Conditional Mutual Information Based Feature Selection for Classification Task

Jana Novovičová[1,2], Petr Somol[1,2], Michal Haindl[1,2], and Pavel Pudil[2,1]

[1] Dept. of Pattern Recognition,
Institute of Academy of Sciences of the Czech Republic
{novovic,somol,haindl}@utia.cas.cz
http://ro.utia.cz/
[2] Faculty of Management, Prague University of Economics, Czech Republic
pudil@fm.vse.cz
http://www.fm.vse.cz

Abstract. We propose a sequential forward feature selection method to find a subset of features that are most relevant to the classification task. Our approach uses novel estimation of the conditional mutual information between candidate feature and classes, given a subset of already selected features which is utilized as a classifier independent criterion for evaluation of feature subsets. The proposed mMIFS-U algorithm is applied to text classification problem and compared with MIFS method and MIFS-U method proposed by Battiti and Kwak and Choi, respectively. Our feature selection algorithm outperforms MIFS method and MIFS-U in experiments on high dimensional Reuters textual data.

Keywords: Pattern classification, feature selection, conditional mutual information, text categorization.

1 Introduction

Feature selection plays an important role in classification problems. In general, a pattern classification problem can be described as follows: Assume that feature space $\mathcal{X}$ is constructed from D features $X_i, i = 1, \dots, D$ and patterns drawn from $\mathcal{X}$ are associated with $|\mathcal{C}|$ classes, whose labels constitute the set $\mathcal{C} = \{c_1, \dots, c_{|\mathcal{C}|}\}$. Given a training data the task is to find a classifier that accurately predicts the label of novel patterns. In practice, with a limited amount of training data, more features will significantly slow down the learning process and also cause the classifier to over-fit the training data because of the irrelevant or redundant features which may confuse the learning algorithm. By reducing the number of features, we can both reduce over-fitting of learning methods and increase the computational speed of classification. We focus in this paper on feature selection in context of classification.

The feature selection task is to select a subset S of d features from a set of available features $X = \{X_i, i = 1, \dots, D\}$, where $d < D$ represents the desired number of features. All feature selection (FS) algorithms aim at maximizing some performance measure for the given class and different feature subsets S.

L. Rueda, D. Mery, and J. Kittler (Eds.): CIARP 2007, LNCS 4756, pp. 417–426, 2007.

Many existing feature selection algorithms can roughly be divided into two categories: *filters* [1], [2] and *wrappers* [3]. Filter methods select features independently of the subsequent learning algorithm. They rely on various measures of the general characteristics of the training data such as distance, information, dependency, and consistency [4]. On the contrary the wrapper FS methods require one predetermined learning algorithm and use its classification accuracy as performance measure to evaluate the quality of selected set of features. These methods tend to give superior performance as they find features better suited to the predetermined learning algorithm, but they also tend to be more computationally expensive. When the number of features becomes very large, the filter methods are usually to be chosen due to computational efficiency. Our interest in this paper is to design a filter algorithm.

Search scheme is another problem in feature selection. Different approaches such as complete, heuristic and random search have been studied in the literature [5] to balance the tradeoff between result optimality and computational efficiency. Many filter methods [6] evaluate all features individually according to a given criterion, sort them and select the best individual features. Selection based on such ranking does not ensure weak dependency among features, and can lead to redundant and thus less informative selected subset of features.

Our approach to FS iteratively selects features which maximize their mutual information with the class to predict, conditionally to the response of any other feature already selected. Our conditional mutual information criterion selects features that are highly correlated with the class to predict if they are less correlated to any feature already selected.

Experiments demonstrate that our sequential forward feature selection algorithm mMIFS-U based on conditional mutual information outperforms the MIFS methods proposed by Battiti [7] and MIFS-U proposed by Kwak and Choi [8], both of which we also implemented for test purposes.

