We are IntechOpen,
the world's leading publisher of Open Access books
Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

BOOK CITATION INDEX
CLARIVATE ANALYTICS
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Feature Selection using Intensified Tabu Search for Supervised Classification

Muhammad Atif Tahir and Jim Smith
Department of Computer Science, Bristol Institute of Technology,
University of the West of England,
UK

1. Introduction

Feature selection algorithms are popular methods to reduce the dimensionality of the feature space and remove the redundant, irrelevant or noisy data. The term feature selection refers to the selection of the best subset of the input feature set. These methods used in the design of pattern classifiers have three goals:
1. to reduce the cost of extracting the features
2. to improve the classification accuracy
3. to improve the reliability of the estimation of the performance, since a reduced feature set requires less training samples in the training process of a pattern classifier [1, 2]

Feature selection produces savings in the measuring features (since some of the features are discarded) and the selected features retain their original physical interpretation [1]. This feature selection problem can be viewed as a multiobjective optimisation problem since it involves minimising the feature subset while maximizing the classification accuracy.

Mathematically, the feature selection problem can be formulated as follows. Suppose X is an original feature vector with cardinality n and $\bar{X}$ is the new feature vector with cardinality $\bar{n}$, $\bar{X} \subseteq X$, $J(\bar{X})$ is the selection criterion function for the new feature vector $\bar{X}$. The goal is to optimize $J()$.

Feature selection problem is NP-hard (Non-deterministic Polynomial-time hard) [3, 4]. Therefore, the optimal solution can only be achieved by performing an exhaustive search in the solution space [5]. However, exhaustive search is feasible only for small n where n is the number of features. A number of algorithms have been proposed for feature selection to obtain near-optimal solutions [1, 2, 6, 7, 8, 9, 10, 30]. 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 according to n belonging to the intervals [0,19], [20,49], or [50,1], respectively [2, 8]. Sequential Forward Selection (SFS) [11] is the simplest greedy sequential search algorithm and has been used for land mine detection using multispectral images [12]. 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 [7]. However, these algorithms suffer from the deficiency of converging to local optimal solutions for large

scale problems when $n > 50$ [2, 8]. 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 [8, 9, 13, 14].

Feature selection algorithms can be broadly divided into two categories [32, 31]: *filters*, and *wrappers*. The filter approach evaluates the relevance of each feature or feature subset using the data set alone and without using any machine learning algorithm [32]. On the other hand, Wrapper approach uses machine learning algorithms to evaluate the relevance of feature subset [31]. An extensive summary of different filter and wrapper approaches to feature selection is provided by [33]. As reported in [31], when the objective is to maximize the classification accuracy of a given feature subset, the features selected should depend not only on relevance of the data but also on the machine learning algorithm. For that reason, throughout this chapter, the wrapper feature selection approach using Naive Bayes classifier has been adopted.

In previous papers [23, 24, 25], we have proposed tabu search (TS) based computational intelligence techniques to solve feature selection problems. Nearest Neighbor classifier was used previously as objective function. In this chapter; instead of instance based classifier; we will explore Naive Bayes Classifier compared with sequential feature selection algorithms. The aim is to maximize the classification accuracy while minimizing the number of features.

This chapter is organized as follows. Section 2 gives an overview about Tabu Search and Fuzzy objective function proposed in [25] followed by fea ture selection using Tabu Search proposed originally by [8] and then modified by Tahir et al [27], by introducing feature search intensification strategy in Section 3. Section 4 discusses experiments carried out followed by discussion on other advanced Tabu Search approaches for classification problems in sec- tion 5. Section 6 concludes the chapter.

2. Overview of Tabu Search, and fuzzy logic

TS was introduced by Fred Glover [15, 16] as a general iterative metaheuristic for solving combinatorial optimisation problems. Tabu Search is conceptually simple and elegant. It is a form of local neighbourhood search. Each solution $S \in \Omega$ has an associated set of neighbours $N(S) \subseteq \Omega$ 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 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 status of a solution is overridden when a 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 the promising regions of the feasible domain. In the second case, an attempt is made to consider solutions in a broad area of the search space. The Tabu Search algorithm is given in Algorithm 1.

