Improving Classification Performance by Merging Distinct Feature Sets of Similar Quality Generated by Multiple Initializations of mRMR

Thomas Bottesch[1,2], Günther Palm[2]

[1]Advanced Research & Technology, Avira Operations GmbH & Co. KG, Tettnang, Germany
{thomas.bottesch}@avira.com
[2]Institute of Neural Information Processing, Ulm University, Ulm, Germany
{thomas.bottesch, guenther.palm}@uni-ulm.de

***Abstract*—The success of machine learning algorithms often depends on the combination of model size, computational cost and interpretability. One way to optimize these properties is feature selection. Computational cost and model size can be reduced by discarding features with low relevance. Furthermore, feature selection can provide a deeper understanding of the feature's importance. This work focuses on the minimal-redundancy-maximal-relevance algorithm (mRMR) which is a filter-method for feature selection that uses pairwise mutual information as a measure to decide which feature is relevant. The algorithm is initialized with the feature with the highest relevance according to the measure and an iterative algorithm selects the next feature which optimizes for a high relevance while maintaining a low redundancy to the previously selected features. This work extensively studies distinct feature sets which can be generated when running the mRMR algorithm multiple times using features of descending relevance as initialization. By exploiting information about the order in which the iterative algorithm chooses the features in the various runs, a strategy is proposed to generate a new combined feature set from all initializations. Applying the proposed strategy to four datasets of different sizes and two classification algorithms shows that the resulting feature sets are significantly better compared to the original mRMR algorithm for the given classification task. The proposed method is well-suited for cases where it is not feasible to use wrapper-methods to increase classification accuracy.**

I. Introduction

Feature selection is one of the most challenging research topics in machine learning. In the context of classification, the task is to find a subset of features from a dataset which delivers similar or sometimes even better classification results compared to using the whole feature set. An optimal solution to find the feature set which gives the best classification results for a specific classifier is a wrapper-method [1] [2]. Using an exhaustive search, the classifier is evaluated on all elements of the power set of all input features. Choosing the element of the power set for which the classifier performed best will always result in the highest achievable classification accuracy. In practice, such an exhaustive search is usually only feasible for a very small set of features combined with a classifier of low training and testing complexity. As a consequence, learning algorithms were developed which embed the feature selection [3] [4]. Such algorithms are faster compared to the wrapper-method, however the resulting features highly depend on the learning algorithm. An alternative approach are filter-methods which do not rely on a learning algorithm, but on a measure which is cheap to compute, while still being able to determine how relevant a feature set is or how much redundancy a feature set has [5] [6] [7]. Measures are for example the correlation or pairwise mutual information. Filter-methods are often characterized by a low computational complexity compared to the previous proposed methods. However, they are generally not able to achieve a competing classification accuracy when compared to wrapper-methods. Filter-methods are often able to generate feature sets which are more general and, therefore, are well-suited for various classifiers. In practice, wrapper as well as filter-methods are often used in combination with search strategies such that the space of feature set candidates shrinks considerably compared to an exhaustive search. Examples for such strategies are the greedy forward selection [8] [9], the greedy backward elimination [10] or combinations of both [11] [12]. In the greedy forward selection strategy, the feature selection algorithm is initialized with one or more features and iteratively the next feature is added according to a measure. The greedy backward elimination is a search strategy to iteratively remove features from a complete feature set. To do so, the feature to be removed has to have the lowest score according to a measure. Due to the large variety of search strategies and feature selection algorithms, the reader is referred to the comprehensive literature [13] [14].

The focus of this work is the minimal-redundancy-maximal-relevance algorithm [15] (mRMR) which can be categorized as a filter-method. The measure used in this algorithm is the pairwise mutual information. The mRMR algorithm is still of high interest to the research community. In [16], mRMR is used in combination with support vector machines in order to do a recursive feature elimination strategy for the task of gene selection. By doing bootstrapping on datasets, [17] created various feature sets using mRMR and showed that an ensemble of classifiers can achieve better classification scores than a feature set which was generated by running mRMR on the complete dataset. In the same paper, a way for creating feature sets by initializing mRMR at different positions was proposed which is picked up and formalized in this work. In [17], the distinct features of all initializations were used for an exhaustive search with a wrapper-method which also generated

978-1-4799-7560-0/15 $31.00 © 2015 IEEE
DOI 10.1109/SSCI.2015.56

IEEE computer society

better classification scores. All of the previous examples use wrapper-methods to increase classification accuracies. The main contribution of this paper is a wrapper-free approach of creating a feature set superior to the feature set found by the traditional mRMR algorithm, for the task of classification, by applying a merge strategy on the feature sets which result from different mRMR initializations.

