

Multi Criteria Wrapper Improvements to Naive Bayes Learning

José Carlos Cortizo¹ and Ignacio Giraldez²

- ¹ Artificial Intelligence and Network Solutions S.L., Universidad Europea de Madrid
jccp@ainetsolutions.com, <http://www.ainetsolutions.com/jccp>
- ² Universidad Europea de Madrid, C/Tajo s/n, Villaviciosa de Odón (Madrid), Spain
ignacio.giraldez@uem.es

Abstract. Feature subset selection using a wrapper means to perform a search for an optimal set of attributes using the Machine Learning Algorithm as a black box. The Naive Bayes Classifier is based on the assumption of independence among the values of the attributes given the class value. Consequently, its effectiveness may decrease when the attributes are interdependent. We present FBL, a wrapper that uses the information about dependencies to guide the search for the optimal subset of features and the Naive Bayes Classifier as the black-box Machine Learning algorithm. Experimental results show that FBL allows the Naive Bayes Classifier to achieve greater accuracies and that FBL performs better than other classical filters and wrappers.

Keywords: Naive Bayes, (In)Dependent Attributes, Wrapper, Feature Subset Selection, Machine Learning, Data Analysis

1 Introduction and Motivation

Machine Learning algorithms try to learn from experience, usually coded as a set of training instances defined as a set of attribute-value pairs. As a general rule, the algorithms' classification accuracy depends on the attributes and values given to those attributes, and it degrades in performance when faced with many features that are not necessary for predicting the desired output ([KO2]).

Work in feature subset selection can be summarized in two main trends: the *filter model* and the *wrapper model*. In the *filter model* (e.g. [KI1], [AL1], [JA1]) the features are filtered according to a certain metric but independent of the Machine Learning algorithm. In the *wrapper model* ([JO1]), the feature subset selection algorithm uses the classification algorithm as a part of the evaluation function.

The Naive Bayes Classifier provides the most probable target value $v_m \in V$ for a new instance according to the concrete values of the attributes on the instance (a_1, a_2, \dots, a_n) . The most probable target value is

$$v_m = \operatorname{argmax}_{v_j \in V} P(v_j | a_1, a_2, \dots, a_n) \quad (1)$$

This expression is rewritten using the Bayes Theorem and ignoring the common denominator:

$$v_m = \operatorname{argmax}_{v_j \in V} P(a_1, a_2 \dots a_n | v_j) P(v_j) \quad (2)$$

In Equation 2 is easy to estimate $P(v_j)$ because it represents the a priori probability of the value v_j of the class and can be obtained dividing the number of instances belonging to that category by the total number of instances. Estimating $P(a_1, a_2 \dots a_n | v_j)$ is not so simple, but assuming independency among the attribute values given the class value (v_j), the the probability $P(a_1, a_2 \dots a_n | v_j)$ can be factorized as the product $P(a_1 | v_j) \dots P(a_n | v_j)$. Replacing this into Equation 2, we obtain the Naive Bayes Classifier [KO1]

$$v_m = \operatorname{argmax}_{v_j \in V} P(v_j) \prod P(a_i | v_j) \quad (3)$$

To obtain the Naive Bayes Classifier, we must assume independency among the attributes values. Our goal, as in [CO1], is to *modify the Naive Bayes Classifier to obtain an extended-Naive Bayes algorithm robust to attribute dependencies*. As [LA1] or [PA1], we propose a wrapper (FBL) to improve Naive Bayes, where the evaluation function is the accuracy obtained by the Naive Bayes. But where [LA1] and [PA1] performed a greedy search, we propose a search guided by the information about the dependencies among attributes. As starting point, we have used linear regression [FI1] (as a first kind of dependency for guiding the search), which is a classical statistical solution to the problem of determining the relationship between two random variables X and Y (or even more than two random variables). We can extend the behaviour of the system adding other kind of dependencies.

In section 2 we present FBL, a strength-of-dependency based wrapper for attribute selection. Section 3 explains the experiments made for testing the FBL algorithm and the comparisons to other attribute selection methods. In section 4 we discuss the results and present our conclusions.

2 FBL Algorithm

We propose a method to improve the Naive Bayes Classifier based on a previous filtering of the attributes used for representing the data (the additional flow required by FBL to classiy instances is represented in Figure 1). It filters out the dependent attributes of a given dataset, as a result, the set of attributes used to represent the data is modified. Then it transforms the original data set so it complies with the new representation.

