Methods of Artificial Intelligence, Gliwice, pp. 33-40, 2009.

Feature Selection for Supervised Classification: A Kolmogorov-Smirnov Class Correlation-Based Filter

Marcin Blachnik[1)], Włodzisław Duch[2)], Adam Kachel[1)], Jacek Biesiada[1,3)]
[1)] *Silesian University of Technology, Electrotechnology Department, Katowice, Krasinskiego 8, Poland;*
[2)] *Department of Informatics, Nicolaus Copernicus University, Grudziądzka 5, Toruń, Poland;*
[3)] *Division of Biomedical Informatics, Children's Hospital Research Foundation, Cincinnati, Ohio 45229*

Feature selection is a challenging problem for computational intelligence. Feature selection algorithms that wrap around learning system suffer from high computational complexity. In this paper a fast redundancy removal filter is proposed based on modified Kolmogorov-Smirnov statistic, utilizing class label information while comparing feature pairs. Kolmogorov-Smirnov Class Correlation-Based Filter proposed here is compared with other filters suitable for removal of redundancy, such as the Fast Correlation-Based Feature Filter (FCBF), Kolmogorov-Smirnov Correlation-Based Filter (K-S CBF) and simple ranking based wrapper. Obtained results do not differ significantly in comparison to basic K-S CBF, but are much better than the results of FCBF algorithm. Proposed algorithm can significantly reduce initial space in wrapper-based feature selection for high-dimensional problems.

Keywords: Feature selection, redundancy filter, classification, machine learning, Kolmogorov-Smirnov statistics

1. INTRODUCTION

Selection of information is one of the most challenging problems facing real-life applications of computational intelligence methods. Good prediction algorithms (Support Vector Machines, neural networks etc.) need tools for feature selection and training vector selection to work efficiently. There are many advantages of applying such techniques: improve overall accuracy, speeding up the training process, and reducing computational complexity of the data model, making it more comprehensible. Real world problems may have thousands of irrelevant features and in such situations feature selection should provide high reduction rate preserving important information hidden in the full dataset. Selection methods should also be fast, especially if they are used in iterative wrapper schemes. To face these requirements a modification of Fast Correlation-Based Feature (FCBF) filter [1] is proposed here, based on modified version of Komolgorov-Smirnov statistic used to reject redundant features.

An overview of various feature selection methods is presented first, describing algorithms used in our empirical experiments. The algorithm based on Komolgorov-Smirnov statistic is described in section 3., paying attention to the limitations of the proposed algorithm resulting from the Komolgorov-Smirnov statistic properties. Empirical comparison between various methods follows this section, with brief conclusion discussing the results and drawing further perspectives closing the paper.

2. FEATURE SELECTION OVERVIEW

Feature selection methods may be of supervised or unsupervised type. Search for the best subset of m features out of all n features is NP hard [2]. For large n exhaustive search testing all possible feature subsets

is prohibitively expensive, and many global search techniques, stochastic methods and sequential search techniques have been developed and implemented in this area [3].

Feature selection methods belong to one of the three main groups: embedded, filter or wrapper methods. Embedded methods are an integral part of specific predictors, for example decision trees or neural networks with regularization. Filters are completely independent of predictors, ranking features or selecting feature subset using some indices of relevance to measure the quality of selected feature subsets. Many statistical correlations, probability distributions, information theory and other types of measures are used to define filters [4]. These methods have low complexity and are universal, but are not always matched in an optimal way to a given predictor. Methods that check performance on various subsets wrapping themselves around particular predictors are called wrappers [2]. They may use incremental, decremental or mixed strategy to select feature subsets, but the necessity of repeatedly training and testing predictors each time a new subset is defined can be very time consuming. A combination of filters and wrappers, called frappers [4], is also possible, using filters first for feature ranking, and in the second stage adding new features in their ranking order and rejecting those that do not improve results of a given predictor.

An algorithm proposed below is a kind of a feature filter utilizing ranking information to remove redundant features. It may be used as a frapper in supervised learning problems.

2.1. Fast Correlation-Based Filter

Predominant correlation, proposed in [1] for the Fast Correlation-Based Filter (FCBF), is defined as follows. Consider relation between features-classes and between pairs of features. The algorithm has two levels. The first part is a typical ranking algorithm using SUC (Symmetric Uncertainty Coefficient) index [5] for estimation of class-feature relevance, and a threshold coefficient to select predominant (highly relevant) features. In the second part features that are redundant to the predominant features are removed. The algorithm is presented below.