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 [?]. Suggested values of tabu list size 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) [13].

Algorithm 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
1.   Start with an initial feasible solution $S \in \Omega$.
2.   Initialize tabu list and aspiration level.
3.   For fixed number of iterations Do
4.   Generate neighbour solutions $V^* \subset N(S)$.
5.   Find best $S^* \in V^*$.
6.   If move S to S^* is not in T Then
7.   Accept move and update best solution.
8.   Update tabu list and aspiration level.
9.   Increment iteration number.
10.   Else
11.   If $Cost(S^*) < AL$ Then
12.   Accept move and update best solution.
13.   Update tabu list and aspiration level.
14.   Increment iteration number.
15.   End If
16.   End If
17.   End For
End

2.1 Fuzzy logic

In this chapter, we present an intermediate TS algorithm, where the quality of a solution is characterized by a fuzzy logic rule expressed in linguistic variables of the problem domain. Fuzzy set theory has recently been applied in many areas of science and engineering. In the most practical situations, one is faced with several concurrent objectives. Classic approaches usually deal with such difficulty by computing a single utility function as a weighted sum of the individual objectives, where more important objectives are assigned higher weights [25]. Balancing different objectives by weight functions is at best controversial. Fuzzy logic is a convenient vehicle for trading off different objectives. It allows the mapping of values of different criteria into linguistic values which characterize the level of satisfaction of the designer with the numerical value of objectives and operation over the interval [0,1] defined by the membership functions for each objective.

Three linguistic variables are defined to correspond to the three component objective functions: number-of-features f_1, number-of-incorrect predictions f_2, and average

classification error rate f_3. One linguistic value is defined for each component of the objective function. These linguistic values characterize the degree of satisfaction of the designer with the values of objectives $f_i(x)$, i = {1, 2, 3}. These degrees of satisfaction are described by the membership functions $\mu_i(x)$ on fuzzy sets of the linguistic values where $\mu(x)$ is the membership value for solution x in the fuzzy set. The membership functions for the minimum number of features, the minimum number of incorrect predictions, and the low classification error rate are easy to build. They are assumed to be non- increasing functions because the smaller the number of features $f_1(x)$, the number of incorrect predictions $f_2(x)$, and the classification error rate $f_3(x)$, the higher is the degree of satisfaction $\mu_1(x)$, $\mu_2(x)$, and $\mu_3(x)$ of the expert system (see Figure 1). The fuzzy subset of a good solution is defined by the following Fuzzy logic rule:

"**IF** a solution has *small number of features* AND *small number of incorrect predictions* AND *low classification error rate* **THEN** it is a good solution" According to the and/ or like ordered-weighted-averaging Logic [17, 18], the above rule corresponds to the following:

$$\mu(x) = \gamma \times min(\mu_i(x)) + (1 - \gamma) \times \frac{1}{3} \sum_{i=1}^{3} \mu_i(x). \tag{1}$$

where γ is a constant in the range [0,1]. The shape of the membershi function $\mu(x)$ is shown in Figure 1. Membership of data in a fuzzy set is defined using values in the range [0,1]. The membership values for the number of features F, the number of incorrect predictions P, and the classification error rate E are computed using equations 2, 3, and 4 respectively.
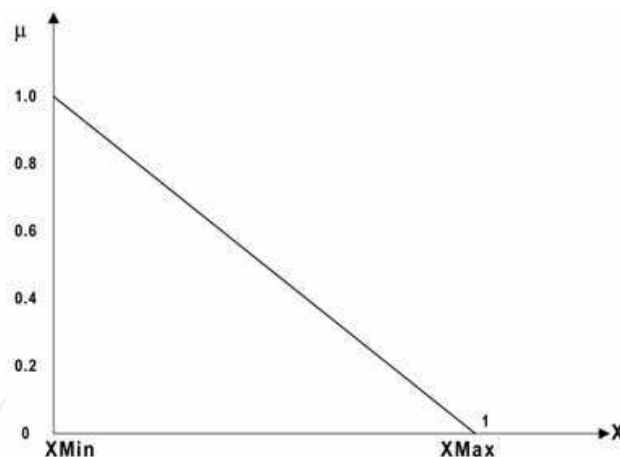

