Proc. 17th European Symposium On Artifi˝cial Neural Networks - Advances in Computational Intelligence and Learning (ESANN 2009), pp. 583-588, d-side publication 2009

ESANN'2009 proceedings, European Symposium on Artificial Neural Networks - Advances in Computational Intelligence and Learning. Bruges (Belgium), 22-24 April 2009, d-side publi., ISBN 2-930307-09-9.

Forward Feature Selection Using Residual Mutual Information

Erik Schaffernicht, Christoph Möller, Klaus Debes and Horst-Michael Gross

Ilmenau University of Technology - Neuroinformatics and Cognitive Robotics Lab
98693 Ilmenau - Germany

Abstract. In this paper, we propose a hybrid filter/wrapper approach for fast feature selection using the *Residual Mutual Information* (RMI) between the function approximator output and the remaining features as selection criterion. This approach can handle redundancies in the data as well as the bias of the employed learning machine while keeping the number of required training and evaluation procedures low.

In classification experiments, we compare the *Residual Mutual Information* algorithm with other basic approaches for feature subset selection that use similar selection criteria. The efficiency and effectiveness of our method are demonstrated by the obtained results on UCI datasets.

1 Introduction

In supervised learning, there are often tasks that provide plenty of input variables, many of which are not required for predicting the target value. These irrelevant or redundant features complicate the learning process for neural networks and other learning machines, since the highdimensional decision space increases the problem of overfitting, may interfere with the generalization abilities and requires more time for the adaptation process. Feature selection methods are designed to select a sufficiently small subset of meaningful features to mitigate these problems.

The taxonomy of feature selection methods distinguishes between three different types: the *Filter* methods, the *Embedded* methods and the *Wrapper* methods (see [1] and [2]).

Filter methods operate on the data to find the intrinsic relations between the variables and the target, prior to and independent of any application of a learning machine. Examples include the well known Principal Component Analysis, the linear correlation coefficient or information theoretic measures like *Mutual Information* [3]. Filter methods are often the cheapest approaches in terms of computational demands.

Embedded methods use certain learning machines, like neural networks, which are adapted with all data channels available. After the training process is complete, the importance of the inputs can be inferred from the structure of the resulting classifier. This includes e.g. weight pruning in neural network architectures with *Optimal Brain Damage* [4] or *Bayes Neural Networks* [5], or *Recursive Feature Elimination* for SVMs [6].

Wrapper methods employ arbitrary learning machines, which are considered black boxes. The feature selection algorithm is wrapped around the black box, hence the name. An interesting feature subset is determined by a search strategy and evaluated with the classifier adapted to this subset. The bias of the learning machine is taken into account opposed to the pure feature relevance determined by a filter method. The advantages of the bias implicit view were discussed e.g. in [1].

The essential disadvantage of wrappers that prevents thire use for large data sets are the associated computational costs. Hence, recent developments aim at combining filter and wrapper methods to speed up the wrapper algorithm by a preprocessing with filter methods. For example [7] applies a *Mutual Information* based relevancy and redundancy filter to prune the input variables for a subsequent genetic wrapper search strategy. The *Chow-Liu trees* employed in [8] are used to limit the candidate variables in each deterministic forward selection step. The construction of these trees is again a filter like preprocessing and operates with *Mutual Information*, too.

Along those lines, we propose a new hybrid filter/wrapper algorithm for feature selection that operates with the *Residual Mutual Information* (RMI) as filter component in a forward wrapper. The next section will describe the algorithm in detail. Thereafter, in section 3 experimental results using the method are given in comparison to other approaches.

2 Residual Mutual Information

The presented RMI method for feature selection is based on the basic *Mutual Information* criterion (MI) and a standard *Sequential Forward Selection* (SFS). Thus both parent algorithms are introduced briefly before the RMI symbiosis is explained.

The symbols used in this section are defined as follows: T is the one dimensional vector of target values, Y is the corresponding network output and R describes the residual $R = T - Y$. F is the set of n available features, the subset of chosen features is denoted by S with $S \subseteq F$, while $F \setminus S$ is the candidate set of features not chosen for training.

2.1 Mutual Information Ranking

Mutual information measures the dependence between two variables, in the context of feature selection between one feature F_i and the target variable T.

$$I(F_i, T) = \int_{f_i} \int_t P(f_i, t) \log \frac{P(f_i, t)}{P(f_i)P(t)} dtdf_i$$

If the MI is zero, both variables are independent and contain no information about each other, thus the feature is irrelevant for the target. Higher MI values indicate more information about the target and hence a higher relevance. Simple feature ranking chooses a certain number of the highest ranked features or all features above a threshold as relevant features for the learning machine. More details can be found e.g. in [3].

