Using Feature-Based Models with Complexity Penalization for Selecting Features

Amir Jalalirad¹ · Tjalling Tjalkens¹

Received: 22 December 2015 / Revised: 20 April 2016 / Accepted: 13 June 2016 / Published online: 30 June 2016
© The Author(s) 2016. This article is published with open access at Springerlink.com

Abstract Feature selection and inference through modeling are combined into one method based on a network that can be used to point out irrelevant, redundant and dependent features in the data. It is shown that this network method is efficient in terms of reducing the number of calculations for estimating the probabilities under different model assumptions by breaking the data into fractions. We prove that the probability estimations within the network method lead to the detection of non-informative features with probability one if the data is sufficiently large. The proposed method's accuracy in detecting complex relations between features, selecting informative features and classifying datasets with different dimensions is assessed through experiments using both synthetic and real data. The results from the network method compare favorably with those from the well-known and powerful feature selection algorithms. It is further shown that the network method can handle complex relations between the features that are intractable for other algorithms.

Keywords Model selection · Feature selection · Data classification · Sequence probability estimation

1 Introduction

In a broad sense, classification algorithms can be divided into two groups of transduction and induction methods. In transduction methods, parts of the input data are grouped into a number of classes such that the classification error is minimized on that specific set of input data. These methods avoid cutting the data for training and testing and the inference is directly made without searching for a rule that separates the input labels [13]. Also, these methods consider the labeled and unlabeled data in one run which will to some extent prevent misclassification of the data points near the class boundaries. However, these methods do not suggest a structure over the data that can be used for generalization and the entire transduction algorithm must run again if a new data point is added. On the other hand induction methods, which are the focus of this research, search for an inference rule that governs the relation between the input data and the labels. This rule remains the same for new data points and there will be no need to run the algorithm again. The inferred rule not only helps with classification of other similar data, but also provides insight over the structure of the data as extra information.

Induction methods may approximate functions that maximize data-fitting criteria such as support vector machines, neural networks, Bayesian networks, etc. [2, 4] or may make a combination of the models that are relevant in modeling the data by weighting and averaging such as Bayesian mixture methods, ensemble methods, etc. [9, 11, 22] One of the simplest methods of induction inference is fitting the Naïve Bayes model to the data. This model assumes that all the features of the input are conditionally independent given the class label. Although the Naïve Bayes Classifier (NBC) which is based on this assumption produces satisfactory

✉ Amir Jalalirad
a.jalalirad@tue.nl
Tjalling Tjalkens
t.j.tjalkens@tue.nl

¹ Eindhoven University of Technology, Eindhoven,
The Netherlands

results in some applications [18, 19], it is shown that considering the possibility of dependencies between the input features boosts the classification accuracy considerably. We propose a method based on [12] that selects the most informative subgroup of features, in addition to detecting all the dependent features in the data by comparing every possible feature-based model.

Contrary to the intuition that more features correspond to more information and thus accomplish a more accurate classification task, feature selection leads to improvement in performance of learning algorithms. The reason lies in the fact that as the number of features increases, the algorithms need more training data to be able to infer a rule over the relation between all those features and the class label without over-fitting. Moreover, features that contain no information about the class label can contribute to misclassification and slow down the learning and decision process. Therefore, developing methods of selecting a subgroup of features that classify the data more accurately than the whole feature set, is an effective field of research.

Feature selection algorithms are numerous and diverse in effectiveness. Here we discuss a few representative ones and compare their results with our proposed algorithm later. Algorithms such as Ranking filter [5], Relief [15] and ReliefF [21] effectively reduce the input data dimensionality but are susceptible to repetitive features since these features change the problem space and reduce general separability. Other algorithms such as Minimal Redundancy Maximal Relevance (MRMR) [20] and Mutual Information based Feature Selection (MIFS) [1] are robust against duplicate features but may remove features that are not informative individually but are informative in combination with other features. Suppose the case where the class label is the exclusive OR of two features. Each of those two features may seem non-informative individually, but they should be kept in the selected set of features together. Algorithms like Conditional Mutual Information Maximization (CMIM) [8], Interaction Gain for Feature Selection (IGFS) [6] and Conditional Mutual Information-based Feature Selection (CMIFS) [3] deal with the aforementioned problems accordingly. However, in these algorithms the feature selection measure is based on mutual information. When the features are categorical or discretized, mutual information distance measure causes the algorithms to favor the features that have more value levels [10, 23]. Correlation based Feature Selection (CFS) [10] and Fast Correlation Based Filter (FCBF) [23] are two of the algorithms that deal with this problem by using symmetrical uncertainty, a normalized variant of mutual information, as their distance measure. The former uses forward selection to search through the feature set while the latter uses backward elimination. In forward selection, the algorithm starts from an empty set of candidate features and continues selecting

