Novel Automatic Filter-Class Feature Selection for Machine Learning Regression

Morten Gill Wollsen[1](✉), John Hallam[2], and Bo Nørregaard Jørgensen[1]

[1] Center for Energy Informatics, The Maersk Mc-Kinney Moller Institute,
University of Southern Denmark, Odense, Denmark
{mgw,bnj}@mmmi.sdu.dk

[2] Center for BioRobotics, The Maersk Mc-Kinney Moller Institute,
University of Southern Denmark, Odense, Denmark
john@mmmi.sdu.dk

Abstract. With the increased focus on application of Big Data in all sectors of society, the performance of machine learning becomes essential. Efficient machine learning depends on efficient feature selection algorithms. Filter feature selection algorithms are model-free and therefore very fast, but require a threshold to function. We have created a novel meta-filter automatic feature selection, Ranked Distinct Elitism Selection Filter (RDESF) which is fully automatic and is composed of five common filters and a distinct selection process.

To test the performance and speed of RDESF it will be benchmarked against 4 other common automatic feature selection algorithms: Backward selection, forward selection, NLPCA and PCA as well as using no algorithms at all. The benchmarking will be performed through two experiments with two different data sets that are both time-series regression-based problems. The prediction will be performed by a Multilayer Perceptron (MLP).

Our results show that RDESF is a strong competitor and allows for a fully automatic feature selection system using filters. RDESF was only outperformed by forward selection, which was expected as it is a wrapper which includes the prediction model in the feature selection process. PCA is often used in machine learning litterature and can be considered the default feature selection method. RDESF outperformed PCA in both experiments in both prediction error and computational speed. RDESF is a new step into filter-based automatic feature selection algorithms that can be used for many different applications.

1 Introduction

More data is available now than ever before, with cheaper sensors and the installation of sensors everywhere. The “Internet of Things” and “Big Data” are terms connected to the fact that the amount of data is increasing rapidly. Machine learning regression attempts to learn the relation between parameters of a system based on historical data. Implementing a successful machine learning algorithm requires choosing a representation of the solution, selecting relevant input features and setting parameters associated with the learning method [17].

P. Angelov et al. (eds.), *Advances in Big Data*, Advances in Intelligent
Systems and Computing 529, DOI 10.1007/978-3-319-47898-2_8

Selecting the relevant input features, or *feature selection*, is the process of determining which subset of the combined available input data should be included to give the best performance. Feature selection is a critical task because excluding important input features means the learning algorithm will not be able to model the system. On the other hand, including unnecessary features complicates the learning. Any input that is added increases the search space by at least one dimension [17]. Feature selection can be performed by humans with the necessary domain expertise. In some cases the experts do not exist or the work itself can be expensive and time consuming [17]. A reduced feature set for the learning algorithm also reduces training time and over-generalization [5]. By automating the feature selection process, the time and expertise required is reduced and the practicality of a combined system with a learning algorithm is increased [17].

Feature selection is broadly split into three categories: *wrapper*, *embedded* and *filter* algorithms [11]. Common to all algorithms is that they only select a subset of the input features. Another strategy is to reduce the dimensions of the original feature set; such methods are named *dimension reduction* algorithms. Filtering algorithms are model-free which makes them very fast. Unfortunately they require a threshold that decides which features are selected. This presents a problem because the same threshold cannot be used for all algorithms and selecting a threshold requires a domain expert.

This paper proposes a novel filtering algorithm that automatically select features. The filter is called the Ranked Distinct Elitism Selection Filter (RDESF) and is composed of multiple common filtering algorithms. RDESF is benchmarked against other common feature selection algorithms such as forward search, backward search, PCA and NLPCA. The benchmark is performed on two time-series regression-based problems in both short-term (1 h ahead) and long-term (24 h ahead). The first problem is the prediction of indoor temperature from the SML2010 data set [18], available at the UCI Machine Learning Repository. The first of the two files in data set is used. More information is available at the UCI website [8]. The second problem is prediction of outdoor temperature. This data set is a design reference year from 2001-2010 created by The Danish Meteorological Institute [16]. Besides the weather parameters, we have added an input whether or not the sun is up, based on [9]. We have also added the earth's azimuth with reference to the sun, to have an input that differentiates the seasons. Both data sets are publicly available.