2.2. Rank wrapper algorithm

For comparison with filter algorithms *rank-wrapper* (RW) algorithm is used. This algorithm belongs to the frapper category, using ranking coefficient in the filter part, and the performance of a given predictor to determine the k best features. The algorithm is sketched in (2): 1) ranking coefficient between each feature and class is calculated; 2) all features are ordered according to the value of the ranking coefficient; 3) the first feature ($k = 1$) is selected and the result of the predictor (Naive Bayes, 1NN, or SVM classifier) is evaluated on the current feature subset using stratified cross-validation. If the accuracy of a given model after adding k's feature increases the feature is added to the selected subset, $k \leftarrow k + 1$ and the next feature is tested. Testing if the increase is statistically significant can considerably decrease the size of the selected subsets.

3. KOLMOGOROV-SMIRNOV FEATURE FILTER DETAILS

Komolgorov-Smirnov Correlation Based Filter (K-S CBF) [6, 7] is a tool for robust feature selection. Typical approaches to feature selection are based on fast and computationally efficient feature ranking, but are not that stable and accurate, or are based on various computationally expensive search strategies. Some algorithms benefit from removing redundant features. In FCBF [1] (Algorithm 1) analysis of redundancy is based on pairwise correlations between features F_i and F_j, while in K-S CBF[6, 7] pairwise correlation between features has been replaced by goodness of fit between probability distributions of that features. If the probability distributions are similar then one with the higher ranking coefficient is taken to the final feature subset, and all those with lower ranking are removed from the final subset. In contrast to K-S CBF the K-S CCBF algorithm introduced here uses goodness of fit analysis not only between pairs of features (F_i and F_j), but also utilize class label information comparing probability distribution of features within classes.

Algorithm 1 Fast Correlation-Based Filter algorithm.

Require: $S(F, C) = S(F_1, F_2, \ldots, F_N, C)$ // a training class-labeled data
Require: γ // threshold value
  $m \leftarrow$ GetNumberOfFeatures(F);
  for $i = 1 \ldots m$ **do**
    $r_i \leftarrow$ CalculateRankingCoeficient(F_i, C)
    if $r_i < \gamma$ **then**
      $S \leftarrow S \setminus F_i$ // remove weak features
    else
      $R \leftarrow R \cup r_i$ // remember feature rank
    end if
  end for
  $S' \leftarrow$ SortFeaturesAccordingToRankValues(S, R)
  $R' \leftarrow$ SortRankValues(R)
  $i \leftarrow 1$
  while $(F_i \leftarrow$ GetFeature$(S, i)) \neq null$ **do**
    $j \leftarrow i + 1$
    while $(F_j \leftarrow$ GetFeature$(S, j)) \neq null$ **do**
      $j \leftarrow j + 1$
      $f =$ CalculateCorrelation(F_i, F_j)
      if $f < R_j$ **then**
        $S \leftarrow S \setminus F_j$
      end if
    end while
    $i \leftarrow i + 1$
  end while
  return F

Algorithm 2 Ranking Wrapper algorithm.

Require: $S(F, C) = S(F_1, F_2, \ldots, F_N, C)$ // training data
  $m \leftarrow$ GetNumberOfFeatures(F);
  for $i = 1 \ldots m$ **do**
    $R_i \leftarrow$ CalculateRankingCoeficient(F_i, C)
  end for
  $S' \leftarrow$ SortFeaturesAccordingToRankValues(S, R)
  for $k = 1 \ldots m$ **do**
    $S' \leftarrow$ FetFeatures$(S, 1 \ldots k)$
    $tmpAcc \leftarrow$ CalculateCVAccuracy(M, S')
    if $acc < tmpAcc$ **then**
      $acc \leftarrow tmpAcc$
    else
      return S'
    end if
  end for

3.1. Kolmogorov-Smirnov statistic

Komolgorov-Smirnov statistics (KS) [5] is based on calculating the highest difference between cumulative distributions between two random variables g and h:

$$KS(g, h) = \sqrt{\frac{n_g n_h}{n_g + n_h}} \sup_k |G_k - H_k| \tag{1}$$

where n_g, n_h is the number of samples for each random variable, k is the number of bins in discrete probability distribution, G and H are cumulative probability distributions of random variables g and h, respectively. Here random variables are a pair of features F_i and F_j taken from the dataset. If the null hypothesis is valid ($KS(F_i, F_j) < \lambda_\alpha$) one of the features is redundant and can be rejected from the feature subset.