This paper is organized as follows. In Section II the basics of the mRMR algorithm are introduced, comprising of the optimization objective and the way the traditional algorithm is initialized. The mathematical formulation is then extended in Section III by presenting multiple initializations. In Section IV a strategy to merge the feature sets generated with the help of different initializations is proposed. In Section V, the initialization properties are analyzed in detail and an empirical evaluation of the merging strategy is given. Finally, a conclusion is given in Section VI.

II. Basics

Given a set of N features $X = \{x_i; i = 1...N\}$, where x_i is a random variable describing the distribution of the feature i, the task is to select $S \subset X$ where $l = |S|$ denotes the desired number of features to be selected by the feature selection algorithm. To keep the text uncluttered, in the following the random variables will be referred to as features.

The mRMR algorithm [15] uses the mutual information between two features x and y which is defined by:

$$I(x;y) = \iint p(x,y) \log \frac{p(x,y)}{p(x)p(y)} dxdy, \tag{1}$$

where $p(x)$, $p(y)$ and $p(x,y)$ denote probabilistic density functions. The goal of mRMR is to retrieve a feature set S which optimizes a combined objective consisting of the relevance D to the target class variable c of all chosen features

$$D(S) = \frac{1}{|S|} \sum_{x_i \in S} I(x_i;c), \tag{2}$$

as well as the redundancy R of all chosen features within S as defined in:

$$R(S) = \frac{1}{|S|} \sum_{x_i, x_j \in S} I(x_i;x_j). \tag{3}$$

The relevance D and redundancy R are used to define the Mutual Information Quotient (MIQ)

$$\max_{S \subset X} \left(\frac{D(S)}{R(S)} \right), \tag{4}$$

which is the combined objective according to which features will be selected. This work focuses on MIQ. However, there are multiple possibilities for defining combined objectives. In [15], the optimization in Equation 4 was realized with the following incremental algorithm:

The set $S_{m-1} \subset X$ contains $m-1$ features which were already chosen. The incremental algorithm then chooses the m-th feature by evaluating:

$$x_j = \operatorname*{argmax}_{x_j \in X \setminus S_{m-1}} \left[\frac{I(x_j;c)}{\frac{1}{m-1} \sum_{x_i \in S_{m-1}} I(x_i;x_j)} \right]; m > 1. \tag{5}$$

The chosen feature x_j is added to the set S_{m-1}. The selection of feature x_j is optimized such that is has a high relevance to the class variable c while maintaining a low redundancy to all features already selected in S_{m-1}. The algorithm is initialized with S_1 which consists of only one feature:

$$x^{init} = \operatorname*{argmax}_{x_j \in X} I(x_j;c) \tag{6}$$

$$S_1 = \{x^{init}\}. \tag{7}$$

Taking a closer look at Equation 5 and Equation 7 it becomes clear that $S_1 \subset S_2 \subset ... \subset S_{m-2} \subset S_{m-1}$. Note that the subscript of S always equals the number of features within S. The computational complexity of the incremental search is $O(l^2 \cdot N \cdot V)$ where V is the number of samples and l the desired number of features. The complexity measures how many computations are needed to achieve the feature selection.

III. Extension to multiple initializations

By design mRMR is a greedy algorithm. The algorithm runs deterministic to the same local optimum in recurrent executions of the algorithm when initialized from the feature specified in Equation 5. With given domain knowledge, S_1 can be initialized with a predefined set of features which are known to work well for classification. Features which are then added to this predefined set would have a high relevance and would not be redundant to the existing features. The initializations, originally proposed in [17] and mathematically formalized in this work, do not rely on domain knowledge, but extend Equation 6 and Equation 7 in a straight forward manner. They are given by:

$$R^i = \left\{ x^{init_1}, x^{init_2}, ..., x^{init_{i-1}} \right\} \qquad ; i > 1, R^1 = \emptyset \tag{8}$$

with x^{init_k} defined as:

$$x^{init_k} = \operatorname*{argmax}_{x_j \in X \setminus R^k} I(x_j;c) \qquad ; k > 0 \tag{9}$$

$$S_1^{init_k} = \{x^{init_k}\}. \tag{10}$$

According to these definitions, $S_1^{init_1}$ is composed of the feature with the highest relevance, $S_1^{init_2}$ is composed of the feature with the second highest relevance and so on. It has to be noted that in case of $k = 1$ the resulting initialization $S_1^{init_1}$ reduces to the initialization used in mRMR (see Equation 6). In the following, feature sets which resulted from running the incremental algorithm initialized by Equation 9 are referred to as initializations.

