Scalable Feature Selection for Multi-class Problems

Boris Chidlovskii and Loïc Lecerf

Xerox Research Centre Europe
6, chemin de Maupertuis, F–38240 Meylan, France

Abstract. Scalable feature selection algorithms should remove irrelevant and redundant features and scale well on very large datasets. We identify that the currently best state-of-art methods perform well on binary classification tasks but often underperform on multi-class tasks. We suggest that they suffer from the so-called accumulative effect which becomes more visible with the growing number of classes and results in removing relevant and unredundant features. To remedy the problem, we propose two new feature filtering methods which are both scalable and well adapted for the multi-class cases. We report the evaluation results on 17 different datasets which include both binary and multi-class cases.

1 Introduction

Feature selection is the technique of selecting a subset of relevant features for building robust learning models[1,4,5,7,8,10]. The feature selection targets removing most *irrelevant* and *redundant* features from the data, by which it helps improve the performance of learning models by

1. alleviating the effect of the curse of dimensionality,
2. enhancing generalization capability,
3. speeding up learning process and
4. improving model interpretability. [1]

Existing supervised feature selection algorithms fall into one of the three following groups: filter models, wrapper models or hybrid models [1,7,9,10]. The *filter model* relies on general characteristics of the training data to select some features without involving any learning or mining algorithm, therefore it does not inherit any bias of a learning algorithm. The *wrapper model* requires one predetermined learning algorithm and uses its performance to determine which features to select. The wrapper model needs to learn a classifier and may suffer from the bias problem [13]. It tends to give superior performance as it finds features better suited to the predetermined learning algorithm, but it also tends to be more computationally expensive. When the number of features becomes very large, the filter model is usually a choice due to its computational efficiency. Algorithms in a *hybrid model* try to combine the advantages of both models [11,1,14].

[1] http://en.wikipedia.org/wiki/Feature_selection

W. Daelemans et al. (Eds.): ECML PKDD 2008, Part I, LNAI 5211, pp. 227–240, 2008.

In this work we focus on the filter model and propose new feature selection algorithms which effectively remove irrelevant and redundant features and are computationally efficient. The entropy-based algorithms we propose are well suited for a variety of problems in different domains where data instances are characterized by a mixture of nominal, categorical and numeric features. Moreover, they play the central role in the *automatic feature extraction* from massive datasets where the automatic analyzer can generate thousands or billions of features. A part of the generated features is irrelevant to the classification task; even a larger part of features is redundant.

Our work is motivated by the automatic feature extraction from the image, OCR- and layout-oriented documents where a very large number of features are produced in order to capture different characteristics of visual and textual data elements. The problem we face is the selection from the very large feature set to build robust and accurate learning models.

Within the filter model, one can distinguish between feature weighting algorithms and subset selection algorithms, based on whether they evaluate the goodness of features individually or through feature subsets. *Feature weighting algorithms* assign weights to features individually and rank them using their relevance to the target concept. There are a number of different definitions on feature relevance in machine learning literature [7]. A feature is good and thus will be selected if its weight of relevance is greater than a threshold value. Relief [6] is one of the well known algorithms relying on relevance evaluation. Its key idea is to estimate the relevance of features according to how well their values distinguish between the instances of the same and different classes.

Subset selection algorithms search through candidate feature subsets guided by a certain evaluation measure which captures the goodness of each subset. An optimal (or near optimal) subset is selected when the search stops. Evaluation measures that have been shown effective in removing both irrelevant and redundant features include the consistency measure [2] and the correlation measure [5]. Combined with different search strategies, such as exhaustive, heuristic, and random search these evaluation measures form different algorithms [2,5,10]. The time complexity is exponential in terms of data dimensionality for exhaustive search and polynomial for heuristic search. However, with the quadratic or higher time complexity in terms of dimensionality, existing subset selection algorithms do not scale well when dealing with very large datasets.

To overcome the problems and to meet the demand for feature selection for high dimensional data, a novel *class of scalable algorithms* has recently appeared [11,10,15]; these algorithms can effectively identify both irrelevant and redundant features with less time complexity than subset selection algorithms. The scalability requires that the feature filtering algorithm allows a little overhead beyond the minimal cost of scanning the dataset with N features and M data items. Since a dataset may be very large in both N and M, scalable algorithms have a complexity sub-quadratic in NM.

Existing scalable algorithms show up a good performance in binary classification tasks but may cause the performance loss when the classification problem

addresses multiple classes. In the following sections, we try to analyze the reasons of this loss and propose two new methods for the scalable feature selection. The algorithms allow us to achieve the two goals: they keep the algorithms scalable and perform well on both binary and multi-class tasks.