Steps taken to calculate KS include:

- Discretization of both features F_i,F_j into k bins.
- Estimation of probabilities in each bin.
- Calculation of cumulative probability distributions for both features.
- Use of the equation 1 to calculate KS statistic.

3.2. Kolmogorov-Smirnov statistic for multi class problems

Because KS statistic is designed to compare only two variables, for our purpose it had to be redefined to utilize also information about classes. In our approach the KS statistics took the form:

$$KS_c(g,h) = \max_c \left(KS\left(g\left(c\right), h\left(c\right)\right)\right) \tag{2}$$

where c is the class label, $g(c)$ are samples of random variable g that belong to the class c, $h(c)$ are samples of random variable h that belong to class c. Calculated statistics is then compared with a threshold $KS_c(g,h) < \lambda_\alpha$ to discover possible redundancy between features.

3.3. Kolmogorov-Smirnov Class Correlation-Based Filter

The K-S Class Correlation-Based Filter (K-S CCBF) algorithm consists of two stages. In the first stage feature ranking is performed using one of the ranking coefficients. SUC coefficient has been selected because of it's stability (fast convergence of estimates to stable values [4]), it has also been used in the FCBF algorithm [1]. Results of ranking are sorted into descending ordered, ending the first stage. In the second stage redundant features are removed utilizing ranking information. In the loop features that are high in the ranking order are compared with all those with lower rankings checking the $KS_c < \lambda_\alpha$ condition. If this condition is true redundant feature lower in ranking is removed from the original set F. If the condition is not valid the algorithm leaves both features. After this analysis only the most significant features are left in the feature subset, and all redundant features are removed. The algorithm can be presented as follows:

K-S CBF Algorithm:

Relevance analysis

1. Order features based on decreasing value of SUC(f,C) index.

Redundancy analysis

2. Initialize F_i with the first feature in the list
3. Find and remove all features for which F_i forms an approximate redundant cover using K-S test.
4. Set F_i as the next remaining feature in the list and repeat step 3 until the end of the list.

Fig. 1. A two-step Kolmogorov-Smirnov Correlation Based Fiter (K-S CBF) algorithm.

Algorithm 3 Kolmogorov-Smirnov Class Correlation-Based Filter algorithm.

Require: $S(F_1, F_2, \ldots, F_N, C)$ // training data
Require: λ_α // threshold value
  $m \leftarrow$ GetNumberOfFeatures(F);
  for $i = 1 \ldots m$ **do**
    $R_i \leftarrow$ CalculateSUCCoeficient(F_i, C)
  end for
  $S' \leftarrow$ SortFeaturesAccordingToRankValues(S, R)
  $i \leftarrow 1$
  while $(F_i \leftarrow$ GetFeature$(S, i)) \neq null$ **do**
    $j \leftarrow i$
    while $(F_j \leftarrow$ GetFeature$(S, j)) \neq null$ **do**
      $j \leftarrow j + 1$
      $ks =$ CalculateKSCStatistic(F_i, F_j)
      if $ks < \lambda_\alpha$ **then**
        $S \leftarrow S \setminus F_j$ // remove feature F_j
      end if
    end while
    $i \leftarrow i + 1$
  end while
  return S

3.4. Limitations of K-S CCBF

K-S CCBF algorithm has several limitations. First, K-S statistic requires features to be of ordinal types, so it cannot be used with symbolic or nominal features. This limitation results from the fact that cumulative distributions may vary with different ordering of values, causing the problem of comparison of maximum differences between cumulative functions.

However, such problem does not appear if the analyzed dataset has all features of the same type. In this case ordering all symbolic values in the same way allows for use of K-S statistic because it is relatively constant. Still the fact that cumulative distribution may vary according to the ordering of symbolic values may cause fluctuations in the number of selected features in the final set, and this creates ambiguity how to fix the threshold value λ_α that determines the validity of the hypothesis.