The Naive Bayes Classifier works under the assumption of independent attributes, and that is why we perform a first stage where we detect all the dependencies between attributes for a later processing, trying to achieve a representation free of dependent attributes. This is performed at the first stage called "Dependency Analysis". The complete dependency search and clean algorithm can be decomposed into four main steps detailed next.

First step: Definitions and initialization. First we define and initialize some variables and lists. Being $S = \{A_1, A_2 \dots A_n\}$ the original set of attributes

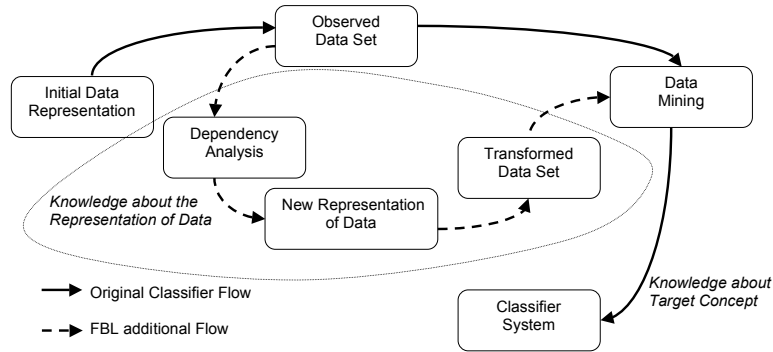


Fig. 1. Naive Bayes original data flow and FBL extended flow for performing a complete classification.

and $V = \{v_1, v_2 \dots v_m\}$ the possible class values, $e_j = \{\vec{x}, v\}$ is a training example where $\vec{x} = (a_1, a_2 \dots a_n)$ is a point that belongs to the input space (X) and $v \in V$ is a point belonging to the output space (V). We also initialize an empty list, L_{IG} , a list of attributes ordered by the inverse of their Information Gain ([KU1]).

Second step: Dependencies analysis. We search for dependencies of the form $a_i = \alpha + \beta A_v$ where $a_i \neq a_v \wedge a_i, a_v \in S$ y $\alpha, \beta \in \mathbb{R}$. For each a_i we calculate the squared correlation (R_i^2) between its attribute values (this value measures how well the regression curve fits the points), then $d_i = (a_i, R_i^2)$. We define $L = \{d_i\}$ for $i = 1 \dots n$, which contains all the possible dependencies between pairs of attributes ($\sum_{i=1}^{N-1} i$ possible dependencies) and is in a strength-of-dependency (R^2) decremental order.

Third step: Dependency based filtering. At this step we use L to obtain the final attribute set by deleting (or not) the most dependent attributes until no accuracy improvement is achieved. At each step, the algorithm considers two dependencies, one is the next dependency not previously seen in L , and the other is the next to this one. We use two dependencies at each stage because some dependent attributes are also very informative when dealing with the target class and should not be removed. When an attribute (A_i) is deleted from the dataset, L is updated deleting all the dependencies where A_i appears as one of the two dependent attributes, and also L_{IG} is updated deleting A_i . When in any step no performance of the classifier could be achieved deleting any of the related attributes, the first stage of the algorithm concludes.

Fourth step: IG based filtering. A further attribute filtering is performed by considering the Information Gain values of the attributes. For every attribute in the resulting L_{IG} we consider if we should delete it by comparing the accuracy of the classifier over the dataset versus the accuracy of the classifier over the dataset when removing the attribute. Once finished the whole process, we obtain

```

input : A DataSet
output: A DataSet with the same number of instances that the input but
        only a subset of the original attributes

1  $F \leftarrow S$ ;
2  $L_{IG} \leftarrow \text{IGInverseOrder}(S)$ ;
3  $L \leftarrow \{\}$ ;
4  $P_{init} \leftarrow \text{NaiveBayes}(S)$ ;
5 for  $A_i \in S$  do
6   for  $A_j \in S - A_i$  do
7      $L_S \leftarrow \{A_i, A_j, \mathbf{g}(A_i, A_j)\}$ ;
8      $L \leftarrow L \cup L_S$ ;
9   end
10 end
11  $L \leftarrow \text{Order}(L)$ ;
12  $pos \leftarrow 0$ ;
13  $fin \leftarrow false$ ;
14 while ( $fin = false$ ) && ( $pos < L.length$ ) do
15    $L_1 \leftarrow L[pos]$ ;
16    $Attributes \leftarrow \{L_1.A_i, L_1.A_j\}$ ;
17   if  $pos < (L.length - 1)$  then
18      $L_2 \leftarrow L[pos + 1]$ ;
19      $Attributes \leftarrow Attributes \cup \{L_2.A_i, L_2.A_j\}$ ;
20   else
21      $Atr \leftarrow \text{GetBest}(Attributes)$ ;
22   end
23   if  $Atr = null$  then
24      $fin \leftarrow true$ ;
25   else
26      $L_{IG} \leftarrow L_{IG} - Atr$ ;
27      $L \leftarrow \text{DeleteDependencies}(L, Atr)$ ;
28      $F \leftarrow \text{DeleteAttribute}(F, Atr)$ ;
29      $T \leftarrow \text{Transform}(F)$ ;
30      $P_i \leftarrow \text{NaiveBayes}(T)$ ;
31   end
32 end
33  $pos \leftarrow 0$ ;
34 while  $pos < L_{IG}.length$  do
35    $Atr \leftarrow L_{IG}[pos]$ ;
36    $F_{aux} \leftarrow \text{DeleteAttribute}(F, Atr)$ ;
37    $P_p \leftarrow \text{NaiveBayes}(F_{aux})$ ;
38   if  $P_p > P_i$  then
39      $P_i \leftarrow P_p$ ;
40      $F \leftarrow F_{aux}$ ;
41   end
42 end
43 return  $P_i$ 