2 Information-Theoretic Feature Selection

In this section we briefly introduce some basic concepts and notions of the information theory which are used in the development of the proposed feature selection algorithm.

Assume a D-dimensional random variable $Y = (X_1, \ldots, X_D) \in \mathcal{X} \subseteq \mathcal{R}^D$ representing feature vectors, and a a discrete-valued random variable C, representing the class labels. In accordance with Shannon's information theory [9], the uncertainty of a random variable C can be measured by entropy $H(C)$. For two random variables Y and C, the conditional entropy $H(C|Y)$ measures the uncertainty about C when Y is known. The amount by which the class uncertainty is reduced, after having observed the feature vector Y, is called the *mutual information*, $I(C, Y)$. The relation of $H(C)$, $H(C|Y)$ and $I(C, Y)$ is

$$I(C, Y) = I(Y, C) = H(C) - H(C|Y) = \sum_{c \in \mathcal{C}} \int_{\mathbf{y}} p(c, \mathbf{y}) \log \frac{p(c, \mathbf{y})}{P(c)p(\mathbf{y})} d\mathbf{y}, \quad (1)$$

where $P(c)$ represents the probability of class C, $\mathbf{y}$ represents the observed feature vector Y, $p(c, \mathbf{y})$ denotes the joint probability density of C and Y.

The goal of classification is to minimize the uncertainty about predictions of class C for the known observations of feature vector Y. Learning a classifier is to increase $I(C, Y)$ as much as possible. In terms of mutual information (MI), the purpose of FS process for classification is to achieve the highest possible value of $I(C, Y)$ with the smallest possible size of feature subsets.

The FS problem based on MI can be formulated as follows [7]: Given an initial set X with D features, find the subset $S \subset X$ with $d < D$ features $S = \{X_{i_1}, \ldots, X_{i_d}\}$ that minimizes conditional entropy $H(C|S)$, i.e., that maximizes the mutual information $I(C, S)$.

Mutual information $I(C, S)$ between the class and the features has become a popular measure in feature selection [7], [8], [10], [11]. Firstly, it measures general dependence between two variables in contrast with the correlation. Secondly, MI determines the upper bound on the theoretical classification performance [12],[9].

To compute the MI between all candidate feature subsets and the classes, $I(C, S)$ is practically impossible. So realization of the greedy selection algorithm is computationally intensive. Even in a sequential forward search it is computationally too expensive to compute $I(C, S)$.

To overcome this practical obstacle alternative methods of $I(C, S)$ computation have been proposed by Battiti [7] and Kwak and Choi [13], [8], respectively. Assume that S is the subset of already selected features, $X \setminus S$ is the subset of unselected features. For a feature $X_i \in X \setminus S$ to be selected, the amount of information about the class C newly provided by feature X_i without being provided by the already selected features in the current subset S must be the largest among all the candidate features in $X \setminus S$. Therefore, the conditional mutual information $I(C, X_i|S)$ of C and X_i given the subset of already selected features S is maximized. Instead of calculating $I(C, X_i, S)$, the MI between a candidate for newly selected feature $X_i \in X \setminus S$ plus already selected subset S and the class variable C, Battiti and Kwak and Choi used only $I(C, X_i)$ and $I(X_s, X_i)$, $X_s \in S$.

The estimation formula for $I(C, X_i|S)$ in MIFS algorithm proposed by Battiti [7] is as follows:

$$I_{Battiti}(C, X_i|S) = I(C, X_i) - \beta \sum_{X_s \in S} I(X_s, X_i). \tag{2}$$

Kwak and Choi [8] improved (2) in their MIFS-U algorithm under the assumption that the class C does not change the ratio of the entropy of X_s and the MI between X_s and X_i

$$I_{Kwak}(C, X_i|S) = I(C, X_i) - \beta \sum_{X_s \in S} \frac{I(C, X_s)}{H(X_s)} I(X_s, X_i). \tag{3}$$

In both (2) and (3), the second term of the right hand side is used to estimate the redundant information between the candidate feature X_i and the already selected features with respect to classes C. The parameter β is used as a factor for controlling the redundancy penalization among single features and has a great influence on FS. The parameter was found experimentally in [7]. It was shown by Peng et al. [11] that for maximization of $I(C, S)$ in the sequential forward selection a suitable value of β in (2) is $1/|S|$, where $|S|$ denotes the number of features in S.