Fig. 1. Membership function for fuzzy subset X, where, in this application, X is the number of features F, the number of incorrect predictions P, or the classi⁻cation error rate E.

$$\mu_1(x) = \begin{cases} 1 & \text{if } F \leq FMin \\ \frac{FMax-F}{FMax-FMin} & \text{if } FMin \leq F \leq FMax \\ 0 & \text{if } FMax \leq F \end{cases}$$

$$\mu_2(x) = \begin{cases} 1 & \text{if } P \leq PMin \\ \frac{PMax-P}{PMax-PMin} & \text{if } PMin \leq P \leq PMax \\ 0 & \text{if } PMax \leq P \end{cases}$$

$$\mu_3(x) = \begin{cases} 1 & \text{if } E \leq EMin \\ \frac{EMax-E}{EMax-EMin} & \text{if } EMin \leq E \leq EMax \\ 0 & \text{if } EMax \leq E \end{cases}$$

The maximum number of features (*FMax*) is the size of the feature vector and the minimum number of features (*FMin*) is 1 or *F*=2. The maximum number of incorrect predictions (*PMax*) and the maximum classification error rate (*EMax*) is determined by applying Naive Bayes Classifier for the initial solution. The minimum number of incorrect predictions (*PMin*) is 0 while the minimum classification error rate (*EMin*) is 0%.

3 Feature selection using Tabu Search

3.1 Initial solution
Feature selection vector is represented by a 0/ 1 bit string where 0 shows the feature is not included in the solution while 1 shows the feature is included. All features are included in the initial solution.

3.2 Neighborhood solutions
Neighbors are generated by randomly adding or deleting a feature from the feature vector of size *n*. For example, if 11001 is the current feature vector, then the possible neighbors with a candidate list size of 3 can be 10001, 11101, 01001. Among the neighbors, 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.

3.3 Tabu moves
A tabu list is maintained to avoid returning to previously visited solutions. With this approach, if a feature (move) is added or deleted at iteration *i*, then adding or deleting the same feature (move) for *T* iterations (tabu list size) is Tabu.

3.4 Aspiration criterion
Aspiration criterion is a mechanism used to override the tabu status of moves. It temporarily overrides the tabu status if the move is su±ciently good. In our approach, if a feature is added or deleted at iteration *i* and this move results in a best cost for all previous iterations, then this feature is allowed to add or delete even if it is in the tabu list.

3.5 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.

3.6 Intensification of the search
For intensification, the search is accentuated in the promising regions of the feasible domain. Intensification is based on some intermediate-term memory. Since, the solution space is

extremely large (initial feature vector $n > 100$), it is important to intensify the search in the promising regions by removing poor features from the search space. The following steps are used to intensify the search

- STEP1: Store M best solutions in intermediate memory for T_1 number of iterations.
- STEP2: Remove features that are not included in the best M solutions for N times.
- STEP3: Re-run the tabu search with the reduced set of features for another T_2 iterations.
- STEP4: Repeat steps 1-3 until the optimal or near-optimal solution is achieved.

where values of M and N can be determined empirically through experiments. As an example, assume that the following four best solutions as shown in Figure 2 are found by tabu search during T_1 iterations. Features f_1 and f_3 are always used while feature f_5 is never used for good solutions. For $N = 2$, the reduced feature set consists of only f_1, f_2, f_3, and f_6. Thus, tabu search will search for the near-optimal solutions in reduced search space avoid visiting non promising regions.

f1	f2	f3	f4	f5	f6	f7
1	0	1	0	0	1	0
1	0	1	1	0	1	0
1	1	1	0	0	0	1
1	1	1	0	0	1	0

Σ	4	2	4	1	0	3	1

Fig. 2. An example showing intensification steps for tabu search. Σ is the number of occurrences of each feature in the best solutions.

