in: Artificial Neural Networks and Machine Learning - ICANN 2011, Espoo, Finland, LNCS, vol. 6792, Part II, pp. 181-188, Springer 2011

Weighted Mutual Information for Feature Selection

Erik Schaffernicht and Horst-Michael Gross

Ilmenau University of Technology
Neuroinformatics and Cognitive Robotics Lab
98693 Ilmenau, Germany
Erik.Schaffernicht@Tu-Ilmenau.de

Abstract. In this paper, we apply weighted Mutual Information for effective feature selection. The presented hybrid filter wrapper approach resembles the well known AdaBoost algorithm by focusing on those samples that are not classified or approximated correctly using the selected features. Redundancies and bias of the employed learning machine are handled implicitly by our approach.

In experiments, we compare the weighted 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.

1 Introduction

In supervised learning, there are often tasks that provide plenty of input variables, many of which are not required for the classification or approximation task in question. These irrelevant or redundant features complicate learning processes. The highdimensional decision space increases the problem of overfitting, it may interfere with generalization abilities, and requires more time for the learning.

Feature selection methods are designed to select a sufficiently small subset of meaningful features to mitigate these problems. There are three different types of feature selection approaches: the *Filter* methods, the *Embedded* methods and the *Wrapper* methods [1], [2].

Filter methods operate on the data to find correlation between the variables and the target, independent of any 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 employ specific learning machines, like neural networks, which learn with all input channels. 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], Bayes Neural Networks [5], or Recursive Feature Selection for SVMs [6].

T. Honkela et al. (Eds.): ICANN 2011, Part II, LNCS 6792, pp. 181–188, 2011.

Wrapper methods [9] employ arbitrary learning machines, which are considered black boxes. The feature selection algorithm is wrapped around the black box. 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 often prevents their use for large data sets are the associated computational costs. Recent developments aim at combining filter and wrapper methods to speed up the wrapper algorithm by a preprocessing with filter methods, while keeping the bias implicit view.

In this paper, we propose a hybrid filter/wrapper algorithm that employs weighted Mutual Information as the filter component for a forward selection.

In the next section, we will discuss Mutual Information, its weighted version and methods to estimate it. Given these foundations, the algorithm for selecting features will be presented and discussed in Section 3. Before experimental results are provided in Section 5, we will discuss related work in Section 4.

2 Weighted Mutual Information

The well known Mutual Information (MI) measures the dependence between two random variables, in the context of feature selection between one feature F_i and the target T (e.g. the class labels). Lower case letter indicate the realization of the random variables.

$$I(F_i, T) = \int_{f_i} \int_t p(f_i, t) \log \frac{p(f_i, t)}{p(f_i)p(t)} dt df_i. \tag{1}$$

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].

Weighted Mutual Information

The idea of a weighted form of Mutual Information is mentioned in [10], but it has not gained popularity because the number of meaningful applications is limited. It is defined as

$$wI(F_i; T; W) = \int_{f_i} \int_t w(s_j) p(f_i, t) \log \frac{p(f_i, t)}{p(f_i)p(t)} dt df_i. \tag{2}$$

For each sample s_j, a combination of input value f_i and target value t, a weight $w(s_j) \geq 0$ is imposed. This results in a specific relevance of each unique sample not unlike a prior on how informative a certain combination is. We will use this to weight the influence of the different input samples.

Estimation of Weighted Mutual Information

The main problem with the above Eq. 1 is the estimation of the required probability densities from the available data. The straightforward approach to compute the MI uses histograms for approximation, simplifying the formula to sums instead of integrals or to apply kernel density estimation methods. A comparison between these and more sophisticated methods for estimating the MI can be found in [7] and, specifically applied to the feature selection domain, in [8].

The estimation of the weighted Mutual Information is straightforward only for some of the estimation approaches. Instead of Eq. 2 it is easier to estimate the equivalent formulation

$$wI(F_i;T;W) = \int_{f_i}\int_{t} w(s_j)p(f_i,t)\log\frac{w(s_j)p(f_i,t)}{w(s_j)p(f_i)p(t)}dtdf_i. \tag{3}$$

Effectively, this realizes the weighted Mutual Information by manipulating the probability distributions. Each sample contributes to the probability density according to its weight only (zero weight samples don't contribute anything), which can be compared to the particle representation used in Particle Filters.

Using the histogram approach, which discretizes the data according to some strategy and then replaces the integrals in Eq. 1 with sums over the discrete histogram bins, the use of the weight is trivial. Each sample does not contribute equally to its bin, but according to its weight. Similar is the weighted variant of kernel density estimation, where in practice the sum of pairwise interacting kernels is calculated. The sample weighting is implemented by manipulating the kernel put at the sample's position.

