

# FEATURE SELECTION BASED ON INFORMATION THEORY, CONSISTENCY AND SEPARABILITY INDICES.

Włodzisław Duch<sup>1</sup>, Krzysztof Grąbczewski<sup>1</sup>, Tomasz Winiarski<sup>1</sup>, Jacek Biesiada<sup>2</sup>, Adam Kachel<sup>2</sup>

<sup>1</sup>Dept. of Informatics, Nicholas Copernicus University, Grudziądzka 5, 87-100 Toruń, Poland.  
<http://www.phys.uni.torun.pl/kmk>

<sup>2</sup>The Silesian University of Technology, Dept. of Electrotechnology, Division of Computer Methods, ul. Krasińskiego 8, 40-019 Katowice, Poland.

## ABSTRACT

Two new feature selection methods are introduced, the first based on separability criterion, the second on consistency index that includes interactions between the selected subsets of features. Comparison of accuracy was made against information-theory based selection methods on several datasets training neurofuzzy and nearest neighbor methods on various subsets of selected features. Methods based on separability seem to be most promising.

## 1. INTRODUCTION

Challenging applications of data mining methods in bioinformatics, chemistry and commercial domains demand inexpensive methods for filtering features that should be used for modeling data. In bioinformatics a very large ( $\sim 10^4 - 10^5$ ) number of features are associated with gene activity (over 30.000 genes in humans and even more in some plants), while properties of proteins may be described by more than 100.000 features. All these features may be important for some problems, but for a given task only a small subset of features is relevant. In commercial applications the situation is similar. Therefore computationally inexpensive methods of filtering features are urgently needed. Filtering features means either ranking or selecting subsets of features. Methods of feature ranking treat each feature in an independent way, trying to determine how useful they may be. Methods of feature selection try to find a subset of features that should lead to the best results. Exhaustive search to evaluate performance with all possible subsets of features is the golden standard here, but the number of all subsets for  $n$  features is  $2^n$ , making such search unrealistic for larger  $n$ . Finding useful subsets of features is equivalent to assigning binary weights to inputs.

Filtering methods are inexpensive and independent of the final system used for data modeling. Computationally

more demanding, but sometimes more accurate, “wrapper methods” [1] require evaluation of each potentially useful subset of features by computational intelligence (CI) systems that are used on a given data. The name “wrapper” is used also for a class of parameter adaptation methods calling a “black box” classifier to evaluate results of parameter changes. Two essential components of such methods are parameter search and evaluation of results requiring test runs. Computational complexity of filtering methods is usually much lower than in the case of wrapper approach. Feature selection methods may also be based on specific properties of classification methods (cf. backpropagation with regularization [2, 3]).

Feature filtering methods frequently are based on information theoretical methods. If a feature carries no information in respect to the task performed and to other features that are already selected, it may be safely filtered out. Several methods based on information theory and other approaches are presented in the next section. Although quite popular, they have some disadvantages that led us to development of two new methods, based on the separability criterion and consistency index. These methods are described in the third section. Numerical comparisons on two well known datasets are presented in section four. The paper is finished with a number of conclusions.

## 2. INFORMATION THEORY AND OTHER FILTERS

**Ranking of features** determines the importance of individual features, neglecting possible feature interactions. Ranking methods may use correlation coefficients, may be based on mutual information between features and classes, or on some functions of classifier’s outputs.

Consider the joint probability  $p(C_i, f), i = 1 \dots K$  of finding the feature value  $X_j = f$  for vectors  $\mathbf{X}$  that belong to some class  $C_k$ . The amount of information contained in this joint distribution, summed over all classes, gives an

---

Support by the Polish Committee for Scientific Research, grant 8 T11C 006 19, is gratefully acknowledged.

estimation of the importance of the feature:

$$I(C, X_j) = - \sum_{i=1}^K \int p(C_i, f) \lg_2 p(C_i, f) df \quad (1)$$

$$\approx - \sum_{k=1}^{M_j} p(r_k(f)) \sum_{i=1}^K p(C_i, r_k(f)) \lg_2 p(C_i, r_k(f))$$

where  $r_k(f)$  is a partition of the continuous feature range into  $M_j$  intervals (a subset of discrete feature values), and  $p(r_k(f))$  is the probability of finding vectors with  $X_j = f \in r_k(f)$ . Low values of  $I(C, X_j)$  indicate that vectors from single class dominate in some intervals, making the feature more valuable for prediction.

