An Efficient Search Strategy for Feature Selection Using Chow-Liu Trees

Erik Schaffernicht[1], Volker Stephan[2], and Horst-Michael Groß[1]

[1] Ilmenau Technical University
Department of Neuroinformatics and Cognitive Robotics, 98693 Ilmenau, Germany
[2] Powitec Intelligent Technologies GmbH, 45219 Essen-Kettwig, Germany
Erik.Schaffernicht@Tu-Ilmenau.de

Abstract. Within the taxonomy of feature extraction methods, recently the Wrapper approaches lost some popularity due to the associated computational burden, compared to Embedded or Filter methods. The dominating factor in terms of computational costs is the number of adaption cycles used to train the black box classifier or function approximator, e.g. a Multi Layer Perceptron. To keep a wrapper approach feasible, the number of adaption cycles has to be minimized, without increasing the risk of missing important feature subset combinations.

We propose a search strategy, that exploits the interesting properties of Chow-Liu trees to reduce the number of considered subsets significantly. Our approach restricts the candidate set of possible new features in a forward selection step to children from certain tree nodes. We compare our algorithm with some basic and well known approaches for feature subset selection. The results obtained demonstrate the efficiency and effectiveness of our method.

1 Introduction

If irrelevant features are used to adapt a Multi Layer Perceptron (MLP) to a certain task, the classifier has to handle a more complex decision surface. This leads to increased time requirements for a successful adaption, it may decrease the precision of the results and worsens the problem of overfitting. Therefore feature selection methods are applied to find and sort out the irrelevant features in the input data.

It is very common to characterize feature selection methods as "Filter", "Embedded" or "Wrapper" approaches (see [1] and [2]). Filter based approaches operate on the data to find intrinsic interrelations of the variables, prior to any application of a learning machine. On one hand, this includes data driven approaches like Principal Component Analysis or Non-Negative Matrix Factorization [3]. On the other hand, supervised methods are applied which investigate the correlation between the input data and the class labels or a target value. Examples are the linear correlation coefficient, Fisher discriminant analysis or information theoretic approaches.

Embedded methods use a specific learning machine, that is adapted with all data channels available. After the training process is complete, the importance

J. Marques de Sá et al. (Eds.): ICANN 2007, Part II, LNCS 4669, pp. 190–199, 2007.

of the inputs can be inferred from the structure of the resulting classifier. This includes e.g. weight pruning in neural network architectures with OBD [4], Bayes Neural Networks [5] or Random Forests [6].

The wrapper approach uses a learning machine, too, but the machine is arbitrary, since in this case it is considered a black box and the features selection algorithm wraps around the classifier, hence the name. A search strategy determines an interesting feature subset to train the learning machine. The resulting error rate is used as evaluation criterion in the subset search process. Since feature relevance and optimality with respect to the classification error rate is not always equivalent (as reported e.g. in [1]), it can be of advantage to use the same algorithm in the feature selection process and the classification task. The downside is, that this approach is prone to overfitting, a problem that has to be dealt with in additionally.

In a recent feature extraction competition (see results in [7]) the successful competitors used Embedded or Filter methods, while Wrappers were almost completely absent. In [7] the authors conclude, that Wrappers where omitted, not because of their capability, but their computational costs. Every time the search strategy determines a new candidate subset of features, the learning machine has to be adapted at least once, often even more, to produce reliable results. The time used to train the classifier is the dominating factor in terms of computational time. Since the used learning machine is considered a black box, it is not possible to optimize within the classifier without losing generality. Therefore, we aim to minimize the number of classifier evaluations imposed by the search algorithm without a significant increase of the risk of missing important feature subsets.

Our proposed method achieves this goal by constructing a Chow-Liu tree (CLT) [8] from the available data, see section 2. Then the obtained underlying tree structure is used by a forward search algorithm to create feature subsets. Through the inherent properties of the tree representation, the number of candidate subsets is considerably smaller, than e.g. in standard sequential forward selection methods [11]. As an positive side effect, possibly redundant features can be inferred directly from the CLT.

The main contribution of this work is the use of the CLT structure to minimize the number of evaluation steps. The paper is organized as follows. Section 2 explains the foundations of Chow-Liu trees, while the application of Chow-Liu trees in the context of feature selection is discussed in section 3. Some implication of the proposed method are discussed in section 4. Thereafter, we present some experimental results achieved with our method in comparison to other search strategies. Additionally, section 5 discusses related work of relevance, before we conclude in the final section.