Other popular methods, like Kraskov's k-nearest neighbour estimator [11], which are not based on the Kullback-Leibler divergence formulation, but entropy estimators, are not easily modified to compute the weighted Mutual Information.

3 Feature Selection with Weighted Mutual Information

The basic idea of the feature selection algorithm proposed here is the following: Given a classifier/approximator and its error, the choice which feature to include next to improve the performance should be based on the errors made and not on all available data. This is done by weighting the correct and wrong classified samples differently.

The simple case is if a classifier only produces discrete class information. In the next step, any correctly classified samples are left out for the computation of the MI between samples and target, because they have a weight of zero. Only the wrongly classified samples are used, they have an equal weight.

The use of a continuous predictor allows for a different weighting for each sample according to the residual. For example, a sample that is classified as positive, but which is near the boundary to the negative class will yield a non zero residual despite being in the correct class. But its influence should be smaller compared to a sample that is on the wrong side of the decision boundary.

Algorithm 1. $S = \text{wMI}\ (X, Y)$

Input: data set of observations X and the corresponding labels Y
Output: final feature set S and the final classifier

$S \leftarrow \emptyset$
$W \leftarrow 1$ {Same weight for all samples.}
while stopping criterion not true **do**
  $F_{max} = \arg\max_{F_i} [wI(F_i, T, W)]$ {Find feature with maximum weighted MI}
  $S \leftarrow S \cup F_{max}$ {Add feature to the subset}
  $F \leftarrow F \setminus F_{max}$ {Remove feature from the candidate set}
  Classifier $\leftarrow$ TrainClassifier(X, S, Y)
  $Y' \leftarrow$ ApplyClassifier (Classifier, X)
  $W \leftarrow |Y - Y'|$ {Residual for each sample is the new weight.}
  CheckStoppingCriterion()
end while

Thereto, we apply the weighted variant of the Mutual Information in a forward selection framework. Starting with an empty feature set, the Mutual Information, more precisely the weighted Mutual Information with an equal weight for all samples, is computed between all available features and the target. The feature yielding the maximal Mutual Information is selected like in a simple ranking algorithm. Then the classifier/aproximator is trained with this variable. The resulting residual for each data sample is of interest, because it is used to define the weights for the next selection round employing weighted Mutual Information. The next feature is chosen based on the maximum weighted Mutual Information and the classifier/approximator is retrained including the new feature channel. This repeats until the stopping criterion is met. A pseudocode description of this cycle is given in algorithm 1.

This resembles one of the basic ideas of the well known AdaBoost algorithm [12]. All samples that are misclassified are given a higher importance for the next round, while all correct samples are less important. The reasoning is simple: all correct classified samples are sufficiently explained by the current subset of features and there is the need to find those features that explain the misclassified samples.

The scaling of the values Y for real valued targets can be arbitrary, since the absolute value of the weighted mutual information is not important but its relative value to the other features, which are computed using the same weights.

Taking a look at global function approximators, like MLPs, this is not a problem. They are able to find again the decision surface they found the round before albeit it is now in a subspace of the space spanned by the features including the newly chosen one. The new dimension adds more options to find a better decision surface, but the same result is always achievable.

For classifiers with local activation functions, like RBF networks or nearest neighbour classifieres, it is a bit more complicated, especially for very low dimensional cases. The neighborhood of a sample can change dramatically by the addition of a new feature. Obviously, this effect is less dramatic in higher

dimensional spaces since new features affect the neighbourhood less. But the overall performance of the classifier may decrease in the early stages as a result. The algorithm will try to correct this based on the new residuals and chose features that compensate for the newly introduced errors, but as a consequence the error rates jump up and down.

Finding a good stopping criterion can be difficult but crucial, especially for local approximators. The algorithm starts with an empty feature subset and adds one variable to the final subset in each step until a predefined number of features is reached, or the approximation result does not improve further. If the best new subset improves more than a threshold ε, which can be negative to allow a decreasing performance, the new subset is confirmed, and another round begins, otherwise the algorithm terminates. Other possible stopping criteria are a fixed number of rounds, which equals the number of chosen features in the end, or a certain approximation error of the resulting classifier/approximator.

4 Related Work

The use of *Mutual Information* for feature selection is quite common. Besides the simple ranking approach [3], which cannot handle redundancies at all, a notable representative is the MIFS algorithm [13] and its extensions. The MIFS algorithm approximates the *Joint Mutual Information* by sums over pairwise *Mutual Information* between features. These approaches are pure filter methods and don't take the learning machine into account, and hence, do not compensate for the bias (as in the sense of the bias-variance dilemma) introduced.

The wrapper methods with the basic forward and backward selection methods [9] care for the bias, but require much time to search for good feature sets. Some proposed methods, like floating search algorithms, increase the number of searched subsets by combining forward and backward steps, or add or remove multiple features at once. This increases the required time even further and is not feasible for larger data sets. The other direction tries to reduce the number of tested subspaces, hopefully without missing the interesting ones.

