Remainder Subset Awareness for Feature Subset Selection

Gabriel Prat-Masramon
Faculty of Computer Science
Polytechnical University of Catalonia
Barcelona, Spain
gprat@lsi.upc.edu

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

2007-05-29

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. The family of algorithms known as *plus-l-minus-r* and its immediate derivatives (like *forward selection*) are very popular and often the only viable alternative when used in *wrapper* mode. In consequence, 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 introduction of new application domains and the growth of the number of features involved. An example of these new domains is web page categorization, a domain currently of much interest for internet search engines where thousands of terms can be found in a document. Another example is found in appearance-based image classification methods which may use every pixel in the image. Classification problems with many features are also very common in medicine and biology; e.g. molecule classification, gene selection or medical diagnostics.

Feature selection can help in solving a classification problem with irrelevant and/or redundant features for many reasons. First it can make the task of data visualization and understanding easier by eliminating irrelevant features which can mislead the interpretation of the data. It can also reduce costs since the measurement or recording of some of the features can be avoided; this is especially important in domains where some features are very expensive to obtain, e.g., a costly or invasive medical test. In addition, a big benefit of feature selection is in defying the curse of dimensionality to help the induction of good classifiers from the data. When many *unuseful*, i.e. irrelevant or redundant, features are present in the training data, classifiers are prone to finding false regularities in the input features and learn from that instead of learning from the features that really determine the instance class (this is also valid when predicting the instance target value in the case of regression).

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. An important family of algorithms for feature subset selection perform an explicit search in the space of subsets by iteratively adding and/or removing features one at a time until some stop condition is met. The idea is then to

use the evaluation of the inducer in the so-called *remainder set* (the set complementary to the current subset of selected features) as an additional source of information. This information is used in conjuntion to conventional algorithms, that only use the evaluation on the current subset of selected features. In our experimental results, such simple modification achieves significant improvements in the maximization of the objective function for classification tasks. The modified algorithms obtain similar numbers of selected features in general, or a further reduction if purely *forward* algorithms are used.

The rest of the paper is organized as follows: first we briefly review the feature subset selection problem (Section 2). Section 3 introduces the concept of the remainder set of features and suggests a modification of the previously presented algorithms. In the next section the experimental design is defined and the results are exposed and discussed. Finally, section 5 extracts conclusions about these results and introduces some future work.

2 Feature Subset Selection

There are two main approaches to feature selection: *filter* methods and *wrapper* methods. These two families of methods only differ in the way they evaluate the candidate sets of features. While the former uses a problem independent criterion, the latter uses the performance of the final classifier to evaluate the quality of a feature subset. The basic idea of the filter methods is to select the features according to some prior knowledge of the data. For example, selection of features based on the *conditional probability* that an instance is a member of a certain class given the value of its features [1]. Another criterion commonly used by filter methods is the *correlation* of a feature with the class, i.e. selecting features with high correlation [3]. In contrast, wrapper methods suggest a set of features that is then supplied to a classifier, which uses it to classify the training data and returns the classification accuracy or some other measure thereof [5].

It is common to see feature subset selection in a set Y of size n as an *optimization problem* where the search space is $\mathcal{P}(Y)$ [6]. In this setting, the *feature selection problem* is to find an optimal subset $X^* \in \mathcal{P}(Y)$ which maximizes a given criterion $J : \mathcal{P}(Y) \rightarrow [0, 1]$ as seen in eq. (1). Having $J(X)$ as the evaluation criterion is a common characteristic of many feature selection algorithms. The criterion J may be problem-independent or may be the classifier that will be used to solve a classification problem, and thus this setting is valid either for filter or wrapper algorithms. In any case, we will refer to $J(X)$ as the *usefulness* of feature subset X.

$$X^* = \arg\max_{X \in \mathcal{P}(Y)} J(X) \tag{1}$$

In the literature, several suboptimal algorithms have been proposed for doing this. Among them, a wide family is formed by those algorithms which, departing from an initial solution, iteratively add or delete features by locally optimizing the objective function. The search starts with an arbitrary set of features (e.g. the full set or the empty set) and moves iteratively to neighbor solutions by adding or removing features. These *sequential* algorithms with no backtracking can be cast in the general class of *hill-climbing algorithms*. They leave a sequence of visited states $X_{k_1}, \ldots X_{k_m}$, where the size of every X_{k_i} is k_i and m is the number of visited states. The difficulty of the feature selection problem can be illustrated by the following facts:

1. In general, it is not the case that $J(X_{k_{i+1}}) \geq J(X_{k_i})$ (not even for the simplest algorithms like SFG or SBG).
2. There may exist "ugly" feature subsets along the way X_{k_i} such that $J(X_{k_i}) < J(\emptyset)$, for some $i \geq 0$.
3. Take X_{k_i} and $X_{k_{i+1}}$ such that $X_{k_i} \subset X_{k_{i+1}}$ and $k_{i+1} = k_i + 1$. In general, it is not the case that $J(X_{k_{i+1}}) \geq J(X_{k_i})$.
4. Calling X^*_{k+1} a current optimal solution with $k+1$ features, it is not the case that $J(X^*_{k+1})$ contains $J(X^*_k)$. This is known as the *nesting* problem.

