A feature set measure based on Relief

Antonio Arauzo-Azofra
Departamento de Ingeniería Rural
University of Cordoba
e-mail: arauzo at uco.es

José Manuel Benítez and Juan Luis Castro
Department of Computer Science and Artificial Intelligence
University of Granada
e-mail: {jmbs, castro} at decsai.ugr.es

Abstracts: *Feature selection methods try to find a subset of the available features to improve the application of a learning algorithm. Many methods are based on searching a feature set that optimizes some evaluation function. On the other side, feature set estimators evaluate features individually. Relief is a well known and good feature set estimator. While being usually faster feature estimators have some disadvantages. Based on Relief ideas, we propose a feature set measure that can be used to evaluate the feature sets in a search process. We show how the proposed measure can help guiding the search process, as well as selecting the most appropriate feature set. The new measure is compared with a consistency measure, and the highly reputed wrapper approach.*

Keywords: Feature, Attribute, Selection, Measure, Learning Algorithms.

1 Introduction

Feature selection help us to focus the attention of an induction algorithm in those features which are better to predict a target concept. Although, theoretically, if the full statistical distribution were known, using more features could only improve results, in practical learning scenarios it may be better to use a reduced set of features [5]. Sometimes a large number of features in the input of induction algorithms may turn them very inefficient as memory and time consumers, even turning them inapplicable. Besides, irrelevant data may confuse algorithms making them reach false conclusions, leading in this way to get worse results.

Apart from increasing efficiency and applicability of induction algorithms, the costs of data acquisition may also be reduced when a smaller number of features is selected, and the understandability of the results of induction algorithm improved.

All those advantages have made feature selection attracts much attention, and many methods have been developed. Some feature selection methods are based on attribute estimation. This is assigning a value of relevanceness to each attribute and then selecting those with higher values. Among these methods probably Relief[4] is the most well known and deeply studied algorithm. Relief estimates are better than usual statistical attribute estimates, like correlation or covariance, because it takes into account attribute interrelationships.

On the other side, no attribute estimator can handle redundancies among features, as features are evaluated individually. To overcome this problem, many feature selection methods consider the whole set of features. This leads to the need of performing a search on the possible feature sets.

A large family of feature selection methods based on searching fit on the modular decomposition we expose on section 2, where an evaluation function of feature sets is used by the search process.

In this paper, we propose and evaluate a new measure for feature sets, based on ideas taken from Relief.

In order to perform the empirical evaluation two main uses of feature set quality measures are identified: to select the best set and to guide the search. A measure that is able to choose the best set is not necessary better to guide the search as will be shown with an example.

In section 2, we show a modular decomposition of a feature selection search based algorithms and the measures for feature sets. Section 3 describes the original Relief and its extensions. After that, key ideas of Relief are identified and the new measure is presented in section 4. An empirical study is developed in section 5, and conclusions in section 6.

2 Search and feature set measures

The problem of feature selection can be seen as a search problem on the powerset of the set of available features [7]. The goal is finding a set of features that allows us to improve a learning activity. The process followed by many feature selection methods based on searching can be divided into two main parts: a search method through the feature sets space, and an evaluation function of a given set of selected features.

In the search process we can identify three parts: the choice of a starting point, the process of generating the next set to explore, and a stopping criterion. Figure 1 shows this modular decomposition of the feature selection process. It is based on the issues that Langley [7] identified. The divisions are also similar to those proposed by Dash and Liu [1].

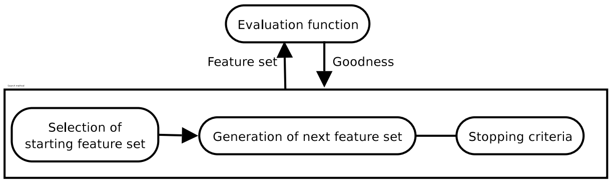

Figure 1: Feature selection process

The evaluation function, given a feature subset ($S \subset F$, being F the set of all features) and the training dataset (T), returns a measure of the goodness of that feature set. The main uses of the evaluation functions are to guide the search and to choose the selected feature set among those evaluated.

$$Eval : S \times T \longrightarrow \mathbb{R} \tag{1}$$

The wrapper approach [5] aims to improve results by using the learning algorithm as the evaluation function. While the wrapper approach has proven very useful, with good results in some circumstances, it is still interesting to study other evaluation measures for several reasons: an evaluation function may be more efficient, the wrapper approach is inapplicable with algorithms that suffer from the curse of dimensionality and many features, and some evaluation measures may be better than the wrapper approach to guide the search process.