4. Experiments

We have performed a number of experiments and comparisons on several public data sets from the UCI [21] and non-public data sets from DynaVis Project [22] in order to demonstrate the performance of the classification system using Tabu Search. A short description of the used benchmarks is mentioned in Table 1. Classification results have been obtained by using N- Fold Cross Validation. In N-Fold CV, each dataset is divided into N blocks using N-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.

In addition, comparisons with several feature selection algorithms were also performed as mentioned below.

- Sequential Forward Search (SFS) [11]: 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) [7]: SFFS improved the SF method by introducing backward steps after each forward step as long as the objective criteria increases.
- TS1: Tabu Search without Intensification [8, 27]
- TS2: Tabu Search with Intensification [25]

Data sets	Name	Samples	Features	Classes
UCI	Australian	690	14	2
	Breast Cancer	569	30	2
	German	1000	20	2
	Ionosphere	351	34	2
	Musk	476	166	2
	Sonar	208	60	2
DynaVis	CD Print 1	1534	74	2
	CD Print 2	1534	74	2
	CD Print 3	1534	74	2
	CD Print 4	1534	74	2
	Egg	4238	74	2
	Rotor	225	72	2

Table 1. Data sets Description. S = Samples, F = Features, C = Classes.

Table 2 shows a comparison of feature selection algorithms (SFS an SFFS) with TS-1 and TS-2 for UCI Datasets. Naive Bayes is used as a base classifier. From the table, it is clear that feature selection using TS has achieved higher accuracy when compared with SFS and SFFS in all data sets where $F > 50$. The most significant improvements are in terms of number of features. For German, Breast Cancer, and Ionosphere; only 9, 8 and 11 fea- tures are used respectively without comprising any accuracy when compared with SFS and SFFS. In Sonar, with only 18 features out of 60, the classification accuracy is 84.5% using TS-1 as compared to the accuracy of 80.8% and 82.2% using SFS and SFFS respectively. The number of features used by SFS and SFFS are 13 and 27 respectively for sonar. Similarly, in Musk, with only 37 features out of 166, the classification accuracy is 85.9% using TS-1 as compared to the accuracy of 83.2% and 84.0% using SFS and SFFS respectively. The number of features used by SFS and SFFS are 98 and 89 respectively for musk.

When compared tabu search with and without intensification; some improvement is achieved in terms of the number of features. For Australian; German and Breast Cancer; both TS-1 and TS-2 are identical. For Ionosphere, 11 features are used by TS-2 instead of 13 by TS-1 but with the same accuracy. In Sonar, with only 10 features out of 60, accuracy of 86.5% is achieved as compared to 18 features by TS-1 with accuracy 84.5%. In Musk, with only 36 features out of 166, accuracy of 86.8% is achieved as compared to 37 features by TS-1 with accuracy 85.9%.

Table 2 shows a comparison of feature selection algorithms (SFS and SFFS) with TS1 and TS2 for DynaVis Datasets. In all Data sets except rotor, some improvement in terms of accuracy and number of features are achieved using TS-2. In Rotor, with only 8 features out of 72, accuracy of 95.3% is achieved using TS-2 as compared to 92.4 (35 features) by SFS, 92.9 (8 features) by SFFS, and 95.1 (17 features) by TS-1.

Table 4 shows the computation cost for using various data sets and feature selection algorithms. The main computation cost is the evaluation of objective function or naive bayes classifier. Hence; number of evaluations are used as the main criterion to compare FS algorithms. As number of features increase; the number of evaluations are also increased. The number of iterations used for TS-1 are 500 for $F < 30$, 1000 for $F < 70$, and 2000 for $F > 70$. It should be noted that although it appears that TS-2/ NB requires highest number of evaluations except in few cases when compared with SFFS but these evaluations are performed with fewer features. As an example; for Musk initially 26000 evaluations are used using 166 features, then 26000 but using 121 features, then 26000 using 104 features, and finally 26000 using only 92 features.