Expressed more succinctly, any partial order with respect to J in the set $\mathcal{P}(X)$ is conceivable. Among the proposed algorithms for attacking this problem are the sequential forward generation (SFG) and sequential backward generation (SBG), the *plus l - take away r* or $PTA(l,r)$ proposed by Stearns [10] or the floating search methods [9]. They both introduce methods for the generation of the sets of features by combining steps of SFG with steps of SBG but keep using a certain $J(X)$ as evaluation criterion.

$X_0 \leftarrow \emptyset$ //Initial subset
$i \leftarrow 0$
repeat
  //Subset generation
  $\vec{S}_{i+1} \leftarrow \{X \mid X = X_i \cup \{x\} \wedge x \in Y \setminus X_i\}$
  //Subset evaluation
  $X_{i+1} \leftarrow \arg\max_{X \in \vec{S}_{i+1}} J(X)$
  $i \leftarrow i + 1$
//Stopping criterion
until $J(X_i) \leq J(X_{i-1}) \vee i = n$
return X_{i-1} //Selected subset

Algorithm 1: SFG

$X_0 \leftarrow Y$ //Initial subset
$i \leftarrow 0$
repeat
  //Subset generation
  $\vec{S}_{i+1} \leftarrow \{X \mid X = X_i \setminus \{x\} \wedge x \in X_i\}$
  //Subset evaluation
  $X_{i+1} \leftarrow \arg\max_{X \in \vec{S}_{i+1}} J(X)$
  $i \leftarrow i + 1$
//Stopping criterion
until $J(X_i) \leq J(X_{i-1}) \vee i = 0$
return X_{i-1} //Selected subset

Algorithm 2: SBG

Algorithm 1 and **Algorithm 2** below describe two of the classic feature selection algorithms using this point of view: sequential forward generation (SFG) and sequential backward generation (SBG). In these algorithms X_0 is the starting set of features of the algorithm, $\vec{S}_k$ the set of sets of features generated during the subset generation phase and X_k the selected set of features at iteration k. It can be seen that the subset evaluation phase in the two algorithms is exactly the same while the initialization and the subset generation phases change. Note that at all times the size of X_k is k and thus $X_n = Y$ and $X_0 = \emptyset$.

3 The Remainder Set of Features

As the goal of feature selection is to find an optimal subset X^* as seen in (1), it seems plausible to choose an X_k for each iteration as in (2) in a stepwise and greedy way, which is exactly what the previously described feature selection algorithms do:

$$X_k = \arg\max_{X \in \vec{S}_k} J(X), \qquad k = 1, \ldots, n \tag{2}$$

In real problems, features are far from independent, thus not always the best feature set in every iteration has to be the best option. Quite possibly there is some combination of features that would be a better choice that the feature which maximizes $J(X)$ in this iteration. In this vain, the forward steps in the previous algorithms are not taking into account some information they could use. Only the *usefulness* of every generated subset of features is measured, as in (2). However, 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 variable to be added or removed at every step. It is our conjecture that a way to enhance the detection of feature interactions is to see how the addition of a feature to X_k (a removal, from the point of view of Y_k) affects the *usefulness* of the remainder set. The idea is to add that feature most useful to X_k and whose removal is most harmful to Y_k. An analogous reasoning can be made in backward steps by interchanging the roles of the current and remainder subsets. The general idea is called *Remainder Subset Awareness* for obvious reasons.

