracy but also has the ability to reduce the size of the feature vector.

Fig. 9 shows the classification error rate vs number of iterations for Glass and Sonar data sets during the solution search space using Tabu Search. The figure clearly shows that Tabu Search focuses 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.

4.4. Run time parameters

Table 4 shows the Tabu run time parameters chosen after the preliminary experimentation was completed. The number of iterations is 500 for all data sets. Odd values of $K = 1$–9 is used for K-NN classifier in data sets with two classes while values of $K = 1$–10 are used for fuzzy K-NN classier in data sets with more than three classes: Balance, Glass, and Vehicle. The best value of K is searched by a hybrid TS/K-NN classifier as it is encoded

Table 4
Tabu search run time parameters and computation time for training and testing

	$V^* = (M^*N + P)$	T	Training (min)	Testing (ms)
Australian	$10^*2 + 4 = 24$	4	140.0	2.30
Balance	$10^*2 + 2 = 22$	2	16.3	1.99
Diabetes	$10^*2 + 3 = 23$	3	169.3	2.44
Glass	$10^*2 + 3 = 23$	3	13.1	0.366
Heart	$10^*2 + 4 = 24$	4	25.5	1.04
Ionosphere	$10^*2 + 6 = 26$	6	23.34	2.06
Iris	$10^*2 + 2 = 22$	2	4.5	0.313
Liver	$10^*2 + 3 = 23$	3	15.24	0.596
Sonar	$10^*2 + 9 = 29$	9	24.45	0.678
Vehicle	$10^*2 + 5 = 25$	5	153.38	3.05

V^* = Number of neighborhood solutions, T = Tabu List Size, M^*N = neighbors for first part of encoding scheme, P = neighbors for second part of encoding scheme.

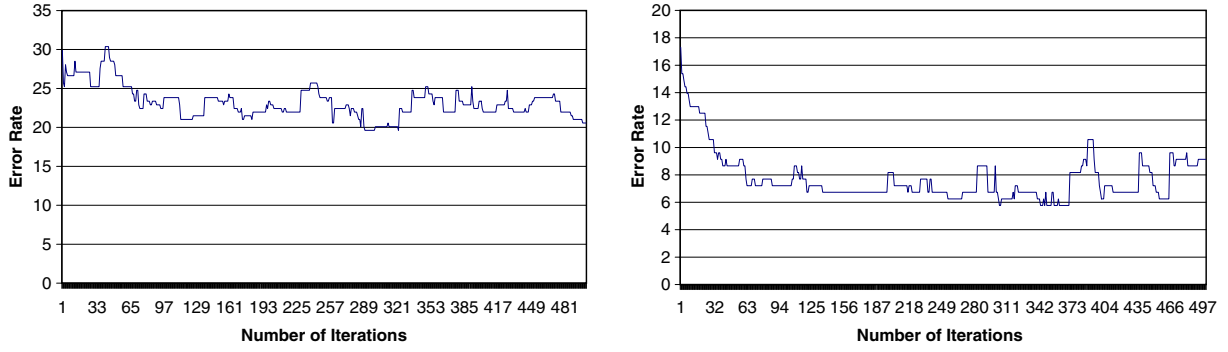

Fig. 9. Error rate vs iterations for Glass and Sonar data sets.

in the encoding solution of the Tabu Search as described in Section 3.1. One of the parameters of TS is the size of Tabu list. As discussed in Section 2, Tabu list size is related to the problem size. The Tabu list size is determined using:

$$T = ceil(\sqrt{F}) \quad (3)$$

where T is the Tabu List Size and F is the number of features. Other parameters used in our proposed algorithm are M, N, and P. M random weights are assigned to N features in each iteration. The values of M and N used for all data sets are 10 and 2, respectively, and these parameters are chosen after a preliminary experimentation. Along with N weighted features, P features are also added/deleted in each iteration. The value of P is determined using Eq. (3). Table 4 also shows the computation time for training and testing using various data sets. The training of the proposed hybrid TS/K-NN classifier is an off-line procedure which is used to find the best weights of features while keeping the classification error rate low. Once the TS finds the best weights for features, a K-NN classifier is used to determine the class of the new sample.

4.5. Case study: prostate cancer classification

The most extensive application of our proposed technique is the classification of prostate cancer using multispectral images. Prostate cancer has become the second most commonly diagnosed cancer in the male population after lung cancer, with approximately 22,800 new cases diagnosed every year in the UK alone. Currently, prostate needle biopsy remains the only conclusive way to make an accurate diagnosis of prostate cancer (Eble and Bostwick, 1996). Recently Roula et al. described a novel approach in which additional spectral data is used for the classification of prostate needle biopsies (Roula, 2002). The aim of the approach is to help pathologists reduce the diagnosis error rate. Instead of analyzing conventional grey scale or RGB colour images, spectral bands have been used in the analysis. This result in a feature vector of size greater than 100. The goal is to classify the following four groups

- Stroma: STR (muscular normal tissue).
- Benign Prostatic Hyperplasia: BPH (a benign condition).
- Prostatic Intraepithelial Neoplasia: PIN (a precursor state for cancer).
- Prostatic Carcinoma: PCa (abnormal tissue development corresponding to cancer).