Data set	Method	Accuracy	Features Used	Standard Deviation
Australian	NB	80	14	3.40
	SFS/NB	87.7	7	4.00
	SFFS/NB	87.7	7	4.00
	TS-1/NB	87.7	7	4.00
	TS-2/NB	87.7	7	4.00
German	NB	73.6	20	5.32
	SFS/NB	75.9	10	4.53
	SFFS/NB	76.5	11	2.88
	TS-1/NB	76.5	9	3.63
	TS-2/NB	76.6	9	2.67
Breast Cancer	NB	93.3	30	2.09
	SFS/NB	97.4	5	2.48
	SFFS/NB	98.1	9	1.32
	TS-1/NB	98.1	8	1.32
	TS-2/NB	98.1	8	1.32
Ionosphere	NB	82.0	33	4.61
	SFS/NB	93.4	10	4.48
	SFFS/NB	93.7	12	4.22
	TS-1/NB	93.7	13	4.22
	TS-2/NB	93.7	11	4.22
Sonar	NB	68.3	60	11.39
	SFS/NB	80.8	13	7.00
	SFFS/NB	82.2	27	8.91
	TS-1/NB	84.5	18	8.60
	TS-2/NB	86.5	10	7.82
Musk	NB	73.3	166	4.08
	SFS/NB	83.2	98	4.85
	SFFS/NB	84.0	89	5.23
	TS-1/NB	85.9	37	4.30
	TS-2/NB	86.8	36	3.98

Table 2. Comparison of TS/ NB with NB, SFS/ NB, and SFFS/ NB Classifier for UCI Datasets.

Two run time parameters i.e. the tabu list size and Number of neighbour- hood Solutions are determined using the following equation:

$$T = V^* = \text{ceil}(\sqrt{F}) \tag{5}$$

where T is the Tabu List Size, V^* is the number of neighbourhood solutions and F is the number of features.

5. Other advanced Tabu Search approaches for supervised classification

5.1 Simultaneous feature selection and feature weighting using hybrid tabu search/ *K*-nearest neighbor classifier

In [24], 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

Data set	Method	Accuracy	Features Used	Standard Deviation
CD Print1	NB	87.2	74	2.63
	SFS/NB	92.3	8	1.91
	SFFS/NB	92.6	14	1.87
	TS-1/NB	92.5	21	1.69
	TS-2/NB	92.8	21	1.70
	TS-2/NB	92.1	8	2.00
CD Print2	NB	92.6	74	2.04
	SFS/NB	95.6	10	1.54
	SFFS/NB	95.7	14	1.32
	TS-1/NB	95.7	11	1.39
	TS-2/NB	95.8	13	1.40
	TS-2/NB	95.7	8	1.46
CD Print3	NB	89.0	74	2.43
	SFS/NB	92.7	14	1.98
	SFFS/NB	92.8	12	1.94
	TS-1/NB	93.1	15	1.86
	TS-2/NB	93.4	9	1.83
CD Print4	NB	90.4	74	1.00
	SFS/NB	93.2	9	1.20
	SFFS/NB	93.2	6	1.23
	TS-1/NB	93.6	12	1.28
	TS-2/NB	93.7	14	1.29
	TS-2/NB	93.2	6	1.25
Egg	NB	86.6	74	7.49
	SFS/NB	92.3	11	4.47
	SFFS/NB	92.4	14	4.60
	TS-1/NB	92.5	13	4.82
	TS-2/NB	92.5	13	4.82
	TS-2/NB	92.4	11	4.94
Rotor	NB	82.7	71	8.02
	SFS/NB	92.4	35	6.79
	SFFS/NB	92.9	8	7.46
	TS-1/NB	93.8	20	6.84
	TS-2/NB	95.1	17	5.85
	TS-2/NB	93.3	8	7.22

Table 3. Comparison of TS/NB with NB, SFS/NB, and SFFS/NB Classifier for DynaVis Datasets.