2 Generation of Tree-Based Distributions

The basic idea of Chow-Liu trees (CLT) was presented in [8] and can be summarized as follows. In order to approximate a n-dimensional probability distribution, a first-order dependency tree with $n-1$ relationships is constructed. Within

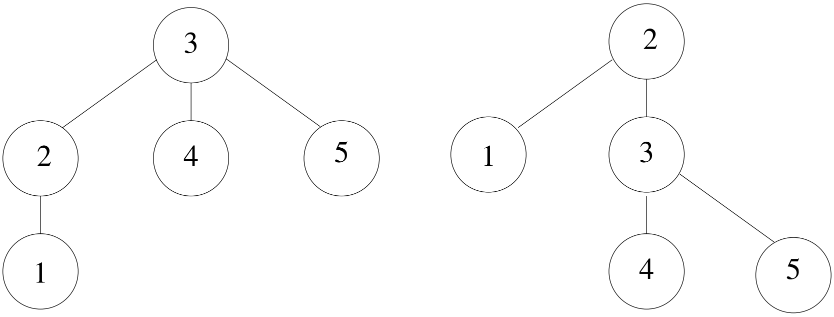

Fig. 1. Twice the same dependence tree with different root nodes. On the left the probability is expressed by $P(x) = P(x_3)P(x_4|x_3)P(x_5|x_3)P(x_2|x_3)P(x_1|x_2)$, on the right it is $P(x) = P(x_2)P(x_1|x_2)P(x_3|x_2)P(x_4|x_3)P(x_5|x_3)$.

the tree the underlying distribution is expressed as product of second-order distributions. The resulting representation can be used e.g. in pattern recognition tasks [8]. An example for a simple CLT is shown in figure 1. Please note that the choice of the root node is arbitrary. The algorithm to compute the tree structure minimizes the information difference between the original data and the dependency tree. It was shown, that the method is a maximum likelihood estimator for empirical data.

The problem of finding the optimal tree distribution is formulated as follow: Be $X = \left\{x^1, x^2, \ldots, x^N\right\}$ the given samples data set in the features space F and we are looking for the tree T_{opt} that maximizes the log likelihood of the data:

$$T_{opt} = \arg\max_T \sum_{i=1}^{N} \log T(x^i) \tag{1}$$

The solution is obtained in three steps. The algorithm is outlined below:

Algorithm 1. Chow-Liu-Tree(X)

```
Input: data set of observations $X$
Output: tree approximation $T_{opt}$

Determine the marginal distributions $P(x_i, x_j)$
Compute the mutual information matrix $I$
Compute the maximum-weight spanning tree $T_{opt}$
```

In the first part the pairwise mutual information I_{ij} between each pair of features $i, j \in F$ using the pairwise marginal distributions is computed:

$$I_{ij} = \sum_{x_i x_j} P(x_i, x_j) \log \frac{P(x_i, x_j)}{P(x_i)P(x_j)}, i \neq j \tag{2}$$

We used a histogram based approach to compute the pairwise marginal distributions and the mutual information, but kernel density estimation approaches are valid as well. For a more in depth discussion of different estimation methods, the interested reader is referred to [9].

The second part of the algorithm runs a maximum-weight spanning tree method on the mutual information matrix I, that is considered an adjacency matrix for this purpose. Beginning with the two nodes that have a maximum mutual information connection, further nodes are added with the next highest MI values. Any edges that would form a cycle are ignored. The resulting tree contains all nodes, while the sum of all weights of edges (which correspondes to the mutual information) in the tree is maximized. A modified Kruskal or Prim algorithm [10] can be applied for this task.

The obtained solution is non-ambiguous, if all the mutual information weights are different. Otherwise, if several weights are equal, the solution T_{opt} is possibly non-unique, but all alternatives still satisfy equation 1 and therefore this non-uniqueness property does not cause a problem.

In their work Chow and Liu showed, that the resulting dependence tree is indeed an optimal tree approximation of the underlying distribution.

3 Chow-Liu Trees for Feature Selection

We propose a supervised method for feature selection based on Chow-Liu trees. After further detailing our approach, we will discuss the benefits of using CLTs. We assume, that for each sample $x_i \in X$ we have a label $y_i \in Y$. We combine both information in a single matrix $Z = X \cup Y$, because for the purpose of constructing the tree, the labels are considered another input dimension. Then algorithm 1 is applied to compute the dependence tree. Each node of the tree now represents a feature or rather the label data.