IV. Merging initialization feature sets

The first features added to S by the iterative algorithm from Equation 5 have a very high relevance, a very low redundancy or both. The more features added, the less relevant or more redundant to the already chosen features they become. The first selected features can be very different when initializing with the strategy from Section III. However, it seems natural that if a feature is selected by the iterative algorithm by many initializations in an early stage, that this feature has a higher importance than features seen only in e.g. one of the initializations. The idea is to create a new feature set which consists of the first j features that were selected by the iterative algorithm across all initializations. Therefore, the feature sets of the initializations are merged as follows:

$$I_j = S_j^{init_1} \cap ... \cap S_j^{init_k} \tag{11}$$
$$K = [I_1, I_2 \setminus I_1, ..., I_u \setminus I_{u-1}] \tag{12}$$
$$Z = delete\_empty\_sets(K) = [z_1, ..., z_n] \tag{13}$$

I_j contains all features that appear in all $S_j^{init_k}$. K is an ordered list of sets without duplicates where u denotes the size of the list. The first element of K contains all features which appeared in all $S_j^{init_1}$. The second element of K contains all features which appeared in all $S_j^{init_2}$, which are not in the first element of K and so on. With these definitions, the j-th element of K contains the features that appeared in all $S_j^{init_k}$, which did not appear in all initializations at the same time before. The function $delete\_empty\_sets$ has an ordered list of sets as input and returns an ordered list of features. The function creates an empty list and iterates over the elements of K starting at the first element I_1. If an element of K is an empty set, its skipped. If its not empty, all features of the set are added to the list. The first feature within the list is the first one that occurred in all initializations at the same time. The second feature in the list is the one which occurred in all initializations after the first one and so on. The result of this operation is the ordered list Z given by Equation 13. To complete the notation, the resulting S_j^{merged} is defined as:

$$S_j^{merged} = \{z_1, ..., z_j\} \qquad ; j > 0. \tag{14}$$

Hence, the set S_j^{merged} contains the j features which first occurred at the same time in all initializations.

V. Empirical evaluation

A. Datasets

The following datasets, which consist of discrete features, have been selected to study the initialization properties and the classification performance:

Dataset	#Samples	#Features	#Class
NCI [18] [19]	60	9703	9
Lymphoma [20]	96	4026	9
Dexter [21]	600	11035	2
FarmAds [22]	4143	54877	2

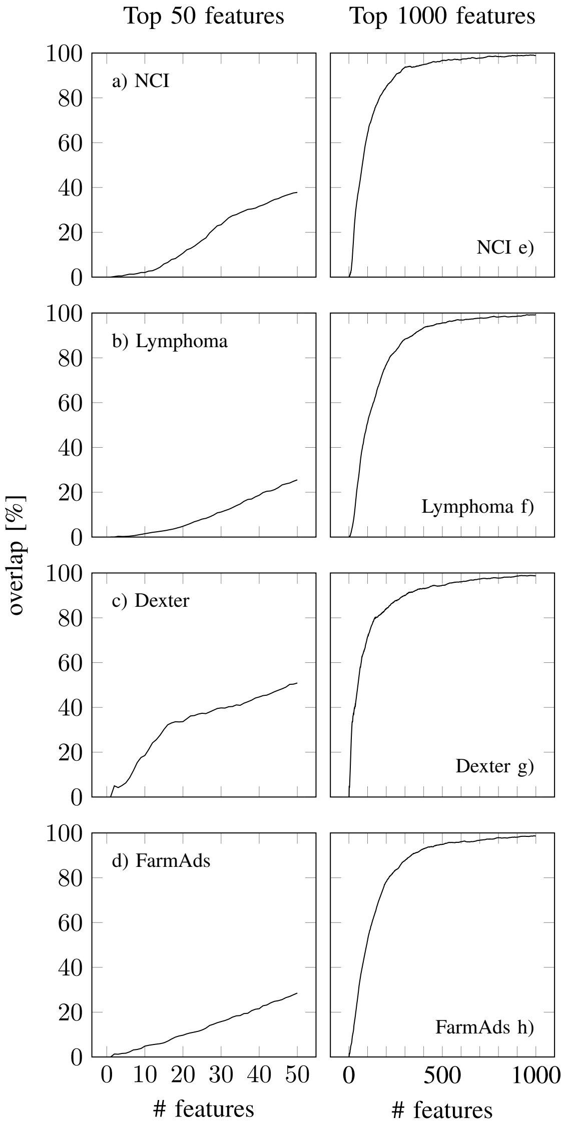

Fig. 1. Visualizes the overlap of feature-sets generated by various $S_j^{init_k}$ when using 20 initializations: a) - d) shows the overlap within the top 50 features, e) - h) the overlap within the top 1000 features.