Information gained by considering the joint distribution of classes and  $X_j$  feature values is a difference between  $I(C) + I(X_j)$  and  $I(C, X_j)$ :

$$IG(X_j) = -I(C, X_j) - \sum_{i=1}^K p(C_i) \lg_2 p(C_i)$$

$$- \sum_{k=1}^{M_j} p(r_k(f)) \lg_2 p(r_k(f)) \quad (2)$$

A feature is more important if its information gain is larger. Various modifications of the information gain have been considered in the literature on decision trees (cf. [4]), such as the gain ratio  $IGR(X_j) = IG(X_j)/I(X_j)$  or the Mantaras distance  $1 - IG(X_j)/I(C, X_j)$  (cf. [5]). Another ratio  $IGN(X_j) = IG(X_j)/I(C)$ , called also "an asymmetric dependency coefficient" is advocated in [6].

Mutual information between feature  $f$  and classes:

$$M_I(C, f) = \sum_{i=1}^K \sum_{k=1}^{M_f} p(C_i \wedge r_k(f)) \lg_2 \frac{p(C_i \wedge r_k(f))}{p(C_i) \cdot p(r_k(f))}$$

where  $r_1(f), r_2(f), \dots, r_N(f)$  is a partition of the range of  $f$  values into bins and  $p(C_i \wedge r_k(f))$  is the probability that vector  $X$  from class  $C_i$  has feature  $f$  in the bin  $r_k$ . The sum runs over all  $M_f$  bins and all  $K$  classes. Mutual information is equal to the Kullback-Leibler divergence between the joint and the product probability distribution, i.e.  $M_I(P_X, P_Y) = D_{KL}(P_{XY}|P_X P_Y)$ .

**Selection of features** by taking those with the highest ranking does not include the fact that features may be highly redundant. Interactions between features should be taken into account. Mutual information between two features  $f, s$  is defined as:

$$M_I(f, s) = \sum_{k,j=1}^K p(r_k(f) \wedge r_j(s)) \lg_2 \frac{p(r_k(f) \wedge r_j(s))}{p(r_k(f)) \cdot p(r_j(s))}$$

The algorithm for finding the best subset of  $k$  features due to Battiti [7] computes the mutual class-feature information  $M_I(C, f)$  for every feature  $f \in F$  (initially the set of all features) and the set of classes  $C = \{C_1, \dots, C_K\}$ .

The feature  $f$  that maximizes  $M_I(C, f)$  is found (like in ranking) and moved from the set  $F$  to the set in  $S$  (initially an empty set). Mutual information  $M_I(f, s)$  is calculated between features  $f \in F$  and  $s \in S$  and a new feature is chosen, one that maximizes the difference  $M_I(C, f) - \beta \sum_{s \in S} M_I(f, s)$  where  $\beta$  is a parameter in the interval  $[0.5, 1]$ . Smaller values of  $\beta$  stress the importance of high mutual information between the feature and set of classes; large values stress more mutual information with the features already included in the set  $S$  [7].

Correlation-based feature selection (CFS) is based on a similar principle: features should be highly correlated with the class but not with each other. Correlation between features may be estimated using entropy distance measure  $D_I(X, Y) = I(X|Y) + I(Y|X)$  or symmetrical uncertainty coefficient  $U(X, Y) = 1 - D_I(X, Y)/(I(X) + I(Y)) \in [0, 1]$ . In numerical tests CFS comes close to the wrapper approach for the Naive Bayes method [8].