3 Relief and its extensions

Originally proposed by Kira and Rendell [4], Relief is a feature selection method based on attribute estimation. Relief assigns a grade of relevance to each feature, and those features valued over a user given threshold are selected. Original Relief only handled boolean concept problems, but extensions have been developed to work in classification problems (Relief-F [6]) and in regression (RRelief-F [10]). The general algorithm that Relief and all its extensions follow is shown in figure 2. The extensions differ in the neighbours that are searched and in how the evaluation is performed from the example pairs.

As Relief-F generalizes the behaviour of Relief to classification, we will describe it directly. Relief-F finds one nearest neighbour of E_1 from every class. On these neighbours, Relief evaluates the relevance of every feature $f \in F$ accumulating it into $W[f]$ with equation (2). The nearest neighbour from the same class is a hit H, and from different class a miss, $M(C)$ of class C. At the end $W[f]$ is divided by m to get the average evaluation in $[-1, 1]$.

$$W[f] = W[f] - \text{diff}(f, E_1, H) + \sum_{C \neq class(E_1)} P(C) \times \text{diff}(f, E_1, M(C)) \tag{2}$$

The $\text{diff}(f, E_1, E_2)$ function calculates the grade in which the values of feature f are different in examples E_1 and E_2, as given in equation (3), where $value(f, E_1)$ denotes the value of feature f on example E_1, and $max(f)$ the maximun value f gets. The distance used considering nearest neighbours is the sum of differences, given by diff function, of all features.

$$\begin{array}{rl|c} & \textbf{f is discrete} & \textbf{f is continuous} \\ \text{diff}(f, E_1, E_2) = & \begin{cases} 0 & \text{if } value(f, E_1) = value(f, E_2) \\ 1 & \text{otherwise} \end{cases} & \frac{|value(f,E_1) - value(f,E_2)|}{max(f) - min(f)} \end{array} \tag{3}$$

In RRelief-F, k nearest neighbours are taken and their contribution is weighted according to their distance to E_1. They contribute to positive and negative evaluation of features weighted each by the diff function on the concept feature.

The drawback of any feature estimator approach to feature selection is that it is unable to detect redundant attributes or any other redundancy relations, as the features are valued individually. For example, if there is one duplicate feature, feature estimators give the same value to both features, and it will not be possible to reject one of them, while using both is clearly useless.

```
RELIEF($Dataset$, m, ...)

1. For 1 to m:
   1.1 $E_1$ = random example from $Dataset$.
   1.2 $Neighbours$ = Find some of the nearest examples to $E_1$.
   1.3 For $E_2$ in $Neighbours$:
       1.3.1 Perform some evaluation between $E_1$ and $E_2$
2. Return the evaluation
```

Figure 2: General Relief algorithm

4 Relief Feature Set Measure

Given its good results and the extensive work that have been published about Relief [6, 10, 9], we think it is interesting to apply its main ideas to evaluate complete feature sets. In our view, the key ideas of relief are:

- Raising relevance degree to those features that have different values on example pairs that have different concept value.
- Penalization of features. In parallel to previous idea, Relief reduces relevance degree to those features with different values on pairs that have the same concept value.
- Pairs are selected from near examples. Given an example, Relief takes other examples, with the same and different class, from its neighbourhood. This is probably the point where the success of Relief resides. The bias of considering near examples makes it work in a non-myopic way [11], and consider the interrelation with other features.
- Random sampling is used to get each example used in evaluation. In this way, running time is reduced while accuracy is not significantly degraded. It is still recommended to use every example if the dataset is small or if we can afford it. As each example is taken in a step, Relief could take more examples on the fly, if more time is available, to improve its estimates. This makes Relief an anytime algorithm as mentioned in [9], section 6.2.

Based on these ideas from Relief, we propose a feature set measure to be used as an evaluation function in the search process. We will refer to this measure as Relief Feature Set Measure (RFSM). The computation of this new measure will be done with the common algorithm skeleton shown in figure 2. The evaluation result is now a single value W_S for the feature set S, instead of the vector of values $W[f]$ for every feature.

