Feature ranking methods based on information entropy with Parzen windows

Authors:

Jacek Biesiada, Włodzisław Duch, Adam Kachel, Krystian Mączka, Sebastian Pałucha

Abstract *– A comparison between several feature ranking methods used on artificial and real dataset is presented. Six ranking methods based on entropy and statistical indices, including χ^2 and Pearson's correlation coeffcient, are considered. The Parzen window method for estimation of mutual information and other indices gives similar results as discretization based on the separability index, but results strongly dependent on the σ smoothing parameter. The quality of the feature subsets with highest ranks is evaluated by using decision tree, Naive Bayes and the nearest neighbour classifiers. Significant differences are found in some cases, but there is no single best index that works best for all data and all classifiers. To be sure that a subset of features giving the highest accuracy has been selected the use of many different indices is recommended.*

Keywords *– discretization, feature selection, Parzen windows, feature ranking methods*

Introduction

Feature selection is a fundamental problem in many different areas, especially in bioinformatics, document classification, forecasting, object recognition or in modelling of complex technological processes. In such applications datasets with thousands of features are not uncommon. All features may be important for some problems, but for some target concept only a small subset of features is usually relevant. To overcome the curse of dimensionality problem dimensionality of the feature space should be reduced. This may be done by creating new features that contain maximum information about the class label from the original ones, or selecting only the subset of relevant features. The latter methodology is named feature selection, while the former is called feature extraction, and it includes linear (Principal Component Analysis, Independent Component Analysis etc.) and non-linear feature extraction methods.

Feature selection algorithms may be divided into filters [17, 18], wrappers [7] and embedded approaches [19]. Wrapper methods require application of a classifier (which should be trained on a given feature subset) to evaluate quality of selected features, while filters method evaluate this quality independently from the classification algorithm. Embedded methods perform feature selection during learning of optimal parameters (for example, neural network weights between the input and the hidden layer).

A common measure of relevance in many feature selection algorithms is based on the mutual information, considered to be a good indicator of relations between the input feature and the target variable [1] . To compute the mutual information between continuous variables one must integrate the probability density functions (*pdf*s) of input and output variables. In the literature one can find three different methods to estimate mutual information based on histogram estimators [20, 22], kernel estimators [23] and parametric methods. Histogram method may be used in estimation of the *pdf*s [1, 6] if sufficient data is given. It is well known that accurate calculation of mutual information using histogram methods requires a very large computer memory. For that reason high-quality estimation of the *pdf*s based on histograms is practically impossible. In this paper, in order to avoid practical obstacles and evaluate mutual information with high accuracy Gaussian Parzen Window method [10] is used instead of discretization to estimate the distribution of feature values.

Some classification algorithms have inherited ability to focus on relevant features and ignore irrelevant ones. Decision trees are primary example of a class of such algorithms [2], [16], but also multi-layer perceptron (MLP) neural networks with strong regularization of the input layer may exclude the irrelevant features in an automatic way [4]. Such methods may also benefit from independent feature selection. On the other hand, some algorithms have no provisions for feature selection. The k-nearest neighbour algorithm (k-NN) is one family of such methods that classify novel examples by retrieving the nearest training example, strongly relaying on feature selection methods to remove noisy features.

This paper is structured as follows. Section 2 contains discussion of entropy-based feature ranking methods and approximation of probability density functions (pdf) based on methodology of Parzen windows. In Section 3 the datasets used in numerical experiments are described. Results of comparing six entropy-based methods and χ^2 indices with estimations based on Parzen windows as well as with simple Pearson's correlation coefficient between feature values and target variables are presented in Section 4. A summary of results and plans for feature work are given in the last section.

Theoretical framework

Ranking methods and information theory

A typical ranking process consists of four steps:

1. Initialize set $\mathcal{F}$ to the whole set of p features. $\mathcal{S}$ is an empty set.
2. For all features $f \in \mathcal{F}$ compute $J(f)$ coefficient.
3. Find feature f that maximizes $J(f)$ and move it to $\mathcal{S} \leftarrow \mathcal{S} \cup \{f\}, \mathcal{F} \in \mathcal{F} \backslash \{f\}$
4. Repeat until the cardinal of $\mathcal{S}$ is p.

where $J(f)$ is a criterion function (specific for a given algorithm) which gives a measure of dependency between features (f) and classes (C).