PCA will act as a baseline for the benchmark, but other commonly used filters are included to give a good indication of RDESF's capabilities. MATLAB and PCA is often used in machine learning litterature because MATLAB has implementations of Artificial Neural Networks and also includes PCA. In addition MATLAB is very easy and intuitive to use. We want to show that there are better and faster alternatives to PCA.

2 Method

The performance of RDESF will be benchmarked against commonly used feature selection algorithms described in the litterature. The benchmarking will be based on prediction error and computational speed.

2.1 Feature Selection Algorithms

The following feature selection algorithms are used:

Principal Component Analysis (PCA). PCA is a feature dimension reduction technique. The features are mapped into a smaller dimensional space to hopefully reveal structures in the underlying data [15]. The principal components are calculated using the singular value decomposition (SVD) method. Selecting a subset of the components is done by removing those components with a standard deviation close to zero with respect to machine precision.

Non-Linear Principal Component Analysis (NLPCA). Where PCA is a linear mapping between the original and the reduced dimension space, NLPCA offers a non-linear mapping, and thus any non-linear correlations between the features will be kept [7]. The non-linear mapping is performed with an artificial neural network (ANN) with three hidden layers. The middle hidden layer is a bottleneck layer and the other hidden layers are mapping layers. The bottleneck layer contains the number of nodes that the input set is reduced to. The number of nodes in the bottleneck layer of the ANN is determined by the Guttmann-Kaiser criterion, which picks components with eigenvalues above 1.0 [4]. The number of nodes in the mapping layers is set to the number of input features plus one, to avoid any bottlenecking in those layers. By training the network to approximate the input through this bottleneck layer, the bottleneck layer contains information for subsequent layers to reconstruct the input [7]. The network is trained using the backpropagation algorithm [12] until an error of 0.001 % has been achieved with a maximum of 500 iterations. After the training is complete, a new ANN is created from the first three layers. The first mapping layer is now the hidden layer and the previous bottleneck layer is now the output layer. The entire data set is then run through the new ANN, and the output of the new ANN is the reduced data set.

Forward and Backward Selection. The forward and backward selection are both wrapper-class feature selection algorithms. This means that the learning algorithm for which features are selected is included in the feature selection process. In forward selection the features are added one at a time. If the testing error decreases when a feature is added, the feature is kept. In backward selection the process is reversed where features are removed, and if the error increases, the removed features are included again [11]. As with the NLPCA, the data is randomly divided 50/50 into a training set and a test set. The error is the corrected Akaike information criterion (AICc) [1]. The formula for AICc is:

$$AICc = n \cdot \ln(RMSE) + 2(k+1) + \frac{2k(k+1)}{n-k-1} \tag{1}$$

with n being the sample size, k being the number of features and $RMSE$ being the root mean square error.

Ranked Distinct Elitism Selection Filter (RDESF). RDESF is our novel filter feature selection algorithm. It is a meta filter that combines commonly used filtering algorithms to combine their strengths and to create a broad-ranging generic filter that can be used in many application. The included filters return a ranked score of the input features based on their individual measurement. This means that a threshold is required to select the relevant features. To overcome this issue, an elitism selection is used inspired by Genetic Algorithms. The top 10 % highest ranked features are selected from every included filters. From the combined features a distinct selection based on set theory is performed to remove duplicate features. In our case the selection is a union. The process of RDESF is:

1. Let every included filter score the features based on their respective measurement
2. Rank the scores and select the best 10 % from each filter
3. Perform a union selection on the combined selected features from the filters

The included filters are Shannon entropy, Granger Causality, Mutual Information (MI), Pearson and Spearman:

Shannon entropy. The Shannon entropy is a measure of unpredictability, which means that features with unique probabilities will be ranked higher. The entropy of a feature, $H(X)$ is defined as:

$$H(X) = -\sum_{i} P(x_i) \log P(x_i) \tag{2}$$

with $P(x_i)$ being the probability density function, X is the feature and x_i are the samples of that feature [14].

Granger Casuality. A variable X "Granger-causes" Y if Y can be better predicted using the histories of both X and Y than it can using the history of Y alone [3]. The Granger causality score is calculated by creating two linear regressions; one that only contains the samples from Y and one that also includes the samples from X. F-tests are used to reject the null hypothesis that X does not Granger-cause Y. By using an F-distribution every input X_i can be scored as to how much it causes Y.