informative features to add to the set. In backward elimination, the algorithm starts with the full set of all the features and continues removing non-informative features from that set. In [16], the authors show that when the distance measure is cross-entropy or Kullback-Leibler divergence, backward elimination outperforms forward selection.

In this paper, we propose a feature selection method based on penalizing the probability of models proportional to their complexity. This method removes non-informative as well as repeated features and does not select individual features over more informative groups of features. Using this method, both redundant (features with information already contained in other features) and irrelevant (features that do not contain information about the class label) features will be removed while features with more values will not be favored since they contribute to make a more complex model. To evaluate the performance of the proposed algorithm, we run experiments on synthetic and real datasets and compare the results with those of ReliefF, MRMR, CFS and FCBF.

The remainder of the paper is organized as follows. In Section 2, we explain how we estimate the probability of a sequence of data and how this probability estimation method is utilized to model the data and select features. In Section 3, an algorithm introduced in a previous work is further developed to suit feature selection purposes. Extensive experiments are conducted on synthetic and real data to evaluate the proposed algorithm and compare it to some of the well-known feature selection algorithms in Section 6. Finally, conclusions are drawn in Section 7.

2 Notations, Definitions and Probability Estimations

Throughout this paper, vectors are denoted by bold letters. Subscripts denote the index of elements of a vector and superscripts refer to the length of sequences. A superscript in parentheses refers to the indexing in a sequence. For instance, $F_i^{(j)}$ refers to the i^{th} feature from the j^{th} instance of data. If feature i and j are dependent, they are denoted by the super-feature $F_{i,j}$. We assume that the dataset consists of independent identically distributed instances or objects. Each object includes a class label and a vector of features, $\mathbf{O} = (C, \mathbf{F})$. For a sequence of length N of these objects,

$$\begin{aligned} P(\mathbf{O}^N) &= P(C^N, \mathbf{F}^N) = \prod_{i=1}^N P(\mathbf{O}^{(i)}) \\ &= \prod_{i=1}^N P(C^{(i)}) P(\mathbf{F}^{(i)} | C^{(i)}) = P(C^N) P(\mathbf{F}^N | C^N). \end{aligned} \quad (1)$$

Since the parameters that define occurrences of each of the symbols are unknown, we use Dirichlet probability estimation method with parameters set to $\frac{1}{2}$ to estimate the probability of the data sequence. Our choice of the estimation method will be justified later when we use some of its specific properties. The Dirichlet estimated probability for a sequence X^N with alphabet $\mathcal{X} = \{x_1, x_2, \dots, x_s\}$ is

$$P_E(X^N) = \frac{\Gamma(\frac{s}{2})}{\pi^{\frac{s}{2}}} \cdot \frac{\prod_{i=1}^s \Gamma(n_i + \frac{1}{2})}{\Gamma(N + \frac{s}{2})}, \quad (2)$$

where n_i denotes the number of times $X = x_i$ in X^N and the subscript E stands for estimation. Now it is easy to estimate $P(C^N)$ in Eq. 1. Hence, estimating the probability of the data-set $\mathcal{O}^N$, narrows down to estimating $P(F^N|C^N)$ in Eq. 1. For different dependent features in the feature vector of the data, our estimation $P_E(F^N|C^N)$ will be different. Similar to [12], we make this point clear through an example.