NCI and $Lymphoma$ are multi-label microarray gene expression datasets. The classes in both datasets are different kinds of cancer. Every feature variable in these datasets has been discretized using the mean μ and the standard deviation σ in the following way: all feature values smaller than $\mu - \sigma/2$ were set to -1, all values in the range of $\mu - \sigma/2$ to $\mu + \sigma/2$ were set to 0 and all values bigger than $\mu + \sigma/2$ were set to +1.

The $Dexter$ dataset was used in the NIPS 2003 feature selection challenge, which was formulated as a classification problem in a bag-of-word representation. The feature values were set to 1 if the word occurs in the text else to 0. There are three datasets available for $Dexter$: (1) the train set with 300 samples, (2) the validation set with 300 samples and (3) the test set with 1400 samples. In the experiments only (1)

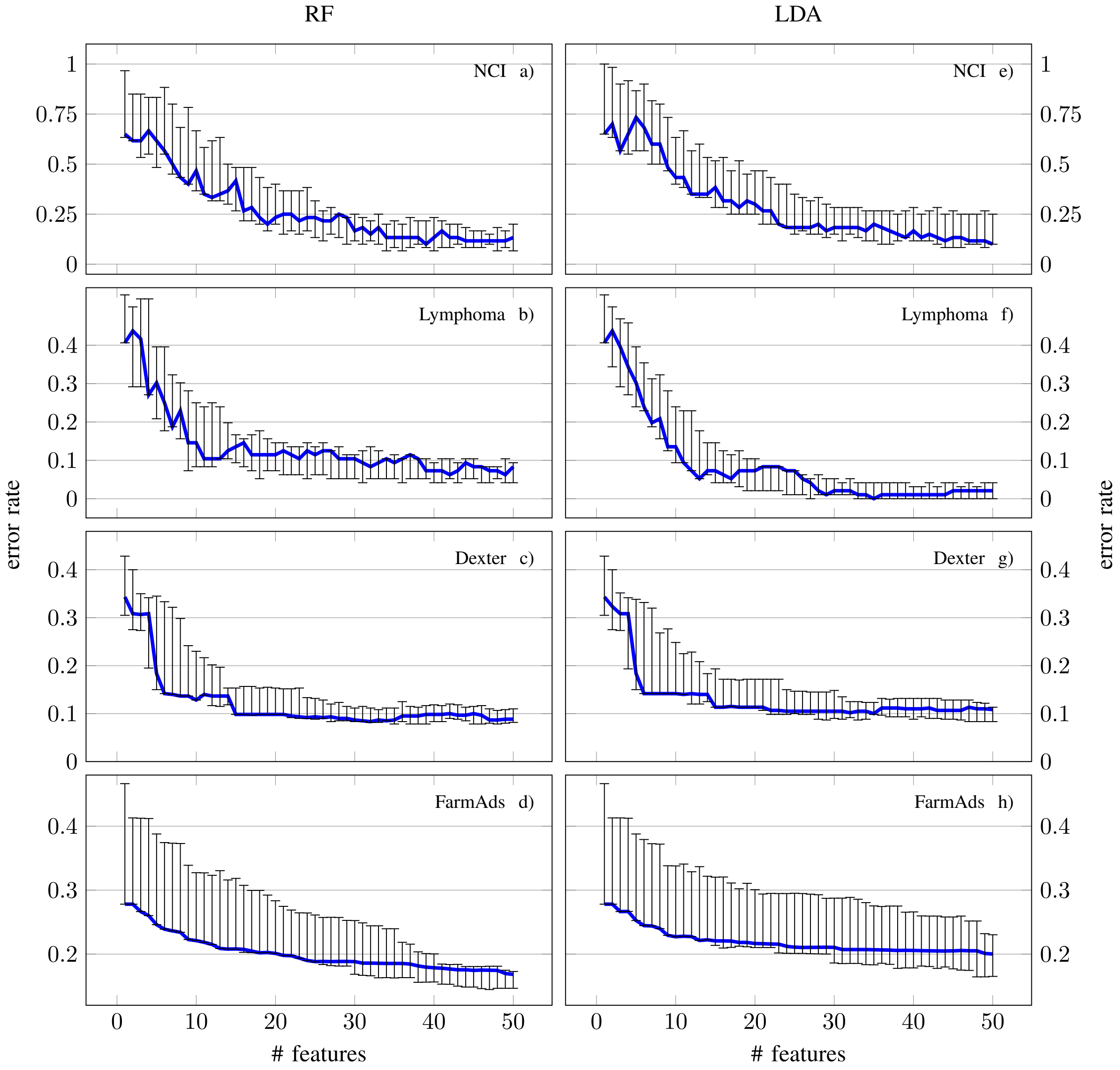

Fig. 2. Comparison of the mean cross-validation error rates, for $S_j^{init_1}$ (mRMR) for RF and LDA. The error bars indicate the min/max mean cross-validation error rate of all initializations. The plots show that in a), b), e) and f) the cross-validation error rate of the initializations vary only little compared to mRMR, while in c), d) e) and h) the variance is larger.

and (2) were used due to the lack of labels for the test set. All features which never occur in (1) and (2) were removed such that the number of features reduced from 20000 to 11035.

Each sample in the $FarmAds$ dataset consists of words from a farm animal related website and words from text-advertisement found on that website. The label is 1 if the content owner approves of the advertisement else it is -1. The features correspond to the occurrence of words. If a word occurs in the advertisement or on the page it is set to 1 else it is 0.

B. Determining Properties of the Initializations

$S_j^{init_k}$ describes a feature selection to the size $|S| = j$ which was initialized using the feature with the k-th highest relevance. While initializing with different features, the combined objective which is described in Equation 4 remains the same. Because of this fact it is clear that for large j all $S_j^{init_k}$ will converge to the same solution even though they were initialized with different features. To determine the overlap of features between the initializations an experiment with $k = 20$ was conducted on the datasets from Section V-A. The overlap was defined as:

$$\frac{|S_j^{init_1} \cap ... \cap S_j^{init_k}| \cdot 100}{j} \qquad ; j = 1, ..., 1000.$$

Figure 1 shows the similarity of feature sets originating from different initializations and is performed on all datasets