Features are also selected during construction of decision trees, with the most important features near the root of the tree, and the least important near the bottom. Pruning leaves only the most important features in the tree. Information theory criteria are used in most popular trees, such as C4.5 [4]. However, Shannon information is not the only, and perhaps not even the most natural, measure of the similarity of probability distributions.

Consistency-based index is the sum, over all bins (partitions), of the number of vectors in the majority class in a given bin, divided by the number of all vectors. This index estimates "class purity", and works best with methods that partition each feature range into bins that contain natural grouping of data (cf. review in [8]).

### 3. NEW METHODS: DECISION TREE AND INTERACTIVE CONSISTENCY INDEX

The Separability Split Value (SSV) criterion [9] selects features that give the largest gain of separability index, equal to the number of correctly separated vectors from different classes. The inexpensive best-first (BFS) search approach is used to build decision tree. The tree node split values, calculated by the maximization of the SSV criterion, provide automatic discretization of continuous intervals. Information-theoretic approaches usually require separate discretization step to determine  $r_k(f)$  intervals.

The SSV tree may place a given feature at different levels and may use a single feature several times. Feature selection has been done here by increasing the degree of pruning [9] and noting the minimal number of tree nodes for which a given feature appears. The most important feature is placed at the highest level and has two nodes (not counting the root). This method includes interactions among feature subsets. The tree may also be used to rank features eval-

uating the classification results that one may obtain with a single feature only, but since the tree algorithm is quite fast (at least in the best-first search mode) there is no reason to use such ranking.

The second method presented here, the Interactive Consistency Index (ICI) method, starts from computing the  $IC(f)$  indices for all features  $f$ :

$$IC(f) = \frac{1}{M_f} \sum_{k=1}^{M_f} \max_C p(r_k(f))p(C_i, r_k(f)) \quad (3)$$

Partitions  $r_k(f)$  may be created by standard techniques used for histogram partitioning (equiwidth, equidepth, least variance etc,) or by using the SSV criterion [9] on the single feature  $f$ . Such partitioning guarantees that for data that is separable using feature  $f$  only, the index  $IC(f) = 1$ . In the worst case if feature  $f$  used separately from all others is useless, for  $K$  classes the index may be  $IC(f) = 1/K$ . Rescaling it by  $(K \cdot IC(f) - 1)/(K - 1)$  gives an index with values in  $[0, 1]$  that may be used for ranking.

The ICI index is useful for feature ranking. Feature selection requires evaluation of subset of features. Let  $S = \{s\}$  be the current subset of  $M$  features. New feature  $f$  to be added to the subset should improve the  $IC(S + \{f\})$  index value, but it should also be different than features already included in  $S$ . In context of the consistency index an appropriate measure of difference between two features  $s, f$  is given by the distance function:

$$DC(s, f) = \sum_{i,j} \min_C [p_C(s_i, f_j) - p_C(s_i)p_C(f_j)] \quad (4)$$

where  $DC(s, f) \in [0, 1]$ . The ICI algorithm starts with empty  $S$  and selects the feature with the highest ICI index. Selection of the next feature  $f$  should maximize the ICI index value calculated over partition of  $S + f$ . This method includes interactions between features selected, avoiding redundant features. Hashing techniques have been used to avoid high computational costs of summing over empty  $S + f$  areas.

#### 4. NUMERICAL EXPERIMENTS

The new methods, ICI and SSV feature selection, have been tested against 17 other methods of feature ranking or selection. Due to the space restrictions we report here only results obtained with information gain (IGn) ranking [6] and Battiti selection method (BA) [7] on two datasets [10]: Monk-1 artificial data and hypothyroid problems. Monk-1 data has only 6 features of which 5, 1, 2 are used to form a rule determining the class. 124 training cases are used for feature selection, and 432 cases are used for testing. Since all features are symbolic discretization is not needed.