The first index uses normalized information gain, called also the "asymmetric dependency coefficient" (ADC) [15]:

$$ADC(C, f) = \frac{MI\,(C, f)}{H\,(C)} \tag{1}$$

where for K classes information entropy $H(C)$ and $H(f)$, and mutual information $MI(C, f)$ between C and f is defined according to Shannon formula [14] as:

$$\begin{aligned} H(C) = & \quad -\sum_{i=1}^{K} p(C_i)\lg_2 p(C_i) \\ H(f) = & \quad -\sum_{x} p(f = x)\lg_2 p(f = x) \\ MI(C, f) = & \quad -H(C, f) + H(C) + H(f) \end{aligned} \tag{2}$$

Here the sum over x is used only for features f that take discrete values, for continuous features it should be replaced by an integral or discretization should be performed first to estimate probabilities $p(f = x)$.

The ranking algorithm proposed by Setiono [12] uses a normalized gain ratio:

$$U_S(C, f) \;=\; \frac{MI(C, f)}{H(f)} \tag{3}$$

where $H(f, C)$ is the joint entropy of f and C variables.

Another normalization may be used to calculate gain ratio dividing by the class-feature entropy :

$$U_H(C, f) = \frac{MI(C, f)}{H(f, C)} \tag{4}$$

The last index based on mutual information is called "symmetrical uncertainty" [11]:

$$SU(C, f) \;=\; 2\left[\frac{MI(C, f)}{H(C) + H(f)}\right] \in [0, 1] \tag{5}$$

The value 1 implies that f and C are strong dependent and the value 0 implies that f and C are independent.

Mantaras [8] has proposed an interesting criterion D_{ML} which fulfils all axioms of a distance and may be defined as:

$$D_{ML}(f_i, C) \;=\; H(f_i|C) \;+\; H(C|f_i) \tag{6}$$

where $H(f_i|C)$ and $H(C|f_i)$ is the conditional entropy defined by Shannon [14] as $H(X|Y) = H(X, Y) - H(Y)$.

Another criterion based on weighted joint entropy index was introduced by Chi [3] and is defined as:

$$Chi(f) = -\sum_{k=1}^{N} p(f = x_k) \sum_{i=1}^{K} p(f = x_k, C_i)\lg_2 p(f = x_k, C_i) = E_f\left[H(f = x_k, C)\right] \tag{7}$$

This is an average of entropy calculated in each discretization bin over the distribution of feature values.

Ranking methods based on statistical measures

Statistical measures of dependence between random variables provide an alternative to indices based on information theory. A common measure of such dependence (in this case the relationships between the feature and the class values) may be based on the χ^2 statistics. The χ^2 coefficient is given by:

$$\chi^2(f,C) = \sum_{ij} \frac{(p(f=x_j, C_i) - p(f=x_j)\cdot p(C_i))^2}{p(f=x_j)\cdot p(C_i)} \tag{8}$$

where $p(\cdot)$ are probabilities. Growing χ^2 values signify stronger dependence between feature values and class labels, and therefore this index may be used for ranking features. The χ^2 statistics has been previously used in the discretization process by Setiono and Liu [13].

The last statistical measure used in this paper is the Pearson's linear correlation coefficient:

$$\rho(X,Y) \;=\; \frac{\text{cov}(X,Y)}{\sigma(X)\,\sigma(Y)} \tag{9}$$

where the standard deviation $\sigma(X), \sigma(Y)$ of X, Y and covariance $\text{cov}(X,Y)$ are defined as:

$$\begin{aligned} \sigma(X) &= \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{x}-x_i)^2} \\ \sigma(Y) &= \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}-x_y)^2} \\ \text{cov}(X,Y) &= \frac{1}{N}\sum_{i=1}^{n}(\hat{x}-x_i)(\hat{y}-y_i) \end{aligned} \tag{10}$$

$\rho(X,Y)$ is equal 0 for independent X and Y and is equal ±1 when the variables are linearly dependent. Absolute value of correlation coefficient is used here because the direction of correlation is not important for feature ranking. It is important to realize that linear correlation may completely fail for some distributions of feature and class values.