Example 1 Consider a set of data with two binary features F_1 and F_2 . Table 1 shows an example of ten instances of this data-set. If the two features are independent, the probability of this example sequence is estimated according to Eq. 2 as follows.

$$\begin{aligned} P_E(F^{10}) &= P_E(F_1^{10}) \cdot P_E(F_2^{10}) \\ &= \frac{\Gamma(4 + \frac{1}{2})\Gamma(6 + \frac{1}{2})}{\pi\Gamma(10+1)} \cdot \frac{\Gamma(8 + \frac{1}{2})\Gamma(2 + \frac{1}{2})}{\pi\Gamma(10+1)} = 4.81 \times 10^{-7}. \end{aligned}$$

If the two features are dependent, then the probability of the feature vector is estimated as the probability of a super-feature with alphabet of size four.

$$\begin{aligned} P_E(F^{10}) &= P_E(F_{1,2}^{10}) \\ &= \frac{\Gamma(3 + \frac{1}{2})\Gamma(1 + \frac{1}{2})\Gamma(5 + \frac{1}{2})\Gamma(1 + \frac{1}{2})}{\pi^2\Gamma(10+2)} = 3.47 \times 10^{-7}. \end{aligned}$$

Example above shows that using the Dirichlet estimation method, the probability of a sequence of feature vectors $P(F^N)$ can be estimated in more than one way if the feature

dependencies are unknown. Consider two non-intersecting subgroups from the feature vector index set g_1 and g_2 . Let $P_E(F_{g_1}^N)$, $P_E(F_{g_2}^N)$ and $P_E(F_{g_1, g_2}^N)$ be the estimated joint sequence probability of the features corresponding to g_1 , g_2 and $g_1 \cup g_2$ respectively. For estimated probabilities using Dirichlet method with parameters set to $\frac{1}{2}$, it is proved in [12] for sufficiently large N that with probability one,

$$P_E(F_{g_1, g_2}^N) < P_E(F_{g_1}^N) \cdot P_E(F_{g_2}^N), \quad (3)$$

when the two groups are independent, and

$$P_E(F_{g_1, g_2}^N) > P_E(F_{g_1}^N) \cdot P_E(F_{g_2}^N), \quad (4)$$

when the two groups are dependent. The inequality pair of Eqs. 3 and 4 show that Dirichlet probability estimation method can be used to find the dependencies between features of the data. Following that, it will be shown that this estimation method can also detect irrelevant and redundant features in a similar manner.

Irrelevant Feature An irrelevant feature is a member of the feature vector that is independent of the class label. If feature F_1 is irrelevant,

$$P(F_1^N, C^N) = P(F_1^N) P(C^N).$$

Therefore, according to the inequality pair of Eqs. 3 and 4, for a group of irrelevant features with index set g ,

$$P_E(F_g^N, C^N) < P_E(F_g^N) P_E(C^N). \quad (5)$$

Redundant Feature A redundant feature is a member of the feature vector that is conditionally independent of the class label given another feature. The first feature is redundant given the second feature if

$$P(F_1^N, C^N|F_2^N) = P(F_1^N|F_2^N) P(C^N|F_2^N).$$

Therefore, according to the inequality pair of Eqs. 3 and 4, if a group of features with index set g_1 is redundant given another group of features with index set g_2 where $g_1 \cap g_2 = \emptyset$, then

$$P_E(F_{g_1}^N, C^N|F_{g_2}^N) < P_E(F_{g_1}^N|F_{g_2}^N) P_E(C^N|F_{g_2}^N) \quad (6)$$

Another form of inequality (6) is achieved by multiplying

$$\text{both sides by } \frac{P_E(F_{g_2}^N|C^N)}{P_E(C^N|F_{g_2}^N)}.$$

$$P_E(F_{g_1, g_2}^N|C^N) < P_E(F_{g_1}^N|F_{g_2}^N) P_E(F_{g_2}^N|C^N) \quad (7)$$

For a data-set containing feature vectors of fixed length, if comparisons similar to Eqs. 5 and 6 are made for every subset of the feature set, groups of irrelevant and redundant features will be detected. To make the comparisons in a uniform way, we introduce the concept of a model.

Table 1 An example of ten objects consisting of two binary features.