In each case subsets of features have been generated analyzing the training set using the normalized information gain [6], Battiti's information gain with feature interaction [7], and using two new methods presented here, the SSV separability criterion and the ICI method. An additional ranking has been provided with  $k$  nearest neighbor method using SBL program [11] as a wrapper, with feature dropping method to determine feature importance.  $k$ NN with optimization of  $k$  and similarity measure, the Feature Space Mapping (FSM) neurofuzzy system [12], and several statistical and neural methods (not reported here due to the lack of space) were used to calculate accuracy on the test set using the feature sets with growing number of features. The best feature selection method should reach the peak accuracy for the smallest number of features.

Both  $k$ NN ( $k=1$ , Canberra distance) and FSM achieve 100% on the Monk-1 data using the 3 important features, but not all methods found them. Our reference method based on feature dropping in the  $k$ NN gave feature number 1 as a clear winner. All other methods start correctly from feature 5, achieving 25% higher accuracy with single feature (Fig. 1), but dropping below the SBL ranking for two features. Same ranking was found using the SSV criterion and the beam search method for tree construction. Rankings based on information gain (in several versions that we have tried) failed to find the 3 important features correctly. Battiti's approach (BA in Table 1 and 2, and Fig. 1 and 2) after correctly recognizing the importance of feature 5 and 1 failed for all recommended  $\beta$  values to recognize the importance of feature 2.

The **hypothyroid dataset** has been created from real medical screening tests for hypothyroid problems [10]. Since most people were healthy 92.7% of test cases belong to the normal group, and 7.3% of cases belonging to the primary hypothyroid or compensated hypothyroid group. Hypothyroid data offers a good mixture of nominal (15) and numerical (6) features. A total of 3772 cases are given for training (results from one year) and 3428 cases for testing (results from the next year). We have provided comparison with results of other classifiers elsewhere [2], here the data is used only for evaluation of feature selection.

This is a much more difficult case due to the strong correlations between features. We have used both equiwidth and equidepth discretization of continuous features, but the results were similar. Dropping features in SBL gives very good results, although SSV finds a subset of 3 features (17, 21, 3) that give higher accuracy with both  $k$ NN and FSM methods. Overall SSV finds very good subsets of features leading to best results for small number of features. IGn selects all important features but does not include any feature interaction; as a result high accuracy is achieved with at least 5 features. On the other hand adding feature interactions in the Battiti method, even with small  $\beta$ , leaves out

| Method           | Most – Least Important |   |   |   |   |   |
|------------------|------------------------|---|---|---|---|---|
| SBL              | 1                      | 2 | 5 | 3 | 6 | 4 |
| BA $\beta = 0.7$ | 5                      | 1 | 3 | 4 | 6 | 2 |
| ICI ranking      | 5                      | 1 | 2 | 3 | 4 | 6 |
| ICI selection    | 5                      | 1 | 2 | 3 | 4 | 6 |
| IGn              | 5                      | 1 | 4 | 2 | 3 | 6 |
| SSV              | 5                      | 1 | 2 | 3 | 4 | 6 |

Table 1: Results of feature ranking on the Monk 1 data using six methods.

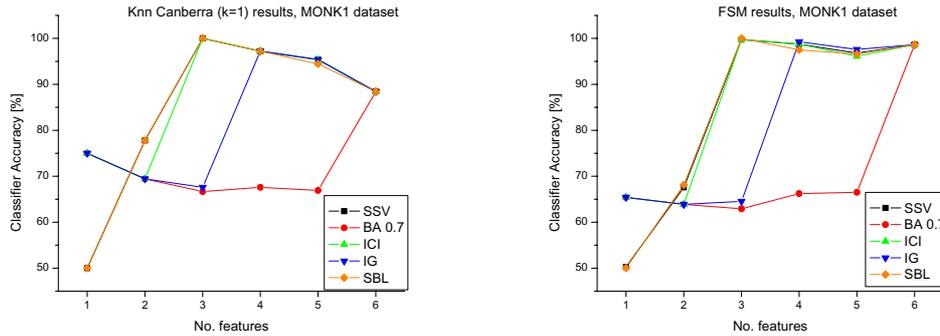


Figure 1: Monk 1 artificial data, results obtained on subsets of features with ranking by 4 methods. Left figure: results from kNN, Canberra distance, k=1; right figure: results from FSM neurofuzzy network.