Combining MI-based filters with wrapper methods is one approach. In [14] wrappers are used to refine candidate subsets provided by a incremental filter method based on MI related measures. [15] 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 [16] are used to limit the candidate variables in each deterministic forward selection step. The construction of these trees is again a filterlike preprocessing and operates with *Mutual Information*, too. All of these methods operate with the MI between the input channels and the labels, or the input channels with each other, but do not take into account the actual classifier output.

One of the few methods that uses the output of a learning machine for MI computation in the context of feature selection is presented in Torkkola [17]. The idea is based on the information theoretic learning framework and computes the

Quadratic Mutual Information between the output and the desired target to adjust a feature transformation using a gradient ascent. The method itself is a filter approach, because the learning machine only learns the feature transformation and does not provide the classification or approximation results.

The most similar existing approach is the *Residual Mutual Information* (RMI) algorithm [18]. It computes the basic Mutual Information between the target and the residual produced by a learning machine. In this case, the idea of using the residual is different. It implies, that input variables that have information about the error the classifier or approximator is going to make are useful for reducing that error and hence those variables are chosen.

5 Experiments

For evaluating the algorithm presented in this paper, it is compared to related approaches on data sets from the UCI repository [19] and one larger artifical dataset generated with known properties (200 input variables, low intrinsic dimensionality (7), linear, non-linear and XOR functional dependencies, redundancies and noise on all variables). The used classifier were the $k = 3$ nearest neighbour classifier and a Multi Layer Perceptron with two hidden layers with 20 and 10 hidden units, respectively. These classifiers took the role of the black box for the wrapper as well as final classification instance after the feature selection process. The hyperparameters for the classifiers were fixed for all experiments, since the model selection problem is not considered here. Thus, there were no adjustments for the different data sets or algorithms and there will be a rather large bias in the error. During the feature selection process, a 3-fold crossvalidation data split was used to estimate the valid features, while for the final predictor evaluation itself we used a 10-fold cross-validation [20]. All problems are binary classification tasks, hence the *balanced error rate* (BER) was used as error measure.

The proposed method using the weighted Mutual Information (WMI) is compared to different algorithms: basic *Sequential Forward Selection* (SFS), *Mutual Information for Feature Selection* (MIFS with $\beta = 0.15$) and *Chow-Liu trees for feature selection* (CLT-FS) and residual *Mutual Information* (RMI). References for those algorithms are included in the previous section. The results are summarized in Table 1 for the kNN and Table 2 for the MLP.

By looking at the numbers in the tables, it is obvious that the conjecture from section 3 regarding local classifiers is true. The performance of the proposed algorithm with the nearest neigbour approach is moderate at best. When combined with the MLP much better results are achieved using the weighted Mutual Information approach.

The number of required steps of adapting and evaluating a classifier are the least besides the filter and the RMI method. Especially for big data sets the linear dependency on the number features is benefical compared to the quadratic dependency of the SFS or the logarithmic one for the CLT method [16].

Table 1. The results for the different data sets and feature selection methods using the kNN. The balanced error rate is given in percent. The number of chosen features and the number of classifier training and evaluation cycles) for one crossvalidation step are shown in parentheses.

Data Set	Ionosphere	Spambase	GermanCredit	Breast Cancer	Artifical Data
Features	34	57	24	30	200
Samples	351	4601	1000	569	3000
All	23.78(34/-)	10.84(57/-)	36.33(24/-)	3.55(30/-)	35.36(200/-)
SFS	12.04(5/189)	8.44(12/663)	31.61(7/164)	4.21(6/189)	16.20(8/1764)
MIFS	11.80(5/-)	8.65(19/-)	33.90(6/-)	4.36(5/-)	28.65(3/-)
CLT-FS	12.19(6/39)	15.97(6/76)	34.89(5/28)	4.42(4/30)	25.89(8/202)
RMI	13.82(5/6)	23.62(3/4)	35.45(5/6)	4.49 (3/4)	24.30(4/5)
WMI	11.57(5/6)	10.73(10/11)	33.31(8/9)	4.48(6/7)	29.52(5/6)

Table 2. The results for the different data sets and feature selection methods using the MLP. The balanced error rate is given in percent. The number of chosen features and the number of MLP training and evaluation cycles are shown in parentheses.