The proposed measure uses diffS function, of equation (4), to measure the ability of the set of features S to differentiate between the two example pairs, as diff function measured the ability of f in Relief.

$$\text{diffS}(S, E_1, E_2) = \text{operator}_{f \in S}\{\text{diff}(f, E_1, E_2)\} \tag{4}$$

The *operator* could be some agreggation operator of the diff values. In discrete features, these are boolean values indicating if each feature is able to differentiate the examples. A set of features is able to differentiate the examples if any of its features is able to do so. Therefore we consider the ideal operator for discrete values is the logic *or*. Any t-conorm could be used to generalize the or behaviour to continuous values. As it seems reasonable to think that the grade in which a set of features differentiate two examples is the grade of that feature that differentiate them most, we have used the *max* operator.

On the penalization part of the evaluation, we were concerned by the results of some informal experiments in which the number of features chosen was too small, leading to poor classification results. We have chosen to use the *min* operator, not only because it reduced the penalization, but mainly because it gives to the measure the property of monotonicity. The monotonic property requires that if S_i, S_j are feature sets and $S_i \subset S_j$, then $M(S_i, D) \leq M(S_j, D)$, where M is the measure and D a dataset. This is useful for some search algorithms like branch & bound.

Finally, in order to evaluate the feature set, the diffS is used weighted using class probabilities with equation 2, as in Relief-F, for classification problems, and with the same weighting as RRelief-F for regression or approximation problems.

We expect RFSM being able to give higher values to those feature sets that are potentially good. Those sets that, while not being better than others to learn, can become better when some other features are included. For example, in a boolean dataset, with equiprobable random features, and the concept value being the function $(x_1 \oplus x_2) \oplus x_3$ ($\oplus$ denotes the exclusive or), any of the relevant features alone is statistically independent to the concept. Any learning algorithm would not be able to induce anything from a relevant feature alone, or from a set of only two of the relevant features. Therefore, neither wrapper, nor any other measure based only on the features to evaluate would be able to help guiding a search that starts on the empty set of features, and starting the search on the full set of features is not always possible or recommendable.

5 Empirical study

The object of our study is to determine the performance of the new Relief Feature Set Measure (RFSM) in a feature selection process. We explore the two main uses of measures identified in section 2.

The performance of the measure to guide the search is evaluated by using a greedy search method. Greedy search strongly relies on the measure to select the search path, as only one path will be explored with no possible back-track. We believe that a search that is so sensitive to measure is the best way to see if the measure is helpful in guiding a search process. Besides, greedy search is very efficient. The exact search process is described by specifying its parts: the starting point is the empty set, the next feature sets to explore is the one adding a feature with maximum value of the measure, and the stopping criterion is true when maximum can not be improved.

Dataset	zoo		lung-cancer		house-vot.84		breast-canc.		wine		prostate	
	Acc.	NoF	Acc.	NoF	Acc.	NoF	Acc.	NoF	Acc.	NoF	MSE	NoF
tree	96.0	16.0	43.3	56.0	96.09	16.0	67.8	9.0	93.79	13.0	1.32	8.0
Liu+tree	95.0	4.9	50.8	4.5	96.55	10.0	66.7	7.9	94.93	5.1	1.33	7.2
Wrapper+tree	95.0	4.9	48.3	8.9	95.60	8.2	72.1	3.1	93.90	4.0	1.64	3.7
RFSM+tree	94.0	6.0	50.8	3.5	95.87	11.5	66.7	8.1	94.35	11.0	1.35	7.5
kNN	94.1	16.0	44.2	56.0	93.32	16.0	72.0	9.0	97.19	13.0	0.92	8.0
Liu+kNN	83.1	4.9	45.8	4.5	94.48	10.0	71.7	7.9	94.97	5.1	0.93	7.2
Wrapper+kNN	89.0	9.9	41.7	16.8	94.50	5.4	67.9	5.2	94.30	6.0	0.77	5.3
RFSM+kNN	88.1	6.1	44.2	3.9	95.39	11.8	73.8	8.2	97.75	11.5	0.99	6.9

Table 2: Feature selection on real world datasets

The use of the measure to choose the best feature set is tested on artificial datasets, checking that those sets with all the relevant features get the maximum value of the measure, and on real world datasets comparing the classification performance of the features selected.

In both cases, the performance of the whole feature selection process is evaluated. The results are compared with the wrapper approach, Liu's consistency measure[8] and the performance of learning algorithms without feature selection.

