Feature Selection for High-Dimensional Data: A Pearson Redundancy Based Filter

Jacek Biesiada[1] and Włodzisław Duch[2]

[1] Division of Computer Methods, Dept. of Electrotechnology, The Silesian University of Technology, Katowice, Poland
[2] Dept. of Informatics, Nicolaus Copernicus University, Toruń, Poland;
Contact: Jacek.Biesiada@polsl.pl; Google: Duch.

Summary. An algorithm for filtering information based on the Pearson χ^2 test approach has been implemented and tested on feature selection. This test is frequently used in biomedical data analysis and should be used only for nominal (discretized) features. This algorithm has only one parameter, statistical confidence level that two distributions are identical. Empirical comparisons with four other state-of-the-art features selection algorithms (FCBF, CorrSF, ReliefF and ConnSF) are very encouraging.

1 Introduction

For large highly dimensional datasets feature ranking and feature selection algorithms are usually of the filter type [1]. In the simplest case feature filter is a function (such as correlation or information content) returning a relevance index $J(\mathcal{S}|\mathcal{D}, C)$ that estimates, given the data $\mathcal{D}$, how relevant a given feature subset $\mathcal{S}$ is for the task C(usually classification or approximation of data). An algorithmic procedure, such as building a decision tree or finding nearest neighbors, may also be used to estimate this index. The $J(\mathcal{S}|\mathcal{D}, C)$ filter index is calculated directly from data, without any reference to the results of programs that are used for final data analysis. Since the data $\mathcal{D}$ and the task C are usually fixed and only the subsets $\mathcal{S}$ varies an abbreviated form $J(\mathcal{S})$ will be used.

Relevance indices computed for individual features $X_i, i = 1 \ldots N$ establish a ranking order $J(X_{i_1}) \leq J(X_{i_2}) \cdots \leq J(X_{i_N})$. Those features which have the lowest ranks may be filtered out. For independent features this may be sufficient, but if features are correlated many of them may be redundant. Moreover, for some data distributions the best pair of features may not even include a single best feature [2]! Thus ranking does not guarantee that the largest subset of important features will be found. Methods that search for the best subset of features may also use filters to evaluate the usefulness of subsets of features.

The thresholds for feature rejection may be set either for relevance indices, or by evaluation of reduced dimensionality results. Features are ranked by the filter, but

how many are finally taken may be determined using adaptive system as a wrapper. Evaluation of the adaptive system performance (usually crossvalidation tests) are done only for a few pre-selected feature sets, but still this "frapper" (filter-wrapper) approach may be rather costly if many feature subsets are evaluated. What is needed is a simple filter method that may be applied to a large datasets ranking and removing redundant features, parameterized in statistically well-established way. Such an approach is described in this paper. Similar filter for reducting redundant continuous features based on Kolmogoros-Smirnov test has been proposed in [3].

In the next section relevance index based on Pearson's χ^2 test to estimate correlation between the distribution of feature values and the class labels is introduced. Section 3 compares it with four state-of-the-art feature selection algorithms using three bioinformatics datasets.

2 Relevance indices and algorithms

2.1 Correlation-Based Measures

For feature X with values x and classes C with values c, where X, C are treated as random variables, Pearson's linear correlation coefficient is defined as [4]:

$$\varrho(X,C) = \frac{E(XC) - E(X)E(C)}{\sqrt{\sigma^2(X)\sigma^2(C)}} = \frac{\sum_i (x_i - \bar{x}_i)(c_i - \bar{c}_i)}{\sqrt{\sum_i (x_i - \bar{x}_i)^2 \sum_j (c_j - \bar{c}_j)^2}}. \quad (1)$$

$\varrho(X,C) = \pm 1$ if X and C are linearly dependent and zero if they are completely uncorrelated. Probability that two variables are correlated is estimated using the error function [4] $\mathcal{P}(X \sim C) = \text{erf}\left(|\varrho(X,C)|\sqrt{N/2}\right)$. The feature list ordered by decreasing values of the $\mathcal{P}(X \sim C)$ may serve as feature ranking. An alternative approach is to use χ^2 statistics, but in both cases for large number of samples probability $\mathcal{P}(X \sim C)$ is so close to 1 that ranking becomes impossible due to the finite numerical accuracy of computations. With $N = 1000$ samples coefficients as small as $\varrho(X,C) \approx 0.02$ give correlation probabilities $\mathcal{P}(X \sim C) \approx 0.5$. The $\varrho(X,C)$ or χ^2 thresholds for the significance of a given feature may therefore be taken from a large interval corresponding to almost the same probabilities of correlation. Non-parametric, or Spearman's rank correlation coefficients is useful for ordinal data types.

