Fast Correlation Based Filter (FCBF) with a Different Search Strategy

Baris Senliol[1], Gokhan Gulgezen[1], Lei Yu[2] and Zehra Cataltepe[1]

[1]Istanbul Technical University, Computer Engineering Department, Istanbul, Turkey
[2]Binghamton University, Computer Engineering Department, Binghamton, NY, USA
{senliol, gulgezen, cataltepe}@itu.edu.tr, lyu@cs.binghamton.edu

***Abstract*: In this paper we describe an extension of the information theoretical FCBF (Fast Correlation Based Feature Selection) algorithm. The extension, called FCBF#, enables FCBF to select any given size of feature subset and it selects features in a different order than the FCBF. We find out that the extended FCBF algorithm results in more accurate classifiers.**

I. INTRODUCTION

Feature selection methods have been used since 70s in the fields of statistics and pattern recognition [1]. Especially with the wide spread use of machine learning techniques, in fields like document processing and bioinformatics, where a lot of features are available, feature selection methods became popular [1]. Feature selection methods are needed when the number of training examples are too little, or when there are too much data that can be processed efficiently by the machine learning algorithms, or when some features are costly to acquire and hence the minimum number of features are preferred, or when there are noisy or irrelevant features in the data [4].

Feature selection methods have been shown to be effective in removing redundant and irrelevant features, improving learning algorithm's prediction performance, reducing the effects of curse of dimensionality, increasing the understandability of the data and helping to avoid slow execution time of learning algorithms [5].

According to their working principles, feature selection methods can be divided into two: methods which select the best subset of features that has a certain number of features and methods which select the best subset of features according to their own principles, independent of outside size measures [5].

Feature selection methods can also be divided into 3 classes by their interactions with learning algorithms [2, 4, 9]. If a feature selection method works independent from the learning algorithm, it is called a filter method. Filter methods are fast, scalable and can be used with any learning algorithm efficiently because of their independency from them [4, 6, 9]. If a feature selection method uses a learning algorithm to guide its search process to weigh the features, it is called a wrapper method. Wrapper methods are less scalable, may overfit the data more and are slower than filters because they require classifier training and validation. However they result in features producing better accuracy with the specific learning algorithm [4, 6, 8, 9]. If a feature selection method is embedded into a learning algorithm and optimized for it, it is called an embedded method. These methods are faster than wrappers and make the most efficient selection for the learning algorithm that they collaborate with [4, 8].

All feature selection methods go through some phases during selection process which are called the characteristic properties of the feature selection method [6, 8]. There are a total of 6 main characteristic properties which are initial state of search, creating successors, search strategy, feature evaluation method used, including or not including the interdependence of features and halting criterion.

Initial state [6] is the condition of initial subset node in the search tree. It can be empty, full of all features or can be filled with randomly selected features. Creating successors of a state [8] is about making a forward or backward feature selection. The number of features that successors can have is determined according to this criterion. There are also compound methods which combine both forward and backward methods, so successor states may have different number of features than each other. Search strategy [8, 9] is the strategy used to travel in the search tree. It can be exponential (exhaustive) [8], sequential [10] or randomized [11]. Feature evaluation methods [8, 9] are used to give every feature a weight so that features can be compared to each other and an efficient selection is possible. Using interdependence of features [4] determines whether the selection method is univariate or multivariate. Univariate methods only calculate the weight of features with their dependences to classes; multivariate methods calculate, in addition to class relevance, the dependency between each feature pair. Finally, the halting criterion [6, 9] is used to decide on when to stop searching. As mentioned before a certain value k can be given to the method to stop after selecting k features or an internal criterion can be used to stop searching for the best feature subset. All of these characteristic properties determine how a feature selection method works.

In this paper, we introduce a modification to the FCBF (Fast Correlation Based Filter) [3] feature selection algorithm. The modified algorithm, FCBF#, has a different search strategy than the original FCBF and it can produce more accurate classifiers for the size k subset selection problem. We also compare FCBF# to MRMR [7] feature selection algorithm and find out that it results in comparable performance.