2 Scalable Feature Selection

One common approach to measure the correlation between two random variables is based on the linear correlation coefficients. However, linear measures are unable to capture correlations that are not linear in nature; such is the case of categorical and nominal features. Another limitation is that the calculation requires all features contain numerical values. To overcome these shortcomings, the other approach is to use entropy-based measures of the uncertainty of a random variable, where the entropy of a variable Y is defined as $H(Y) = -\sum_y P(y)\ \log_2 P(y)$.

Given two random variables Y and X, we are interested in measuring the information that one variable has about another. The so-called *mutual information* $I(Y;X)$ is given by the Kullback-Leibler (KL) divergence between a joint distribution $P(Y,X)$ and the product of its marginal distributions $P(Y)P(X)$:

$$\begin{aligned} I(Y;X) &= D_{KL}(P(Y,X)||P(Y)P(X)) \\ &= \textstyle\sum_{x,y} P(y,x)\ \log_2 \frac{P(y,x)}{P(y)P(x)} \\ &= \textstyle\sum_{x,y} P(y,x)\ \log_2 P(y|x) - \sum_{x,y} P(y,x)\ \log_2 P(y) \\ &= \textstyle\sum_{x,y} P(y|x)P(x)\ \log_2 P(y|x) - \sum_y P(y)\ \log_2 P(y) \\ &= H(Y) - H(Y|X), \end{aligned} \tag{1}$$

where $P(x)$ is the prior probabilities for $x \in X$ values, $P(y|x)$ is the posterior probabilities of $y \in Y$ given the values of $x \in X$ and the conditional entropy $H(Y|X)$ is the entropy of Y after observing values of X:

$$H(Y|X) = \sum_x P(x) \sum_y P(y|x)\ \log_2 P(y|x). \tag{2}$$

When applied to the feature selection, the mutual information is the amount by which the entropy of one variable decreases from the knowledge of another variable. The mutual information (called also *information gain* [8]) is symmetrical for two variables Y and X. Since it is often biased in favor of features with more values, the values have to be normalized to ensure they are comparable and have the same affect. Therefore, *symmetrical uncertainty* is used instead. Defined as

$$SU(Y,X) = \frac{2\ I(Y;X)}{H(Y) + H(X)}, \tag{3}$$

it does compensate for the bias and normalizes its values to the range [0, 1], where the value 0 indicates that Y and X are independent and the value 1 indicates that the value of either one completely predicts the value of the other.

2.1 Markov Blankets for Redundant Features

An efficient feature selection method should cope with irrelevant and redundant features. After years of intensive research, a consensus has been achieved on determining irrelevant features [7,3,8,15]. Instead, the major difficulty remains around the redundant features. Indeed, unlike irrelevant features, not all of them should be removed while keeping all of them often causes the accuracy loss and overfitting. Therefore, this poses the problem of selecting an optimal subset among redundant features.

Let we dispose a dataset S with feature set F and class set Y. A relevant feature $F_i \in F$ is *redundant* if it has a Markov blanket in F [15], where a *Markov blanket* for feature F_i is a feature subset $\mathcal{M}_i \in F$ which subsumes the information feature F_i has about target Y and all other features in $F - \mathcal{M}_i - \{F_i\}$:

$$P(F - \mathcal{M}_i - \{F_i\}, Y | F_i, \mathcal{M}_i) = P(F - M_i - \{F_i\}, Y | \mathcal{M}_i). \tag{4}$$

The *Markov blanket filtering* [3] is a backward elimination procedure, which at any step removes F_i if there exists a Markov blanket for F_i among the features remaining in F. The process guarantees that a feature removed at previous steps will be still redundant later and removing a feature at later steps will not render the previously removed features necessarily to be included in the optimal subset F_{opt}. Finding the exact Markov blanket for a feature requires an exhaustive enumeration of feature subsets which makes the exact Markov blanket filtering computationally unacceptable for any important feature set.

Scalable filtering algorithms do approximate the Markov blanket filtering. Similarly to the exact feature subset selection, where only relevant features having no Markov blanket are selected, in the *approximate feature subset selection*, one selects the relevant features having no approximate Markov blanket.

The scalable algorithms essentially include two steps where the first step determines and removes irrelevant features and the second step copes with redundant features. Below, we present a generic filtering-based feature selection algorithm. Without loss of generality, we assume using SU measures for removing irrelevant features in F. All methods discussed in this section and developed in the following sections are instances of Algorithm 1.

The algorithm initially calculates SU values for each feature $F_i \in F$, then selects those which are superior to the irrelevance threshold δ and ranks the selected ones in the decreasing order. Steps (4)-(10) take $O(N\ M + N \log N)$ time, where N is the number of features, $N = |F|$, and M is the number of data items.