F_1	F_2
1	0
1	0
1	1
1	0
1	0
0	0
0	0
0	0
0	1
1	0

Model A model is defined as a group of non-empty subsets of the feature vector index set. Each model provides information over the feature vector. For example if we have a feature vector $\mathbf{F} = (F_1, \dots, F_5)$, the model $M = (\{1\}, \{2, 4\})$ claims that the third and the fifth features must be ignored, the second and the fourth features are dependent while being independent of the first feature. For this model, the estimated probability of the whole data-set is

$$\begin{aligned} P_E(\mathbf{O}^N | M) &= P_E(\mathbf{F}^N | C^N, M) P_E(C^N) \\ &= P_E(F_1^N | C^N) P_E(F_3^N) \\ &\quad \times P_E(F_{2,4}^N | C^N) P_E(F_5^N) P_E(C^N), \end{aligned} \quad (8)$$

where the conditioning of the sequence of class labels over the model is ignored because the model does not effect the class label sequence estimation. Now the problem of finding dependent features and removing the irrelevant and redundant features simplifies to finding the model that results in the largest estimated probability. However, the number of models that can be defined over a feature vector of fixed length is so large that makes one by one comparison of models unfeasible. In the next section, we introduce an efficient graph-based approach to compare models.

3 Network Method

The method introduced in [11] and further developed in [12] makes the basis of the network method that we describe here. The network for feature vectors of length three is shown in Fig. 1. Each node of the network corresponds to a part of the feature vector indicated in the node index. There are two sequence probability estimations performed per node in the network. In one, we estimate the probability of the sequence of the features corresponding to the node given the class label sequence. In the other one, the same probability is estimated without the conditioning over the class labels. The computations in the network are performed from the top row to the bottom row of nodes. The results of estimations and comparisons in the upper nodes are transferred to the lower nodes through branches and are used in the new computations and comparisons. As an example, the probability estimations performed in N_3 will be used

Figure 1 The network graph for feature vectors of length three.

in the lower nodes to calculate $P_E(\mathbf{F}^N | C^N, M)$ where M could be $(\{1, 2\}, \{3\})$ or $(\{1\}, \{3\})$ or any other model that contains the subset $\{3\}$. In the node N_i of the first row, the probabilities $P_E(F_i^N)$ and $P_E(F_i^N | C^N)$ are calculated and compared and the maximum is selected. The result of the comparison determines whether the feature F_i is irrelevant according to inequality (5). Similar comparisons are made in all the nodes of the first row and the values are saved to be used in the lower nodes. In the nodes of the rows below the first, some more comparisons are needed. For instance, in the node $N_{i,j}$ of the second row, $P_E(F_{i,j}^N)$ and $P_E(F_{i,j}^N | C^N)$ are estimated. Then the following maximization is made.

$$\begin{aligned} P_{\max}(N_{i,j}) &= \max \left\{ P_E(F_{i,j}^N), P_E(F_{i,j}^N | C^N), \right. \\ &\quad \left. \frac{P_E(F_i^N | C^N) P_E(F_{i,j}^N)}{P_E(F_i^N)}, \frac{P_E(F_j^N | C^N) P_E(F_{i,j}^N)}{P_E(F_j^N)}, \right. \\ &\quad \left. P_{\max}(N_i) P_{\max}(N_j) \right\}. \end{aligned} \quad (9)$$

In Eq. 9, the comparison between the first two terms determines whether the group of features i, j is relevant according to inequality (5). The comparison between the second and the third term determines whether feature j is redundant given feature i according to the inequality (7). Similarly, the comparison between the second and the fourth term determines whether feature i is redundant given feature j . The comparison with the last term is to decide whether the two features are independent. Note that only the first two terms of Eq. 9 are calculated in the current node $N_{i,j}$ and the rest of the terms are calculated using the first term and the estimations that are already done in N_i and N_j . This way, the network avoids re-estimating the sequence probability of the parts of the feature vector that are the same in different models.

The same kind of comparisons and inference on relevance, redundancy and dependency of features will be carried out in the nodes of the lower rows of the graph. It is clear that the whole graph of the network is created by repetition of the base pattern shown in Fig. 2. The same computations and comparisons as Eq. 9 are performed per each base pattern except that i and j are replaced by groups of features g_1 and g_2 . Therefore in each node, decisions are made on the relevance, redundancy and dependencies of different parts of the feature vector. Finally in the only node

Figure 2 The base pattern upon which the network method graph is build.

of the last row, the considered part is as large as the feature vector itself.