Mutual Information (M). The MI is a measure of the distance between the distributions of two variables and hence the dependence of the variables [2]. Choosing features with high dependence to the output could indicate that the feature is good for predicting the output. The MI of an input X to the output Y is defined as [2]:

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \tag{3}$$

with $p(x,y)$ being the joint probability density function and $p(x)$ and $p(y)$ being marginal probability distribution functions of X and Y respectively.

Pearson. The Pearson correlation coefficient is a measure of the dependence between two variables [13]. The pearson correlation coefficient is defined as [13]:

$$\rho_{X,Y} = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y} \tag{4}$$

with cov as the covariance and σ_X and σ_Y are the standard deviations of X and Y respectively.

Spearman. Spearman's rank correlation coefficient is simply the Pearson correlation coefficient applied to ranked data [13]. Ranking the data will be more resistant to outliers [13], and does not assume the data is numerical. The ranking used is the normal ordering of the input.

2.2 Artificial Neural Network

Artificial Neural Networks (ANNs) are universal approximators [6] and have been applied successfully in regression based problems for many years. Multilayer Perceptrons (MLPs) are one of the most used ANNs also known as feedforward networks. An MLP will be used to attempt to solve the regression problems. The ANN uses the tanh activiation function, 20 nodes in the hidden layer and the RPROP training algorithm [10]. A rule of thumb states to use the average of the number of input features and the number of outputs as the number of nodes in the hidden layer. However, we found that fewer nodes results in a better generalization, and we've settled on 20 hidden nodes. The network is trained until an error of 0.001 % is achieved with a maximum of 500 iterations. For further information on the technique the reader is refered to [19]. The focus on this paper is on the feature selection, which is why we have chosen MLP which may be the most basic, but very efficient, type of ANNs. Researchers choice of ANN comes down to personal preference, the type of problem, but also what is *hip* at the time. Modern types of ANNs such as Deep Neural Networks have feature selection functionality as well, but feature selection as a pre-processing step will always decrease the learning complexity, regardless of the ANN type.

10-fold cross-validation is performed, and in each fold the network is trained and tested 10 times to decrease the influence by the randomness in the ANN. The average of those 10 repetitions is used for the comparison.

2.3 Statistics

The error of the prediction is calculated using the root mean square error (RMSE) measurement. The RMSE is defined as [13, p. 497]:

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n}(\hat{y}_t - y_t)^2}{n}} \tag{5}$$

where $\hat{y}_t$ is the predicted value and y_t is the actual value. The number of predictions in the series is denoted n. RMSE punishes negative and positive errors equally.

The feature selection methods will be compared against each other using a Wilcoxon signed-rank test. Because the output from all methods are used in the equally configured ANN, we assume the samples are dependent, and hence we must use a paired test. Because we have a small sample size, we cannot make any safe assumptions about the underlying distribution. For that reason a rank test is necessary. The Wilcoxon signed-rank test works any measurement type and returns a p-value on the null hypothesis that the two sample populations are identical. This also means that any specific error measurements or time measurements will not be presented. The Wilcoxon signed-rank test uses the following test statistic, W:

$$W = \sum_{i=1}^{N_r} [\operatorname{sgn}(x_{2,i} - x_{1,i}) \cdot R_i] \tag{6}$$

where N_r is the number of pairs with equalities removed, R_i is the rank of pair i and x_1 and x_2 are the pair samples.

2.4 Experiments

The feature selection algorithms will be tested against two problems. Both data sets are publicly available and will function as benchmarks for future comparison and experimentation. The problems are time-series problems and we assume the parameters change naturally over time. For this reason it is necessary to add the delayed input and output to the available input features. We did a grid search on a reduced version of one of the data sets, and found that a delay of 12 h gives the best performance. We assume this delay will perform equally well for the full data set as well as the other data set. With the delay the first problem will have a total of 285 input features and the second problem will have a total of 142 features. To perform the long term prediction the output is shifted accordingly.

Besides the prediction error the computational speed will also be measured. This gives an indication of the implications of using the feature selection algorithms. The time will also be measured 10 times in each cross fold to even out any randomness. The timer is started before the feature selection and stopped after the output from the ANN has been denormalized. There is no significant time difference between short-term and long-term predictions, so only the measured time from the short-term prediction will be used for comparison.

The data is first delayed and then divided into the crossfold bins. In each crossfold bin the feature selection is performed on the training part followed by a normalization of the entire data in the bin to prepare it for the ANN. After the ANN training the same features are selected from the test part and the ANN is run using this data.