II. FEATURE SELECTION ALGORITHMS

Although there is a significant number of size k and best subset feature selection methods, in the scope of our work we study 3 different algorithms. First one is MRMR [7] which is an efficient size k feature selection method and the second one is FCBF [3] which is a fast best feature subset selection method. We also introduce a new approach to FCBF (FCBF#) which turns FCBF from best subset selection method into a size k feature selection method.

A. MRMR

MRMR (Minimum Redundancy Maximum Relevance Feature Selection) [7] is a multivariate (see section 1) feature selection method which starts with an empty set, uses mutual information to weight features and forward selection technique with sequential search strategy to find the best subset of features. It has a parameter k which enables it to stop when there are k features in the selected feature subset.

Mutual Information (MI) [7] is a symmetrical information theoretic measure that measures the amount of information that can be obtained about one random variable by observing another. The mutual information of feature f_i relative to feature f_j is given by:

$$I(f_i; f_j) = \sum_{x,y} p(x, y) \log \frac{p(x, y)}{p(x) p(y)} \quad (1)$$

where x is all possible values of f_i and y is all possible values of f_j.

B. FCBF

FCBF (Fast Correlation Based Filter) [3] is a multivariate (see section 1) feature selection method which starts with full set of features, uses symmetrical uncertainty to calculate dependences of features and finds best subset using backward selection technique with sequential search strategy. It has an inside stopping criterion that makes it stop when there are no features left to eliminate. It is a correlation based feature subset selection method which runs, in general, significantly faster than other subset selection methods [3].

Symmetrical Uncertainty (SU) is a normalized information theoretic measure which uses entropy and conditional entropy values to calculate dependencies of features. If X is a random variable and $P(x)$ is the probability of x, the entropy of X is:

$$H(X) = -\sum_i P(x_i) \log_2 \left(P(x_i)\right) \quad (2)$$

Conditional entropy or conditional uncertainty of X given another random variable Y is the average conditional entropy of X over Y:

$$H(X \mid Y) = -\sum_j P(y_j) \sum_i P(x_i \mid y_j) \log_2 \left(P(x_i y_j)\right) \quad (3)$$

$$SU(X, Y) = 2 \left[\frac{H(X) - H(X \mid Y)}{H(X) + H(Y)} \right] \quad (4)$$

An SU value of 1 indicates that using one feature other feature's value can be totally predicted and value 0 indicates two features are totally independent. The SU values are symmetric for both features. In order to calculate SU values, features must be nominal but continuous features can also be used if their values discretized properly [3].

C. A NEW APPROACH TO FCBF: FCBF#

Previous experiments [3] show that FCBF is an efficient and fast algorithm which uses interdependence of features together with the dependence to the class. It selects best subset of features from the full set by means of backward elimination. Especially when inputs are highly correlated, this method may eliminate too many features. We introduce a new approach where we change the elimination method and the new algorithm is called FCBF# [12].

FCBF# changes FCBF's quick and sharp elimination method to a more balanced one to select the best subset which has k features. In FCBF [3], it can be seen that predominant features eliminate all features which has higher correlation to predominant feature than the class label in every step of iteration. It is an efficient way to choose a final subset but if you want to have a size k subset, this method is not efficient as it can eliminate features which are highly correlated with the classes in the first rounds. But if we want to have a size k subset, these features must be eliminated as late as possible.

FCBF# achieves this goal by giving every feature a temporary predominance in the elimination process and making them start eliminating features from the features which are least correlated with the class. Also opposed to FCBF where a feature eliminates all features which have more correlation to that feature than class label, every feature has a chance to eliminate only one feature in every step of iteration which makes elimination process more balanced. After all features have their chances, elimination process starts from the beginning with the remaining features and continues as long as no elimination can be made in a full round or enough elimination is made to get a size k feature subset. Another key difference between FCBF and FCBF# is that FCBF finishes its work after finishing all the iterations but FCBF# starts all over again and again until no elimination can be made or the desired set with k features is determined.

