# Dynamic Oscillating Search Algorithm for Feature Selection

Somol P., Novovičová J., Grim, J.
*Dept. of Pattern Recognition*
*Inst. of Information Theory and Automation*
*Academy of Sciences of the Czech Republic*
*Pod vodárenskou věží 4, CZ 182 08 Prague 8*
{*somol,novovic,grim*}*@utia.cas.cz*

Pudil P.
*Faculty of Management*
*Prague University of Economics*
*Jarošovská 1117/II*
*CZ 377 01, Jindřichův Hradec*
*pudil@fm.vse.cz*

## Abstract

*We introduce a new feature selection method suitable for non-monotonic criteria, i.e., for Wrapper-based feature selection. Inspired by Oscillating Search, the Dynamic Oscillating Search: (i) is deterministic, (ii) optimizes subset size, (iii) has built-in preference of smaller subsets, (iv) has higher optimization performance than other sequential methods. We show that the new algorithm is capable of over-performing older methods not only in criterion maximization ability but in some cases also in obtaining subsets that generalize better.*

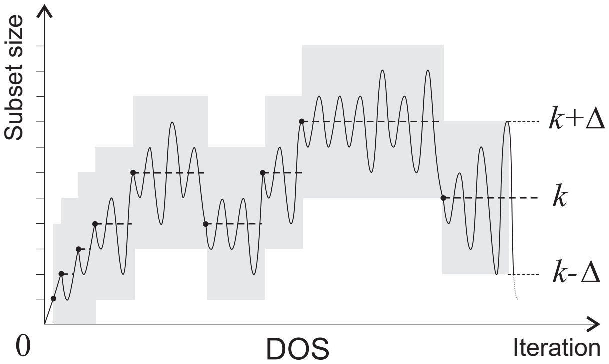

**Figure 1. The DOS course of search**

## 1. Introduction

In feature selection (FS) the search problem of finding a subset of $d$ features from the given set of $D$ measurements, $d < D$, with the aim to improve various properties of pattern recognition systems (i.e., to maximize a suitable criterion function) has been of interest for a long time. Since the optimal methods (exhaustive search or the Branch-and-Bound [2]) are not suitable for non-monotonic criteria nor high-dimensional problems, research has focused on sub-optimal search methods (for recent overviews see [5], [9]). While many approaches to sub-optimal FS are possible (e.g., using evolutionary [3] or Relief-type methods [9]) the family of sequential search methods [2] [9] has been particularly popular due to their good compromise between speed and optimization efficiency, as well as usability with wide variety of criterion functions.

In this paper we introduce a new method extending the principle of Oscillating Search (OS) [10]. While OS requires $d$ to be specified by user (as is the case with most sequential FS methods), the new method determines the best subset size automatically, with preference put on smaller subsets. This ability makes it particularly suitable for Wrapper [4] [5] type of FS, which has recently gained lots of interest. Moreover, the new method has better optimization ability, yielding more often results closer to optimum. Although stronger optimization is naturally accompanied by higher risk of feature over-selection [8], the new method is capable of improving classifier generalization as well.

## 2. Dynamic Oscillating Search

To enable formal description of the Dynamic Oscillating Search (DOS) we follow the notion from [10]. Let $Y$ denote the set of all $D$ features. Let $X_k$ denote the current subset of $k$ features. Let $J(\cdot)$ denote the adopted criterion. The *worst* feature $o$-tuple in $X_k$ should be ideally such a set $\bar{W} \subset X_k$, that

$$J(X_k \setminus \bar{W}) = \max_{W \in \mathcal{W}} J(X_k \setminus W),$$

where $\mathcal{W} = \{W : W \subset X_k, |W| = o\}$. The *best* feature $o$-tuple for $X_k$ should be ideally such a set $\bar{B} \in \mathcal{B}$, where $\mathcal{B} = \{B : B \subset Y \setminus X_k, |B| = o\}$, that

