THE SEPARABILITY OF SPLIT VALUE CRITERION

Krzysztof Grąbczewski and Włodzisław Duch[1]
Department of Computer Methods, Nicolaus Copernicus University
ul. Grudziądzka 5, 87-100 Toruń, Poland

Abstract:
The Separability of Split Value (SSV) criterion is a simple and efficient tool for building classification trees and extraction of logical rules. It deals with both continuous and discrete features describing data vectors and requires no user interaction in the learning process. Extensions of methods based on this criterion are presented. They aim at improvement of reliability and efficiency of the methods and extension of their applications area. Good results for several benchmark datasets were obtained.

1 INTRODUCTION

In many applications classification is not sufficient, some form of explanation or domain theory may be required. Generation of logical rules describing a dataset is closely bound up with the task of finding the most useful (for that purpose) linguistic variables. There are many ways of providing linguistic variables and they are strongly dependent on the method that they are part of. For example, in the MLP2LN approach developed in our group [1] combination of two neurons with constrained weights (called L-units) are used, while in decision trees linguistic variables correspond to conditions (usually open intervals) tested in tree nodes.

In this paper some improvements and extensions to the Separability of Split Value (SSV) criterion for classification systems introduced recently [2] are presented. The first two sections present the original criterion and its possible applications. The fourth section presents new improvements and extensions and the fifth presents some new results.

2 THE CRITERION

In this section a criterion allowing to separate objects with different class labels is presented. It's basic advantage is that it can be applied to both continuous and discrete features, which means that methods based on it can operate on raw data without the need for standardization. The best split value is the one that separates the largest number of pairs of objects from different classes. The *split value* (or *cut-off point*) is defined differently for continuous and discrete features. In the case of continuous features the *split value* is a real number, in other cases it is a subset of the set of alternative values of the feature. In all cases we can define *left side* (LS) and *right side* (RS) of a split value s of feature f for given dataset D:

[1] E-mails: {kgrabcze,duch}@phys.uni.torun.pl, WWW: http://www.phys.uni.torun.pl/kmk

$$\begin{aligned} \text{LS}(s,f,D) &= \begin{cases} \{x \in D : f(x) < s\} & \text{if } f \text{ is continuous} \\ \{x \in D : f(x) \notin s\} & \text{otherwise} \end{cases} \\ \text{RS}(s,f,D) &= D - \text{LS}(s,f,D) \end{aligned} \tag{1}$$

where $f(x)$ is the f's feature value for the data vector x. The *separability of a split value* s is defined as:

$$\begin{aligned} \text{SSV}(s) = 2* & \sum_{c \in C} |LS(s,f,D) \cap D_c| * |RS(s,f,D) \cap (D - D_c)| \\ - & \sum_{c \in C} \min(|LS(s,f,D) \cap D_c|, |RS(s,f,D) \cap D_c|) \end{aligned} \tag{2}$$

where C is the set of classes and D_c is the set of data vectors from D which belong to class c. The higher the separability of a split value the better. According to this criterion the best split value is the one which separates the maximal number of pairs of vectors from different classes and among all the split values which satisfy this condition - the one which separates the smallest number of pairs of vectors belonging to the same class. For every dataset containing vectors which belong to at least two different classes, for each feature which has at least two different values, there exists a split value of maximal separability.

When the feature being examined is continuous and there are several different split values of maximal separability close to each other the split value closest to the average of all of them is selected. To avoid such situations it is good to examine split values that are natural for a given dataset (i.e. centered between adjacent feature values that occur in the data vectors). If there are non-maximal (regarding separability) split values between two maximal points or if the feature is discrete, then the selection of the best split value may be arbitrary.

3 APPLICATIONS

The separability criterion can be used to build classification decision trees. Thanks to the ease of finding the best cuts generated trees may be small, and may be converted into a small number of crisp logical rules. The classification trees are built by finding the best split of the dataset (which becomes a node of the tree) and splitting the data into two parts for further recursive analysis. Special pruning technique based on cross-validation calculations is used to make the tree more general.

The criterion can also be used in algorithms aiming at discretization of continuous features. Because it provides some evaluation of possible splits (for each feature) it can select several most valuable splits for each continuous feature, which in fact is a discretization of the feature. In a similar way one can estimate each feature's importance and select the most valuable features if the data consists of too many dimensions to let systems like k-NN (k nearest neighbors) work successively.

4 EXTENSIONS AND IMPROVEMENTS

4.1 Handling of missing data

There are two sources of the missing values in data. First, because the data has not been collected or measured for a good reason (for example, because the initial examination allowed the doctor to make a reasonable hypothesis and some tests were not needed). Second, because the data got lost or become corrupted, but was desired. These two reasons require different data handling. Unfortunately we usually are not provided with information about the reasons for the missing features. Using missing data imputation methods may lead to building a model strongly affected by the imputed data, which is not desired. Most classification methods are not able to build models based only on the data which are available in the dataset. The original definition of the SSV tree algorithm [2] suffered the same drawback, however it can be rectified quite easily.