Our algorithm uses the computed tree structure to guide the search process, that resembles the sequential forward selection strategy (SFS) (see chapter 4.3

Algorithm 2. SequentialForwardSelection(S, C, X, Y, E_S)

```
Input: data set of observations $X$, the corresponding labels $Y$, the current feature subset $S$, the candidate set of new features $C$, and the approximation error $E_S$ for the subset $S$
Output: feature $c_{best}$ to add to the feature subset

for $\forall c_i \in C$ do
  $E_i = \text{TrainClassifier}(X, Y, S \cup c_i)$
end for
if $\exists E_i \in E; E_i + \varepsilon < E_S$ then
  $c_{best} = \arg\min_{c_i}(E)$
else
  $c_{best} = \emptyset$
end if
```

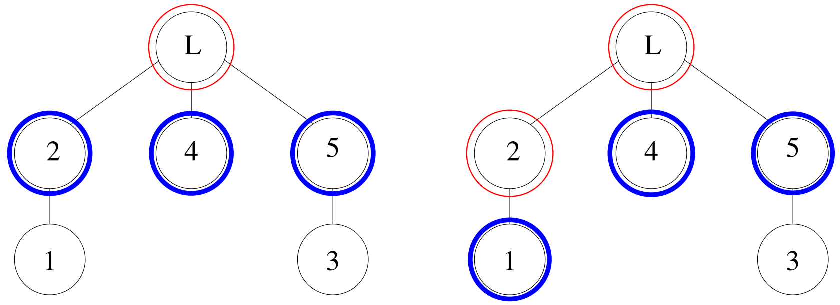

Fig. 2. On the left, the root (representing the label data) is the only member of the node set N, whose children form the candidate set. The SFS step is applied to the candidate set $C_1 = \{f_2, f_4, f_5\}$. The best improvement shall yield the inclusion of feature f_2, which is included in the feature set and is added to N (right). The new candidate set includes all children of f_2 $C_2 = \{f_1, f_4, f_5\}$.

in [7]). 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 is separatly added to the current subset and subsequently evaluated. The feature that induced the best improvement is included in the resulting subset. If the best new subsets improves more than a threshold ε, the new subset is returned, otherwise the algorithm terminates. A SFS step is shown in algorithm 2.

We consider the tree node, that represents the label data Y, as root instance of the dependence tree. All children of this node are treated as set of candidates C for a slightly modified SFS step. After this SFS step the chosen feature is added to the resulting feature subset. Besides this, the corresponding node in the tree is added to the set of nodes N, whose children are considered candidates for the next evaluation step. This is illustrated in figure 2.

Algorithm 3. Feature Selection with CLT(X, Y)

```
Input: data set of observations $X$ and the corresponding labels $Y$
Output: feature subset $S$

$Z \leftarrow X \cup Y$
$T \leftarrow$ Chow-Liu-Tree$(Z)$
$N \leftarrow t_y$ {start with the node corresponding to the label data}
$S \leftarrow \emptyset$ {start with empty feature subset}
repeat
    $C \leftarrow children(N)$ {all children of the current node set are candidates}
    $c \leftarrow$ SequentialForwardSelection$(S, C, X, Y)$
    $N \leftarrow N \cup c_{best} \cup c_{redundant}$ {add the best and possible redundant features to the search path}
    $S \leftarrow S \cup c_{best}$ {add the best node to feature set}
until $c_{best} = \emptyset$ AND $c_{redundant} = \emptyset$
```

The modification of the sequential forward selection is formed by the marking of features that do not improve the classification performance, since these nodes are either irrelevant or redundant features. The SFS step does not only return c_{best}, but any feature, whose inclusion did not improve the approximation performance more than threshold ε. In further candidate sets they are excluded, but added to the set of nodes N, so that their children in the tree are considered canidates in the next evaluation step.

The overall method is detailed in algorithm 3.

4 Discussion

In this section we will discuss the inherent advantages of using the computed tree structure to guide the search process. The Chow-Liu tree is constructed as maximum-weight spanning tree over the pairwise mutual information values of each feature and the labels. Let us assume, all features are statistically independent of each other, so there is no redundancy. A subset of these features contains information about the class label, so the mutual information between these ones and the label data is discriminatingly higher, than between any other pair. During the construction of the dependency tree, these features are connected to the label node, since this maximizes equation 1. All meaningful features are children of the root node. The labels that are irrelevant, are connected to any node, including the root node, with equal probability. Therefore using the CLT approach as filter method, stopping at this point and using only the children of the root node as features is not a good idea, because irrelevant inputs are possibly part of the children set.

