Remainder Subset Awareness for Feature Subset Selection

Gabriel Prat-Masramon and Lluís A. Belanche-Muñoz

Abstract Feature subset selection has become more and more a common topic of research. This popularity is partly due to the growth in the number of features and application domains. It is of the greatest importance to take the most of every evaluation of the inducer, which is normally the more costly part. In this paper, a technique is proposed that takes into account the inducer evaluation both in the current subset and in the remainder subset (its complementary set) and is applicable to any sequential subset selection algorithm at a reasonable overhead in cost. Its feasibility is demonstrated on a series of benchmark data sets.

1 Introduction

In the last few years *feature selection* has become a more and more common topic of research, a fact probably due to the growth of the number of features involved. These problems are very common in medicine and biology; e.g. molecule classification, gene selection or medical diagnostics.

This work addresses the problem of *selecting a subset of features* from a given set by introducing a general-purpose modification for feature subset selection algorithms which iteratively select and discard features. The idea is to use the evaluation of the inducer in the so-called *remainder set* (the set complementary to the subset of selected features) as an additional source of information.

Gabriel Prat-Masramon
Faculty of Computer Science
Polytechnical University of Catalonia
Barcelona, Spain e-mail: grat@lsi.upc.edu

Lluís A. Belanche-Muñoz
Faculty of Computer Science
Polytechnical University of Catalonia
Barcelona, Spain e-mail: belanche@lsi.upc.edu

2 The Remainder Set of Features

It is common to see feature subset selection in a set Y of size n as an *optimization problem* where the search space is $\mathscr{P}(Y)$ [5]. In this setting, the *feature selection problem* is to find an optimal subset $X^* \in \mathscr{P}(Y)$ which maximizes a given evaluation criterion $J : \mathscr{P}(Y) \to [0,1]$ (filter or wrapper). We will refer to $J(X)$ as the *usefulness* of feature subset X [1].

When the goal is to find an optimal subset X^*, it seems plausible to choose an X_k in a stepwise and greedy way (1). That is what the well known sequential forward generation (SFG) and sequential backward generation (SBG) algorithms do [8, 6].

$$X_k = \arg\max_{X \in \mathbf{S}_k} J(X), \qquad k = 1, \ldots, n \tag{1}$$

In real problems features are far from independent, thus not always the best subset in every iteration has to point to the best overall solution. Quite possibly there is some combination of features that *would* lead to a final better solution if chosen now. By considering the current set of features X_k another set is implicitly created, the set of *remaining* features or *remainder set* $Y_k = Y \setminus X_k$. This set can also give information about the new feature to be added or removed at every step. We claim that, in many cases, a way to improve the detection of *feature interactions* is to assess how the addition/removal of a feature to/from X_k (a removal/addition, from the point of view of Y_k) affects the *usefulness* of Y_k.

2.1 Theoretical analysis and examples

The intuitive explanation for using the remainder set is that the optimal set X^* that the algorithm is trying to find could be either in X_k, in Y_k or split among the two. J should give higher values to a set containing X^* and its value should diminish when removing a feature from X^*. Then one should add the best feature to X_k, and whose removal is worse for Y_k, i.e. to maximize $J(X_k)$ and minimize $J(Y_k)$. The general idea is called *Remainder Subset Awareness* (RSA) for obvious reasons. This RSA idea tries to alleviate some of the weaknesses of SFG and SBG:

1. SFG (specially at its first steps) evaluates the features on their own, not taking into account the relationships between them [3]; thus two features that are very good when used together but that are not that good individually may not be selected. Note these two features would *both* belong to the remainder set, that should be then affected by the removal of either.
2. SBG (specially at its first steps) evaluates each feature with all the irrelevant and redundant features that there may be in Y; this may discard a relevant feature early on, due to the disturbing effects of the irrelevant ones over J.

[1] It is very convenient to include resampling in the evaluation criterion; e.g. J could be a cross-validated value.