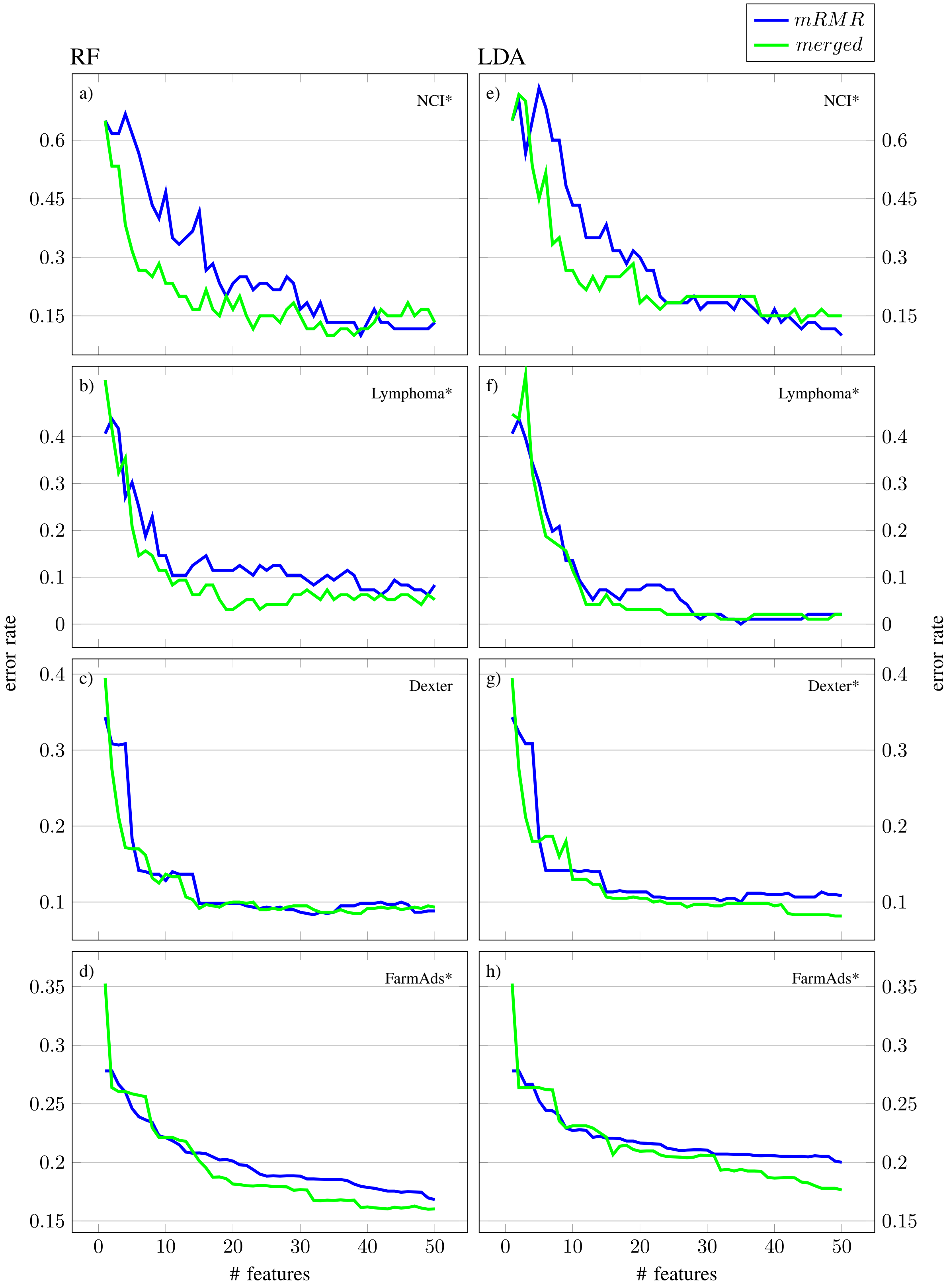

Fig. 3. Comparison of the mean cross-validation error rates, for $S_j^{init_1}$ (mRMR) and S_j^{merged} (merged) for RF and LDA. In plots a), b), d) and h) the merging strategy is able to achieve a visible lower error rate. In f) and h) the merging strategy achieves a marginal lower error rate compared to mRMR. In plot e) the merging strategy is better for $j \leq 25$ and worse afterwards. In plot c) it is unclear which features work better. Results denoted with * are statistical significant according to the Wilcoxon signed-rank test with $p - value \leq 5\%$.

presented in Section V-A. The number of features j from the $S_j^{init_k}$ is visualized on the x-axis. The y-axis displays the overlap in percent of the feature sets. The plots on the left-hand side show the overlap for the feature sets with $j \leq 50$. The plots on the right-hand side show the overlap of the feature sets with $j \leq 1000$.

As suspected, all initializations converge for a high j to the same feature sets in the experiment which is visible in Figure 1 e) - h). For $j = 1000$ the feature sets overlap to more than 99% independent of the dataset. However the motivation for this work is visualized in Figure 1 a) - d). The plots show that for a small j all feature sets overlap only to a small percentage. Especially for the $FarmAds$ and $Lymphoma$ datasets the initializations possess on average more than 70% unique features when choosing less than 50 features. The NCI dataset more than 60% unique features for $j \leq 50$ while the $Dexter$ dataset has more than 50% unique features.

C. Classification Performance of the Initializations

With the substantial differences between the initializations the question remains how suitable their resulting feature sets are compared to $S_j^{init_1}$ (mRMR) for the task of classification. random forest [23] (RF) and linear discriminant analysis [24] (LDA) were chosen to investigate this question.