Second important limitation arises from the sensitivity of cumulative probability distribution to linear transformations. If probability distributions of features f_i and f_j are related by a simple shift $f_j = f_i + a$, where a is a constant, then the test $KS(f_i, f_j)$ will reject null hypothesis of equality of both distributions, while in fact such features are redundant. This problem applies to all goodness of fit tests considered to analyze relation between features. This problem can be partially solved by shifting the mean to zero, and discretization of the features. This does not guarantee full invariance to linear transformations. In particular if the two features are asymmetrical $f_i = -f_j$ then in general the KS statistic will also reject null hypothesis. The best solution is to search for a, b parameters that minimize $KS(f_j, bf_i + a)$ rather than to calculate $KS(f_j, f_i)$ directly, but we have not implemented that.

4. DATASETS AND RESULTS

To verify proposed algorithm the 10-fold cross-validation tests were done wrapping feature selection algorithm and the classification algorithm, both embedded in each fold. The Infosel++ [8]library for feature selection developed by our group has been combined with the Spider Matlab toolbox [9]. Infosel++ is a powerful library designed to support feature selection and algorithm implementation based on information theory and statistical methods. Three algorithms from the Infosel++ library were used, K-S CCBF introduced here, K-S CBF and FCBF. Results obtained for K-S CCBF are also compared with results obtained

from FCBF (Weka implementation [10]), using also the Spider Toolbox to assure equal testing environment. Three quite different classifiers have been selected: the nearest neighbor algorithm (1NN), SVM with Gaussian kernel (LibSVM implementation with C and γ parameters optimized using grid search [[11]]), and Weka implementation of the Naive Bayes classifier. The reference for comparison of the K-S based methods is the ranking wrapper. The purpose here is not to obtain the best results with the most sophisticated tools, but rather to check the influence of redundancy removal on performance, therefore such limited arrangement should be sufficient. In both KS based filters λ_α value was fixed at 0.05. For all problems the cost function was defined as the balanced error rate calculated as the mean error over all classes.

7 datasets from the UCI repository have been used [12] in tests. These datasets are quite diverse, with the number of features ranging from 13 to 60 (nominal, real and mixed), and the number of vectors from 106 to 4601 (Tab.(**1**)). For Cleveland Heart Disease dataset 6 vectors with the missing values have been removed.

Dataset	#No. Features	No. Instances	#No. Classes	Class distribution	Balanced dataset
Heart Disease	13	297	2	160/137	No
Splice	60	3190	3	767/768/1655	No
Promoters	57	106	2	53/53	Yes
Thyroid	21	3772	3	93/191/3488	No
Sonar	60	208	2	97/111	No
Ionosphere	33	351	2	225/126	No
Spam	57	4601	2	1813/2788	No

Table 1. Datasets used in the experiments.

Table (**2**) shows both the mean balanced error rate for each classifier and the mean number of selected features. This allows to analyze not only the prediction accuracy but also the efficiency of the feature selection.

5. CONCLUSIONS

Redundancy has detrimental effect on methods based on distances, and for the kNN classifier results presented in Table **2** show that removing redundancy may indeed have significant effect. K-S CCBF demonstrated significant improvement over K-S CBF for a few datasets (Ionosphere, Sonar and Spam), marginally worse for Heart and Thyroid, and significantly worse for Promoters and Splice datasets. The last two datasets have only nominal features and this evidently leads to unstable behavior of the KS-statistic. The solution is to convert the data to numerical form using probabilistic distance functions first. On all datasets except Thyroid both K-S statistics based filters were better comparing with the original FCBF algorithm. As expected, on average rank wrapper algorithm is the best among tested feature selection algorithms. Ranking wrapper was driven by selection of feature subset in order to fit to a particular classifier. However, it requires much higher computational effort. Analysis of a number of selected features by different algorithms leads to the conclusion that K-S CCBF tends to select more features than K-S CBF, and FCBF algorithm. It is probably due to the λ_α value, that has not been renormalized to the modified statistic defined in 2. Statistical significance at 0.05 level selected used in all tests is not optimal and for some datasets too few or too many features were selected, leading to lower accuracy. Optimization of this parameter will increase complexity of the algorithm. FCBF usually was able to select even less features then the K-S CBF method. Direct comparison of the number of features selected by filters and wrappers is not possible because for wrappers the number of selected features significantly vary for each classifier.