In FCBF#, the comparison criterion which leads to elimination is not changed so its multivariate structure and evaluation method is preserved. FCBF# changes only the search strategy of FCBF so it has the most of the main characteristics of FCBF including selecting a best subset of features instead of size k and main structure of algorithm to preserve efficient running time.

Listing 1 shows the FCBF# algorithm in detail.

III. EXPERIMENTS AND RESULTS

In order to evaluate the performance of FCBF# algorithm, 8 different datasets were used. In the experiments 5 different k size subsets (5, 10, 20, 35 and 50 percent of all features) for every dataset were produced. 10-fold cross validated 3-NNC classifiers were chosen to evaluate classification accuracy.

```
Algorithm: FCBF#
Input:   S(f_1, f_2...f_N)   //training data
         th                  //threshold for pre-elimination
         k                   //size k of feature subset
Output:  I                   //feature subset
begin
I=empty;
for i=1 to N begin
      SU_ic =calculateSU(f_i, C);
      if(SU_ic >th)
            addto(I, f_i);
      end
I=order I descending using features' SU_ic values;
count=0;
endcount=-1;
flag=0;
while flag<>1 AND endcount<>count begin
      count=endcount;
      f_p =firstelement(I);
      while f_p <>NULL AND flag<>1 begin
            f_q =lastelement (I);
            pass=0;
            while f_q <>NULL AND pass <>1 begin
                  if(f_p == f_q ) break;
                  if(calculateSU(f_p, f_q)>= SU_qc )
                        delete(I, f_q);
                        pass=1;
                        count=count+1;
                        if (| I |==k) flag=1;
                  else f_q =previous(I, f_q);
                  end
            f_p =next(I, f_p);
            end
      end
end
```

Listing 1. FCBF# Algorithm

For evaluation process every feature is discretized to ten bins. Experiments were performed in MATLAB with PRTOOLS [16] on a computer with 2.13 GHz processor.

In the experiments, 7 datasets are from the existing databases. Profeat is a dataset which contains the physicochemical properties of 785 amino acid sequences. References to the sources and the general characteristic of datasets can be found in Tables I and II.

TABLE I
DATASETS GENERAL CHARACTERISTICS

Datasets	Sample #	Feature #	Class #
Isolet [14]	7797	617	26
Colon [13]	62	2000	2
Lymphoma [13]	62	4026	3
Musk-Clean2 [3]	6598	166	2
Profeat [15]	785	1447	5
Prost [13]	102	6033	2
Srbct [13]	63	2308	4
Multi-Feat [3]	2000	649	10

TABLE II
DATASETS CORRELATION PROPERTIES (MI)

	Intercorrelations		Class Correlations	
Datasets	Mean	Median	Mean	Median
Isolet	0.0683	0.0328	0.4059	0.3310
Colon	0.8201	0.8233	0.1358	0.1255
Lymphoma	0.9155	0.9193	0.4146	0.3831
MuskClean2	0.3776	0.3154	0.0403	0.0379
Profeat	0.0861	0.0745	0.0540	0.0510
Prost	0.8128	0.7952	0.1166	0.1064
Srbct	0.7095	0.7198	0.4324	0.4194
Multi-Feat	0.0943	0.0607	0.4323	0.4011

In the experiments, first the difference between FCBF and FCBF# when selecting best subsets is evaluated. Second, the difference between elimination orders of FCBF# and FCBF is examined. A stop counter [12] is used in FCBF algorithm to get size *k* feature subsets. Finally, FCBF# is compared with MRMR algorithm which is one of the most recent feature selection algorithms.