In the algorithm, the goodness criteria GOOD() guides the (greedy) process of selecting redundant features among relevant ones. Different algorithms may use different goodness criteria which in turn may results in different computational complexities. The complexity of the second part (steps (11)-(21)) depends on the number of feature pairs the GOOD () criteria is applied on. In the worst case, the criteria removes no features from F_{cand} which leads to the $O(N^2\ G)$ worst case complexity, where G is the complexity of the GOOD criteria for a feature pair, which is $O(M)$ in the general case. Moreover, if, at each iteration (12)-(19)

Algorithm 1. Scalable Feature Filtering Algorithm

INPUT: training dataset S with class set Y and feature set $F = \{F_i\}, i = 1, \ldots, N$, irrelevance threshold δ, goodness criteria GOOD()
OUTPUT: optimal feature subset F_{opt}
for $i = 1, \ldots, N$ **do**
  calculate $SU(Y, F_i)$
  if $SU(Y, F_i) > \delta$ **then**
    append F_i to F_{rel}
  end if
end for
Order features F_i in F_{rel} in the decreasing order of $SU(Y, F_i)$
F_{pivot} = getFirst (F_{rel})
while F_{pivot} is not null **do**
  add F_{pivot} to F_{opt}
  F_{cand} = getNext (F_{rel},F_{pivot})
  while F_{cand} is not null **do**
    if not GOOD(F_{pivot}, F_{cand}) **then**
      remove F_{cand} from F_{rel}
    end if
  end while
  F_{pivot} = getNext (F_{rel},F_{pivot})
end while
return F_{opt}

of Algorithm 1, F_{pivot} removes αN features F_{cand} where $0 < \alpha < 1$, then it can be shown that the expected complexity is $O(MGN \log N)$ [15].

2.2 FCBF and Accumulation Effect

The *Fast Correlation-Based Filtering* (FCBF) is currently the best scalable algorithm for feature selection [15,10]. Among various methods for *approximated Markov blankets* developed so far, the FCBF offers the best trade-off between the efficiently and effectiveness. The FCBF is based on the following Markov blanket approximation rule:

Definition 1. *Feature F_1 is an approximate Markov blanket for feature F_2 if $SU(Y, F_1) \geq SU(Y, F_2)$ and $SU(F_1, F_2) \geq SU(Y, F_1)$.*

The FCBF is an instance of Algorithm 1 where the GOOD () verifies the approximation condition given by Definition 1. Since GOOD () takes $O(1)$ time for a feature pair, the FCBF expected complexity is $O(MN \log N)$ which ensures its high scalability.

When we deployed the FCBF algorithm on a number of various datasets, we empirically identified that the FCBF works often poorly in multi-class cases. Indeed, for all datasets (presented in detail in Section 5), we evaluated the impact of FCBF on the classification accuracy. In each case, we trained (using the cross validation with 5 folds) a classifier, first with the initial feature set F, and

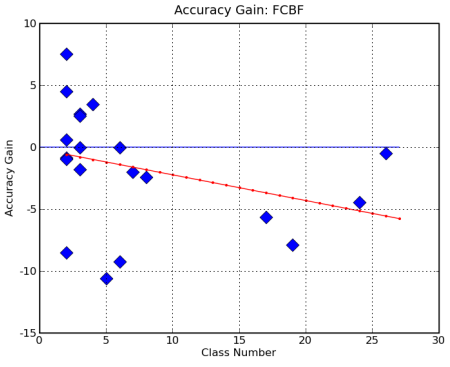

Fig. 1. The FCBF for different class numbers

then with the feature subset F_{opt} selected by the FCBF. For all datasets, we measured the accuracy gain as the difference between the second and first measures. Figure 1 plots the accuracy gain against the number of classes, denoted as $|Y|$, as well as the linear least squares fitting. The figure shows a statistically important correlation between the accuracy loss and the class number. All cases where FCBF does improve the accuracy correspond to small $|Y|$, with 2, 3 or 4 classes. And vice versa, for the large $|Y|$ cases (5 and more), the accuracy loss may achieve 10%.

We have carefully analyzed all cases where the FCBF causes the accuracy loss. Table 1 presents an artificial example which is abstracted from our analysis. The dataset includes 10 items with two features, F_1 and F_2, and three classes, y_0, y_1 and y_2. Feature F_1 correlates with class value y_1; F_2 correlates with y_2 and none correlates with y_0. It is easy to verify that both F_1 and F_2 are relevant to Y and moreover $SU(Y, F_1) = SU(Y, F_2)$. None of the two features is a Markov blanket of another because the information subsumption holds for some y and not for entire Y in (4). However, FCBF approximation rule given by Definition 1 would mistakenly eliminate one of the two features as redundant one.