More formally, by definition of $X^* \equiv \arg\max_{X \in \mathscr{P}(Y)} J(X)$ we have that $J(X^*) > J(X^* \setminus \{x\}), \forall x \in X^*$. But on the other hand, it is not true that $J(X_k \setminus \{z\}) > J(X_k \setminus \{x\}), \forall X_k \supset X^*, \forall x \in X^*, \forall z \notin X^*$. This inequality states that removing a feature in X^* from any set X_k that contains X^* is always more harmful than removing a feature not in X^* from this same set. If this was always true, then SBG would always find X^*, as it would remove one feature not in X^* at each step, until X^* was found. It would be always true only if the J criterion was not affected by the addition of irrelevant of redundant features. It will certainly not be true if the features in $X_k \setminus X^*$ affect the results of J. As stated in the introduction, irrelevant or redundant features often lead classifiers to find false regularities (specially in small sample situations) instead of learning from the features that really determine the target.

Two artificial problems have been chosen to illustrate the potential benefits of the RSA idea. The choice has been made due to their special characteristics that make either SFG or SBG fail to find the best solution. As the structure and best solution to these problems is known the benefits can be clearly explained.

- The CORRAL data set has two classes and six boolean features (A_0; A_1; B_0; B_1; I; C). Feature I is irrelevant, feature C is correlated to the class label 75% of the time, but the other four features can be combined to fully predict the class value. SFG will choose C first as it is the best feature when taken all alone [4]. The hypothesis is that the *usefulness* of the remainder set would be so high if C was chosen that SFG enhanced with the RSA idea would not choose it.
- The ANTICORRAL data set has been generated *ad hoc* for this paper. It is a three class problem with 11 continuous features $(I_1, I_2, \ldots, I_9, C_1, C_2)$. Features I_1 to I_9 follow a normal distribution with mean equal to the class of the example and a standard deviation of 1. Feature C_1 is generated as $C_{1i} = \mathscr{N}(\mu = Y_i, \sigma^2 = 0.5)$, while feature C_2 is generated by the formula: $C_{2i} = C_{1i} - Y_i + \mathscr{N}(\mu = 1, \sigma^2 = 0.2)$. The class can be predicted using C_1 and C_2. The hypothesis is that SBG will readily discard C_2 in the *firsts* steps, due to the influence of the I_j features; while RSA will detect the harm to Y_0 when discarding C_2 and find the best solution.

The two hypothesis were confirmed by the results of the experiments run using the algorithms and the experimental setup explained in the following sections. Table 1 shows mean *error rates* and the *p*-value of the Wilcoxon-Mann-Whitney (WMW) test, indicating that the difference is statistically significant at the 95% level. The table also shows the median number of selected features and its absolute deviation.

Table 1 Results for SFG on CORRAL and SBG on ANTICORRAL. SFG$^+$ and SBG$^+$ are SFG, SBG enhanced with the RSA idea; F, F^+ stand for the final number of features.

Problem	SFG/SBG	SFG$^+$/SBG$^+$	p-value	F	F^+
CORRAL	0.077	0.009	**0.002**	5.0±0.0	4.0±0.0
ANTICORRAL	0.132	0.023	**0.007**	7.0±2.2	2.0±0.0

2.2 Combination function

With the above formulation we have a multi-objective problem, since not always the subset with maximum $J(X_k)$ will be the same as the subset with minimum $J(Y_k)$. Therefore, a trade-off has to be found that partly optimizes both. A reasonable alternative is to choose the subset which maximizes some predefined function f of the two criteria among the two candidate subsets:

$$\underset{X \in \mathbf{S}_k}{\arg\max}\ f\,[J(X), J(Y \setminus X)], \qquad k = 1, \ldots, n \tag{2}$$

The function $f : (0,1)^2 \to (0,1)$ is chosen to be *continuous* in both arguments, *increasing* in the first and *decreasing* in the second. It also should allow control on the relative importance of the two arguments (thus it is non-symmetric). A sequential algorithm can then be modified by *replacing* the function to maximize at each step from the one in (1) to the one in (2).