Since the criterion counts the numbers of correctly (and incorrectly) split pairs, it can be modified to restrict this analysis to the values which are not missing. Thanks to that the features for which there are many missing values have smaller separability values and the features which have no misses are preferred. Such technique requires an additional solution to the problem of splitting data into two parts when a tree node is being added and to the problem of application of a tree to data vectors containing missing values. In the training process, when the best possible split is for feature with some misses two subsets of the training data are created, not necessarily disjoint: the vectors which miss appropriate feature value go to both subsets i.e. both branches of the tree. It is equivalent to ignoring the nodes if the split condition cannot be checked. As a consequence of that when we need to classify an example with a tree, and at some node we cannot determine which way to go, we have to check both ways. If they lead to different decisions the system should answer “don’t know” to the classification question.

4.2 A better pruning technique

The original pruning technique was quite efficient, in most cases giving quite accurate results. However, sometimes it failed to find the best (from the generalization point of view) possible tree reduction. The new pruning method is based on calculation of an index called “valuability" for the tree nodes and data vectors. At the cross-validation training stage we can evaluate the valuability of the tree nodes analyzing their functionality on the validation data samples. The node valuability is then associated with each of the training data vectors corresponding to that node. The valuabilities can be summed at each cross-validation training stage and then used to estimate the valuabilities for each node of the tree trained on the whole dataset.

4.3 Find the best subtrees and make tree committees

The choice of a split point at a given level has strong influence on the subtree. Suboptimal split points at the root of the tree may lead to very complex trees giving low accuracy on the test data. Instead of local decisions for single features search methods may diminish this drawback. Beam search seems to be quite good solution but sometimes it is very time consuming. An alternative method used here is:

- Analyze each possible split value; choose the best n features.
- Analyze tree branches of depth 2 for the selected features; choose the best subtree.
- Repeat the procedure recursively for each leaf.

The stability of classifiers is improved by averaging or combining results of many models. Since generation of decision trees is fast (unless deep search procedure is used) it is worthwhile to use tree committees. There are two advantages: the accuracy of committee systems is higher and probabilities of alternative classes may be estimated. Weighted logical rules explaining the classification may still be found although they are less comprehensible than rules generated by a single tree.

5 RESULTS

Some results obtained with the original SSV based decision tree for several datasets from UCI repository [4] were presented in [2]. Here the results for several new datasets from the same repository are presented.

5.1 Pima Indian diabetes

The database consists of 768 vectors, described by 8 attributes and distributed in 2 classes. 500 vectors (65.1%) belong to the class "healthy" and 268 (34.9%) to class "diabetes". Using C-MLP2LN neural method a rule describing this set with 75% accuracy has been found [5]:

$$\text{IF F2} \leq 151 \wedge \text{F6} \leq 47 \text{ THEN no-diabetes ELSE diabetes}$$

With SSV criterion approach we have found a simpler rule with the same accuracy:

$$\text{IF F2} < 143.5 \text{ THEN no diabetes ELSE diabetes}$$

The comparison of SSV 10 fold crossvalidation results to other similar systems (Statlog [6] results are for 12 fold CV) is presented in table 1. Some of the best decision trees (CART and C4.5) gave less accrate results here. Although the logistic discrimination gives better results it does not give interpretation in form of logical rules.

Method	Accuracy %	Reference
Logdisc	77.7	Statlog
SSV Tree	74.8	this paper
CART	74.5	Stalog
C4.5	73.0	Stalog
Default	65.1	

Table 1: Crossvalidation results for diabetes dataset

Method	Accuracy %	Reference
3-NN	96.7	Karol Grudziński (our group)
MLP+BP	96.0	Sigillito [7]
C4.5	94.9	Hamilton [8]
FSM	92.8	Rafał Adamczak (our group) [9]
SSV Tree	92.0	this paper
DB-CART	91.3	Shang, Breiman [10]
CART	88.9	Shang, Breiman [10]

Table 2: Test ionosphere dataset results

5.2 Ionosphere

The ionosphere dataset has 200 vectors in the training set and 150 in the test set. Each data vector is described by 34 continuous attributes and belongs to one of two classes. Table 2 presents test set accuracies of different systems. This is a difficult dataset for decision trees, requiring rather complex decision borders. In this case the SSV Tree result is better than CART (also its distribution based version, DB-CART), but a bit worse than the C4.5 result.