$$J(X_k \cup \bar{B}) = \max_{B \in \mathcal{B}} J(X_k \cup B).$$

978-1-4244-2175-6/08/$25.00 ©2008 IEEE

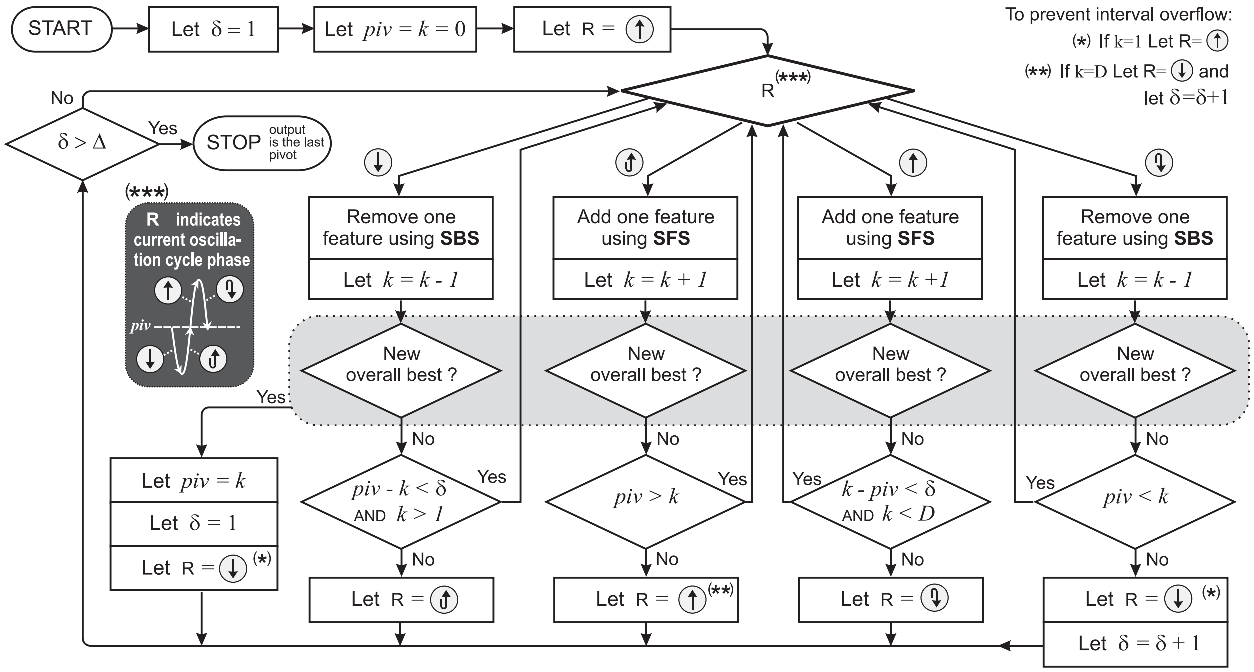

**Figure 2. Simplified diagram of the DOS algorithm assuming $o = 1$.**

In practice we allow sub-optimal finding of the *worst* and *best* $o$-tuples to save computational time.

## 2.1. Algorithm Description

Let REMOVE($\delta$,$o$) denote the sequence of $\delta$ consecutive removals of the *worst* feature $o$-tuples from a feature subset $X_k$ to obtain subset $X_{k-\delta\cdot o}$; let ADD($\delta$,$o$) denote the sequence of $\delta$ consecutive additions of the *best* feature $o$-tuples to a feature subset $X_k$ to obtain subset $X_{k+\delta\cdot o}$. In the following we assume $o$ is set by default to $o = 1$. Higher $o$ values can be specified to obtain the *generalized* DOS version.

The idea of the original OS algorithm is to "oscillate", or repeat consecutive REMOVE($\cdot$) and ADD($\cdot$) steps (and vice versa) to possibly improve a working feature subset of a given size. If the last oscillation cycle led to no improvement, the number of feature $o$-tuples to be consecutively removed and added in one cycle is allowed to increase up to a user-specified limit $\Delta$. In our context we denote the working subset the *pivot* and let the new algorithm change its size, denoted $piv$, in the course of search. This is made possible by introducing a simple rule: whenever a better global solution is found (at any oscillation phase), restart the oscillation process with the new best feature subset taken as the new *pivot*.