```

Algorithm 1: FBL Algorithm

$F = \{A_i, A_j \dots A_z | A_k \in S\}$ that is a subset of S that contains the best possible attributes.

Having F, we modify each element in T deleting all the information that does not correspond with the final attributes (F). Being g the regression method used and t a function that transforms each training instance deleting the values of the attributes that belonging to S are not in F, the FBL algorithm is implemented as shown in Algorithm 1.

3 Experiments

In this section some experiments are present to evaluate our hypothesis and the Wrapper proposed in this paper. To study the behaviour of FBL, 13 synthetic domains were generated from 5 domains (extracted from [BL1]). Dependencies were introduced by adding synthetic attributes dependent from the original ones As can be seen in Table 1, in the 13 synthetic domains the Naive Bayes Classifier works worst than in the original ones where the synthetic dependencies are not present.

Table 1. Comparison of the performance of 6 Machine Learning Algorithms working with data with dependencies. The values represents the variation in the percentage of accuracy from the original dataset to the resultant dataset where synthetic attributes have been added. The datasets are: Contraceptive Method (CM), Breast Cancer (BC), Balance Scale (BS), Glass Identification (GI), Wine Recognition (WR) and TAE. For each dataset we have generated one or some synthetic ones (which can be differentiated by the final number). All the accuracies are calculated using ten-fold cross validation.

Algorithm	CM1	CM2	CM3	BC1	BC2	BC3	BC4	BS1	BS2	BS3	GI1	WR1	TAE
Naive Bayes	-1.15	-2.45	-1.02	-0.14	-0.14	-0.14	-0.14	-9.11	-5.23	-0.68	-0.08	-0.07	-0.04
C4.5	-0.61	1.77	0.68	1.72	3.72	2.72	0.14	0.00	0.45	0.68	0.13	0.01	0.89
C4.5 rules	0.34	0.81	1.08	0.72	1.58	0.29	0.86	-2.74	-1.60	-0.91	0.72	1.58	-0.29
KNN(N=1)	-1.29	-0.61	-1.29	0.04	0.28	0.28	-0.15	-0.23	0.91	1.14	-0.13	-0.28	0.23
KNN(N=4)	-0.81	-0.88	-0.88	0.15	0.72	0.72	-0.43	-1.14	-0.91	-2.50	-1.16	-0.72	0.01
SMO	0.06	0.06	0.00	-0.14	-0.14	-0.14	0.00	0.22	-0.23	-0.46	0.00	-0.04	0.54

In contrast to the other Machine Learning algorithms used, Table 1 shows that Naive Bayes is the only algorithm that invariably worsens its accuracy when interdependent attributes are used. Once proven this, the next step is to study if non-synthetic datasets reveal the same behaviour. To check this, 15 datasets have been selected from [BL1], and have been studied without adding any synthetic feature.

The effectiveness of FBL can be evaluated by comparing the FBL attributes selection to the best possible subset selection. To calculate the best subset we have performed an exhaustive search over the space of attributes of each one of the 15 datasets selected. Table 2 shows the result of this exhaustive search, showing the accuracy of the Naive Bayes Classifier over the best possible subset

Table 2. Comparison between the accuracy obtained by the Naive Bayes (NB), the Naive Bayes over the best possible attribute selection (Best), the FBL algorithm and using filters based on Information Gain (IG) or Chi^2 , PCA or the Langley algorithm in a forward (LF) or a backward mode (LB). O/B means the number of original attributes (O) and the number of attributes in the best subset selection (B). All the results are obtained by 10-fold cross validation.