5.3 NASA Shuttle

NASA Shuttle database is quite large: the training set consists of 43500 vectors and the test set contains 14500 entries. Each instance is described by 9 continuous attributes and is assigned to one of 7 classes. Approximately 80% of the data belongs to class 1. SSV results of various complexity can be obtained, depending on the degree of the tree pruning. 17 rules give already quite accurate description. Typical rules involve only a few features, for example the rules for class 2 are:

1. F1 < 54.5 $\wedge$ F2 > 19.5 $\wedge$ F9 > 5 $\rightarrow$ class 2
2. F7 < 15.5 $\wedge$ F7 > 6.5 $\wedge$ F5 < -16 $\rightarrow$ class 2

Method	Train	Test	Ref.
SSV, 32 rules	100.00	99.99	this paper
NewID dec. tree	100.00	99.99	[6]
FSM, 17 rules	99.98	99.97	Rafał Adamczak [9]
18 SSV rules	99.97	99.96	this paper
k-NN + feature sel.	–	99.95	Karol Grudziński (our group)
C4.5 dec. tree	99.96	99.90	[6]
k-NN	–	99.56	[6]
RBF	98.40	98.60	[6]
MLP+BP	95.50	96.57	[6]
Logistic discrimination	96.07	96.17	[6]
Linear discrimination	95.02	95.17	[6]

Table 3: Comparison of results for the NASA Shuttle dataset.

Results for different systems are compared in the table **??**. SSV results are much better than those obtained from the MLP or RBF networks (as reported in the Stalog project [**?**]) and

comparable with the results of the best decision trees which work very well for this problem. The NewID tree (descendant of the ID3 tree), which gave the best results here, has not been among the first 3 best methods for any other of the 22 datasets analyzed in the Stalog project [6]. Results of the C4.5 decision tree are already significantly worse.

5.4 Statlog Australian credit data

This dataset contains 690 cases classified in 2 classes ($+$ and $-$). Data vectors are described by 14 attributes (6 continuous and 8 discrete). In the Table 4 a comparison of 10 fold cross-validation results for the training and the test partitions are presented.

Method	Train error %	Test error %	Reference
Cal5 dec. tree	13.2	13.1	Statlog
ITrule	16.2	13.7	Statlog
k-NN,k=18, Manhat.	-	13.6	Karol Grudziński
SSV Tree	11.7	14.0	this paper
Linear Discrimination	13.9	14.1	Statlog
CART	14.5	14.5	Statlog
RBF	10.7	14.5	Statlog
Naive Bayes	13.6	15.1	Statlog
MLP	18.7	15.4	Statlog
C4.5	9.9	15.5	Statlog
Bayes tree	—	17.1	Statlog
k-NN	—	18.1	Statlog
LVQ	6.5	19.7	Statlog
Quadisc	18.5	20.7	Statlog
Default	44.0	44.0	Statlog

Table 4: 10 fold crossvalidation results for the Australian credit data

6 SUMMARY

Decision trees belong to the best classification systems [6]. The SSV criterion has proved to be quite efficient and accurate in many classification tasks, in some cases giving better results than C4.5 or CART trees. Similar “dipolar" criterion has been introduced by Bobrowski [11] for neural classificators. Decision trees based on SSV criterion have natural interpretation in form of logical rules describing the classification problem, efficiently dealing with continuous and discrete attributes. Extensions described in this paper (deeper search, treatment of unknown values, committees of trees) are tested now on a real-world data. However, for some data complex decision borders seem to be necessary. They may be provided by the similarity-based systems [12]. Combination of the two approaches may be the best strategy for complex problems.

REFERENCES

[1] W. Duch, R. Adamczak and K. Grąbczewski, Extraction of logical rules from neural networks, *Neural Processing Letters* 7: 211-219, 1998

[2] K. Grąbczewski, W. Duch, A general purpose separability criterion for classification systems, *4-th Conference on Neural Networks and Their Applications*, Zakopane, May 1999, pp. 203-208

[3] R. Andrews, J. Diederich, A.B. Tickle, A survey and critique of techniques for extracting rules from trained artificial neural networks, *Knowledge-Based Systems* 8 (1995) 373–389.

[4] C.J. Mertz, P.M. Murphy, UCI repository of machine learning databases, available at the address
http://www.ics.uci.edu/pub/machine-learning-databases;

[5] W. Duch, R. Adamczak and K. Grąbczewski, Methodology of extraction, optimization and application of crisp and fuzzy logical rules. *IEEE Transactions on Neural Networks* (in print, 2000)

[6] D. Michie, D. J. Spiegelhalter, and C. C. Taylor, editors. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, New York, 1994.

[7] V. G. Sigillito, S. P. Wing, L. V. Hutton, K. B. Baker, Classification of radar returns from the ionosphere using neural networks. *Johns Hopkins APL Technical Digest* 10 (1989) 262-266.

[8] H.J. Hamilton, N. Shan, N. Cercone, RIAC: a rule induction algorithm based on approximate classification, Tech. Rep. CS 96-06, Regina University 1996.

[9] W. Duch, G.H.F. Diercksen, Feature Space Mapping as a universal adaptive system, *Computer Physics Communications* 87 (1995) 341–371

[10] ??N. Shang, L. Breiman, *ICONIP'96*, p.133

[11] ??L. Bobrowski, M. Kretowska, M. Kretowski, Design of neural classifying networks by using dipolar criterions. *3rd Conf. on Neural Networks and Their Applications*, Kule, Poland, 1997.

[12] Duch W. (1998) A framework for similarity-based classification methods, *Intelligent Information Systems VII*, Malbork, Poland, June 1998, pp. 288-291