A. FCBF vs. FCBF# - Best subset selection

In Tables III and IV, it can be seen that FCBF# selects fewer number of features but it takes to run longer than FCBF. But compared to the the running time difference with other algorithms such as CorrFS, ReliefF and ConFS with FCBF [3], the difference between FCBF and FCBF# is not significant. FCBF selects a very small number of features when the intercorrelations are high and feature's relevance to class labels is low. FCBF# selects smaller subsets than FCBF as it gives predominance [3] and hence possibility of elimination to every feature and as a result, the accuracy values generally become lower than FCBF.

B. FCBF vs. FCBF# - Size k feature subset selection

Table V shows that generally FCBF# performs better than FCBF when a certain percentage of features is requested. FCBF#'s accuracy values are higher than FCBF and more stabilized in general because of its balanced elimination of features. This shows that FCBF#'s elimination method is better than a FCBF stop counter method and can be considered as an extension to the algorithm.

C. FCBF# vs. MRMR

In Table VI, it can be seen that FCBF# and MRMR shows close results in all datasets especially for small *k* values. This shows that FCBF# gives results as good as one of the most efficient size *k* feature selection algorithms and proves its efficiency as a size *k* feature selection algorithm.

TABLE V
ACCURACY COMPARISON OF FCBF AND FCBF# FOR DIFFERENT k VALUES

	Selected Feature's Percent to Whole									
	5		10		20		35		50	
Datasets	FCBF	FCBF#	FCBF	FCBF#	FCBF	FCBF#	FCBF	FCBF#	FCBF	FCBF#
Isolet	**0.861**	0.809	**0.869**	0.86	0.833	**0.87**	0.814	**0.866**	0.805	**0.867**
Colon	0.738	**0.796**	0.733	**0.796**	0.7	**0.813**	0.7	**0.796**	0.65	**0.763**
Lymphoma	0.921	**0.983**	0.871	**0.983**	0.854	**0.983**	0.871	**0.983**	0.921	**0.983**
Musk-Clean2	0.932	**0.94**	0.95	**0.961**	0.947	**0.961**	0.949	**0.96**	0.958	**0.96**
Profeat	0.372	**0.47**	0.38	**0.463**	0.374	**0.462**	0.389	**0.454**	0.361	**0.431**
Prost	0.667	**0.883**	0.723	**0.885**	0.7	**0.893**	0.72	**0.883**	0.691	**0.863**
Srbct	0.947	**1**	0.913	**1**	0.863	**0.983**	0.833	**0.93**	0.783	**0.937**
Multi-Feat	**0.919**	0.908	**0.929**	0.924	**0.934**	0.929	0.933	**0.943**	0.938	**0.943**

TABLE VI
ACCURACY COMPARISON OF MRMR AND FCBF# FOR DIFFERENT k VALUES

	Selected Feature's Percent to Whole									
	5		10		20		35		50	
Datasets	MRMR	FCBF#	MRMR	FCBF#	MRMR	FCBF#	MRMR	FCBF#	MRMR	FCBF#
Isolet	**0.837**	0.809	**0.868**	0.86	**0.897**	0.87	**0.911**	0.866	**0.911**	0.867
Colon	0.796	0.796	0.796	0.796	0.8	**0.813**	**0.8**	0.796	0.733	**0.763**
Lymphoma	**1**	0.983	0.983	0.983	0.983	0.983	0.983	0.983	0.983	0.983
Musk-Clean2	0.938	**0.94**	0.95	**0.961**	0.948	**0.961**	0.954	**0.96**	**0.961**	0.96
Profeat	0.43	**0.47**	0.46	**0.463**	**0.484**	0.462	**0.489**	0.454	**0.452**	0.431
Prost	0.883	0.883	0.865	**0.885**	0.883	**0.893**	**0.885**	0.883	**0.883**	0.863
Srbct	1	1	1	1	0.983	0.983	**0.947**	0.93	0.937	0.937
Multi-Feat	0.797	**0.908**	0.905	**0.924**	**0.935**	0.929	**0.947**	0.943	**0.945**	0.943