2.1 Conditional Mutual Information

Our feature selection method is based on the definition of the conditional mutual information $I(C, X_i|X_s)$ as the reduction in the uncertainty of class C and the feature X_i when X_s is given:

$$I(C, X_i|X_s) = H(X_i|X_s) - H(X_i|C, X_s). \tag{4}$$

The mutual information $I(C, X_i, X_s)$ satisfies the chain rule for information [9]:

$$I(C, X_i, X_s) = I(C, X_s) + I(C, X_i|X_s). \tag{5}$$

For all candidate features to be selected in the greedy feature selection algorithm, $I(C, X_s)$ is common and thus does not need to be computed. So the greedy algorithm now tries to find the feature that maximizes conditional mutual information $I(C, X_i|X_s)$.

Proposition 1: The conditional mutual information $I(C, X_i|X_s)$ can be represented as

$$I(C, X_i|X_s) = I(C, X_i) - [I(X_i, X_s) - I(X_i, X_s|C)] \tag{6}$$

Proof: By using the definition of MI we can rewrite the right hand side of (6):

$$\begin{aligned}
&I(C, X_i) - [I(X_i, X_s) - I(X_i, X_s|C)] = H(C) - H(C|X_i) \\
&- [H(X_i) - H(X_i|X_s)] + H(X_i|C) - H(X_i|X_s, C) \\
&= H(C) - H(C|X_i) - H(X_i) + H(X_i|X_s) + H(X_i|C) - H(X_i|X_s, C) \\
&= H(X_i|X_s) - H(X_i|X_s, C) + H(C) - H(C|X_i) - [H(X_i) - H(X_i|C)] \\
&= I(C, X_i) - I(C, X_i) + H(X_i|X_s) - H(X_i|X_s, C).
\end{aligned} \tag{7}$$

The last term of (7) equals to $I(C, X_i|X_s)$.

The ratio of mutual information between the candidate feature X_i and the selected feature X_s and the entropy of X_s is a *measure of correlation* (also known as *coefficient of uncertainty*) between X_i and X_s [9]

$$CU_{X_i,X_s} = \frac{I(X_i, X_s)}{H(X_s)} = \big(1 - \frac{H(X_s|X_i)}{H(X_s)}\big), \tag{8}$$

$0 \leq CU_{X_i,X_s} \leq 1$. $CU_{X_i,X_s} = 0$ if and only if X_i and X_s are independent.

Proposition 2. Assume that conditioning by the class C does not change the ratio of the entropy of X_s and the MI between X_s and X_i, i.e., the following relation holds

$$\frac{H(X_s|C)}{I(X_i, X_s|C)} = \frac{H(X_s)}{I(X_i, X_s)}. \tag{9}$$

Then for the conditional mutual information $I(C, X_i|X_s)$ it holds:

$$I(C, X_i|X_s) = I(C, X_i) - CU_{X_i,X_s}\ I(C, X_s). \tag{10}$$

Proof: It follows from condition (9) and the definition (8) that

$$I(X_i, X_s|C) = CU_{X_i,X_s}\ H(X_s|C). \tag{11}$$

Using the equations (6) and (11) we obtain (10).