3 Results and Discussion

3.1 Experiment 1 - SML2010 Data Set

The results of the performance of predicting the indoor temperature and the computational speed can be seen in Table 1. With 5 % significance the prediction with RDESF outperforms backward selection, NLPCA and PCA on both short-term and long-term. RDESF also outperforms using all available features on short-term and with 10 % significance on long-term. Only forward selection is able to get a better prediction than RDESF and only on short-term. On long-term there is not enough statistical significance to make any statements. It was expected that forward selection performs best because it includes the model in the feature selection process. Interestingly, the backward selection also includes the model in the selection process but does not have the same performance.

Table 1. p-values from the Wilcoxon signed-rank test for methods compared to RDESF for the SML2010 data set

	Backward	Forward	NLPCA	PCA	All input features
Short-term error	0.019	1.0	0.001	0.003	0.014
Long-term error	0.018	0.862	0.001	0.002	0.053
Computational speed	0.001	0.001	0.001	0.001	1.0

Looking at the computational speed in Table 1, RDESF outperforms all other algorithms with 1 % significance. Only using no algorithms at all is faster and therefore best in terms of computational speed. Using all input features equals no pre-calculations before the prediction and the fact that the MLP can be computed in parallel makes this the fastest option. This result will not reproduce for other types of neural networks because they are not suited for a large number of input parameters, for example Support Vector Regression (SVR). A large increase in the number of features will also affect the speed for MLP, however this has not been encountered yet.

3.2 Experiment 2 - Temperature Forecasting

Results from the temperature prediction from the design reference year data set can be seen in Table 2. The results do not change much between short term

Table 2. p-values from the Wilcoxon signed-rank test for methods compared to RDESF for the reference year data set

	Backward	Forward	NLPCA	PCA	All input features
Short-term error	0.652	1.0	0.001	0.003	0.52
Long-term error	0.313	1.0	0.001	0.002	0.423
Computational speed	0.001	0.001	0.001	0.001	1.0

and long term prediction as seen in experiment 1. Our algorithm RDESF clearly outperforms both NLPCA and PCA (with 1 % significance) when it comes to the prediction error. There is not statistical significance for the performance of RDESF against using all available input features nor backward selection. Again RDESF is only outperformed by the forward selection.

Just like in experiment 1, we expected that forward selection would outperform RDESF, because the model is included in the feature selection process. In experiment 1 we see that backward selection does not have the same superior performance as forward selection. Forward and backward selection both have advantages and disadvantages and it might be that the disadvantages of backward selection is influencing the results. One could overcome the disadvantages of both selection algorithms by using stepwise regression that combines forward and backward selection, but that will not be further investigated in this paper.

The computational speed of RDESF outperforms all other algorithms with 1 % significance just as in experiment 1. As expected using all input features is also faster in this experiment. However, keep in mind that using all input features will increase the computational time heavily for other types of neural networks.

3.3 Discussion of RDESF

The filters we chose to include in RDESF are by no means final. Mutual information (MI), Pearson and Spearman ranking are all measures of dependency. Their effectiveness is based on the assumption that a dependence between an input feature and the output will equal a better prediction performance. The Shannon entropy ranks features higher that are unique with respect to their probability density function. Selecting features with a high Shannon entropy score will include features that are different from each other and thereby decreasing the amount of similar information given to the prediction algorithm. The last included filter was the Granger causality which investigates if histories of the input feature and the output are better together. We believe that this mix of different types of filter makes RDESF strong and a generic solution to automatic feature selection of the filter-class. We will continue to investigate the performance of RDESF with other filters.

The numbers of filters included in RDESF has a big influence. Too many filters will decrease the influence of the individual filter. Because of the filter-wise selection, all possible features can be selected if too many filters are included. On the other hand, too few filters will mean the individual filters are too influential. If the filters have measurements, it means that the 10 % from every filter will be almost identical. The meta approach implies that a variety of included filters with different types of measurements will result in a better performance.

The union selection used as the distinct selection in RDESF was the obvious choice for us. Other set theory selections should be investigated such as intersection or even cartesian product. Other elaborate possibilities such as heuristics or voting systems should also be further investigated in future work.

Our initial goal was to beat the prediction performance by using PCA. PCA is included in MATLAB which is often used in machine learning. PCA is often the default feature selection method used and it has clear advantages such as reducing the dimension space and reversibility. It is a very positive result that RDESF outperforms PCA. That RDESF outperforms or evens with using all input features is a good indication that RDESF will perform well for computationally heavy types of ANNs. Support Vector Regression is one of those types of network, and preliminary results show that RDESF performs equally well for SVRs and RBFs. Testing the RDESF with other types of ANNs and other application areas are planned for further research.