Leave-one-out cross-validation was used for the small datasets NCI and $Lymphoma$. A 10-fold cross-validation was used to assess classification error rates of the bigger datasets $Dexter$ and $FarmAds$.

The results of the experiment are shown in Figure 2. The x-axis displays the number of features in the feature sets. The y-axis is assigned to the mean cross validation error rate. Please note that the plots showing the datasets have different scales on the y-axis. Error bars are used to display the min/max mean cross validation error rate for the 20 initializations. The lower bound of the error bar corresponds to the minimum $mean(CV(S_j^{init_k}))$ and the upper bound corresponds to the maximum $mean(CV(S_j^{init_k})$ with $k = 1, ..., 20$. The blue line corresponds to $mean(CV(S_j^{init_1}))$ (mRMR), where CV is a function that generates an array with each element being the cross validation error of a different fold of the samples for the given feature set. The plots on the left-hand side display the results for RF while the plots on the right-hand side display the results for LDA.

Analyzing the results for the small datasets Figure 2 a), b), e) and f) reveals that the variance of the error rate of the initializations is small around $S_j^{init_1}$ (mRMR). For LDA the general trend seems to be that mRMR has a lower error rate compared to the initializations, while for RF the initializations often seem to have a lower error rate.

In case of the bigger datasets $Dexter$ and $FarmAds$ (Fig. 2 c), d), g) and h)) a higher variance of the initializations compared to mRMR can be observed. For both datasets mRMR has the lowest error rate with less than 30 features almost all the time.