**Dynamic Oscillating Search Algorithm**

*Initialization*: Starting from empty set call ADD(3,$o$) to obtain the initial subset. Let $piv = 3o$.

**Step 1**: Let $\delta = 1$. Let the current subset be the *pivot*.

**Step 2**: If $\delta > piv/o - 1$ then go to Step 5.

**Step 3**: REMOVE($\delta$,$o$). If the best of intermediate subsets $X_k$, $k = piv - o, piv - 2o, \ldots, piv - \delta o$ yields higher criterion value than the so-far best (or equal with smaller subset size), go to Step 1.

**Step 4**: ADD($\delta$,$o$). If the best of intermediate subsets $X_k, k = piv - (\delta - 1)o, piv - (\delta - 2)o, \ldots, piv$ yields higher criterion value than the so-far best (or equal with smaller subset size), go to Step 1.

**Step 5**: If $\delta > (D - piv)/o$ then go to Step 8.

**Step 6**: ADD($\delta$,$o$). If the best of intermediate subsets $X_k, k = piv + o, piv + 2o, \ldots, piv + \delta o$ yields higher criterion value than the so-far best (or equal with smaller subset size), go to Step 1.

**Step 7**: REMOVE($\delta$,$o$). If the best of intermediate subsets $X_k$, $k = piv + (\delta - 1)o, piv + (\delta - 2)o, \ldots, piv$ yields higher criterion value than the so-far best (or equal with smaller subset size), go to Step 1.

**Step 8**: No improvement in previous oscillation cycle. Let $\delta = \delta + 1$.

**Step 9**: If $\delta > \Delta$ then STOP, else go to Step 2.

An alternative explanation of the same DOS principle (assuming for simplicity $o = 1$) is given in Fig. 2.

### 2.2. New Algorithm Properties

In the course of search DOS generates a sequence of solutions with ascending criterion values (while smaller subsets are preferred to larger subsets). The search time vs. closeness-to-optimum trade-off can thus be handled by means of pre-mature search interruption.

The number of criterion evaluations is in the O$(n^3)$ order of magnitude. Nevertheless, the total search time depends heavily on the chosen $\Delta$ value, on particular data and criterion settings, and on the unpredictable number of oscillation cycle restarts that take place after each solution improvement. In our experiments (see later) DOS run roughly up to $10\times$ slower than SFFS.

## 3. Evaluating FS Methods' Performance

In older papers the prevailing approach to FS method performance assessment was to evaluate the ability to find optimum, or to get as close to optimum as possible, with respect to some criterion function defined to distinguish classes in classification tasks or to fit data in approximation tasks. Recently, emphasis is put on assessing the impact of FS on generalization performance, i.e., the ability of the devised decision rule to perform well on independent data. It has been shown that similarly to classifier over-training the effect of feature over-selection can hinder the performance of pattern recognition system [8]; especially with small-sample or high-dimensional problems.

We evaluate our new method from both perspectives – its optimization performance and its impact on classification performance with independent test data. To enable this evaluation we employ the so-called 2-Tier Cross-Validation (CV) process, consisting of *outer* and *inner* CV loops. The purpose of the *outer* loop (to be denoted O-CV) is to put aside part of the data for independent testing, while the *inner* loop (to be denoted I-CV) is used on the remaining data in the course of FS process to evaluate classification performance (the actual FS criterion).

### 3.1. Experiments

We compare the DOS algorithm (unrestricted, i.e., $\Delta = D$) with standard sequential methods: Sequential Forward Selection (SFS) [2], Sequential Forward Floating Selection (SFFS) [7] and OS (individually best initialization, $\Delta = 1$) [10]. In case of methods that select subsets of given size $d$ we repeated the search for each $d = 1, \ldots, D$ to eventually choose the best overall result. We used the accuracy of various classifiers as criterion function: Bayesian classifier assuming Gauss distribution, 5-Nearest Neighbor and SVM with RBF kernel [1]. We used three standard datasets [6] of various dimensionalities: *Mammo* data (65 dim., 2 classes: 57 benign and 29 malignant samples), *WDBC* data (30 dim., 2 classes: 357 benign and 212 malignant samples) and *Wine* data (13 dim., 3 classes: 59, 71 and 48 wine grape samples). Both the I-CV and O-CV loops run 5-fold with the higher-dimensional *Mammo* data and 10-fold with the *WDBC* and *Wine* data.