The Parzen window Density Estimate

The Parzen window density estimate of a continuous feature f can be used to approximate the probability density $p(\mathbf{x})$ of a distribution [10], where x is a value of feature f. It involves the superposition of a normalized windows function centred on a set of random samples. Given a set of n d-dimensional training vectors $D = \{\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_n}\}$, the *pdf* estimate using Parzen window is given by:

$$\hat{p}(\mathbf{x}) \;=\; \frac{1}{n}\sum_{i=1}^{n}\phi(\mathbf{x} \;-\; \mathbf{x}_i, h) \tag{11}$$

where $\phi(\cdot)$ is the window function and h is the window width parameter. Parzen showed that $\hat{p}()$ converges to the true density if $\phi(\cdot)$ and h are selected properly [10]. The window function must be normalized to 1:

$$\int \phi(\mathbf{y}, h)\, d\mathbf{y} \;=\; 1 \tag{12}$$

and the width parameter is required to be a function of n such that:

$$\lim_{n\to\infty} h(n) \;=\; 0 \tag{13}$$

and

$$\lim_{n\to\infty} nh^d(n) \;=\; \infty \tag{14}$$

The Gaussian window function is given by:

$$\phi\left(\mathbf{z}, h\right) \;=\; \frac{1}{\left(2\pi\right)^{d/2} h^{d} |\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{\mathbf{z}^{T}\Sigma^{-1}\mathbf{z}}{2h^{2}}\right) \tag{15}$$

In ranking methods the one dimensional Gaussian window was used:

$$\phi\left(\mathbf{x}, \sigma\right) \;=\; \frac{1}{\sqrt{2\pi}\sigma} \exp -\frac{\left(x-\hat{x}\right)^{2}}{2\sigma^{2}} \tag{16}$$

where σ is the standard deviation; several arbitrary values $\sigma = 1.0, 0.5, 0.2, 0.1$ were used for tests. In [21] the choice $\sigma = k/\log n$ has been proposed, where k is a positive constant (for example $k = 1$), and n is the number of the samples. The choice of σ satisfies the condition (13) and (14). For the datasets described below this translates to: hypothyroid $\sigma = 0.279$ ($k = 1$ and $n = 3772$) and abalone $\sigma = 0.276$ ($k = 1$ and $n = 4177$).

In practice it is not clear if there are significant differences between various coefficients measuring similarity of distributions. To answer this questions computational experiments described below were performed.

Datasets used for testing

Artificial datasets called “Gauss1" and “Gauss2" have been generated [5] to compare different feature ranking and feature selection methods on data for which the importance of features is completely known. Two real datasets for tests were used, the “hypothyroid” and the “abalone" data, both taken from the UCI repository of machine learning databases [9].

Artificial data

* Gauss1,Gauss2
  These datasets have four and eight features, respectively. In the first dataset four Gaussian functions with unit dispersion have been used to generate vectors in 4 dimensions, each Gaussian cluster representing a separate class. The first Gaussian is centred at (0,0,0,0), the next at $a(1, 1/2, 1/3, 1/4), 2a(1, 1/2, 1/3, 1/4), 3a(1, 1/2, 1/3, 1/4)$, respectively ($a = 2$ is used below). The dataset contains 4000 vectors, 1000 per each class. In this case the ideal ranking should give the following order: $x_1 > x_2 > x_3 > x_4$.

  The second dataset (Gauss2) is an extension of the first, containing eight features. Additional 4 linearly dependent features have been created by taking $x_{i+4} = 2x_i + \epsilon$, where ϵ is a uniform noise with unit variance. In this case the ideal ranking should give the following order: $x_1 > x_5 > x_2 > x_6 > x_3 > x_7 > x_4 > x_8$.