| Method           | Most Important – Least Important |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|------------------|----------------------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| SBL              | 17                               | 3  | 8  | 19 | 21 | 5  | 15 | 7  | 13 | 20 | 12 | 4  | 6  | 9  | 10 | 18 | 16 | 14 | 11 | 1  | 2  |
| BA $\beta = 0.5$ | 21                               | 17 | 13 | 7  | 15 | 12 | 9  | 5  | 8  | 4  | 6  | 16 | 10 | 14 | 2  | 11 | 3  | 18 | 1  | 20 | 19 |
| IGn              | 17                               | 21 | 19 | 18 | 3  | 7  | 13 | 10 | 8  | 15 | 6  | 16 | 5  | 4  | 20 | 12 | 1  | 2  | 11 | 9  | 14 |
| ICI ranking      | 1                                | 20 | 18 | 19 | 21 | 17 | 15 | 13 | 7  | 5  | 3  | 8  | 16 | 12 | 4  | 2  | 11 | 6  | 14 | 9  | 10 |
| ICI selection    | 1                                | 19 | 20 | 18 | 2  | 21 | 3  | 11 | 16 | 10 | 6  | 14 | 8  | 9  | 4  | 12 | 13 | 17 | 5  | 7  | 15 |
| SSV BFS          | 17                               | 21 | 3  | 19 | 18 | 8  | 1  | 20 | 12 | 13 | 15 | 16 | 14 | 11 | 10 | 9  | 7  | 6  | 5  | 3  | 2  |

Table 2: Results of feature ranking on the hypothyroid dataset; see description in text.

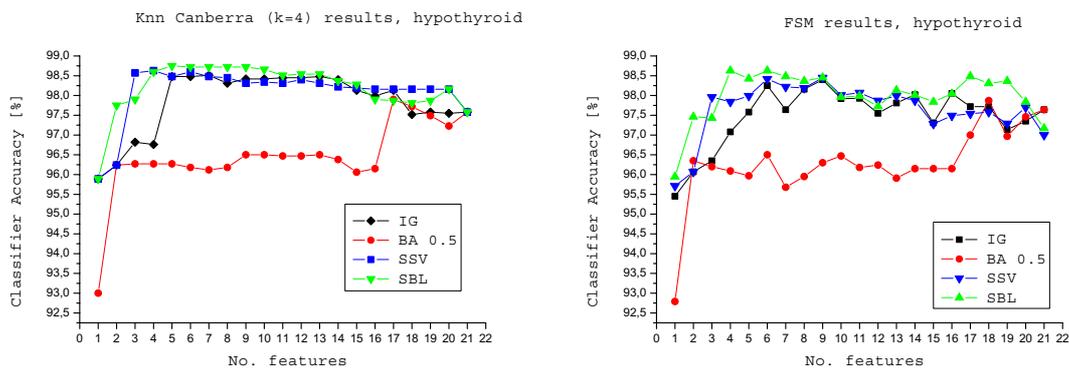


Figure 2: Hypothyroid data, results obtained on subsets of features created by 4 methods. Left figure: results from kNN, Canberra distance, k=4; right figure: results from FSM neurofuzzy network.

important features 3, 18-20, leading to poor accuracy with sets smaller than 17 features. BA has left out some important features that had large mutual information with features 17 and 21, selected as the first two features. ICI ranking and selection incorrectly start from feature number 1. This seems to be a result of naive discretization. IGn behaves correctly, climbing slowly and reaching a plateau and declining when irrelevant features are added. The variance of the FSM results is rather high (few points have been averaged over 10 runs), but that does not change the overall character of curve in Fig. 2.