Fig. 10 shows samples of the four classes.

The data set consists of 592 textured multispectral images with each image taken at 16 spectral channels (Roula, 2002). For each sample, the total number of features is 128. For such a high dimensionality problem, pattern recognition techniques suffer from the well-known curse-of-dimensionality problem (Jain et al., 2000). A novel feature selection technique based on intermediate-memory Tabu Search /1NN classifier is proposed in (Tahir et al., 2004a; Tahir, 2005) to solve this curse-of-dimensionality problem. Table 5 shows the overall classification error with data reduction by using Tabu Search and classification using leave-one-out 1NN classifier. Classification error rate has been reduced to 2.90% as compared to 5.70% reported in Roula (2002). By applying our proposed Feature weighting after Feature selection using TS/K-NN classifier, the classification error has been further improved from 2.90% to 1.80% as shown in Table 6. Fig. 11 shows the classification error rate versus the number of iterations during the solution search space using Tabu Search. For the

Table 5
Classification error by using feature selection through TS/1NN

Classified as	BPH	PCa	PIN	STR	Error (%)
BPH	101	1	0	4	4.71
PCa	1	174	2	0	1.69
PIN	0	2	140	2	2.77
STR	2	2	0	161	2.42
Overall					2.90

Table 6
Classification error by using feature selection followed by feature weighting through TS/1NN

Classified as	BPH	PCa	PIN	STR	Error (%)
BPH	103	3	0	0	2.91
PCa	2	174	1	0	1.69
PIN	0	3	141	0	2.08
STR	0	0	1	164	0.61
Overall					1.82

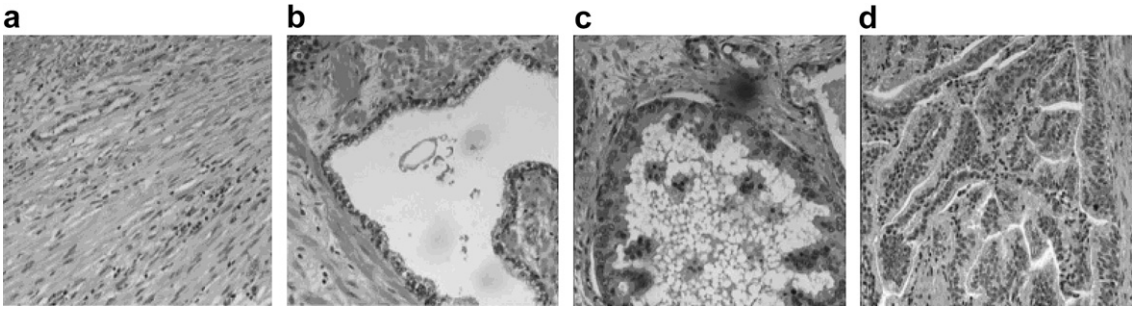

Fig. 10. Images showing representative samples of the four classes. (a) Stroma (b) BPH (c) PIN (d) Cancer.

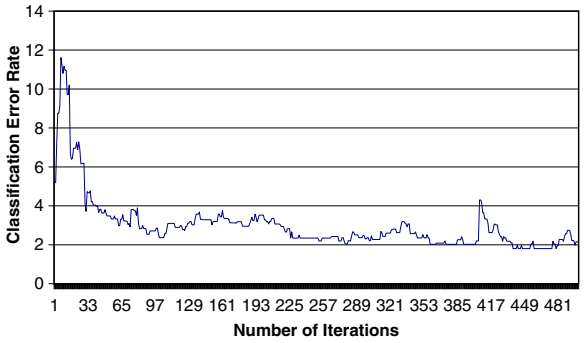

Fig. 11. Error rate vs iterations for prostate cancer dataset.

first few iterations (<50), there is an expected increase in classification error rate because random weights are applied to features that can cause sudden change in behaviour but after few iterations, the proposed hybrid TS/*K*-NN algorithm progressively zooms towards a better solution subspace as weights have stabilized.

5. Conclusion

In this paper, a Tabu Search method is proposed for simultaneous feature selection and feature weighting using *K*-NN rule. The technique has proved effective for improving classification accuracy and has been compared with different classifiers on various data sets. The results have indicated that simultaneous feature selection and extraction not only have the ability to find weights for *K*-NN classifier that result in higher classification accuracy but also have the ability to reduce the size of feature vector. Furthermore, the proposed Tabu Search progressively zoomed towards a better solution subspace as time elapsed, a desirable characteristics of approximation iterative heuristics. The proposed technique is also used to improve the classification accuracy of prostate cancer using multispectral images, an important problem in biomedicine.

References

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

Cover, T.M., Hart, P.E., 1967. Nearest Neighbor Pattern Classification. IEEE Trans. Inform. Theory 13 (1), 21–27.

Cover, T.M., Van Campenhout, J.M., 1997. On the possible orderings in the measurement selection problem. IEEE Trans. Syst. Man Cybernet. 7 (9), 657–661.