The problem seems to be in the way the FCBF approximates the Markov blankets. The uncertainty reduction may vary from one class to another, but the FCBF uses the uncertainty value for entire class set Y to make a selection decision. Actually, our hypothesis is the FCBF suffers from the *accumulation effect* when the uncertainty reduction for different classes are summed up when verifying the Markov blanket condition. The effect is hardly visible for binary classification cases, but it becomes important for multi-class cases where the FCBF tends to remove features which are not redundant at all.

This analysis does suggest that the redundancy of one feature F_i with respect to another feature F_j should be verified for each class $y \in Y$ and we should consider the redundancy correlation on the per-class basis. In the following section, we propose two new methods to remedy the problem, without compromising the efficiency and scalability of the filtering algorithm.

Table 1. Artificial example where the FCBF fails

Y	F_1	F_2
y_0	0	0
y_0	0	0
y_0	0	0
y_0	1	1
y_0	1	1
y_0	1	1
y_1	0	0
y_1	0	1
y_2	0	0
y_2	1	0

3 Fast Targeted Correlation-Based Filter

In order to keep low the computational complexity and to take into account the per-class uncertainty, we first propose the *Fast Targeted Correlation Based Filtering* (FtCBF) method. It modifies the FCBF algorithm by adding an extra condition to the goodness criteria in order to avoid, or at least to minimize, the accumulation effect.

Class $y \in Y$ is called *targeted* by a feature F_i if there exist at least two items in S with different values of F_i with the class y. In terms of the conditional probability, y is targeted by feature $F_i \in F$ if $H(F_i|Y = y) > 0$. The set of classes $y \in Y$ targeted by feature F_i is denoted $S_Y(F_i)$, $S_Y(F_i) = \{y|H(F_i|Y = y) > 0\}$.

We modify the FCBF in order accommodate it to the multi-class cases. We relax the too strong condition imposed by FCBF, by verifying that the target set of the pivot feature subsumes the target set of the candidate, $S_Y(F_{pivot}) \supseteq S_Y(F_{cand})$. Thus the FtCBF applies on the following Markov blanket approximation rule:

Definition 2. *(FtCBF rule). Feature F_1 is an approximate Markov blanket for feature F_2 if $SU(Y, F_1) \geq SU(Y, F_2)$, $SU(F_1, F_2) \geq SU(Y, F_1)$ and $S_Y(F_1) \supseteq S_Y(F_2)$.*

In the artificial example in Table 1, we have $S_Y(F_1) = \{y_0, y_1\}$ and $S_Y(F_2) = \{y_0, y_2\}$. Since none of the two target sets subsumes another, the new approximation rule would retain both features. We will see in the evaluation section, how this modification allows to resist to the accumulation effect and to filter out redundant features without the accuracy loss.

The algorithm for FtCBF is obtained by setting accordingly the GOOD () criteria in Algorithm 1. Since the first term of the FtCBF rule is explicitly realized by the first part of the algorithm, the criteria for FtCBF includes the second and third terms, GOOD(F_{pivot}, F_{cand}) $=SU(F_{pivot}, F_{cand}) \geq SU(Y, F_{cand})$ *and* $S_Y(F_{pivot}) \supseteq S_Y(F_{cand})$.

4 Fast Class Correlation Filter

Another method to avoid the accumulation effect is to analyze the contribution of each class to the conditional entropy $H(Y|X)$ and the symmetric uncertainty $SU(Y,X)$ and to take them in consideration when building the Markov blanket approximation. Here we propose the *Fast Class Correlation Filtering* (FCCF) method which use both per-class uncertainty and feature correlation to build the Markov blanket approximation. We first rewrite the information gain in (1) on the per-class basis as follows:

$$\begin{aligned} I(Y;X) &= H(Y) - H(Y|X) \\ &= \sum_y \sum_x P(x)P(y|x)\ \log_2 P(y|x) - \sum_y P(y)\ \log_2 P(y) \\ &= \sum_{y\in Y} I(Y=y;X), \end{aligned} \tag{5}$$

where $I(Y=y;X)$ is the contribution of class $y \in Y$ to the aggregated information gain $I(Y;X)$, $I(Y=y;X) = \sum_x P(x)P(y|x)\ \log_2 P(y|x) - P(y)\ \log_2 P(y)$. After the normalization, the symmetric uncertainty SU may be equally decomposed on the per-class principle. For two random variables Y and X, we have $SU(y,X) = \frac{H(Y=y)-I(Y=y;X)}{H(Y)+H(X)}$ where $H(Y=y) = p(y)\log_2(y)$ and therefore

$$SU(Y,X) = \sum_{y\in Y} SU(y,X). \tag{6}$$

A relevant feature F_i *strictly subsumes* a relevant feature F_j if $SU(y,F_i) \geq SU(y,F_j)$, for all $y \in Y$. By adding the constraint on the feature correlation $SU(F_i,F_j)$, we obtain the following Markov blanket approximation rule:

Definition 3. *(FCCF rule). Feature F_1 is an approximate Markov blanket for feature F_2 if for any $y \in Y$, $SU(Y=y,F_1) \geq SU(Y=y,F_2)$ and $SU(F_1,F_2) \geq SU(Y,F_1)$.*

To apply the FCCF approximation rule and to preserve the scalability, we have to make some minor changes in Algorithm 1. In step (4)-(8), we extend the calculation of $SU(Y,F_i)$ by calculation of per-class uncertainty vector $[SU(y_1,F_i),\ldots,SU(y_{|Y|},F_i)]$. The ordering of values in step (9) would be then done by the decreasing values of one class, say y_1. The second part of Algorithm 1 is modified accordingly, in order to compare uncertainty vectors $[SU(y_1,F_i),\ldots,SU(y_{|Y|},F_i)]$ and $[SU(y_1,F_j),\ldots,SU(y_{|Y|},F_j)]$ for a pair of features $F_i,F_j \in F$. This need to compare uncertainty vectors leads to the change in the method complexities with respect to FCBF. The FCCF worst case becomes $O(M|Y|N^2)$ and the average case is $O(M|Y|N\log N)$.

This class-by-class uncertainty comparison is naive and quite satisfactory for dozens and hundreds of classes. If the number of classes $|Y|$ accounts for thousands, some sophisticated structures might implemented to reduce the average case to $O(M\log|Y|N\ \log N)$ [12].

5 Evaluation Results

In this section we report results of evaluation tests aimed at verifying how good new feature selection methods are in cases of large numbers of features, instances

and classes. All evaluations are performed in terms of classification accuracy and dimensionality degree.

For evaluation tests, 17 different datasets have been proposed. First, we selected 15 datasets available from UCI Machine Learning repository [2]. Among existing UCI datasets, we prefered ones covering different application domains and, importantly, representing a high variability of class numbers. Since the UCI Machine Learning repository is essentially dominated by 2- and 3-class datasets, we additionally included two multi-class datasets from another source. These two datasets have been created in the framework of the VIKEF European Integrated Project [3] for enabling the integrated development of semantic-based information, content, and knowledge management systems. The first, CPO dataset is a collection of data items extracted from PDF documents and annotated with 8 metadata classes, including `title, author, organization` and `address`. The second, bizCard dataset is composed of data items extracted from personal business cards where each paper-based business card was scanned, OCR-ed and annotated with different metadata classes. For all 17 datasets included in the evaluation tests, Table 2 reports the number of data items, classes and features.

Table 2. Feature selection results for 17 test datasets

Collection			**Initial Set**			**FCBF**			**FtCBF**			**FCCF**		
UCI	$\|Y\|$	M	F	**ME**	**DT**	F_{opt}	**ME**	**DT**	F_{opt}	**ME**	**DT**	F_{opt}	**ME**	**DT**
breast	2	699	9	94.7	**94.98**	8	**95.28**	94.85	8	**95.28**	94.85	8	**95.28**	94.85
credit-a	2	690	15	83.48	85.55	3	82.61	83.51	10	83.91	85	11	**84.57**	**85.51**
heart-c	2	303	13	74.97	76.24	5	**82.51**	**80.37**	10	79.89	**80.37**	11	81.11	**80.37**
hepatitis	2	155	19	79.35	78.52	3	**83.87**	**80.52**	8	80.64	**80.52**	8	82.45	**80.52**
labor	2	57	16	94.54	**80.53**	3	86.01	77.48	9	94.54	77.48	9	**94.71**	77.48
mushroom	2	8124	22	**100**	**100**	5	99.03	99.02	5	99.03	99.02	5	99.03	99.02
balance-sc	3	625	4	**88.32**	78.11	4	**88.32**	**77.78**	4	**88.32**	**77.78**	4	**88.32**	**77.78**
iris	3	150	4	91.33	94.53	1	**94**	**94.67**	4	91.33	94.33	4	91.33	**94.67**
splice	3	3190	61	92.98	93.88	22	**95.52**	**94.21**	28	93.98	94.16	31	94.13	93.26
waveform	3	5000	40	76.12	75.24	1	54.33	57.01	4	73.86	74.1	5	**76.44**	**76.19**
lymph	4	148	18	77.76	76.11	8	81.21	75.01	13	**85.28**	**76.43**	13	84.8	**76.43**
anneal	5	898	38	98.55	**98.62**	5	87.97	**98.62**	19	**99.00**	98.33	20	97.66	98.48
autos	6	205	25	58.05	**79.66**	3	58.05	70.59	24	55.12	69.82	22	**59.13**	72.59
glass	6	214	9	48.79	67.45	1	39.56	54.44	8	47.31	64.11	9	**48.79**	64.58
zoo	7	101	17	95.0	93.29	7	93.0	91.39	10	**95.0**	**94.96**	10	**95.0**	**94.96**
soybean	19	683	35	85.35	**90.57**	9	77.46	80.13	21	**86.09**	87.92	22	85.87	89.91
audiology	24	226	69	**78.22**	77.57	24	73.77	75.9	27	77.12	**78.68**	27	77.12	**78.68**
letter	26	20000	16	**82.58**	87.25	10	80.16	86.77	10	80.16	86.77	11	80.21	**87.66**
CPO	8	7612	42	93.83	90.54	11	91.43	92.04	19	94.18	**93.45**	23	**95.2**	93.22
Bizcard	17	3620	135	**71.48**	**71.65**	21	65.85	62.42	45	68.29	69.63	38	71.24	69.97
Average	6.3		31.1	83.31	**84.37**	7.6	81.57	81.05	14.5	83.57	83.79	14.7	**84.32**	84.02