The best kNN result ( $k=4$ , Canberra) is achieved with 5 features, 17,3,8,19,21, reaching 98.75% on the test set, significantly higher than 97.58% with all features. This seems to be the best kNN result achieved so far on this dataset.

## 5. CONCLUSIONS

Two new feature selection methods have been introduced and compared with a wrapper method, a ranking method based on normalized information gain and selection method based on mutual information that includes correlation among features. Only a few results obtained with several feature selection schemes and classification methods have been presented here. Several conclusions may be drawn from this and our more extensive studies:

- 1) Results of ranking algorithms depend strongly on discretization procedures for continuous features; dependence on the choice of the number of intervals for calculation of information may partially be removed if Gaussian overlapping windows are used instead of intervals, but better ranking methods should be based on separability or entropy-based discretization criteria.
- 2) Decision trees may provide very good selection and ranking; in particular SSV tree consistently selected small subsets of most important features, sometimes giving better results than wrapper methods.
- 3) Selection of relevant feature subsets is more difficult than feature ranking; best-first search is not always sufficient. A good strategy is to use ranking method to find a subset of features and then to use selection method to find a smaller set of features.
- 4) Selection methods that include correlation among features may find smaller subsets of features, but may also miss important features.
- 5) Classification method may benefit to a different degree from different selection methods, therefore finding the best selection method for a given classification method is an open question.
- 6) Methods based on consistency indices may outperform information theory methods but are sensitive to discretization.
- 7) In multiclass problems a better feature selection strat-

egy is to select features useful for discrimination of a single class from the rest; this is especially important for such datasets as thyroid, with 92% of cases in one class.

8) Aggregation (for example by linear combination) of features may be easier than selection.

## 6. REFERENCES

- [1] R. Kohavi, Wrappers for performance enhancement and oblivious decision graphs. PhD thesis, Dept. of Computer Science, Stanford University (1995)
- [2] Duch W, Adamczak R, Grąbczewski K, A new methodology of extraction, optimization and application of crisp and fuzzy logical rules. *IEEE Transactions on Neural Networks* **12**, 277-306 (2001)
- [3] R. Setiono, H. Liu, Improving Backpropagation learning with feature selection. *Applied Intelligence* **6**, 129-139 (1996)
- [4] J.R. Quinlan, C4.5: Programs for machine learning. San Mateo, Morgan Kaufman (1993)
- [5] L. R. de Mantaras, A distance-based attribute selection measure for decision tree induction. *Machine Learning* **6**, 81-92 (1991)
- [6] D.V. Sridhar, E.B. Bartlett, R.C. Seagrave, Information theoretic subset selection. *Computers in Chemical Engineering* **22**, 613-626 (1998)
- [7] R. Battiti, Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks* **5**, 537-550 (1991)
- [8] M.A. Hall, Correlation based feature selection for machine learning. PhD thesis, Dept. of Comp. Science, Univ. of Waikato, Hamilton, New Zealand (1998)
- [9] K. Grąbczewski, W. Duch, The Separability of Split Value Criterion, *5th Conf. on Neural Networks and Soft Computing*, Zakopane, Poland, pp. 201-208 (2000)
- [10] C.L. Blake, C.J. Merz, UCI Repository of machine learning databases (2001) <http://www.ics.uci.edu/mllearn/MLRepository.html>.
- [11] W. Duch, K. Grudziński, The weighted k-NN method with selection of features and its neural realization, *4th Conf. on Neural Networks and Their Applications*, Zakopane, May 1999, pp. 191-196 (1999)
- [12] W. Duch, G.H.F. Diercksen, Feature Space Mapping as a universal adaptive system, *Computer Physics Communications* **87**, 341-371 (1995)