With this formulation we have a multiobjective problem, since not always the sub-

set with maximum $J(X_k)$ will coincide with the subset with minimum $J(Y_k)$, so it will not be possible to satisfy both objectives with the same single solution. In this case, either the two solutions have to be explored or a trade-off has to be found that partly optimizes both objectives. If both solutions are chosen for further exploration, then the search space is highly increased over the original version of the algorithm, and the complexity of the algorithm grows from polynomial to exponential, which is unfeasible. A reasonable alternative is to choose the subset which maximizes some predefined function f of the two criteria among the two candidate subsets, as expressed by:

$$\arg\max_{X \in \vec{S}_k} f\left[J(X), J(Y \setminus X)\right], \qquad k = 1, \ldots, n \tag{3}$$

The function $f : (0,1)^2 \rightarrow (0,1)$ has to be chosen to be continuous in both arguments, increasing in the first and decreasing in the second and to permit control on the relative importance of the two arguments (thus it is non-symmetrical). Following this alternative, an algorithm of the sequential kind can be modified by replacing the evaluation function $J(X)$ with the one in Eq. 3. As an example, the following **Algorithm 3** shows the straightforward *Remainder Subset Aware* version of the original SFG presented in **Algorithm 1**. Other forward/backward algorithms would be modified analogously.

$X_0 \leftarrow \emptyset$ //Initial subset
$i \leftarrow 0$
repeat
  //Subset generation
  $\vec{S}_{i+1} \leftarrow \{X \mid X = X_i \cup \{x\} \wedge x \in Y \setminus X_i\}$
  //Subset evaluation
  $X_{i+1} \leftarrow \arg\max_{X \in \vec{S}_{i+1}} f\left[J(X), J(Y \setminus X)\right]$
  $i \leftarrow i + 1$
//Stopping criterion
until $J(X_i) \leq J(X_{i-1}) \vee i = n$
return X_{i-1} //Selected subset

Algorithm 3: Remainder set aware SFG

The chosen evaluation function f, which combines the *usefulness* of the selected subset of features with that of the remaining subset is shown in Eq. 4.

$$f(x, y) = x^k \times (1 - y)^{1-k}, \; k, x, y \in [0, 1] \tag{4}$$

Note that $k = 1$ recovers the conventional algorithms and $k = 0.5$ corresponds to the geometrical mean between x and $1 - y$. In general, lower values of k give more weight to the evaluation of the inducer in the remainder set.

4 Experimental work

The previous idea is first illustrated using the CorrAl problem, a small dataset with some specific characteristics that make it useful to test feature subset selection algorithms in a known environment [4]. This dataset 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, and the other four features are relevant to the boolean target concept: $(A_0 \wedge A_1) \vee (B_0 \wedge B_1)$. 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 the modified version of SFG would not choose it. After running the experiments with CorrAl, the hypothesis was confirmed: a conventional SFG chose the features in the order $\{C, I, A_0, A_1, B_0, B_1\}$, whereas the modified remainder set aware version chose the order $\{A_0, A_1, B_0, B_1, C, I\}$.

4.1 Experimental settings

Experimental work is now presented in order to assess the described modification with a group of four sequential algorithms, using some well known datasets from the UCI repository of machine learning databases [2]. The family $PTA(l, r)$ has been selected to carry out the experiments, comparing their original versions and the modified ones, which are aware of the remainder set of non-selected features. Four different combinations of values for l and r have been tested as seen on Table 1. Note SFG can be seen as a particular case of $PTA(l, r)$ with $l = 1$ and $r = 0$ and referred

Table 1: Tested values for the $PTA(l, r)$ parameters (forward and backward steps)

Algorithm	Fwd st.	Bwd st.
$PTA(0,1) \equiv SBG$	0	1
$PTA(1,0) \equiv SFG$	1	0
$PTA(1,2)$	1	2
$PTA(2,1)$	2	1

to as $PTA(1, 0)$. The same can be done for SBG calling it $PTA(0, 1)$. The value of k in eq. (4) was set to 0.8 after some preliminary experiments and should be taken only as an educated guess.
Each pair of algorithms has been tested with the following datasets:

Ionosphere Classification of radar returns from the ionosphere. There are 2 classes, 351 instances, 34 numeric features. The targets were free electrons in the ionosphere. "Good" radar returns are those showing evidence of some type of structure in the ionosphere. "Bad" returns are those that do not: their signals pass through the ionosphere.

Mammogram Mammography data donated by the Pattern Recognition and Image Modeling Laboratory at University of California, Irvine. There are 86 cases with 65 features each and a binary class indicating benign or malignant.

Spect The dataset describes diagnosing of cardiac Single Proton Emission Computed Tomography (SPECT) images. Each of the patients is classified into two categories: normal and abnormal. There are 22 binary features extracted from the original SPECT images and 267 instances.

Spectf The same data as the previous dataset but this time a continuous feature pattern of size 44 was created for each patient. The same binary class and the same 267 instances.