DataSet	NB	Best	O/B	FBL	IG	Chi^2	PCA	LF	LB
Abalone	0.240	0.266	9/2	0.266	0.266	0.266	0.245	0.265	0.265
Adult	0.827	0.835	15/13	0.835	0.752	0.752	ND	0.791	0.835
Cmc	0.508	0.554	10/4	0.554	0.508	0.508	0.453	0.553	0.537
Glass	0.495	0.603	10/3	0.579	0.556	0.556	0.542	0.598	0.598
Ionosphere	0.826	≥ 0.984	36/?	0.915	0.863	0.872	0.920	0.984	0.900
Iris	0.960	0.967	5/2	0.967	0.967	0.967	0.933	0.960	0.953
Nursery	0.903	0.903	9/9	0.903	0.876	0.903	0.883	0.903	0.903
OpDigits	0.913	≥ 0.939	64/?	0.913	0.865	0.881	0.939	0.925	0.918
PenDigits	0.857	0.865	16/13	0.865	0.615	0.836	0.886	0.864	0.864
Spam	0.793	≥ 0.908	57/?	0.908	0.793	0.793	0.737	0.837	0.902
TAE	0.503	0.510	6/5	0.510	0.470	0.510	0.470	0.503	0.509
TicTacToe	0.696	0.724	9/5	0.718	0.699	0.699	0.742	0.718	0.728
WdbCancer	0.929	≥ 0.959	31/?	0.956	0.924	0.923	0.937	0.959	0.945
Wine	0.966	0.989	13/10	0.977	0.792	0.977	0.983	0.989	0.977
Yeast	0.577	0.577	7/7	0.577	0.577	0.577	0.558	0.577	0.577

of attributes, the number of attributes of the best subset and also the number of attributes of the original dataset for establishing a comparison. In 13 from 15 datasets we are able to find a subset of attributes that allows the Naive Bayes Classifier to perform better than using the original ones. In addition, we can see the best subsets are composed, generally, by a few number of attributes, which means a drastic reduction of the complexity of the classifier’s model.

Once obtained the superior level for the attributes selection, we compare these results with the behaviour of the FBL algorithm. We have performed some experiments using the FBL algorithm over the same 15 UCI domains, and also we have studied the behaviour of two classical ways of attribute selection as are the filtering using the Information Gain value and also the filtering using the Chi^2 value and the Principal Components Analysis (PCA, [SH1]) method to extract a subset of more or less independent attributes. For the filtering using Chi^2 and IG, we have generated a ranking of the attributes according to those metrics and then, from an empty set of features, we have added one attribute from the ordered list to the set of features until no improvement of the classifier is achieved. For PCA we have studied the behaviour of the Naive Bayes Classifier over the studied domains covering 4 different levels of variance (0.65, 0.75, 0.85 and 0.95) and then we select the best value for each dataset. The results are shown in Table 2.

Finally, Table 3 is a summary table. Having the average accuracy obtained for the 15 datasets proven, the times that each algorithm obtains the best accuracy,

Table 3. Summary table where, for each algorithm proven, it is shown the average accuracy over the 15 datasets proven, the times the algorithm obtains the best accuracy and the percentage of accuracy reduced.

Algorithm	N.B.	FBL	PCA	LF	LB	IG	Chi ²
Avg. Accuracy	0.7331	0.7629	0.7327	0.7619	0.7610	0.7016	0.7346
Trim	0.00%	11.17%	-0.15%	10.78%	10.45%	-11.79%	0.57%
Times Best	ND	8	3	6	4	3	3

and the percentage of improved accuracy, we are able to compare more carefully all the algorithms used in this paper. FBL obtains the best possible value 8 times from 15 possible, being the algorithm that more times obtains the best value. Also obtains the best accuracy value, and we can assure it is the best option from all the proposed in this paper. The forward exhaustive search proposed by Langley is near FBL in accuracy values, but is not as good as FBL and also performs an exhaustive search, wasting too much time. It seems interesting the results obtained by PCA and the Information Gain based selection as, in average, obtain worse results than using the Naive Bayes without any transformation or filter.

4 Discussion and Future Work

We have presented a simple alternative to the Naive Bayes which works well even with domains where strong linear dependencies³ are present. The proposed algorithm performs a non greedy search based on the previous estimation of the dependencies strength between attributes, and the filtering according to these values and also according to the Information Gain values for each attribute.