D. Classification Performance of Merged Feature Sets

The experiment from Section V-C was repeated in order to compare $S_j^{init_1}$ (mRMR) and S_j^{merged}. The results are visualized in Figure 3. The x-axis again shows the number of features while the y-axis shows the mean cross-validation error rate. The blue line corresponds to $mean(CV(S_j^{init_1}))$ (mRMR) while the green line corresponds to $mean(CV(S_j^{merged}))$ (merged). The plots on the left-hand side show the results for RF while the plots on the right-hand side show the results for LDA.

In plots a), b), d), h) the merging strategy visibly outperforms mRMR almost all the time. Interesting regarding the $FarmAds$ dataset is a comparison of the current plots d) and h) and the lower end of the error bars visualized in Figure 2 d) & h). It seems that the merging strategy moved the error rate in the direction of the minimum error rate of the initializations. Also in plot g) and f) the merging strategy has a marginal lower error rate. For plot e) the merging strategy is better for $j < 25$ and marginally worse afterwards. In plot c) it is unclear which feature sets are better. It is visible that in six of the eight plots mRMR is outperformed by the merging strategy.

E. Implementation Details

As stated in Section III, the computational complexity is $O(l^2 \cdot N \cdot M)$. When only calculating l up to 50, which is done often in previous literature, the quadratic complexity of l does not bother very much. However, in order to compute Figure 1, the subset size l ranged from 1 to 1000. The naive implementation with complexity depending on l^2 is far too computational infeasible to compute the feature sets derived from the initializations. By storing and looking up every previously computed $I(x; y)$ the complexity can be reduced to $O(l \cdot N \cdot M)$ at the cost of $l \cdot N$ entries in RAM. The previous statement assumes that a memory lookup costs nothing compared to computing $I(x; y)$. By using parallelization and the lookup strategy, retrieving the feature sets for all datasets took only about 4 hours on a 20-core machine.

VI. Statistical significance

To provide statistical significant results a further experiment was conducted in which the Wilcoxon signed-rank [25] was applied to the results of the last experiment shown in Figure 3.

When using the signed-rank test it is important to make sure that all assumptions required by the test will hold. Especially the interpretation of the result of the test should be analyzed and understood in detail. The resulting $p - value$ serves as an instrument to decide whether two vectors deviate statistically significant or not. For the experiments in this work it was verified that all $p - values$ have the correct meaning. A low $p - value$ corresponds to the cross validation results of the merged strategy deviating (overall) much in the direction of lower error rate. It is mandatory to use the signed-rank test which incorporates continuity correction for random variables with less than 60 elements, which is the case here. The widely accepted threshold of 5% is used in order to decide if the merging strategy is significantly better. All results from Figure 3 are annotated with an * if $p - value \leq 5\%$. In seven of the eight experiments the merging strategy is significantly better according to the signed-rank test. This might be surprising for e) but it is important to notice that the signed-rank test gives large deviations a higher weight.

VII. Conclusion

Wrapper-methods often yield high classification accuracies when selecting a feature subset for the task of classification. Due to the dependency on training a classifier, these methods can be very slow even for classifiers with relatively low computational complexity. Filter-methods on the other hand do not rely on a classifier. They therefore can have a very low computational complexity compared to wrapper-methods. However, generally feature sets found by filter-methods do not reach as high classification accuracies as feature sets found by wrapper-methods. Hence, it is desirable to increase the classification accuracies of feature sets derived from filter-methods which was addressed in this paper. With the first complete mathematical formulation of how to initialize the filter-method mRMR with features of descending relevance (Section III), it was possible to generate k feature sets, which are highly distinct. Furthermore, it was shown that the feature sets which originated from different initializations have similar classification error rates compared to the mRMR algorithm. A caching algorithm was proposed which significantly reduces the computational complexity compared to the traditional mRMR, allowing empirical evaluations on the $FarmAds$ dataset. By combining knowledge from all feature sets, a merging strategy was introduced in order to create a superior feature set for the classification task (see Section IV). The comparison of the merging strategy with mRMR (Section V) together with the test for statistical significance (Section VI) showed that using the merging strategy is a viable option to increase classification accuracy. In applications where wrapper-methods cannot be used due to e.g. the complexity of the classifier, the proposed method can be helpful.

In a future work the number of initializations will be varied to study the change in classification accuracies. The proposed merging strategy only uses features if they appear in all feature sets, which is a very hard constraint. Modifying the merging strategy to allow features which only appear in a subset of all feature sets can also lead to better classification results. This work showed only one possible way to merge the feature sets which resulted from different initializations while there are numerous ways to do this. Future work will present alternative ways of merging these features sets.