Information theory is frequently used to define relevance indices. The Shannon information for distribution of feature values and classes is:

$$H(X) = -\sum_i \mathcal{P}(x_i) \log \mathcal{P}(x_i); \qquad H(C) = -\sum_i \mathcal{P}(c_i) \log \mathcal{P}(c_i) \quad (2)$$

and the joint Shannon entropy is:

$$H(X,C) = -\sum_{i,j} \mathcal{P}(x_i, c_j) \log \mathcal{P}(x_i, c_j) \quad (3)$$

Information filtering is frequently based on mutual information (MI):

$$MI(X,C) = H(X) + H(C) - H(X,C) \tag{4}$$

or on the Symmetrical Uncertainty Coefficient (SU) with similar properties:

$$SU(X,C) = 2\frac{MI(X,C)}{H(X)+H(C)} \tag{5}$$

If a group of k features $\mathbf{X}_k$ has already been selected, correlation coefficient may be used to estimate correlation between this group and the class, including inter-correlations between the features. Denoting the average correlation coefficient between these features and classes as $r_{kc} = \bar{\varrho}(\mathbf{X}_k, C)$ and the average between different features as $r_{kk} = \bar{\varrho}(\mathbf{X}_k, \mathbf{X}_k)$ the relevance of the feature subset is defined as:

$$J(\mathbf{X}_k, C) = \frac{k r_{kc}}{\sqrt{k + (k-1) r_{kk}}}. \tag{6}$$

This formula has been used in the Correlation-based Feature Selection (CFS) algorithm [5] adding (forward selection) or deleting (backward selection) one feature at a time. A definition of predominant correlation proposed by Yu and Liu [6] for Fast Correlation-Based Filter (FCBF) includes correlations beetwen feature and classes and between pairs of features. The FCBF algorithm does a typical ranking using SU coefficient (Eq. 5) to determine class-feature relevance, setting some threshold value $SU \geq \delta$ or number of features $\lfloor n \log(n) \rfloor$ to determine how many features should be taken. In the second part redundant features are removed by defining the "predominant features".

Selection method called ConnSF, based on inconsistency measure, has been proposed by Dash *et al.* [7] and will be used for comparison in Sec. 3. Two identical input vectors are inconsistent if they have identical class labels (a similar concept is used in rough set theory). Intuitively it is clear that inconsistency grows when the number of features is reduced and that feature subsets that lead to high inconsistency are not useful. If there are n samples in the dataset with identical feature values x_i, and n_k among them belong to class k then the inconsistency count is defined as $n - \max_k c_k$. The total inconsistency count for a feature subset is the sum of all inconsistency counts for all data vectors.

A different way to find feature subsets is used in the Relief algorithm [8]. This algorithm estimates weights of features according to how well their values distinguish between data vectors that are near to each other. For a randomly selected vector X from a data set $\mathcal{S}$ with k features Relief searches the dataset for its two nearest neighbors: the nearest hit H from the same class and the nearest miss M from another class. For feature x and two input vectors X, X' the contribution to the weight W_x is proportional to the $D(x, X, X') = 1 - \delta(X(x), X'(x))$ for binary or nominal features, and $D(x, X, X') = |X(x) - X'(x)|$ for continuous features. The process is repeated m times, where m is a user defined parameter. Normalization with m in calculation of W_x guarantees that all weights are in the $[-1, 1]$ interval. In Sec. 3 an extension of this agorithm for multiclass problems, called ReliefF [8] has been used.