Various functions that satisfied the previous conditions have been tested. We choosed the best function based on our experiments which was one of the simplest:

$$f(x,y) = \frac{x \cdot w_x - y \cdot w_y + 1}{2},\ w_x, w_y \in (0,1). \tag{3}$$

These weights have to be selected taking into account the weaknesses of SFG and SBG presented above. We take the weights to be proportional to the *usefulness* of the set we are about to modify: $w_x = J(X_{k-1})$ and $w_y = J(Y \setminus X_{k-1})$. Thus giving more importance to more useful sets: when X_k is better than Y_k, the features that make it even better are preferred; when Y_k is better than X_k (e.g. at the first steps of SFG) those that harm Y_k the most are preferred over others that helped X_k more.

3 Experimental work

Experimental work is now presented in order to assess the described modifications using SFG and SBG and their RSA counterparts SFG$^+$ and SBG$^+$. The algorithms were implemented using the R language [7]. We used well-known datasets from the UCI repository [1], as well as *microarray* gene expression problems, with scarce data and high dimensionality, all of them listed in Table 2. Each experiment consists of an *outer* loop of 5x2-fold feature selection [2]. It keeps half of the examples out of the feature selection process and uses them as a test set to evaluate the quality of the selected features. For every step of the outer loop, two feature selection processes are conducted with the same examples, one with the original algorithm and one with the RSA version. The selected objective function is the 1-nearest neighbor (1NN) learner, since arguably 1NN is one of the inducers that suffers the most in presence of redundant or irrelevant features. However, the modifications do not depend on this choice and others could be possible. This evaluation is resampled in another (*inner*)

Table 2 Used datasets. Left: UCI data sets descriptions. Right: Microarray data sets descriptions

Problem	features	classes	examples
IONOSPHERE	34	2	351
IRIS	4	3	50
MAMMOGRAM	65	2	86
MUSK	166	2	476
SONAR	60	2	208
SPECT	22	2	267
SPECTF	44	2	267
WAVEFORM	21	3	5,000
WDBC	10	2	699

Problem	features	classes	examples
BREAST CANCER	24,481	2	97
COLON TUMOR	2,000	2	62
GCM	16,063	14	190
LEUKEMIA	7,129	2	72
LUNG CANCER	12,533	2	181
PROSTATE CANCER	12,600	2	136

5x2-fold cross-validation loop for a more informed estimation of subset usefulness. Forward methods run until all the features are selected and backward ones until all have been removed. Then the *best* of the obtained sequence of subsets is returned. This subset is evaluated in the corresponding test set using the same 1NN inducer. Finally a WMW test is conducted on the sets of classification errors from each algorithm to determine whether the difference is statistically significant.

The results are displayed in Table 3. The table also shows the median of the size of the final selected subsets and its absolute deviation. Few results are signaled as

Table 3 Detailed results. Figures in boldface correspond to statistically significant improvements.