Sonar There are 208 patterns obtained by bouncing sonar signals off a metal cylinder and rocks at various angles and under various conditions. Each pattern is a set of 60 numbers in the range 0.0 to 1.0. Each number represents the energy within a particular frequency band, integrated over a certain period of time. The class is binary indicating whether the object was a rock or a metal cylinder.

Waveform Artificial dataset where each class is generated from a combination of 2 of 3 "base" waves. There are 5000 instances with 21 features each, all of which include noise, and 3 classes.

Wdbc Breast cancer databases obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg [7]. Features 2 through 10 have been used to represent instances. There are 699 instances with 10 features, each has one of 2 possible classes: benign or malignant.

The experiments were carried out by extending the YALE learning environment [8] in order to implement a conventional PTA and the modified remainder set aware version of it. Each experiment consisted of a feature selection chain with a 1-nearest neighbor learner (using Euclidean distance) and 5-fold cross-validation for estimating feature *usefulness*. The quantity reported is the mean classification error in the five test folds. It is important to mention that there was no stopping criterion in the experiments: forward methods run until all the features were selected and backward ones until all of them were removed. Then the best of the obtained sequence of subsets was returned. The results are displayed in Table 4. The table also shows the standard deviation for these values found in the cross-validation runs and the size of the final selected subsets.

Table 2: Summary results of the experiments. The results are from the point of view of the modified algorithms.

	# Feat.			
Error	=	>	<	Total
better	3%	28%	25%	56%
equal	19%	3%		22%
worse		8%	14%	22%
Total	22%	39%	39%	100%

Table 3: Summary results of the experiments for forward algorithms. The results are from the point of view of the modified algorithms.

	# Feat.			
Error	=	>	<	Total
better		44%	17%	61%
equal	17%	6%		22%
worse		6%	11%	17%
Total	17%	56%	28%	100%

4.2 Experimental results

Tables 2 and 3 show the summary results of the experiments for all the 8 data sets ans 4 algorithms, and for the forward versions of the algorithms only, respectively. The tables cross the number of features selected with the classification error.

Upon looking at the summary tables, the first fact to note from the experiment results is that the remainder aware version of the algorithms outperformed the conventional version in most of the cases. It is seen that performance is in general increased (as expressed by the chosen J) while keeping the number of selected features roughly equal (Table 2). The improvement is even greater if we only look at the *forward* versions of the algorithms (Table 3). In this particular case, performance is increased while lowering the number of selected features. This is mainly due to the fact that the forward methods can easyly make wrong decisions at early iterations as (almost) no feature interaction is taken into account when evaluating individual features. A clear example of this has been exposed at the beginning of this section with the CorrAl dataset. There SFG selected the correlated feature while SBG correctly discarded it as the interaction with the other more relevant features was taken into account. Whenever the conventional and the modified algorithm are in ties or very close to, the modified versions offer a solution with a lower number of features, which is also interesting from the point of view of feature selection (there is an exception to this rule for the particular case of the Sonar data set and PTA (2,1)). The detailed experiment results are displayed in Table 4.

5 Conclusions

This paper has presented a modification for feature subset selection algorithms that iteratively evaluate subsets of features, by making them compute not only the *usefulness* of the selected set but also the *usefulness* of the remainder set. A set of experiments have been conducted in order to compare the modified versions of the algorithms with their original versions. Our experimental 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 the results of the original 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 [11]. In the present situation this fact can be explained by the way the modified version selects subsets of features. For instance, given two features: One that makes a significant reduction of the performance of the remainder set and not a big change on the performance of the selected set. And one that increases the performance of the selected set a bit more than the first one but does not make a big change on the remainder one. A conventional algorithm would always select the latter while the mod-

Table 4: Detailed Experiment Results. The error shown is the average of test set error in the 5 folds, while σ is the standard deviation of these values. Figures in boldface correspond to improvements.