Based on this study, and earlier correlation based filters (CBF) [13], accuracy improvement should not be expected, but rejection of correlated features carrying similar information may lead to significant reduction

Table 2. Mean balanced error rates and mean number of features for 1NN, Naive Bayes and linear SVM classifiers for 3 filters and a ranking wrapper. bold face - best results, italics - worst

Dataset	Algorithms						
	Classifier	FCBF	K-S CBF	K-S CCBF	No FS	Ranking Wrapper	
						BER	Feat. No.
Heart	kNN	*23.14 ± 7.21*	22.05 ± 9.39	22.94 ± 8.63	22.94 ± 8.63	**20.60 ± 9.86**	3.9 ± 2.85
	NBC	17.02 ± 7.64	16.09 ± 6.38	*17.42 ± 5.16*	*17.42 ± 5.16*	**14.21 ± 6.71**	11.7 ± 0.48
	SVM	*20.53 ± 8.20*	**16.80 ± 5.96**	18.88 ± 6.09	18.88 ± 6.09	18.98 ± 6.35	11.2 ± 3.05
	N. feat.	5.2 ± 0.63	12.10 ± 0.57	13.00 ± 0.00	13.00 ± 0.00	-	
Ionosphere	kNN	17.64 ± 3.31	**11.86** ± 3.92	12.59 ± 5.32	18.05 ± 7.52	15.46 ± 5.87	13.00 ± 5.12
	NBC	15.08 ± 4.86	13.38 ± 4.57	**10.58** ± 3.61	*16.49 ± 7.03*	16.49 ± 7.03	32.90 ± 0.32
	SVM	14.79 ± 5.11	10.55 ± 4.87	6.69 ± 5.03	6.56 ± 5.71	**4.80** ± 3.34	30.40 ± 1.26
	N. feat.	2 ± 0.00	7.10 ± 0.32	15.2 ± 1.40	33.00 ± 0.00	-	
Sonar	kNN	*37.46 ± 11.9*	18.07 ± 7.54	13.53 ± 8.27	**13.21** ± 6.96	16.27 ± 7.35	51.00 ± 7.80
	NBC	31.16 ± 7.61	30.84 ± 7.55	31.19 ± 7.77	*31.29 ± 8.93*	**27.69 ± 7.99**	54.30 ± 3.34
	SVM	*28.72 ± 7.90*	21.23 ± 5.64	11.34 ± 9.59	**9.34 ± 9.15**	10.75 ± 8.50	55.00 ± 0.00
	N. feat.	1.1 ± 0.32	9.40 ± 1.08	23.70 ± 1.77	60.00 ± 0.00	-	
Spam	kNN	15.28 ± 2.86	*21.31 ± 0.99*	10.88 ± 1.37	9.34 ± 1.36	**9.14 ± 1.54**	54.40 ± 0.97
	NBC	*20.68 ± 1.92*	19.4 ± 2.71	17.98 ± 1.67	**17.63 ± 1.52**	18.62 ± 1.72	34.00 ± 23.1
	SVM	10.71 ± 1.25	*13.9 ± 1.41*	9.57 ± 1.37	**6.71 ± 1.43**	6.74 ± 1.39	55 ± 0
	N. feat.	17.2 ± 0.92	7.20 ± 0.42	19.40 ± 0.70	57.00 ± 0.00	-	
Thyroid	kNN	**20.53 ± 8.85**	36.50 ± 4.38	36.99 ± 7.38	*37.89 ± 8.37*	29.07 ± 4.70	7.00 ± 4.62
	NBC	*39.23* ± 7.08	27.06 ± 4.47	26.88 ± 4.36	26.22 ± 4.98	**25.88 ± 5.04**	19.60 ± 0.84
	SVM	*29.81 ± 5.42*	**17.96 ± 5.88**	21.69 ± 6.00	19.28 ± 7.02	17.98 ± 6.95	19.00 ± 0.00
	N. feat.	3.7 ± 0.67	9.00 ± 0.00	11.70 ± 0.48	21.00 ± 0.00	-	
Promoters	kNN	**15.00 ± 8.20**	17.17 ± 12.4	24.50 ± 15.8	*27.33 ± 13.0*	18.50 ± 8.94	46.70 ± 3.80
	NBC	13.00 ± 11.7	18.50 ± 14.6	*18.67 ± 15.7*	11.00 ± 11.6	**11.0 ± 10.9**	50.30 ± 3.74
	SVM	**12.17 ± 8.39**	15.33 ± 11.2	28.17 ± 18.8	*50.00 ± 0.00*	15.00 ± 11.2	7 ± 0
	N. feat.	6 ± 0.82	3.50 ± 0.53	8.40 ± 0.52	57.00 ± 0.00	-	
Splice	kNN	19.34 ± 2.81	**17.19 ± 1.97**	28.10 ± 2.85	*29.42 ± 4.28*	26.43 ± 3.61	45.10 ± 2.13
	NBC	7.88 ± 1.72	*8.96 ± 1.92*	**7.04 ± 1.32**	7.19 ± 1.21	7.22 ± 1.20	54.60 ± 4.95
	SVM	33.56 ± 3.96	11.68 ± 2.45	49.36 ± 2.54	*50.27 ± 2.43*	**6.36 ± 1.87**	7.0 ± 0
	N. feat.	21.1 ± 0.74	13.00 ± 0.66	45.10 ± 1.59	60.00 ± 0.00	-	