Real data

* Hypothyroid
  This data contains results from real medical screening tests for the thyroid problems. The class distribution is about 92.5% normal, 5% of the primary hypothyroid and 2.5% of the compensated hypothyroid type. The data offers a good mixture of nominal (15) and continuous (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 of the data collection).

* Abalone
  The age of abalone molluscs should be predicted from their physical measurements. This is essentially an approximation problem, but because the age is quantized into 29 bins (the number of rings, equivalent to the age in full years) it may be treated as a classification problem with a large number of classes. 4177 cases with 8 features are given, including one nominal feature, six continuous measurement values, and feature number one with values created randomly in the $[0, 1]$ range.

Experiments and results

Four datasets described above have been used in numerical tests. In each case seven ranking methods have been used based on information theory (1) to (8) and correlation coefficient (9). Different methods of parameter estimation have been used because the datasets have both discrete and continuous feature. For continuous features Parzen window with different values of σ

smoothing parameter has been used. Figure 1 presents the dependence of the ADC coefficient on the number of bins using the equi-width discretization, and estimated values of the ADC coefficient for different value of the smoothing parameter σ. The values of ADC coefficients for Parzen windows estimations are much larger then for the equi-width discretization (non-monotonic dependence on σ should be noted). Although much larger values may be expected with more sophisticated discretization method [5] this is sufficient for relative comparison of entropy and correlation based methods. This behavior has been verified using the artificial datasets Gauss1 and Gauss2. For prediction accuracy the balanced accuracy is used because the hypothyroid dataset has strong unbalanced data and the abalone has 28 classes.

Results for artificial data

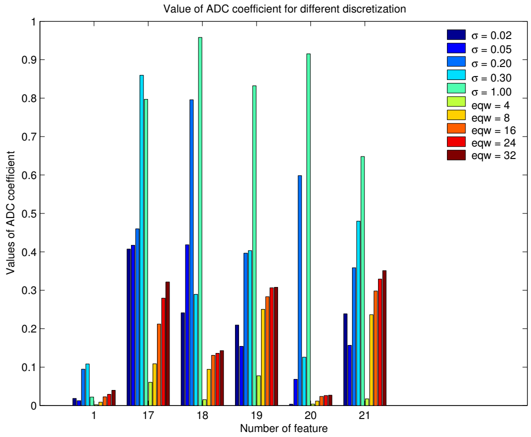

FIGURE 1:
ADC coefficient in function of a number of bins for equi-width discretization and several σ smoothing parameters.

Correct ranking was found by the ρ linear correlation index; for this data monotonic change between the feature values and class numbers makes it possible. Surprisingly the Chi-index (7) placed all redundant features at the end providing a perfect selection, although this is only a ranking algorithm. Average entropy of noisy versions of features 5-8 happens to be in this case larger than the increase of entropy due to the overlap of distributions, but this effect should depend on the value of a parameter determining the overlap. Results for other rankings show preference of Gaussian distributions with larger values of standard deviation (5 more important than 1 etc.). Errors in ranking of features may result from numerical deviations. These observations provide an interesting starting point to search for the corrections to the indices used for rankings.

Method	Most – Least Important							
ADC - index (1)	5	1	6	7	8	2	3	4
U_S - index (3)	5	1	6	7	8	2	3	4
U_H - index (4)	5	1	6	7	8	2	3	4
D_{ML} - index (6)	1	5	6	2	7	8	3	4
Chi - index (7)	1	2	3	4	5	6	7	8
SU -index (5)	5	1	6	7	8	2	3	4
ρ - index (9)	1	5	2	6	3	7	8	4
χ^2 - index (8)	5	1	6	7	8	2	3	4

TABLE 1:
Results of feature ranking on Gauss2 data for $\sigma = 0.3$; see description in the text.