By making comparisons like (9) for all pairs of branches going to the last node, it becomes clear either that all the features are relevant/irrelevant or the feature vector should be divided at a specific place. The best place to divide the feature vector is determined by the maximum of comparisons between all pairs of branches. Then, we follow the pair of branches that made the maximum of the comparisons from the bottom node up to the two nodes above according to the pattern in Fig. 2. Again, we look at the comparison results of these new nodes. The comparison results of each of these new nodes determine whether all the features in the node are relevant/irrelevant or the feature subset corresponding to the node should be divided again. This procedure continues until all the relevant feature subsets are detected.

4 Algorithmic Representation of the Network Method

Algorithm 1 shows a summary of the steps taken in the network method in order to produce the highest data sequence probability. If the explicit information about dependency, relevance, and redundancy of the features given the selected model is required, then the winning index in the comparisons of line 7, 13 and 15 must be recorded in Algorithm 1. In this way, each node contains a pointer that shows if the maximum probability for that node was produced by the conditional probability, unconditional probability or some multiplication of probabilities from incoming branches to that node. One gets to the model selected by the network by starting from the last node and following the direction of the pointers. If the conditional probability in a node is the maximum probability, then the group of features corresponding to that node are included in the selected model by the network. If the unconditional probability in a node is the maximum probability selected, then the features corresponding to that node are irrelevant to the class and are removed from the final model of the network. If the maximum probability in a node is produced by the multiplication of some probabilities coming from one of the input branch pairs to a node, there will be three cases:

Case 1. The pointer from line 13 of the Algorithm 1 shows that the winning probability of the branch is from the case where the feature group corresponding to the left branch is redundant given the feature group corresponding to the right branch. In this case, we continue to look for the maximum probability of the nodes following only the right branch.

Case 2. The pointer from line 13 of the Algorithm 1 shows that the winning probability of the branch is from the case

where the feature group corresponding to the right branch is redundant given the feature group corresponding to the left branch. In this case, we continue to look for the maximum probability of the nodes following only the left branch.

Case 3. The pointer from line 13 of the Algorithm 1 shows that the winning probability of the branch is from the case where maximum probability of the left and right branch are multiplied. In this case, we continue to look for the maximum probability of the nodes following the left and the right branch.

This procedure continues until there is no pointer to follow. At this point, a decision is made for every single feature in the feature vector whether to be included in the final model or not.

Algorithm 1 The Network's Algorithm

Input: $\mathcal{D}$ Table of input data
Output: P_{best} Data sequence probability given the best model

- 1: k = number of features extracted from $\mathcal{D}$
- 2: N = an array of 2^k nodes
 (Network nodes are arranged from top to bottom row)
- 3: **for** $i = 1 : |N|$ **do**
- 4: $N[i].UP$ = unconditional probability of features relating to $N[i]$
- 5: $N[i].CP$ = conditional probability of features relating to $N[i]$
- 6: **if** $i \leq k$ **then**
 (Nodes in the first row of the network)
- 7: $N[i].PM = \max\{N[i].UP, N[i].CP\}$
- 8: **else**
 (Rest of the nodes - branch probabilities must be compared)
- 9: $BranchProb = 0$
- 10: **for** $j = 1 : \text{number of input branch pairs to } N[i]$ **do**
- 11: NLB = The node connected to $N[i]$ by the j^{th} left branch
- 12: NRB = The node connected to $N[i]$ by the j^{th} right branch
- 13:
$$BranchProb = \max\left\{\frac{NLB.CP}{NLB.UP}N[i].UP, \frac{NRB.CP}{NRB.UP}N[i].UP, \right.$$

$$\left. NLB.PM \cdot NRB.PM, BranchProb\right\}$$
- 14: **end for**
- 15: $N[i].PM = \max\{N[i].UP, N[i].CP, BranchProb\}$
- 16: **end if**
- 17: **end for**
- 18: $P_{best} = N[2^k].PM$
 (Bottom-most node of the network)

5 Saving Further on Computations

Even though the network method saves on computations by estimating the probability of each part of the feature vector index set only once, the computation time may still be considerably long since the number of possible models can be extremely large for large feature vector sizes. However, when the features follow some order, a simplified version of the network method can be used. The graph of this network is shown in Fig. 3. In this simplified version, the assumption is that only features that are adjacent in the feature vector

Figure 3 Network for ordered feature vectors; the simplified version of the network for feature vectors of length three.

can be dependent and also features may be redundant given only adjacent features. In other words, models considered in the network for the ordered feature vectors cannot have a dependency group like $\{1, 3\}$, because F_1 and F_3 are not adjacent. Also for the same reason, redundancy of F_1 given F_3 and vice versa is not considered. As illustrated in Fig. 3, the network for ordered feature vectors has fewer nodes and consequently includes fewer calculations.