[2] http://mlearn.ics.uci.edu/MLRepository.html
[3] http://www.vikef.net/

We evaluated the performance of different filter-based feature selection methods by using two well-known classification methods, C4.5 decision tree (DT) and maximum entropy (ME) ones. For the former, we used the Weka package [4] which has its version of C4.5 known as `weka.classifiers.trees.J48`. In all tests, the C4.5 confidence factor was set by default to 0.25. For the maximum entropy classifier, we used the Maximum Entropy modeling toolkit for Python language [5].

Feature selection evaluation. In the first series of tests, we compare the accuracy of classification models trained with initial feature set F, as well as with the feature subsets selected by three methods, FCBF, FtCBF and FCCF. All tests are run using the cross validation protocol. We used 5 folds for all selected datasets. For each folding, a feature selection method is applied to the training set; it returns a feature subset which is used to train a model from the training set. [6] The average over all foldings is reported as the model accuracy for the dataset.

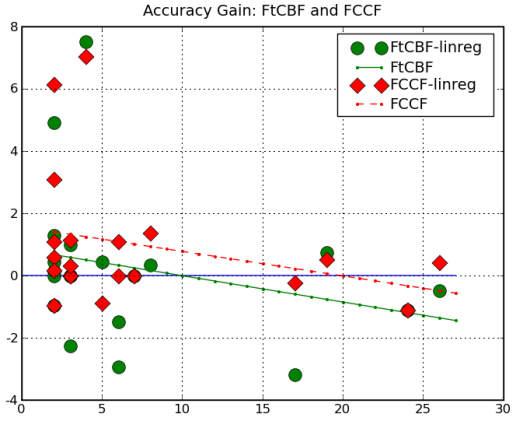

Fig. 2. Accuracy gain versus the class number: FtCBF and FCCF methods

Table 2 reports the evaluation results for the initial feature sets and the feature subsets selected by FCBF, FtCBF and FCCF. For each dataset [7] and each method, the table reports the size of optimal feature subset $|F_{opt}|$ and the accuracy for both C4.5 and maximum entropy classification models. Both

[4] http://www.cs.waikato.ac.nz/m̃l/weka/

[5] http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html

[6] Using the same training data in both feature selection and classifier training may cause the so-called *feature selection bias* [9]. We initially intended to avoid this bias. Unfortunately, a further split of training data would severely penalize the small datasets. Thus we accepted the proposed schema as far as all feature selection algorithms are evaluated in the same conditions.

[7] The UCI datasets are ordered by the increasing number of classes $|Y|$.

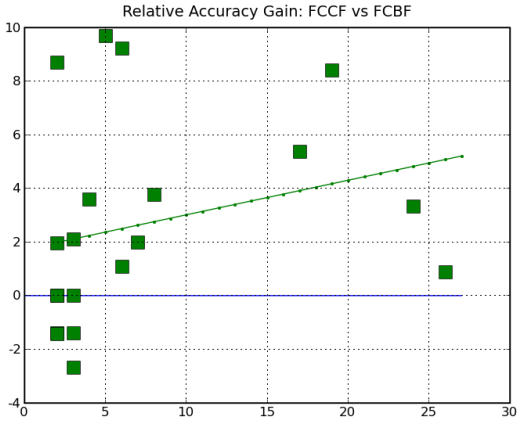

Fig. 3. Relative accuracy gain: FCCF versus FCBF

FtCBF and FCCF behave well on all the datasets, using either of two classification methods. They both appear less sensible to the number of classes and show no significant difference between collections with 2, 3, 4 and more classes.