Results for the hypotyhroid datasets

Hypothyroid dataset has a large training and test part strongly unbalanced (training 92.5% normal, 5% primary and 2.5% for compensated hypothyroid, and similar distribution for test). 21 features are given, 15 are binary and 6 are continuous (no. 1, 17–21), results obtained from medical test. Parzen window smoothing parameter applied to these 6 features was changed from 0.02 to 1.00 has significant influence on the results, with the best results obtained for $\sigma = 0.3 \approx 1/\log n$. The ranking of

Method	Most – Least Important																				
ADC - index, (1)	17	21	19	18	20	1	3	10	16	2	6	7	8	13	5	4	11	12	14	9	15
U_S - index (3)	17	21	18	19	3	7	13	10	8	20	15	1	6	16	5	4	12	2	11	9	14
U_H - index (4)	17	21	18	19	3	20	1	10	7	16	6	8	13	2	5	4	11	12	14	9	15
D_{ML} - index (6)	19	21	1	20	17	18	2	3	11	10	16	6	14	9	8	7	4	5	13	12	15
Chi - index (7)	19	21	20	18	1	17	2	3	11	10	16	6	14	7	9	8	4	5	13	12	15
SU - index (5)	17	21	18	19	3	20	1	10	7	16	6	8	13	2	5	4	11	12	14	9	15
ρ - index (9)	17	21	19	18	10	3	2	20	16	7	13	5	11	6	4	12	8	9	1	15	14
χ^2 - index (8)	17	21	19	18	20	1	3	10	16	2	6	7	8	13	5	4	11	12	14	9	15

TABLE 2:
Results of feature ranking on the hypothyroid data for the $\sigma = 0.3$; see description in the text.

features obtained for the training data by all $J(f)$ indices is presented in Tab. 2. The first 4 features are always continuous, with 17, 19 and 21 consistently ranked as top. Significant differences are observed in the order of the remaining features.

For each ranking method investigation of classification accuracy on the test data as a function of the n best features has been done. Three classifiers were used: NBC (Naive Bayes Classifier) and C4.5 decision tree [16], as implemented in Weka (http://www.cw.waikato.ac.nz/ ml/weka/) and the nearest neighbour (as implemented in the GhostMiner package, http://www.fqspl.com.pl). All of these classifiers give deterministic results, simplifying the comparison (in contrast to these methods, neural classifiers give slightly different results after each restart, therefore averaging and variance of the results should be taken into account).

Classification results are presented in Fig. 2. Feature number 17, predicted by the ρ, ADC , U_S, U_H, χ^2 and SU indices, is definitely more important than 19 and 21. Up to 6 features all these methods give similar results, but after that only ρ and S_U indices provide important features, all other indices leading to a significant degradation of results. The last 3 features evidently confuse the nearest neighbour classifier, therefore they should always be omitted in this type of methods. Features 19 and 21 selected as the most important by D_{ML} and Chi as the two best features gives poor result for all classifiers until feature 17 is added; thus the two best methods for the artificial data completely fail in this case. The peak performance is reached already for two features except for the NBC where 4 features are needed. Because C4.5 removes less useful features automatically pruning its tree accuracy does not drop but stays at the peak level as long as all important features are included.

Results for the abalone datasets

Similar calculations were performed for the abalone dataset. First the ranking algorithms were applied to the whole dataset, and since there is no test datasets classification accuracy was estimated using ten-fold crossvalidation. For the purpose of comparing different ranking methods this approach is sufficient, producing one ranking for each index. For real application this could lead to some overfitting, therefore ranking and classification should be done separately for each training partition. Good generalization may be obtained by selecting only those features that were highly ranked in all data partitions.

The ranking of features for 8 indices is presented in Tab. 3. For the abalone dataset the quality of classification is obviously quite low, but many errors are small, getting the predicted age of the abalone wrong by one or two years. The number of data vectors for ages 6-13 years is significantly larger than for a very young or old abalones, thus many bigger errors are made outside this range.