**Table 1.** *Mammo* **data experiments** *(5-f.CV)*

| Crit. | Meth. | I-CV | O-CV | Size | Time(m) |
|---|---|---|---|---|---|
| Gauss | SFS | 0.799 | 0.607 | 12.2 | 02:31 |
| | SFFS | 0.848 | 0.570 | 12 | 12:30 |
| | OS | 0.815 | 0.605 | 7.8 | 24:18 |
| | DOS | 0.851 | 0.585 | 7.8 | 47:57 |
| | full | | 0.663 | 65 | |
| 5-NN | SFS | 0.883 | 0.746 | 16.4 | 00:09 |
| data scaled | SFFS | 0.930 | 0.838 | 6 | 00:59 |
| | OS | 0.921 | 0.803 | 5.8 | 01:31 |
| | DOS | 0.936 | 0.827 | 7.2 | 03:53 |
| | full | | 0.610 | 65 | |
| SVM | SFS | 0.924 | 0.838 | 25.4 | 00:26 |
| C=4, $\gamma$=0.0078 | SFFS | 0.950 | 0.872 | 9.6 | 01:36 |
| | OS | 0.921 | 0.757 | 22.6 | 05:01 |
| | DOS | 0.953 | 0.860 | 8.6 | 12:57 |
| | full | | 0.816 | 65 | |

**Table 2.** *Wine* **data experiments** *(10-f.CV)*

| Crit. | Meth. | I-CV | O-CV | Size | Time(m) |
|---|---|---|---|---|---|
| Gauss | SFS | 0.598 | 0.513 | 3.1 | 00:00 |
| | SFFS | 0.634 | 0.607 | 3.9 | 00:03 |
| | OS | 0.640 | 0.624 | 3.5 | 00:05 |
| | DOS | 0.647 | 0.657 | 3.8 | 00:17 |
| | full | | 0.431 | 13 | |
| 5-NN | SFS | 0.986 | 0.959 | 7.3 | 00:01 |
| data scaled | SFFS | 0.987 | 0.971 | 7 | 00:04 |
| | OS | 0.984 | 0.959 | 6.8 | 00:11 |
| | DOS | 0.988 | 0.971 | 6.8 | 00:28 |
| | full | | 0.949 | 13 | |
| SVM | SFS | 0.981 | 0.966 | 7.8 | 00:15 |
| C=0.86, $\gamma$=0.43 | SFFS | 0.985 | 0.966 | 8.3 | 00:50 |
| | OS | 0.988 | 0.956 | 8.4 | 02:03 |
| | DOS | 0.988 | 0.966 | 8.7 | 02:18 |
| | full | | 0.983 | 13 | |

### 3.2. Results

The results of our experiments are collected in Tables 1 to 3. Each table contains three sections gathering results for one type of classifier (criterion func-

**Table 3.** *WDBC* **data experiments** *(10-f.CV)*

| Crit. | Meth. | I-CV | O-CV | Size | Time(h) |
|---|---|---|---|---|---|
| Gauss | SFS | 0.962 | 0.933 | 10.8 | 00:00 |
| | SFFS | 0.972 | 0.942 | 10.6 | 00:03 |
| | OS | 0.970 | 0.940 | 9.9 | 00:06 |
| | DOS | 0.973 | 0.951 | 10.7 | 00:06 |
| | full | | 0.945 | 30 | |
| 5-NN | SFS | 0.978 | 0.967 | 12.9 | 00:01 |
| data scaled | SFFS | 0.982 | 0.968 | 16.4 | 00:09 |
| | OS | 0.981 | 0.970 | 15.9 | 00:22 |
| | DOS | 0.983 | 0.958 | 13.6 | 00:36 |
| | full | | 0.968 | 30 | |
| SVM | SFS | 0.979 | 0.970 | 18.5 | 00:05 |
| C=16, $\gamma$=0.031 | SFFS | 0.982 | 0.968 | 16.2 | 00:23 |
| | OS | 0.981 | 0.974 | 16.7 | 00:58 |
| | DOS | 0.983 | 0.968 | 12.8 | 01:38 |
| | full | | 0.972 | 30 | |

tion). The main information of interest is in the column I-CV, showing the maximum criterion value (classification accuracy) yielded by each FS method in the *inner* CV loop, and O-CV, showing the respective classification accuracy on independent test data.