TABLE III
ACCURACY AND TIME OF FCBF AND FCBF#

	Acc		Time (s)	
Dataset	FCBF	FCBF#	FCBF	FCBF#
Isolet	**0.872**	0.804	**19.29**	109.68
Colon	**0.771**	0.754	**0.664**	0.81
Lymphoma	**0.95**	0.913	**7.49**	8.39
Musk-Clean2	0.880	0.880	**0.909**	1.069
Profeat	**0.351**	0.341	**1.31**	1.42
Prost	**0.915**	0.912	**12.51**	14.27
Srbct	**0.933**	0.93	**4.68**	5.85
Multi-Feat	**0.934**	0.929	**11.9**	55.5

TABLE IV
SELECTED FEATURE # AND PERCENT OF FCBF AND FCBF#

	Mean Feat#		Feat# percent	
Dataset	FCBF	FCBF#	FCBF	FCBF#
Isolet	57.8	31.2	0.09	0.05
Colon	1.7	1.8	0.0009	0.0009
Lymphoma	4.6	2.6	0.0011	0.0006
Musk-Clean2	1	1	0.006	0.006
Profeat	6	1.7	0.004	0.001
Prost	1.9	1.4	0.0002	0.0001
Srbct	12.4	7	0.003	0.005
Multi-Feat	148.9	121.2	0.23	0.19

IV. CONCLUSION

In this paper, we define a new feature selection method which is called FCBF#. The new method transforms FCBF from a subset selection method to a size k subset selection method. We compare FCBF# with FCBF in best subset selection, FCBF with stop counter in size k feature subset selection. We also perform experiments with MRMR algorithm to compare FCBF#'s performance against an efficient and recent feature selection method.

REFERENCES

[1] John, G., Kohavi, R., Pfleger, K. Irrelevant features and the subset selection problem. *In Proceedings of the Eleventh International Machine Learning Conference* (1994) 121–129. New Brunswick, NJ: Morgan Kaufmann.

[2] Guyon, I., Elisseef, A. An introduction to variable and feature selection. *J. Machine Learning Res.* 3 (2003) 1157–1182

[3] L. Yu and H. Liu. Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution. *In Proceedings of The Twentieth International Conference on Machine Leaning* (ICML-03), 856-863, Washington, D.C., August 21-24, 2003.

[4] Saeys, Y., Inza, I., Larrañaga, P. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19) (2007), 2507–2517

[5] Liu H., Motoda H. Computational methods of feature selection. Taylor, (2008).

[6] Blum, A.L., Langley, P. Selection of relevant features and examples in machine learning. *Artificial Intelligence* (1997) 245-271

[7] Peng, H.C., 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*, Vol. 27, No. 8, pp.1226–1238, 2005

[8] Molina, L.C., Belanche, L., Nebot, A. Feature Selection Algorithms: A Survey and Experimental Evaluation. ICDM, (2002)

[9] Liu, H., Yu, L. Towards integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, 17(3):1–12, 2005.

[10] Aha, D.W., Bankert, R.L. A comparative evaluation of sequential feature selection algorithms. In Doug Fisher and Hans-J. Lenz, editors, *Learning from Data*, chapter 4, pages 199-206. Springer, New York, 1996.

[11] Setiono, R., and Liu, H. A probabilistic approach to feature selection-a fitler solution. *In Proceedings of International Conference on Machine Learning*, 319-327 (1996).

[12] Senliol, B. Feature Selection Methods on High Dimensional Space. *Undergraduate Thesis for Computer Engineering Program in ITU*, 2008

[13] Dettling, M., Bühlmann, P. Supervised Clustering of Genes. *Genome Biology* (2002), 3: research 0069.1–0069.15.

[14] Blake, C., & Merz, C. UCI repository of machine learning databases. (http://www.ics.uci.edu/»mlearn/MLRepository.html.) (1998)

[15] Li, Z.R., Lin, H.H., Han, L.Y. et al. 2006, PROFEAT: a web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence, Nucleic Acids Research, 34, W32-W37.

[16] Horvath, M. P. Evaluation of PRTools. (2002)