Method	Most – Least Important								
ADC - index, (1)	6	7	8	1	9	3	4	5	2
U_S - index, (3)	6	7	8	9	1	2	5	4	3
U_H - index, (4)	6	7	8	9	1	4	3	5	2
D_{ML} - index, (6)	2	5	4	3	6	9	7	8	1
Chi - index, (7)	2	1	5	8	9	7	6	4	3
SU - index, (5)	6	7	8	9	1	3	4	5	2
ρ - index (9)	9	4	5	3	6	8	7	2	1
χ^2 - index, (8)	6	7	8	1	9	3	4	5	2

TABLE 3:
Results of ranking methods on the abalone dataset with the standard deviation. $\sigma = 0.3$

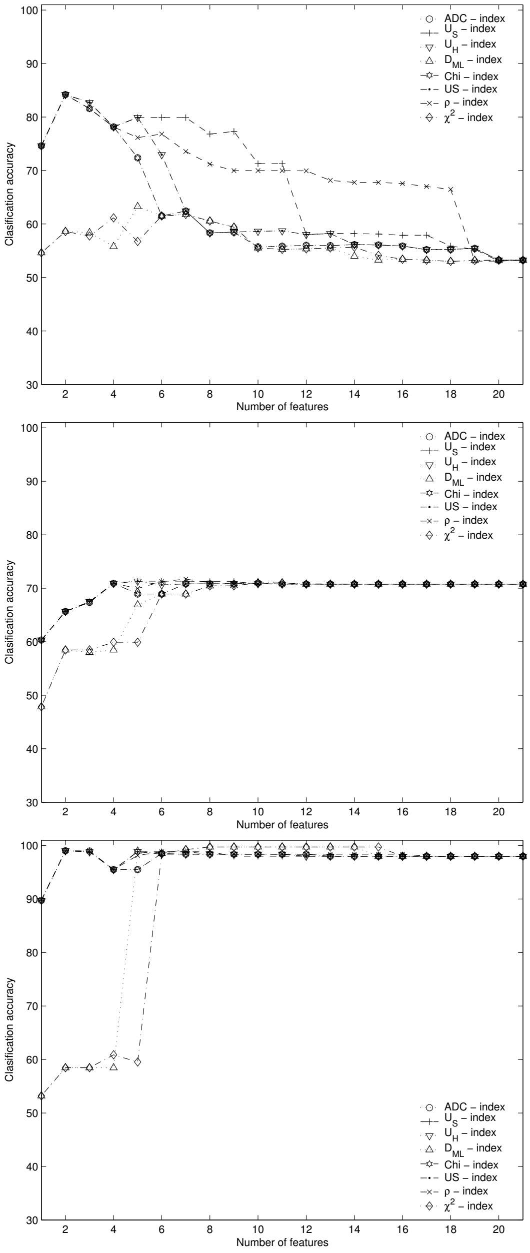

FIGURE 2:
Balanced classification accuracy for the hypothyroid dataset, upper figure – 1NN classifier, middle figure – Naive Bayes classifier, lower figure – C4.5 trees classifier