Now consider adding redundant features f_1 and f_2 to the mix. The mutual information between one feature and the labels is the same as the joint mutual information between both features and the target value:

$$I(f_1, y) \approx I(f_1 \cup f_2, y) \tag{3}$$

They can be characterized by stating, that the mutual information between these feature f_1 and f_2 is greater than the mutual information between the features and the labels:

$$I(f_1, f_2) > \max(I(f_1, y), I(f_2, y)) \tag{4}$$

For this constellation of three nodes, the algorithm for constructing the maximum-weight spanning tree will always include the connection between the two features, since this maximizes the sum over the weights. Due to the tree structure, the root representing the labels can be connected to one of them only. This a plus in system identification tasks, because from the root's perspective, it is connected to the most informative feature and any features redundant to it are located in the same branch of the tree.

Any features that fulfill the condition of equation 4, but violate the redundancy condition 3, will be added to the same branch, even if they sustain new non-redundant information about the labels. Therefore the tree has to be searched down the branches. The search path has to include redundant features, because the informative feature is possibly connected to one.

From the sequential forward selection algorithm the CLT methods inherits the inability to detect any features that are useful only in combination with another (like the XOR problem [2]). The use of strategies to avoid this problem like sequential forward / backward floating selection (SFFS/SFBS) [11] are not very effective, because of the limited size of the candidate set. As a middle course we suggest using the proposed CLT method to construct a feature subset, that is used as starting point of a SBFS search afterwards. Preliminary results show that in this case the SBFS algorithm only performs very few steps, before it terminates as well.

Degenerated Trees

In the worst case, the tree is degenerated in such a way, that all nodes representing the input data are connected to the root node. All nodes contain information about the target and there is no significant dependency between the input channels. In this case the proposed method is reduced to the basic sequential forward selection method with the additional costs for tree construction, but such ill-conditioned input data indicates a problem that couldn't be solved by the means of feature selection methods. The maximum number of adaption cycles AC for the SFS-like subset search is given by

$$AC_{max} = \sum_{i=0}^{n_{sub}} \left(n_{all} - i \right), n_{all} >= n_{sub}. \tag{5}$$

n_{all} is the number of all available features and n_{sub} is the number of features chosen for the final subset.

The other extreme case is a tree that has no splitting nodes, all features are lined up on a single path from the root to the only leaf. In terms of evaluation steps, this is optimal, since at each step the candidate set contains a single node only. So the minimum of adaption cycles is $AC_{min} = n_{all}$.

Both discussed cases are not common for real-world data and will occur in artificial datasets only. Typically, the obtained tree structures are somewhere in between the described extrema. The exact value depends on the underlying tree structure and the data interrelationship and is difficult to estimate. The average number of children per non-leaf node for the Spambase data set from UCI Machine Learning Repository [13] with 57 features is 2.48 with a variance of 5.72. For a number of different data sets ranging from 100 to 1000 features the average children per node is between 1.61 and 2.63, while the variance increased proportional to the amount of features. Hence for the average case, induced by the tree structure we conjecture an additional logarithmic dependency between the features and the number of adaption cycles, compared to AC_{min}.

5 Related Work and Experiments

The application of information theoretic measures like mutual information for feature selection was suggested before in several publications. The construction of classification and regression trees using the *Information Gain* criterion and their application in form of Random Forests [6] as Embedded method is an example.

A very similar idea to the Chow-Liu tree approach is the MIFS algorithm [12]. The MIFS criterion approximates the joint mutual information, between the features and the label data, which is difficult to estimate in high dimensional spaces. The method is a filter approach that evaluates the features with respect to the labels by computing the mutual information between those. Additionally the mutual information between the candidate and the previously chosen features in the subset is taken into account. The feature that maximizes the following term is added to the subset of features

$$\arg\max_{f_i}(I(f_i, y) - \beta \sum_{f_s \in S} I(f_i, f_s)). \tag{6}$$

The parameter β is used to balance the goals of maximizing relevance to the label and minimize the redundancy in the subset. Like the CLT method MIFS uses the pairwise relations of the features and the label data. The main difference is, that MIFS is used as data driven filter method, while the CLT approach is a wrapper using a black box classifier.