The algorithm of the network method for the ordered feature vectors is the same as Algorithm 1 except that in line 2, an array of $\frac{k(k+1)}{2}$ nodes is made and in line 18, the best probability is $N[\frac{k(k+1)}{2}].PM$. The exhaustive enumeration of all the possible models made with k ordered features and their comparison require $\mathcal{O}(k(2.62)^k)$ multiplications and $\mathcal{O}(2.62^k)$ comparisons. The network method reduces both of these orders to $\mathcal{O}(k^3)$. Table 2 shows the effect of such reduction for a few sample sizes of feature vectors.

6 Experiments and Results

To evaluate the feature selection and classification capabilities of the network method, extensive experiments are conducted using both synthetic and real data-sets and the

results of the algorithm are compared to those of well-received and prominent algorithms, namely FCBF, CFS, MRMR and ReliefF. In all the experiments, the number of neighbors and the number of samples for the ReliefF algorithm are set to 5 and 30 respectively as suggested by [21] and [23]. In these experiments, the goal is to investigate how efficient the network method is in removing irrelevant and redundant features. Also, we seek to reveal the competence of the network method in detecting complicated relations between the features. Finally, we aim to check whether it is reasonable to assume feature-base models for real world data-sets and search for the model with the best group of feature subsets.

6.1 Experiments on Synthetic Data-Sets

A widely-used synthetic data-set which is particularly challenging for feature selection algorithms is the CorrAI data-set [14]. This data-set consists of a binary class label and six binary features. The class label of each data instance is produced from its first four feature values according to the following relation,

$$C = (F_1 \wedge F_2) \vee (F_3 \wedge F_4). \quad (10)$$

The fifth feature is uniformly random and therefore irrelevant. The sixth feature matches the class label 75 % of the time and therefore is redundant given the first four features. Thus, the desired feature selection algorithm removes the fifth and the sixth feature and keeps the first four features. We run the network method along with the other feature selection algorithms on different sizes of CorrAI data-set. Since MRMR and ReliefF are feature ranking algorithms, the four highest ranked features are regarded as the selected features of these algorithms. The selected features are used as input to a classifier that classifies the new object $\mathbf{O}_{new} = (C_{new}, \mathbf{F}_{new})$ according to the following relation,

$$C_{new} = \arg \max_c \left\{ P_E(C_{new} = c | \mathbf{O}^N) P_E(\mathbf{F}_{new} | \mathbf{O}^N, C_{new} = c) \right\}. \quad (11)$$

Table 2 Required workload comparison between the network method and exhaustive enumeration of models.

	Multiplications for exhaustive enumeration	Comparisons for exhaustive enumeration	Multiplications or comparisons for the network method
$k = 5$	122	88	100
$k = 10$	39360	10945	825
$k = 20$	1.34×10^9	1.65×10^8	6650
$k = 50$	1.23×10^{22}	5.73×10^{20}	104125
$k = 100$	1.99×10^{43}	4.54×10^{41}	833250
$k = 500$	1.57×10^{211}	7.03×10^{208}	1.04×10^8

Figure 4 shows the accuracy of the algorithms as a function of the number of training objects per class. Each point in the plot represents averaged result of 100 runs of each algorithm. The test data includes 50 objects per class. When at least 20 objects per class are observed in the training, only the ReliefF algorithm has a higher classification accuracy than the network method. MRMR cannot catch the relevant features that are important through the complex relation of Eq. 10, and CFS and FCBF are not able to remove the

Figure 4 Classification accuracy comparison between Network, FCBF, CFS, ReliefF and MRMR as a function of training size for CorrAl data-set.

sixth feature (redundant feature). Both the ReliefF and the network algorithms on the other hand, select the first four features and remove the other two almost all the time when the training data is sufficiently large. However, the classification rate of the ReliefF algorithm is not as high when the features are duplicated in the data.