Through careful implementation of the included filters, RDESF is very fast, especially compared to the computational intensive NLPCA and forward and backward selection. We've implemented RDESF through a mix of the Apache Commons library and an optimized use of data structures in Java. The speed can be further improved by the use of multi-threading, with every filter scoring the features simultaneously.

4 Conclusion

A novel filter-class algorithm for automated selection of features has been proposed. Our algorithm called Ranked Distinct Elitism Selection Filter (RDESF) was tested in two experiments. The filter-class of feature selection are model-free and thereby fast. A big drawback of filters is the requirement of choosing a threshold to select the features. This problem was overcome by creating a meta-filter that selects the top 10 % features for each included filter. To avoid duplicate features a distinct selection was performed, in this case a union selection.

The experiments in which RDESF was tested were time-series regression based problems of prediction a variable. RDESF was only outperformed by forward selection which was expected, because the prediction model is included in the forward selection process. All the other feature selection algorithms which were: Backward selection, NLPCA and PCA were all outperformed by RDESF. In the first experiment RDESF even outperformed using all input features, that indicates a good performance in computationally heavy types of ANNs. The computational speed of RDESF was only outperformed by using all input features which was also expected, since no pre-calculations are required.

RDESF clearly outperformed PCA in both prediction error and computational speed, which was our benchmark baseline. Unlike PCA, RDESF allows for feature analysis in fault detection scenarios because the features are not transformed. This means that RDESF is a strong competitor in the feature selection field, and applicable to a variety of application areas. The meta-filter approach does not require the user to select a threshold for the filter which allows the user to include filters into an automatic feature selection process or system.

References

1. Cavanaugh, J.E.: Unifying the derivations for the akaike and corrected akaike information criteria. Stat. Probab. Lett. **33**(2), 201–208 (1997)
2. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley, New York (1991)
3. Granger, C.: Some recent development in a concept of causality. J. Econometrics **39**, 199–211 (1988)
4. Guttman, L.: Some necessary conditions for common-factor analysis. Psychometrika **19**(2), 149–161 (1954)
5. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. J. Mach. Learn. Res. **3**, 1157–1182 (2003)
6. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Netw. **2**(5), 359–366 (1989)
7. Kramer, M.A.: Nonlinear principal component analysis using autoassociative neural networks. AIChE J. **37**(2), 233–243 (1991)
8. Lichman, M.: UCI machine learning repository (2013). http://archive.ics.uci.edu/ml
9. Nautical Almanac Office: Almanac for Computers 1990. U.S Government Printing Office (1989)
10. Riedmiller, M., Braun, H.: A direct adaptive method for faster backpropagation learning: the RPROP algorithm. In: IEEE International Conference on Neural Networks, pp. 586–591. IEEE (1993)
11. May, R., Dandy, G., Maier, H.: Review of Input Variable Selection Methods for Artificial Neural Networks. InTech, April 2011
12. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by backpropagating errors. Cogn. Model. **5**, 1 (1988)
13. Shanmugan, K.S., Breipohl, A.M.: Random Signals: Detection, Estimation, and Data Analysis. Wiley, New York (1988)
14. Shannon, C.E.: A mathematical theory of communication. SIGMOBILE Mob. Comput. Commun. Rev. **5**(1), 3–55 (2001)
15. Shlens, J.: A tutorial on principal component analysis. arXiv preprint arXiv:1404.1100 (2014)
16. Wang, P.G., Mikael, S., Nielsen, K.P., Wittchen, K.B., Kern-Hansen, C.: Reference climate dataset for technical dimensioning in building, construction and other sectors. DMI Technical reports (2013)
17. Whiteson, S., Stone, P., Stanley, K.O., Miikkulainen, R., Kohl, N.: Automatic feature selection in neuroevolution. In: GECCO (2005)
18. Zamora-Martínez, F., Romeu, P., Botella-Rocamora, P., Pardo, J.: On-line learning of indoor temperature forecasting models towards energy efficiency. Energy Build. **83**, 162–172 (2014)
19. Zhang, G., Patuwo, B.E., Hu, M.Y.: Forecasting with artificial neural networks: the state of the art. Int. J. Forecast. **14**(1), 35–62 (1998)