Domeniconi, C., Peng, J., Gunopulos, D., 2002. Locally adaptive metric nearest-neighbor classification. IEEE Trans. Pattern Anal. Machine Intell. 24 (9), 1281–1285.

Duda, R.O., Hart, P.E., Stork, D.G., 2001. Pattern Classification. Willey-Interscience.

Eble, J.N., Bostwick, D.G., 1996. Urologic Surgical Pathology. Mosby-Year Book, Inc.

Glover, F., 1989. Tabu search I. ORSA J. Comput. 1 (3), 190–206.

Glover, F., 1990. Tabu search II. ORSA J. Comput. 2 (1), 4–32.

Glover, F., Taillard, E., de Werra, D., 1993. A user's guide to Tabu Search. Ann. Operat. Res. 41, 3–28.

Guverenir, H.A., and Akkus, A., 1997. Weighted K-Nearest Neighbor Classification on Feature Projection. In: Proceedings of the Twelfth International Symposium on Computer and Information Sciences. ISCIS XII.

Hastie, T., Tibshirani, R., 1996. Discriminant Adaptive Nearest Neighbor Classification. IEEE Trans. Pattern Anal. Mach. Intell. 18 (6), 607–615.

Jain, A.K., Duin, R.P.W., Mao, J., 2000. Statistical Pattern Recognition: A Review. IEEE Trans. Pattern Anal. Mach. Intell. 22 (1), 4–37.

Keller, J.M., Gray, M.R., Givens, J.A., 1985. A fuzzy K-nearest neighbor algorithm. IEEE Trans. Syst. Man Cybernet. 15 (4), 580–585.

Korycinski, D., Crawford, M., Barnes, J.W, and Ghosh, J., 2003. Adaptive feature selection for hyperspectral data analysis using a binary hierarchical classifier and Tabu Search. In: Proceedings of the IEEE International Geoscience and Remote Sensing Symposium , IGARSS.

Kudo, M., Sklansky, J., 2000. Comparison of algorithms that select features for pattern classifiers. Pattern Recogn. 33, 25–41.

Lowe, D., 1995. Similarity metric learning for a variable kernel classifier. Neural Comput. 7.

Michie, D., Spiegelhalter, D.J., Taylor, C.C., 1994. Machine Learning Neural and Statistical Classification. Ellis Horwood.

Paredes, R., Vidal, E., 2000. A Class-Dependent Weighted Dissimilarity Measure for Nearest Neighbor Classification Problems. Pattern Recogn. Lett. 21 (12), 1027–1036.

Puch, W.F. et al., 1993. Further research on feature selection and classification using genetic algorithms. In: Proceedings of the Fifth International Conference on Genetic Algorithms, ICGA.

Pudil, P., Novovicova, J., Kittler, J., 1994. Floating search methods in feature selection. Pattern Recogn. Lett. 15 (Nov), 1119–1125.

Quinlan, J.R., 1993. C4.5: Programs for Machine Learning. Morgan-Kaufmann.

Raymer, M.L. et al., 2000. Dimensionality Reduction using Genetic Algorithms. IEEE Trans. Evolution. Comput. 4 (2), 164–171.

Roula, M.A., et al., 2002. A multispectral computer vision system for automatic grading of prostatic neoplasia. In: IEEE International Symposium on Biomedical lmaging.

Sait, S.M., Youssef, H., 1999. General iterative algorithms for combinatorial optimization. IEEE Computer Society.

Siedlecki, W., Sklansy, J., 1989. A note on genetic algorithms for large-scale feature selection. Pattern Recogn. Lett. 10 (11), 335–347.

Statlog Corpora. Dept. Statistics and Modelling Science (Stams). Stratchclyde University. Available from: <http://www.liacc.up.pt/ML/statlog/>.

Tahir, M.A. et al., 2005. A Novel Prostate Cancer Classification Technique using Intermediate Memory Tabu Search Eurasip Journal on Applied Signal Processing, Special Issue: Advances in Intelligent Vision Systems: Methods and Applications (ForthComing).

Tahir, M.A., Bouridane, A., Kurugollu, F., Amira, A., 2004a. Feature Selection using Tabu Search for Improving the Classification Rate of Prostate Needle Biopsies. In: Proc. 17th International Conference on Pattern Recognition (ICPR 2004), Cambridge, UK.

Tahir, M.A., Bouridane, A., Kurugollu, F., 2004b. Simulataneous Feature Selection and Weigthing for Nearest Neighbor Using Tabu Search. In: Lecture Notes in Computer Science (LNCS 3177), 5th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL 2004), Exeter, UK.

Wettschereck, D., Aha, D.W., Mohri, T., 1997. A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. Artif. Intell. Rev. 11 (1–5), 273–314.

Whitney, A.W., 1971. A direct method of nonparametric measurement selection. IEEE Trans. Comput. 20 (9), 1100–1103.

Zhang, H., Sun, G., 2002. Feature selection using Tabu Search method. Pattern Recog. 35, 701–711.