We have run several experiments over 15 UCI domains, comparing the accuracy of the FBL algorithm presented in this paper to the accuracy obtained by the Naive Bayes and the accuracies obtained by performing some attribute filters: PCA, the forward and backward attribute selection algorithms proposed by Langley and a Information Gain and a Squared Chi based filterings.

Experimental results show that using the FBL Wrapper allows us to obtain better accuracy results when applying the Naive Bayes Classifier. It should be remarked that on 13 from 15 real datasets we can perform a better classification when those dependencies are not present, which shows we are not working on a synthetic problem, this is a reality. We have proved that a classical attributes extraction technique such as PCA does not make the Naive Bayes to perform better. We have also proved that filters based on Information Gain and Squared Chi metrics are not a good way to make the Naive Bayes perform better and that FBL is as good (or better) than the greedy algorithms proposed by Langley but performing a more guided search that implies to try less attribute sets, resulting in a lower running time.

³ FBL can be extended to use other kind of dependencies

Applications as Spam are excellent scenarios to apply FBL as their domains are composed by words extracted from texts, and those words are not independent attributes as some words should appear, commonly, in pairs, or the appearance of some of them should be much related to the appearance of other words. Any other applications on the Information Retrieval area should be also susceptible to be improved with the use of FBL, as could be [GO1].

Summarizing, the FBL algorithm performs better than Naive Bayes, PCA and other attribute filters under 15 UCI domains as it is able to deal with the dependencies presented on those domains.

References

- [AL1] Almuallim, H., Dietterich, T. G.: Learning with Many Irrelevant Features. 9th National Conference on Artificial Intelligence (1991) Mit Press. 547–552.
- [BL1] Blake, C. L. and Merz, C. J.: UCI Repository of Machine Learning Databases. [<http://www.ics.uci.edu/mllearn/MLRepository.html>]. Irvine, University of California. Department of Information and Computer Science.
- [CO1] Cortizo, J. C. and Giraldez J. I.: Discovering Data Dependencies in Web Content Mining. IADIS International Conference WWW/Internet. (2004) 881–884.
- [FI1] Fisher, R.: Statistical Methods for Research Workers. Macmillan Pub Co (1925)
- [GO1] Gomez, J. M., Buenaga, M., Cortizo, J. C.: The Role of Word Sense Disambiguation in Automated Text Categorization NLDB 2005 Springer Verlag, LNCS **3513**. (2005) 298–309.
- [JA1] Jakulin, A. and Bratko, I.: Analyzing Attribute Dependencies. Proceedings of Knowledge Discovery in Data (PKDD) Springer Verlag, LNAI. (2003) 229–240.
- [JO1] John, G. H., Kohavi, R., Pfleger, K.: Irrelevant Features and the Subset Selection Problem. Proceedings of the International Conference on Machine Learning (1994), 121–129.
- [KI1] Kira, K., Rendell, L. A.: A Practical Approach to Feature Subset Selection. 9th International Conference on Machine Learning (1992) Morgan Kaufman.
- [KO1] Kononenko, I.: Comparison of inductive and naive Bayesian learning approaches to automatic knowledge acquisition. B. Wielinga Editors, Current trends in Knowledge Acquisition. Amsterdam: IOS Press (1990) 190–197.
- [KO2] Kohavi, R., Sommerfield, D.: Feature Subset Selection Using the Wrapper Method: Overfitting and Dynamic Search Space Topology. 1st International Conference on Knowledge Discovery and Data Mining (1995) 192–197.
- [KU1] Kullback, S., Leibler, R. A.: On Information and Sufficiency. Annals of Mathematical Statistics **22**, (1951) 79–86.
- [LA1] Langley, P.: Induction of recursive Bayesian Classifiers. Proceedings of the 1993 European Conference on Machine Learning (1993) 153–164.
- [MO1] Montes, C.: Metodo de Induccion Total. PhD Thesis. Universidad Politecnica de Madrid, Boadilla del Monte, Spain.
- [NE1] Neter, J., Kutner, M. H., Wasserman, W., Nachtsheim C. J.: Applied Linear Statistical Models. Irwin Editors, (1996).
- [PA1] Pazzani, M.: Searching for dependencies in Bayesian Classifiers. Artificial Intelligence and Statistics IV. Springer Verlag, New York, USA. (1997)
- [SH1] Shlens, J.: A tutorial on Principal Component Analysis. Systems Neurobiology Laboratory, Salk Institute for Biological Studies (2005).