Artificial datasets

Two well known artificial datasets are used. They present two difficulties: redundant features in Parity3+3, and a feature that seems useful but it is not necessary in CorrAL. Parity3+3 dataset is the parity of three boolean features ($x_1 \oplus x_2 \oplus x_3$), with the three relevant features duplicated and another six irrelevant random features included. CorrAL dataset represents the function $(x_1 \wedge x_2) \vee (x_3 \wedge x_4)$, with one additional random feature and another correlated with the concept function 75% of the times.

	CorrAL	Parity3+3
Liu's measure	5.0 ±0.00	4.7 ±0.21
Wrapper(tree)	5.2 ±0.13	6.5 ±0.65
Wrapper(knn)	5.0 ±0.15	5.9 ±0.85
RFSM	4.0 ±0.00	3.0 ±0.00

Table 1: No. features ±Std.err.

Table 1 shows the mean number of features selected by each method in a ten fold cross validation. In all cases the feature sets selected contain all the necessary features. Better than we expected before, RFSM has been able to guide the search perfectly, while the other methods have found sets with unnecessary features.

Real world datasets

Six real world datasets from the UCI[3] repository have been used. Three of them are classification problems with discrete features, the next two, classification with discrete and continuous features, and the last one is an approximation problem. In order to apply Liu's measure to continuous features, equal frequency discretization is used.

The learning algorithms used to check the quality of features selected are a classification and regression tree learner (tree) with pruning, and kNN with $k = 21$. The implementation and all details about these algorithms are available from the Orange Data Mining system [2].

Table 2 shows the results of the considered feature selection methods with the learning algorithms. The average accuracy achieved in ten fold cross validation is given in classification percentage or mean squared error, and the average number of features used is also reported.

We can see how the feature selection using RFSM is useful, as the accuracy is kept or improved, while the number of features is reduced. The results are even better than using the wrapper approach in some problems. The combination of kNN classifier with RFSM in general have given better results than using Liu's measure, while, with the tree learner, results are similar.

6 Conclusions and future work

We have described the process that a large family of feature selection algorithms follow, being the evaluation function of feature sets one of their important parts. After that Relief algorithms have been described and their main ideas identified. Based on these ideas, a new feature set measure is proposed and evaluated. On artificial datasets, we show the proposed measure can be better than the wrapper approach to guide a common feature selection search process. In a practical scenario, using real world datasets, the experiments show how feature selection using our measure improves the learning process, and how the new measure performs better than the highly reputed wrapper approach and Liu's consistency measure.

As future work, we consider interesting to study if the good performance of the measure guiding the search are applicable to other search methods. As well as, extending the measure to deal with linguistic variables.

References

[1] M. Dash and H. Liu. Feature selection for classification. *Intelligent Data Analysis*, 1(1-4):131–156, 1997.

[2] J. Demsar and B. Zupan. Orange: From experimental machine learning to interactive data mining. (White paper) http://www.ailab.si/orange, 2004.

[3] S. Hettich and S. D. Bay. The uci kdd archive. http://kdd.ics.uci.edu/, 1999.

[4] Kenji Kira and Larry A. Rendell. A practical approach to feature selection. In *Proceedings of the ninth international workshop on Machine learning*, pages 249–256. Morgan Kaufmann Publishers Inc., 1992.

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

[6] Igor Kononenko. Estimating attributes: Analysis and extensions of RELIEF. In *European Conference on Machine Learning*, pages 171–182, 1994.

[7] Pat Langley. Selection of relevant features in machine learning. In *Procedings of the AAAI Fall Symposium on Relevance*, New Orleans, LA, 1994. AAAI Press.

[8] Huan Liu, Hiroshi Motoda, and Manoranjan Dash. A monotonic measure for optimal feature selection. In *European Conference on Machine Learning*, pages 101–106, 1998.

[9] Marko Robnik-Sikonja and Igor Kononenko. Theoretical and empirical analysis of relieff and rrelieff. *Machine Learning*, 53:23–69, 2003.

[10] M. Robnik Sikonja and I. Kononenko. An adaptation of relief for attribute estimation in regression. In Morgan Kaufmann, editor, *Machine Learning: Proceedings of the Fourteenth International Conference*, pages 296–304, 1997.

[11] R. Sikonja and M. Kononenko. Non-myopic attribute estimation in regression, 1996.