By analogy with Figure 1, Figure 2 plots the accuracy gain and the linear minimar square fitting for all datasets, using FtCBF and FCCF methods [8]. First, the performance of new methods are comparable to the FCBF on 2- and 3-class datasets. Instead, they take an advantage over the FCBF on multi-class datasets.

To show it explicitly, Figure 3 combines the results presented in Figures 1 and 2. It shows the *relative accuracy gain* against the class number. Here, the relative gain is given by the difference in accuracy between FCCF and FCBF. The advantage of FCCF over FCBF grows as the class number increases.

Finally, we mention another aspect of test results reported in Table 2. Figure 4 plots the accuracy gain against the *feature reduction ratio* given by the fraction of features from the initial feature set F selected by a given method, $|F_{opt}|/N$. As one can observe, the FCBF conducts an aggressive policy of redundant feature removal which might be not justified and thus leading to the accuracy loss on some datasets, in particular, the multi-class ones. Instead, FtCBF and FCCF impose additional conditions for removing a feature as redundant one. As result, they are more modest at removing features from the initial set, this however permits to avoid the accuracy loss.

Class shrinkage. In order to extend the comparison of feature selection methods in multi-class tasks, we have undertaken the second series of experiments. The *class shrinkage* procedure was invented in order to test the feature selection methods over the varying number of classes.

The class shrinkage is implemented in the following way. In a dataset with $|Y|$ classes, a pair of distinct classes is randomly selected and all items of one

[8] The plot shows the accuracy values obtained with the maximum entropy models. Results for C4.5 models are not presented here but have a very similar shape.

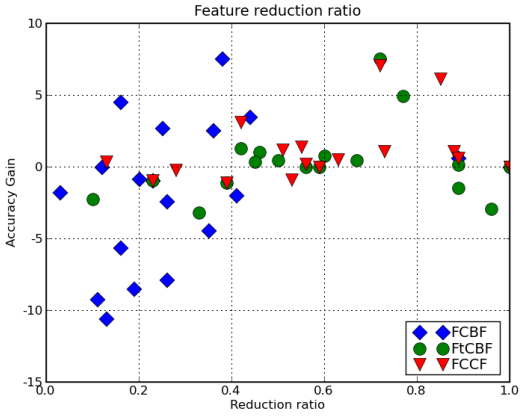

Fig. 4. Accuracy gain versus the feature reduction ratio

class are relabeled with another one. This step reduces the class number by one. The random shrinkage step is repeated $|Y| - 2$ more times, thus producing a sequence of datasets with diminishing number of classes, $k = |Y|, |Y| - 1, \ldots, 2$. For each value k in the sequence, we run all feature selection methods and measure the accuracy of classification models with the initial (Basic) feature set and feature subsets selected by FCBF, FtCBF and FCCF. The shrinkage experiment is repeated 10 times and the average values over all runs are reported as the model accuracy together with the standard deviation.

Below we present results of class shrinkage experiments for multi-class datasets in Table 2. Figures 5 shows Basic, FCBF, FtCBF and FCCF plots for the **soybean** dataset. The right extreme of all plots corresponds to the initial dataset; it was analyzed in Table 2. All three feature selection methods get close to the Basic accuracy on the left extreme, when $k = 2$. Among the three feature selection

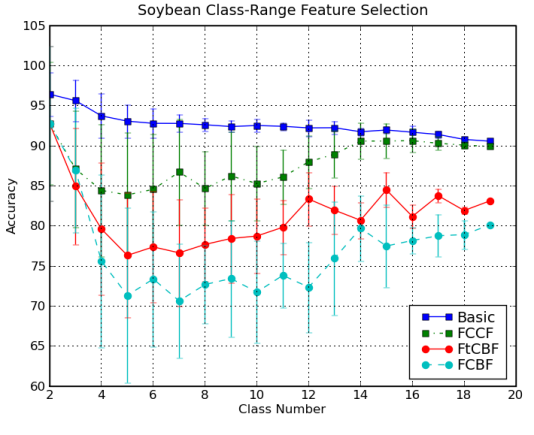

Fig. 5. Class shrinkage with the soybean dataset

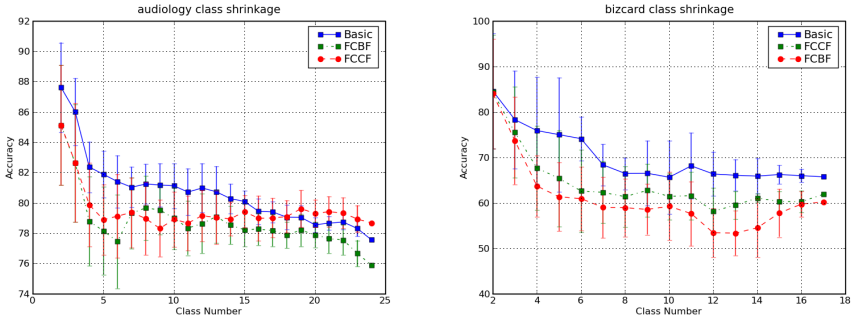

Fig. 6. Class shrinkage with the a) audiology and b) bizcard datasets

methods, both FtCBF and FCCF outperform FCBF. We note very important standard deviation values for all plots, in particular for intermediate class numbers. This is explained by the randomness of class replacement and a large variety of datasets obtained by such a replacement.