2.2 Sequential Forward Selection

Sequential Forward Selection (SFS) is a simple search strategy to find useful features [9]. The basic SFS algorithm starts with an empty feature subset and adds one variable each step until a predefined number of features is reached, or the approximation result does not improve any further. For one step, each candidate feature is separately added to the current subset and subsequently evaluated. The feature that induced the highest improvement is included in the resulting subset. If the best new subsets improves more than a threshold, the algorithm continues with this subset, otherwise it terminates.

2.3 Residual Mutual Information

The first step of the RMI algorithm is the same as for *Mutual Information* ranking. After the information values for all features are computed, a neural network is trained with the single most informative input. Next the residual between network output and desired target is calculated, which in turn is used to calculate the next *Mutual Information* ranking. This yields again an informative input channel, which is added and the procedure repeats. More formal the algorithm can be state the following way:

1. Start with a emtpy subset $S \leftarrow \emptyset$ and initialize $R \leftarrow T$.
2. Find the feature with the maximum *Mutual Information* to the residual $F_{max} = \arg\max_{F_i} [I(F_i, R)]$.
3. Add it to the subset of chosen features and remove it from the candidate set of features: $S \leftarrow S \cup F_{max}$; $F \leftarrow F \setminus F_{max}$
4. Train the neural network with the feature subset S as input and the target T.
5. Compute the new residual $R \leftarrow T - Y$ and return to 2 or end if stopping criterion is fulfilled.

Possible stopping criteria include but are not limited to: a certain number of features chosen, a residual below a threshold, or the maximum *Mutual Information* below a threshold.

2.4 Discussion

The advantage of using the residual as guiding principle to choose informative features is twofold. First, the residual contains the unexplained part of the target values, hence any redundant candidate variables will not correlate with the residual. Second, the actual use of the neural network helps to cope with the bias of the learning machine.

In the optimal case, the used classifier has no bias and all information present in the chosen features are used to form an accurate prediction. The absolute value of the residual is a measurement how good the prediction is. Finding features that can predict the current function approximator's performance have to contain additional information about those samples the current predictor is unable to classify correctly. Otherwise it won't be possible to find dependencies to the residual. Thus any feature that contains information about the performance of the classifier is a useful feature.

During the selection process, the set of features F is split into two sets S and $F \setminus S$. Except for redundancies, this is true for the information contained in the features as well:

$$I(F,T) = I(S,T) + I(F \setminus S, T) - TC(F \setminus S, S, T). \tag{1}$$

The last term TC denotes the *Total Correlation* [10]

$$TC(F \setminus S, S, T) = \int_{F \setminus S} \int_S \int_T P(F \setminus S, S, T) \log \frac{P(F \setminus S, S, T)}{P(F \setminus S)P(S)P(T)}$$

between both subsets and the target, and captures the redundancies between both sets concerning the target.

The goal of the search strategy is to find a subset S with $I(F,T) = I(S,T)$, where the subset has as much information about the target as the complete set of features. This is the case if the features not chosen only add redundancies about the target, thus $I(F \setminus S, T) = TC(F \setminus S, S, T)$. These redundancies are handled by the use of the residual, because if the information of the subset is represented in the output Y, any redundant features cannot be used to gain information about the function approximator's performance.

The second benefit of using the residual is the fact that the bias of the learning machine is represented in the residual. This bias of the classifier introduces a loss of information I_{bias} e.g. due to limits of the chosen model. This implies that only a part of the information is in the output I_{net}.

$$I(S,T) = I_{net}(S,T) + I_{bias}(S,T) \tag{2}$$

Apparently, any information that was lost in the black box due to the model selection leads to a higher error in the output and thus is part of the residual.

To conclude, the MI between the residual and the candidate features $I(I(F \setminus S, R)$ includes the bias $I_{bias}(S,T)$ of the function approximator and redundancies of the remaining features $I(F \setminus S, T) - TC(F \setminus S, S, T)$. We reformulate Eq.(1) and restate the problem:

$$I(F,T) = I_{net}(S,T) + I(F \setminus S, R). \tag{3}$$

The goal is to maximise the information used by the function approximator $I_{net}(S,T)$. Thus if the term $I(F \setminus S, R)$ is zero, all information is in the subset of the features used by the classifier. Hence the algorithm selects the feature with the maximal MI between features and residual to reduce $I(F \setminus S, R)$ as fast as possible.

3 Experiments and Results

For evaluating the RMI algorithm, it was compared to related approaches on data sets from the UCI repository [11]. The used classifier is a Multi Layer Perceptron with two hidden layers with 20 and 10 hidden units respectively. This empirically chosen neural net took the role of the black box for the wrapper as well as final classification instance after the feature selection process. All problems are binary classification tasks and the *balanced error rate* was used for a 10-fold cross-validated evaluation.