For better overview we have created a summary in form of graphs in Figure 3. The graphs show the results of tested FS methods averaged over each tested classifier-dataset combination. The left graph shows in light gray the methods' optimization performance, or the ”dependent” achieved classification accuracy (corresponds to I-CV column in tables), in black the respective accuracy on independent test data (O-CV in tables). The right graph shows the average yielded subset size.

The following properties of the *Dynamic Oscillating Search* can be observed: (i) it constantly outperforms other tested methods in the sense of criterion maximization ability (I-CV), (ii) it tends to produce the smallest feature subsets, (iii) its impact on classifier performance on unknown data varies depending on data and classifier used – in some cases it yields the best results.

## 4 Conclusion

We have introduced the new Dynamic Oscillating Search FS method suitable for Wrapper search setting. It has been shown to bring constant improvement in optimization performance over previous sequential methods. The negative effect of feature over-selection has been investigated experimentally. Despite its high optimization performance the new DOS has been shown capable of yielding the best classification accuracy on independent test data in several experiments.

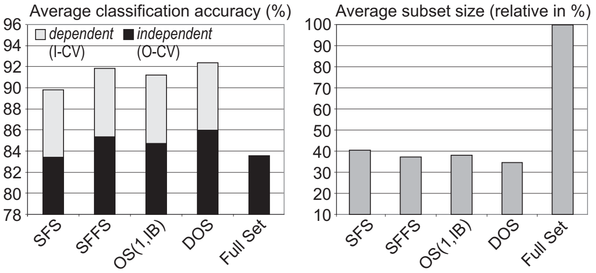

**Figure 3. Experimental results summarized.**

The new method has a built-in mechanism to prefer smaller subset sizes throughout the course of search. The DOS has been experimentally shown to yield smaller subsets than other comparable methods without degrading pattern recognition system performance.

**Acknowledgements**: The work has been supported by grants of the Czech Ministry of Education 2C06019 ZIMOLEZ and 1M0572 DAR, G.A. of the Academy of Sciences of the CR Nos. 1ET400750407 and AV0Z10750506 and the GACR No. 402/03/1310.

## References

[1] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for SVM*, 2001. `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[2] P. A. Devijver and J. Kittler. *Pattern Recognition: A Statistical Approach*. Prentice Hall, Englewood Cliffs, London, UK, 1982.

[3] F. Hussein, R. Ward, and N. Kharma. Genetic algorithms for feature selection and weighting, a review and study. *ICDAR*, 00:1240, 2001.

[4] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artif. Intell.*, 97(1-2):273–324, 1997.

[5] H. Liu and L. Yu. Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, 17(4):491–502, 2005.

[6] D. Newman, S. Hettich, C. Blake, and C. Merz. UCI repository of machine learning databases, 1998.

[7] P. Pudil, J. Novovičová, and J. Kittler. Floating search methods in feature selection. *Pattern Recogn. Lett.*, 15(11):1119–1125, 1994.

[8] Š. J. Raudys. Feature over-selection. In *Structural, Syntactic, and Statistical Pattern Recognition*, volume LNCS 4109, pages 622–631, Berlin / Heidelberg, Germany, 2006. Springer-Verlag.

[9] A. Salappa, M. Doumpos, and C. Zopounidis. Feature selection algorithms in classification problems: An experimental evaluation. *Optimization Methods and Software*, 22(1):199–212, 2007.

[10] P. Somol and P. Pudil. Oscillating search algorithms for feature selection. *ICPR*, 02:406–409, 2000.