Figure 5 reports the **audiology** and **bizcard** for the initial feature set and FCBF and FCCF methods [9]. Both new methods tend to behave well in class shrinkage experiments on other multi-class datasets. The only exception is the *letter* dataset, where all feature selection methods show no significant difference.

6 Conclusion

We have proposed two new scalable feature selection methods which guarantee a good tradoff between efficiency and effectiveness for multi-class cases. Both new methods outperform the state-of-art FCBF method which may suffer from the accumulation effect. Either method proposes an approximate Markov blanket rule which relaxes the FCBF's aggressive criteria for removing redundant features. The evaluation on 17 datasets demonstrate that this relaxation pays off when the number of classes becomes important.

All scalable filtering algorithms remove as irrelevant any feature whose uncertainty value with respect to the class variable falls under the irrelevance threshold. This unfortunately ignores all possible interactions between the features [16]. Improving the filtering algorithms that takes into consideration the feature interaction but does not hurt the scalability represents a challenging problem.

References

1. Das, S.: Filters, wrappers and a boosting-based hybrid for feature selection. In: Proc. 18th Intern. Conf. Machine Learning, pp. 74–81 (2001)
2. Dash, M., Liu, H.: Feature selection for clustering. In: Proc. Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 110–121 (2000)

[9] The FtCBF plots are not reported here; they are close to FCCF plots for both datasets.

3. Koller, D., Sahami, M.: Toward optimal feature selection. In: ICML 1996: Proc. 13th International Conference on Machine Learning, pp. 284–292. Morgan Kaufmann Publishers Inc., San Francisco (1996)
4. Geng, X., Liu, T.-Y., Qin, T., Li, H.: Feature selection for ranking. In: SIGIR 2007: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 407–414. ACM, New York (2007)
5. Hall, M.A.: Correlation-based feature selection for discrete and numeric class machine learning. In: ICML 2000: Proceedings of the Seventeenth International Conference on Machine Learning, pp. 359–366 (2000)
6. Kira, L., Rendell, L.: The feature selection problem: Traditional methods and a new algorithm. In: Proc. 10th National Conf. Artificial Intelligence, pp. 129–134 (1992)
7. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artificial Intelligence 97(1-2), 273–323 (1997)
8. Kononenko, I.: Estimating attributes: Analysis and extensions of relief. In: Bergadano, F., De Raedt, L. (eds.) ECML 1994. LNCS, vol. 784, pp. 171–182. Springer, Heidelberg (1994)
9. Liu, H., Motoda, H.: Computational Methods of Feature Selection. Chapman and Hall/CRC, Boca Raton (2007)
10. Liu, H., Yu, L.: Toward integrating feature selection algorithms for classification and clustering. IEEE Transactions on Knowledge and Data Engineering 17(4), 491–502 (2005)
11. Ruiz, R., Aguilar-Ruiz, J.S., Riquelme, J.C.: Efficient incremental-ranking feature selection in massive data. In: Liu, H., Motoda, H. (eds.) Computational Methods of Feature Selection, pp. 147–166. Chapman and Hall/CRC, Boca Raton (2007)
12. Samet, H.: Foundations of Multidimensional And Metric Data Structure. Morgan Kaufmann Publishers, Reading (2006)
13. Singhi, S.K., Liu, H.: Feature subset selection bias for classification learning. In: ICML 2006: Proceedings of the 23rd international conference on Machine learning, pp. 849–856. ACM, New York (2006)
14. Xing, E.P., Jordan, M.I., Karp, R.M.: Feature selection for high-dimensional genomic microarray data. In: ICML 2001: Proceedings of the Eighteenth International Conference on Machine Learning, pp. 601–608. Morgan Kaufmann Publishers Inc., San Francisco (2001)
15. Yu, L., Liu, H.: Efficient feature selection via analysis of relevance and redundancy. J. Mach. Learn. Res. 5, 1205–1224 (2004)
16. Zhao, Z., Liu, H.: Searching for interacting features. In: Proc. Intern. Joint Conf. Artificial Intelligence, IJCAI, pp. 1156–1161 (2007)