The RMI method is compared to three different methods: *Sequential Forward Selection* (SFS), *Mutual Information for Feature Selection* (MIFS) and *Chow-Liu trees for feature selection* (CLT-FS). The basic SFS idea was explained in section 2.2. Instead of a mutual information ranking, the more sophisticated MIFS filter approach [12] was used. This method has the advantage of being able to handle redundancies in the input channels, like all the other algorithms. This is achieved by approximating the *Joint Mutual Information* by sums over pairwise *Mutual Information* between features. The third reference is the CLT-FS [8] where the wrapper search process is speeded up by restricting the candidates based on a tree structure constructed by MI values between features and targets. Additionally, the results of the neural network without any feature selection (All) are provided. The results are summarized in table 1.

Data Set	Ionosphere	Spambase	GermanCredit	Breast Cancer
Features	34	57	24	30
Samples	351	4601	1000	569
All	20.08(34/-)	13.81(57/-)	41.70(24/-)	13.78(30/-)
SFS	18.47(3/130)	17.39(8/477)	39.06(4/110)	13.44(4/140)
MIFS	24.54(5/-)	16.29(18/-)	37.47(6/-)	12.48(5/-)
CLT-FS	18.12(6/38)	17.26(9/97)	38.52(3/24)	9.37(8/37)
RMI	17.08(5/6)	13.93(54/55)	39.73(15/16)	8.58 (5/6)

Table 1: The results for the different data sets and feature selection methods. The balanced error rate is given in percent. The number of chosen features and the number of instances when a network and evaluation training cycle is required during the selection process, are shown in parentheses.

The results in the table show that the RMI approach is quite capable of producing results on par with the other approaches, yet it is considerably faster regarding the required number of instances to train and evaluate the classifier. One interesting observation is that despite using the same stopping criterion (no further improvement compared to the previous classifier) for all four test, in two of them (GermanCredit and Spambase), a high number of features was chosen by the RMI method. This seems like a conservative behavior of keeping as many features as possible. But for both of the other test data sets, RMI selected a number of features very similar to the best competing methods. Changing the threshold for the termination condition relieves the problem to some degree, but it is rather unclear, why this contrast exists in the first place.

According to the algorithm, the number of adaption cycles that are required for the RMI approach equals the amount of chosen features plus one. The consequence is that not every feature is evaluated, but all untested features are even less informative than that one, whose evaluation terminated the algorithm.

4 Conclusion

We proposed a hybrid Filter/Wrapper approach for fast feature selection. The *Mutual Information* between the features and the residual error of the learning machine is used to determine the feature relevance, taking into consideration redundancies between features and the bias of the used black box function approximator.

The resulting algorithm is very fast, since the expensive task of adapting the function approximator is done at most once per feature. The resulting performance is on par with the other approaches, despite fewer adaption cycles. This leads to the conclusion that the use of the residual information is expedient.

References

[1] Kohavi, R., John, G.H.: Wrappers for feature subset selection. *Artifical Intelligence* **97** (1997) 273–324

[2] Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. *Journal of Machine Learning Research* **3** (2003) 1157–1182

[3] Torkkola, K.: Information-Theoretic Methods. *Feature Extraction Foundations and Applications* StudFuzz 207, 167–185 Springer 2006

[4] LeCun, Y., Denker, J., Solla, S., Howard, R. E., Jackel, L. D.: Optimal Brain Damage. *Advances in Neural Information Processing Systems* **2**. Morgan Kaufmann (1990)

[5] Neal, R.M.: Bayesian Learning for Neural Networks. Springer. (1996)

[6] Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene Selection fo Cancer Classification Using Support Vector Machines. *Machine Learning* **46**, 2002

[7] Van Dijck, G., Van Hulle, M.M.: Speeding Up the Wrapper Feature Subset Selection in Regression by Mutual Information Relevance and Redundancy Analysis. *International Conference on Artificial Neural Networks* (ICANN 2006) Lecture Notes in Computer Science 4131, 31–40 Springer 2006

[8] Schaffernicht, E., Stephan, V., Gross, H.-M.: An Efficient Search Strategy for Feature Selection Using Chow-Liu Trees. *International Conference on Artificial Neural Networks* (ICANN 2007) Lecture Notes in Computer Science 4669, 190–199 Springer 2007

[9] Reunanen, J.: Search Strategies. *Feature Extraction Foundations and Applications* StudFuzz 207, 119–136 Springer 2006

[10] Watanabe, S.: Information Theoretical Analysis of Multivariate Correlation. *IBM Journal of Research and Development* **4** (1960) 66–82

[11] Newman, D.J., Hettich, S., Blake, S.L., Merz, C.J.: UCI Repository of machine learning databases (1998) http://www.ics.uci.edu/∼mlearn/MLRepository.html

[12] Battiti, R.: Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks* **5** (1994) 537–550.