Problem	SFG	SFG$^+$	F	F^+	SBG	SBG$^+$	F	F^+
IONOSPHERE	0.133	0.122	5±3.0	5.5±1.5	0.144	0.128	10.50±5.9	6.5±1.5
IRIS	0.075	0.072	2±1.5	2±1.5	0.080	0.070	2±1.5	1±0.0
MAMMOGRAM	0.291	0.286	9±5.9	11.5±3.7	0.302	0.265	14±11.9	11.50±3.0
MUSK	0.133	0.140	50±11.1	47.5±7.4	0.161	0.157	31.5±17.0	49.5±18.5
SONAR	0.214	**0.183**	21.5±10.4	23±8.15	0.190	0.180	17±4.5	27±4.5
SPECT	0.227	0.211	11±3.0	8±3.0	0.240	0.230	4.5±2.2	7±4.5
SPECTF	0.263	0.255	10.5±5.2	7.5±5.2	0.277	0.270	16±5.9	11±7.4
WAVEFORM	0.223	**0.216**	15±3.0	16.5±3.0	0.224	**0.215**	16±1.5	17.00±3.0
WDBC	0.086	0.085	19±5.9	18±4.5	0.083	0.085	17.5±8.2	15±8.9
BREAST CANCER	0.289	0.286	29±15.6	49±22.2	0.328	0.315	23±16.3	21±17.0
COLON TUMOR	0.258	0.252	20±14.8	11.5±11.9	0.219	0.216	30±34.8	25±30.4
GCM	0.501	0.491	45±22.2	44±27.4	0.561	**0.503**	34±25.2	60.5±29.7
LEUKEMIA	0.094	0.092	2.5±0.7	2±2.2	0.092	0.089	8.5±3.7	3±1.48
LUNG CANCER	0.031	0.028	2±0.0	7.5±3.0	0.045	0.032	9±5.2	3.5±2.2
PROSTATE CANCER	0.103	0.134	12±5.19	14.5±5.2	0.132	0.157	29±20.8	19.5±15.6

statistically significant according to the WMW test at the 95% level (p-value lower than 0.05). Two of them when comparing SFG to SFG$^+$ and another two when comparing SBG to SBG$^+$. In all these cases the statistically significant differences signal the RSA enhancement as better than the original SFG or SBG algorithms. The RSA versions of the algorithms outperformed the conventional versions in the 78.5% of the experiments. It is seen that for SFG performance is in general increased while

keeping the number of selected features roughly equal. Only a 28.5% of the results had more features using the modified versions. Thus, whenever the algorithms are in ties or very close to, the modified versions offer a solution with lower number of features, which is attractive from the point of view of feature selection.

4 Conclusions

Our results indicate a general *improvement* in performance while keeping the size of the final subset roughly equal or lower. The fact that the modified version does not always improve on the results should not be a surprise. According to the *No free lunch* theorems, if an algorithm achieves superior results on some problems, it must pay with inferiority on other problems. However, it is possible to modify a search algorithm to obtain a version that is generally superior in performance to the original version [9]. In the present situation this fact can be explained by the way the modified version selects subsets of features. Consider two features: one that makes a significant reduction in performance at the remainder set and not a big change in the performance of the current set; and one that increases the performance of the selected set a bit more than the first feature but does not make a big change on the remainder set. A conventional algorithm would always select the latter while the modified version would likely select the former. That could lead the modified version to avoid local extrema and ultimately end in a better subset; however, when a set close to the optimal subset has been selected, the modification may cause the algorithm to loose precision in choosing the last features. For future work we plan to fine-tune the proposed combination function in order to avoid this weakness.

References

1. Blake, Merz, C.J.: UCI repository of machine learning databases (1998)
2. Dietterich, T.G.: Approximate statistical tests for comparing supervised classification learning algorithms. Neural Computation **10**, 1895–1923 (1998)
3. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. J. Mach. Learn. Res. **3**, 1157–1182 (2003)
4. John, G.H., Kohavi, R., Pfleger, K.: Irrelevant features and the subset selection problem. In: Proc. of the 11th ICML, pp. 121–129. Morgan Kaufmann, New Brunswick, NJ, USA (1994)
5. Langley, P.: Selection of relevant features in machine learning. In: Proceedings of the AAAI Fall Symposium on Relevance, pp. 140–144. AAAI Press, New Orleans, LA, USA (1994)
6. Pudil, P., Novovicová, J., Kittler, J.: Floating search methods in feature selection. Pattern Recognition Letters **15**(11), 1119–1125 (1994)
7. R Development Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2008)
8. Whitney, A.W.: A direct method of nonparametric measurement selection. IEEE Trans. Comput. **20**(9), 1100–1103 (1971)
9. Wolpert, D., Macready, W.G.: No free lunch theorems for optimization. IEEE Trans. Evolutionary Computation **1**(1), 67–82 (1997)