2.2 Pearson's Redundancy Based Filters.

The Pearson χ^2 test measures the difference between the probability distribution of two binned random variables. If a feature is redundant than the hypothesis that its distribution is equal to already selected feature should have high probability. n independent observations of two random variables X, X' are given in the training data, where for the Pearson χ^2 test to be valid n should be more than 100. The test for X, X' feature redundancy proceeds as follows:

- Frequencies f_i, f'_i of occurrences of feature values in each bin are recorded (counting unique feature values).
- Based on the frequency counts emiprical probability distributions F_i and F'_i are constructed and $\chi^2(X, X')$ matrix is constructed:

$$\chi^2(X, X') = \sum_{i=1}^{k} \frac{(F_i - F'_i)^2}{F'_i} \tag{7}$$

A large value of χ^2 or a different number of unique feature values indicates that features are not redundant. When p-value $p(\chi^2) > \alpha$ then the two distributions are equivalent with α significance level, and thus one of the features is redundant. The best p-value could be estimated indepedently for each classifier using crossvalidation techniques. Below several estimates for different values of α are made to find the optimal value for each classification method. This represents the frapper approach of using filter for ranking and adding wrapper in the final determination of the number of selected features.

Pearson's Redundancy Based Filter (PRBF) algorithm is presented in Fig. 1 First, the relevance is determined using the symmetrical uncertainty (other relevance criteria may also be used), and then χ^2 test is applied to remove redundancy.

Algorithm PRBF:
Relevance analysis
1. Calculate $SU(X, C)$ relevance indices and create an ordered list $\mathcal{S}$ of features according to the decreasing value of their relevance.
Redundancy analysis
2. Take as X the first feature from the $\mathcal{S}$ list
3. Find and remove all features for which X is approximately equivalent according to the Pearson χ^2 test
4. Set the next remaining feature in the list as X and repeat step 3 for all remaining features in the $\mathcal{S}$ list.

Fig. 1. A two-step Pearson's Redundancy Based Filter (PRBF) algorithm.

3 Empirical Studies.

To evaluate the performance of the PCBF algorithm both artificial and real datasets have been used with a number of classification methods. Two artificial datasets,

Gauss4, and Gauss8, have been used in our previous study [9]. Gauss4 is based on sampling from 4 Gaussian functions with unit dispersion in 4 dimensions, each cluster representing a separate class. The first function is centered at $(0, 0, 0, 0)$, the next at $(1, 1/2, 1/3, 1/4)$, $(2, 1, 2/3, 1/2)$, and $(3, 3/2, 3, 3/4)$, respectively. The dataset contains 4000 vectors, 1000 per each class. In this case the ideal ranking should give the following order: $X_1 > X_2 > X_3 > X_4$.

Gauss8 used here is an extension of Gauss4, adding 4 additional features that are approximately linearly dependent $X_{i+4} = 2X_i + \epsilon$, where ϵ is a uniform noise with a unit variance. In this case the ideal ranking should give the following order: $X_1 > X_5 > X_2 > X_6 > X_3 > X_7 > X_4 > X_8$ and the selection should reject all 4 linearly dependent features as redundant. The PRBF and the ConnSF [7] algorithms had no problem with this task, but FCBF [6] selected only 3 features, CorrSF [5] selected only first two, and ReliefF [8] left only feature 1 and 5, giving them both the same weight 0.154 (for features 2 and 6 the weight was 0.060, dropping to 0.024 for feature 3, 6 and to 0.017 for features 4, 8).

Title	Selected features					
	Full set	FCBF	CorrSF	ReliefF	ConnSF	PRBF
Features	1 to 8	1+2+3	1+2+5	1+5	1 to 4	1 to 4
C4.5	78.85 ± 0.36	79.21 ± 0.29	78.64 ± 0.31	76.15 ± 0.09	78.85 ± 0.36	78.85 ± 0.36
NBC	82.07 ± 0.07	81.57 ± 0.08	80.25 ± 0.07	76.98 ± 0.06	82.08 ± 0.07	82.07 ± 0.07
1NN	73.48 ± 0.25	73.57 ± 0.22	71.33 ± 0.25	68.19 ± 0.34	73.48 ± 0.25	73.48 ± 0.25
SVM	81.97 ± 0.08	81.54 ± 0.10	80.77 ± 0.07	76.98 ± 0.07	81.88 ± 0.08	81.87 ± 0.09
Average	79.09 ± 0.19	78.97 ± 0.17	77.75 ± 0.18	74.57 ± 0.14	79.07 ± 0.19	79.07 ± 0.20

Table 1. Accuracy of 4 classifiers on selected subsets of features for the Gauss8 dataset.