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:

$$D(x,y) = \sum_{i=1}^{m} (x_i - y_i)^2 \qquad (6)$$

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

Dataset	NB	SFS/NB	SFFS/NB	TS-1/NB Iterations $* (\sqrt{F} + 1)$	TS-2/NB TS-1/NB $\times$ {2,3,4}
Australian	1	106	326	2000	4000
German	1	211	929	2500	5000
Breast Cancer	1	466	2902	6000	12000
Ionosphere	1	562	12705	6000	12000
Sonar	1	1831	13637	8000	24000
Musk	1	13862	182399	26000	104000
CD Print 1	1	2776	73472	18000	54000
CD Print 2	1	2776	26472	18000	54000
CD Print 3	1	2776	23232	18000	54000
CD Print 4	1	2776	19902	18000	54000
Egg	1	2776	376593	18000	54000
Rotor	1	2629	53811	18000	54000

Table 4. Computation cost for various feature selection algorithms.

The classification accuracy obtained from TS/ K-NN classifier is then compared and assessed with published results of several commonly-employed pattern classification algorithms. The results have indicated that simultaneous feature selection and weighting 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.

5.2 Round robin Tabu Search algorithm for multi-class problems

In [23], a novel round robin classification algorithm using a Tabu Search/ Nearest Neighbor (TS/ 1NN) classifier to improve the classification accuracy of multi- class problems. Round robin classification is a technique which is suitable for use in multi-class problems. The technique consists of dividing the multi-class problem into an appropriate number of simpler binary classification problems [26]. Each binary classifier is implemented as TS/ 1NN classifier and the final outcome is computed using a simple voting technique. A key characteristic of this approach is that, in a binary class, the classifier tries to find features that distinguish only that class. Thus, different features are selected for each binary classifier, resulting in an overall increase in classification accuracy. In contrast, in a multi-class problem, the classifier tries to find those features that distinguish all classes at once. Results have indicated a significant increase in the classification accuracy for prostate cancer multi-class problem.

5.3 Feature selection for heterogeneous ensembles of nearest neighbour classifiers using hybrid Tabu Search

A new ensemble technique is proposed in [29, 28] to improve the performance of nearest neighbour (NN) classifier. This 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). This ensemble classifier is evaluated using various benchmark data sets from UCI Machine Learning Repository. Results indicate a significant increase in the performance when compared with different well-known classifiers.

6. Conclusion

Feature selection (FS) algorithms are popular methods to reduce the dimensionality of the feature space and remove the redundant, irrelevant or noisy data and improves the classification accuracy. In this chapter; we have dis- cussed our recently proposed tabu search based algorithms for feature se- lection problems and have compared with other sequential feature selection algorithms. Tabu search is refined with respect to traditional approaches: A feature search intensification strategy is introduced. Search intensification is realized by incorporation of intermediate term memory in the search process. We have also discussed advanced tabu search approaches for supervised classification. Like many combinatorial optimization problems such as VLSI, Time Tabling, and Scheduling in which TS is quite useful, TS is also quite effective in data mining feature selection problems.

7. Acknowledgement

8. References

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

M. Kudo and J. Sklansky. Comparison of algorithms that select features for pattern classifiers. *Pattern Recognition*, 33, 25-41, 2000.

E. Amaldi, and V. Kann. On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. Theoretical Computer Science, 209, 237-260, 1998.

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

T. M. Cover, J. M. Van Campenhout. On the possible orderings in the measure- ment selection problem. *IEEE Transactions on Systems, Man, and Cybernetics*, 7(9), 657-661, 1977.