in the number of selected features, preserving the original information. Earlier approaches to CBF were based on correlation analysis between pairs of features (FCBF) while in here, as proposed by Biesiada and Duch [6] the correlation measure may be replaced by the Kolmogorov-Smirnov statistic. Results obtained here support this idea, with the exception of datasets that have nominal or binary values (Promoters, Splice). The improved K-S CCBF algorithm proposed in this paper is similar to K-S CBF algorithm. They are both based on K-S statistic, however K-S CCBF utilizes also the class label information comparing pairs of features. Application of KS-statistic including external variable information such as class labels required modification of the original statistic.

In future we plan to test and compare other feature selection redundancy analysis algorithms such as [14], [15], and wrappers [16] using different searching strategies on real high-dimensional datasets like microarray data, or datasets provided for NIPS 2003 feature selection challenge with up to 100,000 features. Moreover, we plan to use feature filters (especially redundancy removal filters) as initial step in search for optimal feature selection. In that case, feature subsets obtained after redundancy removal may be used for initialization of various search strategies using wrappers.

Acknowledgement: This work was supported by the Polish Committee for Scientific Research grant 2007-2010 No.: N N519 1506 33. We are grateful to the referees for interesting remarks.

REFERENCES

[1] Yu, L., Liu, H.: Efficient feature selection via analysis of relevance and redundancy. Journal of Machine Learning Research (2004) 1205–1224

[2] Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L. (eds): Feature extraction, foundations and applications. Springer, Berlin, 2006.

[3] Somol, P., Pudil, P.: Feature selection toolbox. Pattern Recognition **35**, 2749–2759, 2002.

[4] Duch, W.: Filter methods. In Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L., eds.: Feature extraction, foundations and applications. Springer, pp. 89–118, 2006.

[5] Sheskin D.: Handbook of parametric and nonparametric statistical procedures, CRC Press, 2004.

[6] Biesiada, J., Duch, W., Feature Selection for High-Dimensional Data: A Kolmogorov-Smirnov Correlation-Based Filter Solution" Advances in Soft Computing, Computer Recognition Systems, pp. 95-105, 2005.

[7] Biesiada J, Duch W, A Kolmogorov-Smirnov correlation-based filter solution for microarray gene expressions data. Springer Lecture Notes in Computer Science Vol. 4985, 285-294, 2008.

[8] Kachel, A., Blachnik, B., Duch, W., Biesiada, J., Infosel++: Information Based Feature Selection C++ Library, Software available at http://metet.polsl.pl/~jbiesiada/infosel - in preparation.

[9] Weston J., Elisseeff A., BakIr G. and Sinz F., Spider 1.71 2006, Software available at http://www.kyb.tuebingen.mpg.de/bs/people/spider/

[10] Witten I., Frank E. Data mining – practical machine learning tools and techniques with JAVA implementations. Morgan Kaufmann Publishers, 2000.

[11] Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm

[12] UCI repository of machine learning databases http://www.ics.uci.edu.pl/~mlearn/MLRespository.html. Irvine, CA: University of California, Department of Information and Computer Science, 1998.

[13] Hall M.A., Smith L.A. Feature subset selection: a correlation based filter approach. In N. Kasabov and et al., editors, Proc Fourth International Conference on Neural Information Processing and Intelligent Information Systems, pages 855-858, Dunedin, New Zealand, 1997.

[14] Battiti R., Using mutual information for selecting features in supervised neural net learning. IEEE Trans. on Neural Networks, 5:537–550, 1994.

[15] Peng H., Long F., and 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.

[16] Kohavi R. and John G., Wrappers for Feature Subset Selection. In Artificial Intelligence journal, special issue on relevance, Vol. 97(1-2), 273-324, 1997.