We can see from (10) that the second term is the weighted mutual information $I(C, X_s)$ with the weight equal to the measure of correlation CU_{X_i,X_s}. We propose the modification of the estimation $\tilde{I}(C, X_i|S)$ for $I(C, X_i|S)$ of the following form

$$\tilde{I}(C, X_i|S) = I(C, X_i) - \max_{X_s \in S} CU_{X_i,X_s}\ I(C, X_s). \tag{12}$$

It means that the best feature in the next step of the sequential forward search algorithm is found by maximizing (12)

$$X^{+} = \arg \max_{X_i \in X \setminus S} \{I(C, X_i) - \max_{X_s \in S} CU_{X_i,X_s}\ I(C, X_s)\}. \tag{13}$$

3 Proposed Feature Selection Algorithm

The sequential forward selection algorithm mMIFS-U based on the estimation of conditional mutual information given in (12) can be realized as follows:

1. *Initialization:*
   Set $S =$ "empty set", set $X =$ "initial set of all D features".
2. *Pre-computation:*
   For all features $X_i \in X$ compute $I(C, X_i)$.
3. *Selection of the first feature:*
   Find feature $X^{\star} \in X$ that maximizes $I(C, X_i)$;
   set $X = X \setminus \{X^{\star}\}$, $S = \{X^{\star}\}$.
4. *Greedy feature selection:*
   Repeat until the desired number of features is selected.
   (a) *Computation of entropy:*
   For all $X_s \in S$ compute entropy $H(X_s)$, if it is not already available.
   (b) *Computation of the MI between features:*
   For all pairs of features (X_i, X_s) with $X_i \in X$, $X_s \in S$ compute $I(X_i, X_s)$, if it is not yet available.
   (c) *Selection of the next feature:*
   Find feature $X^{+} \in X$ according to formula (13).
   Set $X = X \setminus \{X^{+}\}$, $S = S \cup \{X^{+}\}$.

4 Experiments and Results

Feature selection has been successfully applied to various problems including text categorization (e.g., [14]). The *text categorization* (TC) task (also known as *text classification*) is the task of assigning documents written in natural language into one or more thematic classes belonging to the predefined set $\mathcal{C} = \{c_1, \ldots, c_{|\mathcal{C}|}\}$ of $|\mathcal{C}|$ classes. The construction of a text classifier relies on an initial collection of documents pre-classified under $\mathcal{C}$. In TC, usually a document representation using the *bag-of-words* approach is employed (each position in the feature vector representation corresponds to a given word). This representation scheme leads to very high-dimensional feature space, too high for conventional classification methods. In TC the dominant approach to dimensionality reduction is feature selection based on various criteria, in particular *filter*-based FS.

Sequential forward selection methods MIFS, MIFS-U and mMIFS-U presented in Sections 3 and 2 have been used in our experiments for reducing vocabulary size of the vocabulary set $\mathcal{V} = \{w_1, \ldots, w_{|\mathcal{V}|}\}$ containing $|\mathcal{V}|$ distinct words occurring in training documents. Then we used the Naïve Bayes classifier based on multinomial model, linear Support Vector Machine (SVM) and k-Nearest Neighbor (k-NN) classifier.

4.1 Data set

In our experiments we examined the commonly used Reuters-21578 data set[1] to evaluate all considered algorithms. Our text preprocessing included removing all non-alphabetic characters like full stops, commas, brackets, etc., lowering the upper case characters, ignoring all the words that contained digits or non alphanumeric characters and removing words from a stop-word list. We replaced each word by its morphological root and removed all words with less than three occurrences. The resulting vocabulary size was 7487 words. The ModApte train/test split of the Reuters-21578 data contains 9603 training documents and 3299 testing documents in 135 classes related to economics. We used only those 90 classes for which there exists at least one training and one testing document.

4.2 Classifiers

All feature selection methods were examined in conjuction with each of the following classifiers:

Naïve Bayes. We use the multinomial model as described in [15]. The predicted class for document d is the one that maximizes the posterior probability of each class given the test document $P(c_j|d)$,

$$P(c_j|d) \propto P(c_j) \prod_{v}^{|\mathcal{V}|} P(w_v|c_j)^{N_{iv}}.$$

[1] http://www.daviddlewis.com/resources/testcollections/reuters21578.