Data Set	Ionosphere	Spambase	GermanCredit	Breast Cancer	Artifical Data
Features	34	57	24	30	200
Samples	351	4601	1000	569	3000
All	20.08(34/-)	13.81(57/-)	41.70(24/-)	13.78(30/-)	33.65(200/-)
SFS	18.47(3/130)	17.39(8/477)	39.06(4/110)	13.44(4/140)	20.11(7/1572)
MIFS	24.54(5/-)	16.29(19/-)	37.47(6/-)	12.48(5/-)	26.87(3/-)
CLT-FS	18.12(6/38)	17.26(9/97)	38.52(3/24)	9.37(8/37)	24.08(9/217)
RMI	17.08(5/6)	13.93(54/55)	39.73(15/16)	8.58(5/6)	21.74(6/7)
WMI	16.97(5/6)	16.41(9/10)	39.52 (6/7)	8.03(3/4)	19.29(6/7)

6 Conclusion

In this paper, we introduced an algorithm using the weighted Mutual Information for effective feature selection.

The proposed algorithm aims at a low number of cycles a classifier has to be trained and evaluated during the feature selection process. The number for adapting and testing cycles is only linear in the number of features for the WMI algorithms, while achieving similar results compared to wrapper approaches using more cycles.

Based on the results for the kNN and the MLP, we conjecture, that using the residual is less useful if it is a binary value only. More sophisticated information in the residual can be exploited better.

Based on tests presented, it is hard to determine if using the classifier output in form of the residual (RMI algorithm) or as a weighting for the selection (WMI) is the better overall approach, but at least for classifiers with global activation functions the weighted Mutual Information approach is preferable. More extensive studies are required to answer this question conclusively.

References

1. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artifical Intelligence 97, 273–324 (1997)
2. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of Machine Learning Research 3, 1157–1182 (2003)
3. Torkkola, K.: Information-Theoretic Methods. In: Feature Extraction Foundations and Applications StudFuzz 207, pp. 167–185. Springer, Heidelberg (2006)
4. LeCun, Y., Denker, J., Solla, S., Howard, R.E., Jackel, L.D.: Optimal Brain Damage. In: Advances in Neural Information Processing Systems, vol. 2, pp. 598–605. Morgan Kaufmann, San Francisco (1990)
5. Neal, R.M.: Bayesian Learning for Neural Networks. Springer, Heidelberg (1996)
6. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene Selection fo Cancer Classification Using Support Vector Machines. Machine Learning 46 (2002)
7. Khan, S., Bandyopadhyay, S., Ganguly, A.R., Saigal, S., Erickson, D.J., Protopopescu, V., Ostrouchov, G.: Relative performance of mutual information estimation methods for quantifying the dependence among short and noisy data. Physical Review E 76, 026209 1–15 (2007)
8. Schaffernicht, E., Kaltenhaeuser, R., Verma, S.S., Gross, H.-M.: On estimating mutual information for feature selection. In: Diamantaras, K., Duch, W., Iliadis, L.S. (eds.) ICANN 2010. LNCS, vol. 6352, pp. 362–367. Springer, Heidelberg (2010)
9. Reunanen, J.: Search Strategies. In: Feature Extraction Foundations and Applications StudFuzz 207, pp. 119–136. Springer, Heidelberg (2006)
10. Guiasu, S.: Information Theory with Applications. McGraw-Hill Inc., New York (1977)
11. Kraskov, A., Stögbauer, H., Grassberger, P.: Estimating Mutual Information. Physical Review E 69 (2004)
12. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences 55, 119–139 (1997)
13. Battiti, R.: Using mutual information for selecting features in supervised neural net learning. IEEE Transactions on Neural Networks 5(4), 537–550 (1994)
14. Peng, H., Long, F., Ding, C.: Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. IEEE Trans. Pattern Analysis and Machine Intelligence 27, 1226–1238 (2005)
15. Van Dijck, G., Van Hulle, M.M.: Speeding up the wrapper feature subset selection in regression by mutual information relevance and redundancy analysis. In: Kollias, S.D., Stafylopatis, A., Duch, W., Oja, E. (eds.) ICANN 2006. LNCS, vol. 4131, pp. 31–40. Springer, Heidelberg (2006)
16. Schaffernicht, E., Stephan, V., Gross, H.-M.: An efficient search strategy for feature selection using chow-liu trees. In: de Sá, J.M., Alexandre, L.A., Duch, W., Mandic, D.P. (eds.) ICANN 2007. LNCS, vol. 4669, pp. 190–199. Springer, Heidelberg (2007)
17. Torkkola, K.: Feature Extraction by Non Parametric Mutual Information Maximization. Journal of Machine Learning Research 3, 1415–1438 (2003)
18. Schaffernicht, E., Moeller, C., Debes, K., Gross, H.-M.: Forward feature selection using Residual Mutual Information. In: 17th European Symposium on Artificial Neural Networks, ESANN 2009, pp. 583–588 (2009)
19. 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`
20. Reunanen, J.: Overfitting in Making Comparisons Between Variable Selection Methods. Journal of Machine Learning Research 3, 1371–1382 (2003)