In Table 1 results of Naive Bayes Classifier (NBC) (Weka implementation, [10]), the nearest neighbor algorithm (1NN) with Euclidean distance function, C4.5 tree [12] and the Support Vector Machine with a linear kernel are given (Weka and SVM, Ghostminer 3.0 implementation[3]).

Title	Features	Instances	Classes
Lung-cancer (Lung)	58	32	3
Promoters	59	106	2
Splice	62	3190	3

Table 2. Summary of the datasets used in empirical studies.

For the initial comparison on real data three biomedical datasets from the UCI Machine Learning Repository [11] were used. A summary of all datasets is presented in Table 2. They have rather modest number of nominal features and range from 32 to 3190 samples. Lungs dataset is extremely small and 5 out of 32 instances containing

[3] http://www.fqspl.com.pl/ghostminer/

missing values have been removed. The purpose is to see the influence of the number of samples on the quality of results for similar number of nominal features.

For each data set all five feature selection algorithms are compared (FCBF [6], CorrSF [5], ReliefF [8], ConnSF [7], and PRBF) and the number of features selected by each algorithm is given. 5 neighbors, 30 instances and treshold 0.1 were used for ReliefF, as suggested by Robnik-Sikonia and Kononenko [8]. For CorrSF and ConnSF forward search strategy has been used, and for FCBF, ReliefF, and the PRBF forward search strategy based on ranking.

Dataset	Selected features					
	Full set	FCBF	CorrSF	ReliefF	ConnSF	PRBF
Lung-cancer	58	6	7	11	**4**	*12*
Splice	62	22	**6**	*24*	10	19
Promoters	59	6	**4**	*12*	**4**	6
Average	59.6	11.3	**5.6**	*15.6*	6	12.2

Table 3. The number of selected features for each algorithm; bold face – lowest number, italics – highest number.

In Table 4 results of Naive Bayes Classifier (NBC) (Weka implementation, [10]), the nearest neighbor algorithm (1NN) with Euclidean distance function, C4.5 tree [12] and the Support Vector Machine with a linear kernel and $C = 1$ (estimated to be close to optimal value for these datasets) are collected. The overall average balanced accuracy (accuracy for each class, averaged over all classes) and the standard deviation obtained from averaging 20 repetitions of 10-fold cross-validation calculations with different initializations is reported in Tables below. For datasets with significant differences in *a priori* class distributions balanced accuracy is more sensitive measure than the overall accuracy.

In Table 5 classification results for various sigificance levels are presented. Surprisingly the best results have been obtained for a very small level $\alpha = 0.001$, removing the largest number of redundant features.

4 Conclusion

A new algorithm for finding non-redundant binned feature subsets based on the Pearson χ^2 test has been introduced. PRBF has only one parameter, statistical significance or the probability that the hypothesis that distributions of two features is equivalent is true. In the first step SU indices Eq. 5 have been used for ranking, and in the second step redundant features are removed in an unsupervised way, because durning reduction of redundant features infromation about the classes is not used. Our initial tests are encouraging: on the artificial data perfect ranking has been recreated and redundant features rejected, while on the real data, with rather modest number of features selected results are frequently the best, or close to the best, comparing with four state-of-the-art feature selection algorithms. The new algorithm seems to work especially well with the linear SVM classifier. Computational demands of PRBF algorithm are similar to other correlation-based filters, and much lower than ReliefF.