Here $P(c_j)$ is the prior probability of the class c_j, $P(w_v|c_j)$ is the probability that a word chosen randomly in a document from class c_j equals w_v, and N_{iv} is the number of occurrences of word w_v in document d. We smoothed the word and class probabilities using Bayesian estimate with word priors and a Laplace estimate, respectively.

Linear Support Vector Machines. The SVM method has been introduced in TC by [16]. The method is defined over the vector space where the classification problem is to find the decision surface that "best" separates the data points of one class from the other. In case of linearly separable data the decision surface is a hyperplane that maximizes the "margin" between the two classes. The normalized word frequency was used for document representation:

$$tfidf(w_i, d_j) = n(w_i, d_j) \cdot \log\big(\frac{|\mathcal{D}|}{n(w_i)}\big), \tag{14}$$

where $n(w_i)$ is the number of documents in $\mathcal{D}$ in which w_i occurs at least one.

K-Nearest Neighbor. Given an arbitrary input document, the system ranks its nearest neighbors among training documents, and uses the classes of the k top-ranking neighbors to predict the class of the input document. The similarity score of each neighbor document to the new document being classified is used as a weight if each class, and the sums of class weights over the nearest neighbors are used for class ranking. The normalized word frequency (14) was used for document representation.

4.3 Performance Measures

For evaluating the multi-label classification accuracy we used the standard multi-label measures *precision* and *recall*, both *micro-averaged*. Estimates of micro-averaging precision and recall are obtained as

$$\hat{\pi}_{mic} = \frac{\sum_{j=1}^{|\mathcal{C}|} TP_j}{\sum_{j=1}^{|\mathcal{C}|}(TP_j + FP_j)}, \quad \hat{\rho}_{mic} = \frac{\sum_{j=1}^{|\mathcal{C}|} TP_j}{\sum_{j=1}^{|\mathcal{C}|}(TP_j + FN_j)}.$$

Here TP_j, (FP_j) is the number of documents correctly (incorrectly) assigned to c_j; FN_j is the number of documents incorrectly not assigned to c_j.

4.4 Thresholding

There are two variants of multi-label classification [17], namely ranking and "hard" classifiers. *Hard classification* assigns to each pair document/class (d, c_k) the value YES or NO according to the classifier result. On the other hand *ranking classification* gives to the pair (d, c_j) a real value $\phi(d, c_j)$, which represents the classifier decision for the fact that $d \in c_k$. Then we sort all classes for the document d according to $\phi(d, c_j)$ and the best τ_j classes are selected where τ_j is the threshold for the class c_j. Several thresholding algorithms to train the τ_j exist.