Dataset	**Steps**		**Conventional Algorithm**			**Modified Algorithm**		
	fw	bw	error	σ	#Features	error	σ	#Features
corrAl	0	1	0,00%	0,00%	4	0,00%	0,00%	4
corrAl	1	0	3,20%	6,40%	5	**0,00%**	0,00%	4
corrAl	1	2	0,00%	0,00%	4	0,00%	0,00%	4
corrAl	2	1	0,00%	0,00%	4	0,00%	0,00%	4
Ionosphere	0	1	9,68%	2,41%	9	**7,12%**	2,85%	13
Ionosphere	1	0	**6,27%**	2,15%	11	7,70%	1,95%	7
Ionosphere	1	2	**7,11%**	2,33%	12	7,42%	2,12%	7
Ionosphere	2	1	6,26%	1,09%	14	**5,11%**	2,44%	7
Mammogram	0	1	**15,03%**	5,65%	4	16,27%	2,29%	29
Mammogram	1	0	12,75%	6,74%	17	**12,68%**	5,29%	15
Mammogram	1	2	14,97%	7,41%	13	**11,57%**	5,05%	13
Mammogram	2	1	11,57%	3,42%	26	**8,10%**	4,61%	22
Spect	0	1	20,59%	1,07%	1	**20,22%**	1,81%	4
Spect	1	0	23,23%	3,11%	6	**20,59%**	1,07%	1
Spect	1	2	23,21%	1,38%	10	**20,60%**	1,68%	7
Spect	2	1	**21,35%**	1,92%	5	22,09%	2,68%	6
Spectf	0	1	20,98%	4,04%	11	**18,72%**	5,52%	13
Spectf	1	0	19,48%	4,98%	10	**17,97%**	3,85%	6
Spectf	1	2	20,58%	3,00%	22	**16,48%**	4,34%	27
Spectf	2	1	20,59%	3,53%	8	**17,64%**	5,07%	24
Sonar	0	1	13,94%	5,72%	39	**10,10%**	4,12%	28
Sonar	1	0	8,19%	4,70%	42	**7,21%**	2,61%	29
Sonar	1	2	12,02%	3,99%	22	**9,11%**	3,11%	26
Sonar	2	1	**7,20%**	5,42%	36	10,56%	4,87%	42
Waveform	0	1	**20,60%**	0,83%	18	21,32%	1,26%	17
Waveform	1	0	21,16%	0,85%	16	**20,60%**	0,83%	18
Waveform	1	2	21,00%	1,22%	15	21,00%	1,22%	15
Waveform	2	1	21,62%	1,11%	15	**21,14%**	0,48%	17
Wdbc	0	1	5,27%	2,41%	11	**4,39%**	2,66%	25
Wdbc	1	0	3,86%	2,39%	24	**3,34%**	2,17%	13
Wdbc	1	2	**3,51%**	3,04%	13	4,04%	1,89%	27
Wdbc	2	1	3,86%	1,80%	27	3,86%	2,26%	14

ified version would maybe select the former. That could lead the modified version to avoid local maxima by not selecting the best feature in this iteration feature and end with a better subset; but when the algorithm has selected a set close to optimal subset, the modification may cause the algorithm to loose precision in choosing features. This loss of precision can be greater when the remainder set of features is very small compared with the selected set so few feature interactions are taken into account in this remainder set. Thus a future line of work is to make the weight of the remainder set performance vary with its size to compensate for this fact.

References

[1] M. Ben-Bassat. *Use of Distance Measures, Information Measures and Error Bounds in Feature Evaluation*, volume 2, pages 773–791. North Holland, 1982.

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

[3] Mark A. Hall. *Correlation-based Feature Selection for Machine Learning*. PhD thesis, University of Waikato, 1999.

[4] George H. John, Ron Kohavi, and Karl Pfleger. Irrelevant features and the subset selection problem. In William W. Cohen and Haym Hirsh, editors, *Machine Learning, Procs. of the 11th Intl. Conf.*, pages 121–129, Rutgers University, New Brunswick, NJ, USA, July 10-13 1994. Morgan Kaufmann.

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

[6] P. Langley. Selection of relevant features in machine learning. In *Procs. of the* AAAI *Fall Symposium on Relevance*, pages 140–144, New Orleans, LA, USA, 1994. AAAI Press.

[7] O. L. Mangasarian and W. H. Wolberg. Cancer diagnosis via linear programming. 23(5):1–18, 1990.

[8] Ingo Mierswa, Michael Wurst, Ralf Klinkenberg, Martin Scholz, and Timm Euler. Yale: rapid prototyping for complex data mining tasks. In Tina Eliassi-Rad, Lyle H. Ungar, Mark Craven, and Dimitrios Gunopulos, editors, *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, August 20-23, 2006*, pages 935–940. ACM, 2006.

[9] Pavel Pudil, Jana Novovicová, and Josef Kittler. Floating search methods in feature selection. *Pattern Recognition Letters*, 15(11):1119–1125, 1994.

[10] S.D. Stearns. On selecting features for pattern classifiers. In *Procs. of the 3rd Intl. Conf. on Pattern Recognition (ICPR 1976)*, pages 71–75, Coronado, CA, 1976.

[11] David Wolpert and William G. Macready. No free lunch theorems for optimization. *IEEE Trans. Evolutionary Computation*, 1(1):67–82, 1997.