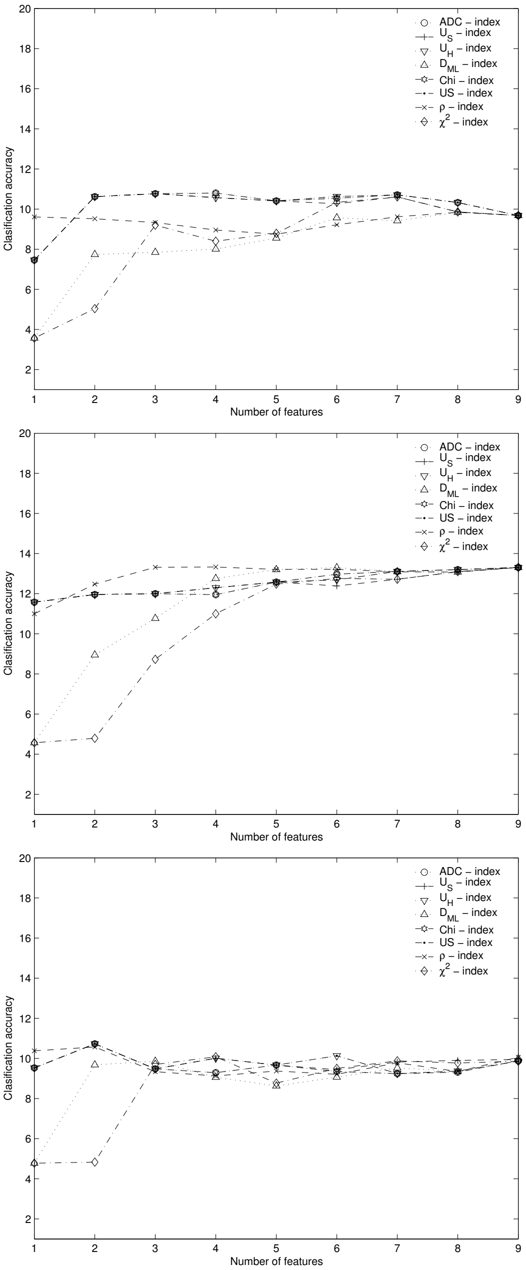

FIGURE 3:
Balanced classification accuracy for the abalone dataset, upper figure – 1NN classifier, middle figure – Naive Bayes, lower figure – C4.5 tress classifier.

Classification accuracy for the three classifiers and growing number of features is presented in Fig. 3. Feature no. 1, added to the abalone dataset with random values has been ranked as last only by the D_{ML} and ρ indices. Unfortunately all ranking methods give very similar poor results for balanced accuracy. In this case simple ranking is not sufficient to obtain good results.

Conclusions

Ranking methods may filter features leading to reduced dimensionality of the feature space. This is especially effective for classification methods that do not have any inherent feature selections build in, such as the nearest neighbour methods or some types of neural networks. Seven entropy-based and two statistical indices have been used for feature ranking, evaluated and compared using three different types of classifiers on artificial and real benchmark data. Accuracy of the classifiers is significantly influenced by the choice of ranking indices (Fig. 2 and 3). For the two experiments on the real data presented here, and other experiments that have not been reported, different ranking methods emerge as the winner. The simple statistical test based on correlation coefficient gives usually the best results, but very similar results may be obtained using the entropy based indices. From computational point of view the correlation coefficient is slightly less expensive then entropy based indices (although in practice this may not be so important), and it does not require discretization of continuous features.

The algorithms and datasets used in this paper were selected according to precise criteria: entropy-based algorithms and several datasets, either real or artificial, with nominal, binary and continuous features. The two real datasets illustrated the fact that the best indices (ρ and U_S) on one of them may select the worst feature as the first on the other dataset. The classifiers used for evaluation of feature subsets generated by ranking procedures were deterministic, to avoid additional source of variance.

What conclusions may one draw from this study? There is no best ranking index, for different datasets and different classifiers accuracy curves as a function of the number of features used may significantly differ. Evaluation of ranking indices is fast. The only way to be sure that the highest accuracy is obtained in practical problems requires testing a given classifier on a number of feature subsets, obtained from different ranking indices. The number of tests needed to find the best feature subset is very small comparing to the cost of wrapper approach for larger number of features.

Several improvements of the ranking methods presented here are possible:

- Results of ranking algorithms depend on discretization procedure for continuous features, therefore better discretization should be used.
- Crossvalidation techniques may be used to select features that are important in rankings on all partitions.
- Features with lowest ranking values of various indices in all crossvalidations may be safely rejected.
- The remaining features should be analyzed with selection methods that allow for elimination of redundant and correlated features.
- More ranking indices similar to ρ that evaluate similarity of statistical distributions may be introduced, not only for linear dependency.
- A superindex based on the average of a group of ranking indices may be introduced.

These conclusions and recommendations will be tested on larger datasets using various classification algorithms in the near future.

Acknowledgement

This work was partially supported by the Polish Committee for Scientific Research (KBN) grant no. 3 T11C 071 28.

References

[1] Battiti R., “Using mutual information for selecting features in supervised neural net learning.” IEEE Trans. Neural Networks, vol. 5, no. 4, pp. 537 - 550, July 1994.

[2] Breiman L., Friedman J.H., Olshen R.H., Stone C.J., *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA, 1984.

[3] Chi J., “Entropy based feature evaluation and selection technique”, Proc. of 4^{th} Australian Conf. on Neural Networks (ACNN’93), pp. 181-196, 1993.

[4] 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, vol. 12, pp. 277-306, 2001.