Acknowledgment

A big thank you to Michael Glodek who assisted with very useful comments as well as suggestions on how to structure this paper.

References

[1] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial intelligence*, vol. 97, no. 1, pp. 273–324, 1997.

[2] S. Maldonado and R. Weber, "A wrapper method for feature selection using support vector machines," *Information Sciences*, vol. 179, no. 13, pp. 2208–2217, 2009.

[3] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, Mar. 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=944919.944968

[4] Y. Saeys, T. Abeel, and Y. Van de Peer, "Robust feature selection using ensemble feature selection techniques," in *Machine learning and Knowledge Discovery in Databases*. Springer, 2008, pp. 313–325.

[5] M. A. Hall, "Correlation-based feature selection for machine learning," Ph.D. dissertation, The University of Waikato, 1999.

[6] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in *Advances in neural information processing systems*, 2005, pp. 507–514.

[7] P. Estévez, M. Tesmer, C. Perez, J. M. Zurada *et al.*, "Normalized mutual information feature selection," *Neural Networks, IEEE Transactions on*, vol. 20, no. 2, pp. 189–201, 2009.

[8] D. Ververidis and C. Kotropoulos, "Fast and accurate sequential floating forward feature selection with the Bayes classifier applied to speech emotion recognition," *Signal Processing*, vol. 88, no. 12, pp. 2956–2970, 2008.

[9] A. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample performance," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 2, pp. 153–158, 1997.

[10] D. W. Aha and R. L. Bankert, "A comparative evaluation of sequential feature selection algorithms," in *Learning from Data*. Springer, 1996, pp. 199–206.

[11] M. Kächele, D. Zharkov, S. Meudt, and F. Schwenker, "Prosodic, spectral and voice quality feature selection using a long-term stopping criterion for audio-based emotion recognition," in *Pattern Recognition (ICPR), 2014 22nd International Conference on*. IEEE, 2014, pp. 803–808.

[12] T. Zhang, "Adaptive forward-backward greedy algorithm for learning sparse representations," *Information Theory, IEEE Transactions on*, vol. 57, no. 7, pp. 4689–4708, 2011.

[13] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 17, no. 4, pp. 491–502, 2005.

[14] M. Dash and H. Liu, "Feature selection for classification," *Intelligent Data Analysis*, vol. 1, no. 1, pp. 131–156, 1997.

[15] H. Peng, F. Long, and C. Ding, "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.

[16] P. Mundra, J. C. Rajapakse *et al.*, "SVM-RFE with mRMR filter for gene selection," *NanoBioscience, IEEE Transactions on*, vol. 9, no. 1, pp. 31–37, 2010.

[17] N. De Jay, S. Papillon-Cavanagh, C. Olsen, N. El-Hachem, G. Bontempi, and B. Haibe-Kains, "mRMRe: an R package for parallelized mRMR ensemble feature selection," *Bioinformatics*, vol. 29, no. 18, pp. 2365–2368, 2013.

[18] U. Scherf, D. T. Ross, M. Waltham, L. H. Smith, J. K. Lee, L. Tanabe, K. W. Kohn, W. C. Reinhold, T. G. Myers, D. T. Andrews *et al.*, "A gene expression database for the molecular pharmacology of cancer," *Nature Genetics*, vol. 24, no. 3, pp. 236–244, 2000.

[19] D. T. Ross, U. Scherf, M. B. Eisen, C. M. Perou, C. Rees, P. Spellman, V. Iyer, S. S. Jeffrey, M. Van de Rijn, M. Waltham *et al.*, "Systematic variation in gene expression patterns in human cancer cell lines," *Nature Genetics*, vol. 24, no. 3, pp. 227–235, 2000.

[20] A. A. Alizadeh, M. B. Eisen, R. E. Davis, C. Ma, I. S. Lossos, A. Rosenwald, J. C. Boldrick, H. Sabet, T. Tran, X. Yu *et al.*, "Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling," *Nature*, vol. 403, no. 6769, pp. 503–511, 2000.

[21] I. Guyon, S. Gunn, A. Ben-Hur, and G. Dror, "Result analysis of the NIPS 2003 feature selection challenge," in *Advances in Neural Information Processing Systems*, 2004, pp. 545–552.

[22] M. Lichman, "UCI machine learning repository," 2015. [Online]. Available: http://archive.ics.uci.edu/ml

[23] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[24] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.

[25] J. a. Litchfield and F. Wilcoxon, "A simplified method of evaluating dose-effect experiments," *Journal of Pharmacology and Experimental Therapeutics*, vol. 96, no. 2, pp. 99–113, 1949.