For a number of examples from the UCI repository [13] we compared the performance for the CLT, SFS and MIFS algorithms. As classifier we used a standard MLP with two hidden layers with 20 and 10 neurons respectively. After applying the feature extraction methods, the network was adapted using the selected features only. The balanced error rate

$$BER = \frac{1}{2}\left(\frac{false\ neg}{false\ neg + true\ pos} + \frac{false\ pos}{false\ pos + true\ neg}\right) \tag{7}$$

for the classification problems was calculated using 10-fold cross-validation. This measure accounts for any unbalanced class distributions. For comparison we adapted a network with all available features, too.

The stopping criteria for SFS and CLT were identical, see section 3. For the MIFS algorithm we introduced an additional random channel, independent from the labels and the rest of the features. The feature selection was stopped when the algorithm attempted to add this probe to the subset. In our tests we used $\beta = 0.15$.

The results for the experiments are shown in Table 1.

The MIFS algorithm shows a mixed performance. Given that it is a filter approach, MIFS does not have the advantage of using the classifier itself. Thus the information theoretic approach yields features, that are not optimal in every case for the training of the MLP. This seems to be the case for some examples (see the results for the Ionosphere data set), but not all data sets. The number

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 evaluation steps are shown in parentheses.

Data set	Features	Samples	All	CLT	SFS	MIFS
	f	n	balanced error rate(features/evaluation steps)			
Ionosphere	34	351	20.08(34/-)	18.12(6/38)	18.47(3/130)	24.54(5/-)
Spambase	57	4601	13.81(57/-)	17.26(9/97)	17.39(8/477)	16.29(18/-)
GermanCredit	24	1000	41.70(24/-)	38.52(3/24)	39.06(4/110)	37.47(6/-)
Breast Cancer	30	569	13.78(30/-)	9.37(8/37)	13.44(4/140)	12.48(5/-)

of chosen features is higher, than for both wrapper approaches. Given its nature as filter approach, MIFS is the fastest algorithm considered.

CLT tends to produce smaller error rate results compared to the SFS algorithm, while the size of the feature set chosen by CLT is slightly higher. This observation seems to be somewhat counterintuitive, although both approaches act greedy when choosing the next feature, the difference is, that SFS does its selection on the global level, while CLT choses on a local level (the current candidate set). This can help avoiding local minima in the search, but comes at the cost of adding more features to the subset. The real advantage becomes clear if the number of evaluation steps is compared. The CLT algorithm performs only fractions of adaption cycles needed by the SFS method (given by equation 5).

6 Conclusion

We proposed a search strategy based on Chow-Liu trees for feature selection methods. The tree structure is utilized in a forward search strategy by restricting the candidate sets to the children of certain nodes in the tree. This way, some advantages of the information theoretic approach used to construct the tree are exploited.

This results in a significant reduction of performed evaluation steps in a wrapper feature selection strategy compared to standard methods like sequential forward selection, while retaining a similar performance error. Compared to the MIFS approach, a filter method using the mutual information in a similar way, the results for CLT based feature selection are slightly better, but in terms of computational costs the MIFS algorithm is cheaper. Within the domain of wrapper approaches the speed of the CLT based feature selection method is significant.

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. Lee, D.D., Seung, H.S.: Algorithms for Non-negative Matrix Factorization. Advances in Neural Information Processing Systems, vol. 13, pp. 556–562. MIT Press, Cambridge, MA (2001)
4. LeCun, Y., Denker, J., Solla, S., Howard, R.E., Jackel, L.D.: Optimal Brain Damage. Advances in Neural Information Processing Systems, vol. 2. Morgan Kaufmann, San Francisco (1990)
5. Neal, R.M.: Bayesian Learning for Neural Networks. Springer, Heidelberg (1996)
6. Breiman, L.: Random Forests. Machine Learning 45, 5–32 (2001)
7. Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L.: Feature Extraction: Foundations and Applications. Studies in fuzziness and soft computing, vol. 207. Springer, Heidelberg (2006)
8. Chow, C.K., Liu, C.N.: Approximating Discrete Probability Distributions with Dependence Trees. IEEE Transactions on Information Theory 14, 462–467 (1968)
9. Scott, D.W.: Multivariate density estimation: theory, practice, and visualization. John Wiley & Sons, New York (1992)
10. Cormen, T.H., Leierson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, 2nd edn. MIT Press, Cambridge, MA (2001)
11. Reunanen, J.: Search Strategies. In: Feature Extraction: Foundations and Applications. Studies in fuzziness and soft computing, ch. 4, vol. 207, Springer, Heidelberg (2006)
12. Battiti, R.: Using mutual information for selecting features in supervised neural net learning. IEEE Transactions on Neural Networks 5(4), 537–550 (1994)
13. 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`