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

P. Pudil, J. Novovicova, and J. Kittler. Floating search methods in feature selec- tion. *Pattern Recognition Letters*, 15, pp. 1119-1125, 1994.

H. Zhang and G. Sun. Feature selection using tabu search method. *Pattern Recognition*, 35, 701-711, 2002.

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

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

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

G. A. Clark et al. Multispectral image feature selection for land mine detection. *IEEE Transactions on Geoscience and Remote Sensing*. 38(1), pp. 304-311, 2000.

S. M. Sait and H. Youssef. General Iterative Algorithms for Combinatorial Optimization. IEEE Computer Society, 1999.

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

F. Glover. Tabu search I. *ORSA Journal on Computing*, 1(3), 190-206, 1989.

F. Glover (1990). Tabu search II. *ORSA Journal on Computing*, 2(1), 4-32, 1990.

H. J. Zimmerman. Fuzzy Set Theory and Its application. Kluwer Academic Publishers, 3rd Edition, 1996.

Salman A. Khan, Sadiq M. Sait, Habib Youssef. Topology Design of Switched Enterprise Networks Using Fuzzy Simulated Evolution Algorithm. *Engineering Applications of Artificial Intelligence*, Elsevier Science Ltd, pp. 327-340, 2002.

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

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

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

URL: www.dynavis.org

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

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

M. A. Tahir et al. 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*, 14, 2241-2249, 2005.

J. Furnkranz. Round Robin Classification. *Journal of Machine Learning Re- search*, 2, 721-747, 2002.

M. A. Tahir et al. Feature Selection using Tabu Search for Improving the Classification Rate of Prostate Needle Biopsies. *IEEE Proceedings of the International Conference on Pattern Recognition*, Cambridge, 2004.

M. A. Tahir and J. Smith. Feature Selection for Heterogeneous Ensembles of Nearest Neighbour Classifiers using Hybrid Tabu Search Advances in Metaheuristics for Hard Optimization, Springer Verlag, 2008.

M. A. Tahir and J. Smith. Improving Nearest Neighbor Classifier using Tabu Search and Ensemble Distance Metrics Proceedings of the sixth IEEE International Conference on Data Mining (ICDM), HongKong, 2006.

D. A. Bell and H. Wang A formalism for relevance and its application in Feature Subset Selection, Machine Learning, 41, 175-195, 2000.

R. Kohavi and G. John Wrappers for Feature Subset Selection, Artificial Intelligence, 97(1{2), 273-234, 1997.

G. H. John and R. Kohavi and K. Pfleger Irrelevant features and the subset selection problem, Proceedings of the Eleventh International Conference on Machine learning, New Brunswick,NJ, Morgan Kaufmann, 121-129, 1994.

A. Blum and P. Langley Selection of Relevant Features and Examples in Machine Learning, Artificial Intelligence, 97(1-2), 245-271, 1997.
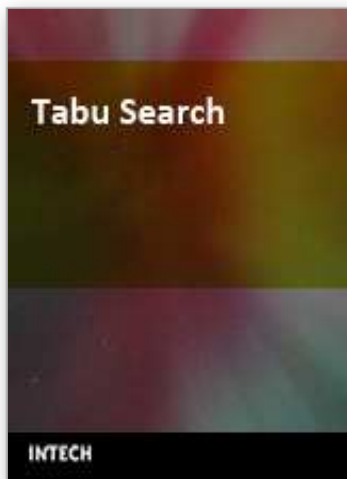
Tabu Search

Edited by Wassim Jaziri

The goal of this book is to report original researches on algorithms and applications of Tabu Search to real-world problems as well as recent improvements and extensions on its concepts and algorithms. The book' Chapters identify useful new implementations and ways to integrate and apply the principles of Tabu Search, to hybrid it with others optimization methods, to prove new theoretical results, and to describe the successful application of optimization methods to real world problems. Chapters were selected after a careful review process by reviewers, based on the originality, relevance and their contribution to local search techniques and more precisely to Tabu Search.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

INTECH
open science | open minds