Method	C 4.5 tree					
Dataset	Full set	FCBF	CorrSF	ReliefF	ConnSF	$PRBF_{0.001}$
Lung	**80.52 ± 3.53**	*76.30 ± 2.88*	**80.52 ± 3.53**	**80.52 ± 3.53**	**80.52 ± 3.53**	77.37 ± 3.49
Splice	94.16 ± 0.26	**94.30 ± 0.24**	*93.07 ± 0.16*	94.02 ± 0.19	93.83 ± 0.21	94.03 ± 0.22
Promoters	*79.20 ± 1.90*	81.04 ± 1.81	80.85 ± 2.65	81.09 ± 2.06	80.47 ± 2.21	**82.69 ± 1.57**
Method	Naive Bayes					
Dataset	Full set	FCBF	CorrSF	ReliefF	ConnSF	$PRBF_{0.001}$
Lung	*61.27 ± 4.67*	87.37 ± 2.10	**90.98 ± 1.95**	83.43 ± 2.55	71.28 ± 3.93	88.09 ± 1.96
Splice	94.95 ± 0.08	**96.10 ± 0.06**	*93.33 ± 0.05*	95.54 ± 0.08	94.30 ± 0.08	94.62 ± 0.08
Promoters	*90.47 ± 1.40*	94.43 ± 0.52	**94.58 ± 0.86**	91.27 ± 1.18	92.45 ± 1.30	91.18 ± 0.93
Method	1 Nearest Neighbor					
Dataset	Full set	FCBF	CorrSF	ReliefF	ConnSF	$PRBF_{0.001}$
Lung	*47.55 ± 5.61*	78.83 ± 2.98	**82.17 ± 4.23**	78.59 ± 3.71	74.33 ± 5.11	70.60 ± 5.02
Splice	*80.16 ± 0.47*	85.14 ± 0.44	84.60 ± 2.19	83.54 ± 0.44	**87.13 ± 0.64**	84.37 ± 0.65
Promoters	*81.27 ± 2.40*	85.24 ± 2.51	**88.63 ± 1.90**	81.04 ± 1.81	85.38 ± 2.62	85.33 ± 3.02
Method	SVM					
Dataset	Full set	FCBF	CorrSF	ReliefF	ConnSF	$PRBF_{0.001}$
Lung	*47.90 ± 5.71*	84.48 ± 2.74	**90.00 ± 0.00**	**90.00 ± 0.00**	80.63 ± 2.07	80.78 ± 2.07
Splice	*92.35 ± 0.31*	**95.78 ± 0.15**	93.74 ± 0.03	95.49 ± 0.24	94.24 ± 0.16	94.99 ± 0.17
Promoters	91.51 ± 1.65	93.68 ± 1.15	**95.76 ± 0.94**	87.78 ± 2.38	*87.31 ± 1.08*	90.66 ± 1.96

Table 4. Balanced accuracy for the 4 classification methods on features selected by each algorithm; bold face – best results, italics – worst.

The χ^2 test works well for $n > 100$ samples, therefore the results for very small Lung-cancer data are rather poor.

For simplicity of interpretation only data with nominal features have been used, avoiding discretization. Features were ranked according to the SU relevance index. In real applications with very large number of features a cutoff point for ranking should be defined and optimized using crossvalidation tests to determine optimal threshold value. Further reduction of the selected feature subsets using tests for redundancy requires another estimation of the significance parameter that may be done in crossvalidation test and will depend on classifier used. Such frapper (filter-wrapper) approach is not too costly and may be completely automatic. The same algorithm may be used with other indices for relevance indication. Moreover, redundancy reduction based on χ^2 test may be used in unsupervised methods of data analysis. Various variants of this and similar test exist [4], including versions for small samples. This combination of filters, wrappers and redundancy evaluation is a fertile ground for information selection, with many possibilities that remain to be explored. Further tests on much larger bioinformatics data will be reported soon.

Acknowledgement. This work was financed by the Polish Committee for Scientific Research grant 2005-2007 to WD; JB has been supported by the Polish Fundation of Science and grant (2006-2009) No.: 6ZR9 2006 C/06742.