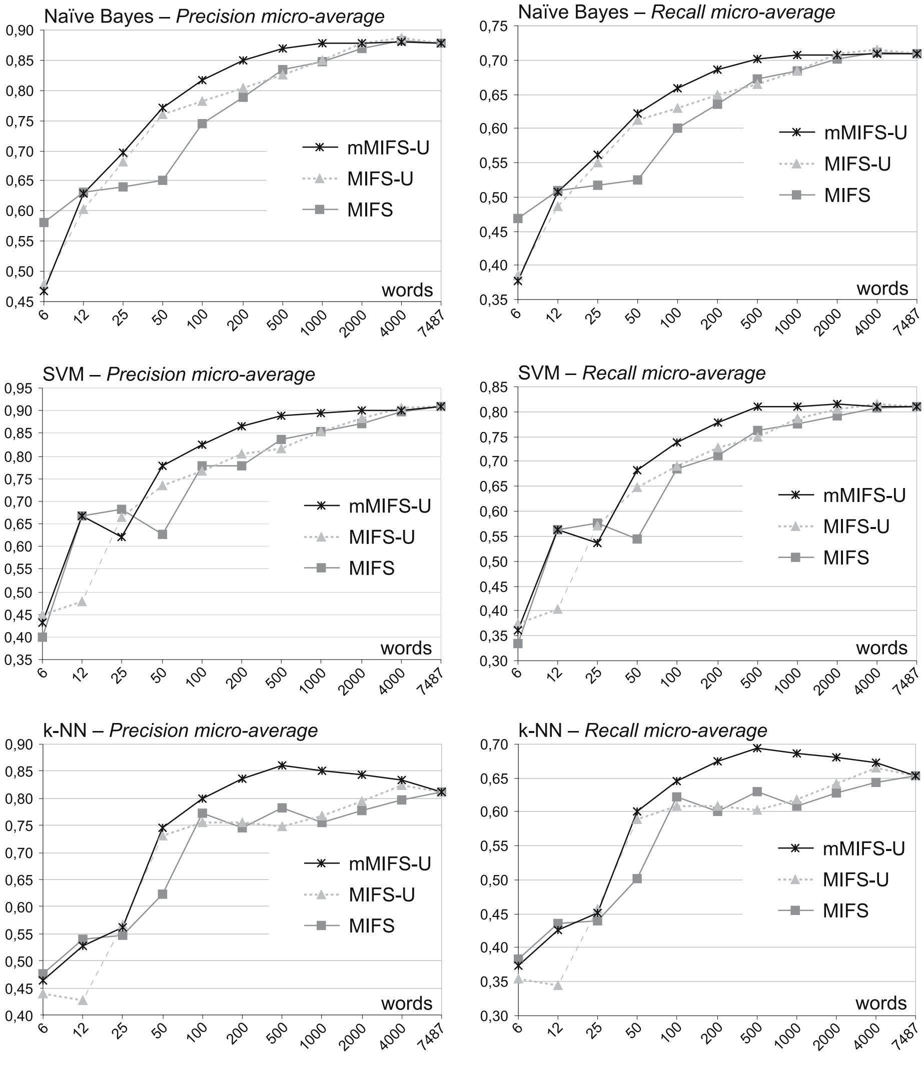

Fig. 1. Classifier performance on Reuters data (90 classes), with Apte split, and RCut-thresholding. Charts of micro-averaged precision, (left-side) and micro-averaged recall (right-side) of Naïve Bayes classifier (1st row), Support Vector Machine (2nd row) and k-Nearest Neighbour (3rd row). Horizontal axes indicate numbers of words.

The commonly used methods RCut, PCut and SCut are described and compared in the paper [18]. It is shown that thresholding has great impact on the classification result. However, it is difficult to choose the best method. We used the RCut thresholding, which sorts classes for the document and assigns YES to the best τ top-ranking classes. There is one global threshold τ (integer value between 1 and $|\mathcal{C}|$) for all classes. We set the threshold τ according to the average number of classes per one document. We used the whole training set for evaluating the value τ.

The Naïve Bayes and k-NN classifiers are typical tools for ranking classification, with which we used thresholding. In contrast, SVM is the "hard" classifier because there is one classifier for each class which distinguishes between that class and the rest of classes. In fact, SVM may assign a document to no class. In that case we reassign the document to such class that is best according to SVM class rating. This improves the classification result.

4.5 Experimental Results

In total we made 21 experiments, each experiment was performed for eleven different vocabulary sizes and evaluated by three different criteria. Sequential FS (SFS) is not usually used in text classification because of its computational cost due to large vocabulary size. However, in practice we can often either employ calculations from previous steps or make some pre-computations during initialization. Since FS is typically done in an off-line manner, the computational time is not as important as the optimality of the found subset of words and classification accuracy. The time complexity of SFS algorithms is less than $O(|\mathcal{V}^{'}||\mathcal{V}|^2)$ where $|\mathcal{V}^{'}|$ is the number of desired words and $|\mathcal{V}|$ is the total number of words in the vocabulary. The required space complexity is $S(|\mathcal{V}|^2/2)$ because we need to store the mutual information for all pairs of words (w_i, w_s) with $w_i \in \mathcal{V} \setminus S$ and $w_s \in S$. The charts in Figure 1 show the resulting micro-averaged precision and recall criteria. In our experiments the best micro-averaged performance was achieved by the new mMIFS-U methods using modified conditional mutual information.