[5] Duch W., Winiarski T., Biesiada J., Kachel A., “Feature Ranking, Selection and Discretization". Int. Conf. on Artificial Neural Networks (ICANN) and Int. Conf. on Neural Information Processing (ICONIP), Istanbul, June 2003, pp. 251-254.

[6] Duch W., Wieczorek T., Biesiada J., Blachnik M., “Comparision of feature ranking methods based on information entropy.” Proc. of. Int. Joint Conference on Neural Networks (IJCNN), Budapest 2004, IEEE Press, pp. 1415-1420.

[7] Kohavi R., John G.H., “Wrappers for feature Subset Selection." Artificial Intelligence, vol. 97, pp 273-324, 1997.

[8] Lopez de Mantaras R., “A Distance-Based Attribute Selecting Measure for Decision Tree Induction." Machine Learning vol. 6, pp. 81-92, 1991.

[9] Mertz C.J., Murphy P.M., UCI repository of machine learning databases http://www.ics.uci.edu.pl/~mlearn/MLRespository.html. Irvine, CA: University of California, Dep. of Information and Computer Science.

[10] Parzen E., “On estimation of a probability density function and mode”, Ann. Math. Statistic, vol. 33, pp. 1065-1076, 1962.

[11] Press, W.H., Flnnery B.P., Teukolsky S.A. and Vetterling W.T. “Numerical recipies in C.” Cambridge University Press, Cambridge.

[12] Setiono R., Liu H., “Improving backpropagation learning with feature selection". Applied Intelligence: The International Journal of Artifical Intelligence, Neural Networks, and Complex Problem-Solving Technologies, vol. 6, pp. 129-139, 1996.

[13] Setiono R., Liu H., “Chi2: Feature selection and discretization of numeric attributes." In: Proc. 7th IEEE International Conf. on Tools with Artificial Intelligence, Washington D.C., pp. 388-391, Nov. 1995.

[14] Shannon C.E., Weaver W., *The Mathematical Theory of Communication*. Urbana, IL, University of Illinois Press, 1946.

[15] Shridhar D.V., Bartlett E.B., Seagrave R.C., “Information theoretic subset selection." Computers in Chemical Engineering, vol. 22, pp. 613-626, 1998.

[16] Quinlan J.R. *C4.5: Programs for machine learning*. San Mateo, Morgan Kaufman, 1993.

[17] Almuallim H., Dietterich T.G. “Learning with many irrelevant features”. In: *Proc. AAAI-91*, Anaheim, CA, pp. 547-552, 1991.

[18] Kira K., Rendell L.A. “The feature selection problem: tradional methods and a new algorithm”. In: *Proc. AAAI-92*, San Jose, CA, pp. 122-126, 1992.

[19] Blum A.I., Langley P. “Selection of relevant features and examples in machine learning". *Artificial Intelligence*, vol 97, pp. 245-271, 1997.

[20] Mars N.J.L., van Aragon G.W. Time delay estimation in non-linear systems using average amount of mutual information analysis. *Signal Processing*, vol. 4(2), pp. 139-153, 1982.

[21] Meilhac C. and Naster C. Relevance feedback and category search in image databese. Proc. IEEE Int. Conf. on Content-based Access of Video and Image databese. Florence, Italy, pp. 512-517, June 1999.

[22] Moddemeijer R. On estimation of entropy and mutual information of continuous distribution. *Signal Processing*, vol. 16(3), pp. 233-246, 1989.

[23] Moon Y.I., Rajagopalan B., Lall U. Estimation of mutual information using kernel density estimators. *Phys. Rev. E.*, vol. 52(3), pp. 2318-2321, 1995.

Włodzisław Duch
Department of Informatics, Nicolaus Copernicus University,
Grudziądzka 5, Toruń, Poland,
and School of Computer Engineering, Nanyang Technological University, Singapore.
WWW - Google: Duch

Jacek Biesiada, Adam Kachel, Sebastian Pałucha, Krystian Mączka
Department of Electrotechnology, The Silesian University of Technology,
ul. Krasińskiego 8, 40-019 Katowice, Poland
WWW - http://metet.polsl.katowice.pl