α	0.001	0.01	0.05	0.1	0.15	0.2
Lung	12	14	16	16	18	20
C4.5	77.37 ± 3.49	77.37 ± 3.49	77.37 ± 3.49	77.37 ± 3.49	77.37 ± 3.49	77.37 ± 3.49
NBC	88.09 ± 1.96	85.56 ± 1.14	83.22 ± 2.45	83.22 ± 2.45	84.89 ± 1.85	84.47 ± 2.04
1NN	70.60 ± 5.02	72.17 ± 4.64	68.49 ± 3.72	68.49 ± 3.75	65.88 ± 3.61	63.69 ± 4.37
SVM	80.78 ± 2.07	76.45 ± 3.33	75.08 ± 3.27	75.08 ± 3.27	72.16 ± 3.09	70.20 ± 4.04
Splice	19	24	27	28	30	31
C4.5	94.03 ± 0.22	94.03 ± 0.22	94.19 ± 0.21	94.19 ± 0.21	94.19 ± 0.21	94.22 ± 0.20
NBC	94.62 ± 0.08	94.62 ± 0.08	95.11 ± 0.11	95.08 ± 0.07	94.96 ± 0.10	95.25 ± 0.07
1NN	84.37 ± 0.65	84.37 ± 0.65	81.40 ± 0.48	80.46 ± 0.58	80.66 ± 0.42	81.14 ± 0.41
SVM	94.99 ± 0.17	95.00 ± 0.17	94.49 ± 0.22	94.44 ± 0.19	94.20 ± 0.17	94.42 ± 0.22
Promoters	6	8	11	13	13	14
C4.5	82.69 ± 1.57	82.41 ± 1.69	79.72 ± 2.09	79.77 ± 1.72	79.77 ± 1.72	79.53 ± 1.73
NBC	91.18 ± 0.93	91.98 ± 0.94	92.78 ± 1.24	91.65 ± 0.98	91.65 ± 0.98	92.45 ± 0.69
1NN	85.33 ± 3.02	85.10 ± 2.90	88.68 ± 1.81	86.13 ± 2.37	86.13 ± 2.37	85.33 ± 2.18
SVM	90.66 ± 1.96	90.09 ± 2.09	86.93 ± 2.04	87.88 ± 1.45	87.88 ± 1.45	88.35 ± 2.04

Table 5. Number of features for different levels of significance, and balanced accuracy (bacc) $\pm$ std(bacc) for C4.5, NBC, 1NN and SVM clasiffiers.

References

1. W. Duch, *Filter Methods.* In: Feature extraction, foundations and applications. Eds: I. Guyon, S. Gunn, M. Nikravesh, L. Zadeh, Studies in Fuzziness and Soft Computing, Physica-Verlag, Springer, pp. 89-118, 2006.
2. T.M. Cover. The best two independent measurements are not the two best. *IEEE Transactions on Systems, Man, and Cybernetics*, 4:116–117, 1974.
3. J. Biesiada, W. Duch, Feature Selection for High-Dimensional Data: A Kolmogorov-Smirnov Correlation-Based Filter Solution. Advances in Soft Computing, Computer Recognition Systems (CORES 2005), pp. 95-105, 2005.
4. W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical recipes in C. The art of scientific computing*. Cambridge University Press, Cambridge, UK, 1988.
5. M.A. Hall. *Correlation-based Feature Subset Selection for Machine Learning*. PhD thesis, Department of Computer Science, University of Waikato, Waikato, N.Z, 1999.
6. L. Yu and H. Liu. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *12th Int. Conf. on Machine Learning (ICML-03), Washington, D.C.*, pp. 856–863, Morgan Kaufmann, CA 2003.
7. M. Dash and H. Liu. Consistency-based search in feature selection. *Artificial Intelligence*, 151:155–176, 2003.
8. M. Robnik-Sikonja and I. Kononenko. Theoretical and empirical analysis of relieff and rrelieff. *Machine Learning*, 53:23–69, 2003.
9. W. Duch, T. Winiarski, J. Biesiada, and A. Kachel. Feature ranking, selection and discretization. In *Proceedings of Int. Conf. on Artificial Neural Networks (ICANN)*, pages 251–254, Istanbul, 2003. Bogazici University Press.
10. I. Witten and E. Frank. *Data minig – practical machine learning tools and techniques with JAVA implementations*. Morgan Kaufmann, San Francisco, CA, 2000.
11. C.J. Mertz and P.M. Murphy. *The UCI repository of machine learning databases*. Univ. of California, Irvine, 1998. http://www.ics.uci.edu.pl/ mlearn/MLRespository.html.
12. J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufman, CA, 1993.