We repeat the experiment with a slightly different data-set. This time, four redundant features are added to the CorrAl data that are duplicates of the first four features. Hence, a desired feature selection algorithm still selects four features, one from each of these four groups: $\{F_1; F_7\}$, $\{F_2; F_8\}$, $\{F_3; F_9\}$ and $\{F_4; F_{10}\}$. Figure 5 shows the classification accuracies of the algorithms for the new data-set as a function of the number of training objects per class. Clearly, the performance of the ReliefF algorithm has degraded since in most of the runs, it is unable to select four features that determine the class label. On the contrary, the classification rate of the network method remains high and still only four important features are selected. No major

Figure 5 Classification accuracy comparison between Network, FCBF, CFS, ReliefF and MRMR as a function of training size for a modified version of CorrAl data-set that has ten features; the four important features of the original CorrAl data-set are duplicated.

change is observed in the performance of the three other algorithms. It is also worth noting that the network algorithm, not only selects the four relevant features, but also as extra information declares that these four features are dependent features.

Another family of data-sets that feature selection algorithms can hardly handle, is based on sum by modulo p problems. For each data-set based on a modulo- p - I problem, the class label is determined by the following relation between the features,

$$C = \left(\sum_{i=1}^I F_i \right) \mod p. \quad (12)$$

The features are integers that can have values in the range $[0, p - 1]$. According to Eq. 12, a modulo- p - I data-set has p different class labels. It is easy to see that the binary exclusive OR problem is the special case of modulo- p - I problems when p and I are set to two.

To evaluate and compare the feature selection algorithms on this kind of problems, we generate data-sets with nine uniformly distributed random features where the class label is generated according to Eq. 12 from the first I features. Different data-sets are created with symbol alphabet size p , ranging from two to six and relevant features I , ranging from two to eight. The five algorithms are run 50 times on each data-set. The experiments show that MRMR, CFS, and FCBF are never able to detect the important features in such problems and therefore a classification method based on the features selected by these algorithms classifies the data randomly. Both ReliefF and network algorithms are able to detect relevant features in this kind of data-sets, however the network method requires less training data and achieves successful results on a wider range of values of p and I . Figure 6 shows the rate of selecting the exact set of relevant

Figure 6 Rate of selecting the relevant features in modulo- p - I data-sets as a function of required objects per class for ReliefF and network algorithms.

Table 3 The number of objects per class required by the network and the ReliefF algorithms to select the exact relevant features 90 % of the time for modulo- p - I data-sets with nine features.

	$I = 2$		$I = 3$		$I = 4$		$I = 5$		$I = 6$		$I = 7$		$I = 8$	
$p = 2$	5	16	8	38	13	71	19	152	24	185	41	212	46	266
$p = 3$	6	16	11	86	25	300	50	995	150	> 1000	336	> 1000	342	> 1000
$p = 4$	8	18	19	115	64	462	195	> 1000	643	> 1000	> 1000	> 1000	> 1000	> 1000
$p = 5$	10	18	31	138	139	950	548	> 1000	> 1000	> 1000	> 1000	> 1000	> 1000	> 1000
$p = 6$	12	20	56	154	257	> 1000	> 1000	> 1000	> 1000	> 1000	> 1000	> 1000	> 1000	> 1000

The bold numbers on the left refer to the network method

features for both algorithms in 50 runs, as a function of the number of objects per class for two different values of p and I . Note that in all of these experiments, the first I highest ranked features by the ReliefF algorithm are regarded as selected features of the algorithm. Clearly, the network method needs less training data to detect the relevant set of features. A general evaluation on similar data-sets is shown in Table 3. This table shows the number of objects per class required by ReliefF and network algorithms to select the exact relevant features in at least 90 % of the runs. The number of objects required by the network method are shown in bold. The values in Table 3 are calculated with the assumption that the maximum number of objects available per class is 1000.

According to Table 3, in 20 % (7 out of 35) of the p - I value pairs, the network method detects the relevant features within 1000 objects per class range whereas none of the other algorithms are able to. Furthermore in all these data-sets, the network algorithm also correctly reveals that the relevant features are dependent, while the other algorithms do not provide any information on the dependencies.