5 Conclusion

In this paper we proposed a new sequential forward selection algorithm based on novel estimation of the conditional mutual information between the candidate feature and the classes given a subset of already selected features.

- Experimental results on textual data show that the modified MIFS-U sequential forward selection algorithm (mMIFS-U) performs well in classification as measured by precision and recall measures and that the mMIFS-U performs better than MIFS and MIFS-U on the Reuters data.
- In this paper we also present a comparative experimental study of three classifiers. SVM overcomes on average both Naïve Bayes and k-Nearest Neighbor classifiers.

Acknowledgements. The work has been supported by EC project No. FP6-507752, the Grant Agency of the Academy of Sciences of the Czech Republic (CR) project A2075302, and CR MŠMT grants 2C06019 and 1M0572 DAR.

References

1. Yu, L., Liu, H.: Feature selection for high-dimensional data: A fast correlation-based filter solution. In: Proceedings of the 20th International Conference on Machine Learning, pp. 56–63 (2003)
2. Dash, M., Choi, K., Scheuermann, P., Liu, H.: Feature selection for clustering - a filter solution. In: Proceedings of the Second International Conference on Data Mining, pp. 115–122 (2002)
3. Kohavi, R., John, G.: Wrappers for feature subset selection. Artificial Intelligence 97, 273–324 (1997)
4. Liu, H., Yu, L.: Toward integrating feature selection algorithms for classification and clustering. IEEE Transactions on Knowledge and Data Engineering 17(3), 491–502 (2005)
5. Dash, M., Liu, H.: Consistency-based search in feature selection. Artificial Intelligence 151(1-2), 155–176 (2003)
6. Jain, A.K., Duin, R.P.W., Mao, J.: Statistical pattern recognition: A review. IEEE Transactions on Pattern Analysis and Machine Intelligence 22, 4–37 (2000)
7. Battiti, R.: Using mutual information for selecting features in supervised neural net learning. IEEE Transactions on Neural Networks 5, 537–550 (1994)
8. Kwak, N., Choi, C.H.: Input feature selection for classification problems. IEEE Transactions on Neural Networks 13(1), 143–159 (2002)
9. Cover, T., Thomas, J.: Elements of Information Theory, 1st edn. John Wiley & Sons, Chichester (1991)
10. Fleuret, F.: Fast binary feature selection with conditional mutual information. Journal of Machine Learning Research 5, 1531–1555 (2004)
11. Peng, H., Long, F., Ding, C.: Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(8), 1226–1238 (2005)
12. Fano, R.: Transmission of Information: A Sattistical Theory of Communications. John Wiley and M.I.T.& Sons (1991)
13. Kwak, N., Choi, C.: Improved mutual information feature selector for neural networks in supervised learning. In: Proceedings of the IJCNN 1999, 10th International Joint Conference on Neural Networks pp. 1313–1318 (1999)
14. Forman, G.: An experimental study of feature selection metrics for text categorization. Journal of Machine Learning Research 3, 1289–1305 (2003)
15. McCallum, A., Nigam, K.: A comparison of event models for naive Bayes text classification. In: Proceedings of the AAAI-1998 Workshop on Learning for Text Categorization (1998)
16. Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS, vol. 1398, pp. 137–142. Springer, Heidelberg (1998)
17. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys 34(1), 1–47 (2002)
18. Yang, Y.: A study on thresholding strategies for text categorization. In: Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2001), New Orleans, Louisiana USA (September 9-12, 2001)