Table 4 Description of the data-sets from the UCI Machine Learning Repository.

Data-set name	Features	Instances	Classes
Wilt	5	4839	2
Car Evaluation	6	1728	4
Bank Marketing	16	45211	2
Firm	16	10800	4
Parkinsons	22	195	2
Ionosphere	33	350	2
Splice	60	3175	3
Urban land cover	147	675	9
Isolet	617	7797	26
Multiple Features	649	2000	10

6.2 Experiments on Real Data-Sets

In this part, we compare the five feature selection algorithms using ten benchmark data-sets from the UCI Machine Learning Repository [17]. In selecting data-sets from the repository, our aim was to maintain diversity in the number of features, objects and classes. Table 4 shows an overview of the data-sets used in this experiment. The objects that include missing values and features with alphabet size of one are removed from the data-sets. We apply the MDL discretization method [7] to data-sets with features that have continuous values. The five algorithms run on each of the data-sets and the features selected by them are used as input to a classifier based on Eq. 11. For feature ranking algorithms, ReliefF and MRMR, the number of selected features is set to the value that leads to the highest classification accuracy. Since in some of the data-sets the number of features is large, the simplified version of the network method (network for ordered feature vectors) is used in this experiment.

Table 5 shows the 10-fold cross validation accuracy of classification for each algorithm on each of the data-sets. A two-tailed paired Student's t-test is performed between the classification accuracy results of the network method and each of the other methods to assess the significance of the differences in accuracies. The p -value of these tests are shown on the right side of each cell. The difference in classification accuracy is more significant if the p -value is lower. In the last row of the table, the number of data-sets on which each algorithm wins from, loses to, or makes a tie with the network method (W/L/T) at significance level 0.1 is noted.

The results show that the network method achieves high classification accuracy compared to some of the well-known algorithms. This further reckons that fitting feature-based models to the data is a reasonable practice for many data-sets with a variety in size and application.

Table 5 Classification accuracy over UCI data-sets with 10-fold cross validation using five different feature selection algorithms.

	FCBF	<i>p</i> -val	CFS	<i>p</i> -val	MRMR	<i>p</i> -val	ReliefF	<i>p</i> -val	Network
Wilt	94.61	0.02	94.61	0.02	94.61	0.02	94.61	0.02	95.85
Car Evaluation	85.77	0.00	83.92	0.00	85.93	0.00	85.30	0.00	93.92
Bank Marketing	89.85	0.00	89.85	0.00	90.04	0.09	89.32	0.00	90.29
Firm	77.94	0.00	75.67	0.00	77.94	0.00	77.94	0.00	80.31
Parkinsons	88.13	0.87	89.16	0.56	81.92	0.04	87.03	0.87	87.58
Ionosphere	91.43	0.24	91.71	0.09	89.71	1.00	90.86	0.22	89.71
Splice	96.25	0.00	96.19	0.00	95.09	0.20	95.37	0.09	94.58
Urban land cover	85.66	0.27	87.59	0.53	82.51	0.00	84.89	0.04	87.27
Isolet	89.07	0.00	92.11	0.01	91.21	0.06	89.18	0.00	91.77
Multiple Features	97.80	0.10	98.35	0.00	96.65	0.04	96.85	0.11	97.15
(W/L/T)	1/5/4		4/4/2		0/8/2		1/6/3		–

7 Conclusions

The previously introduced network algorithm is further developed to mark the dependencies between the features of the data and remove irrelevant and redundant features, all in one coherent process of selecting the best feature-based model. We have shown that this algorithm is capable of detecting complicated relations between the features. This also enables the algorithm to remove redundant features given a group of dependent informative features. The algorithm is tested through extensive experiments using synthetic and real data and compared to common competitive feature selection algorithms. The network method achieves satisfactory results that enunciate the practicality of fitting feature-based models to data-sets in order to detect relevance, redundancy and dependency of features.

Acknowledgments This work is partly presented at MLSP 2015.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Battiti, R. (1994). Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, 5(4), 537–550.
- Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.
- Cheng, H., Qin, Z., Feng, C., Wang, Y., & Li, F. (2011). Conditional mutual information-based feature selection analyzing for synergy and redundancy. *ETRI Journal*, 33(2), 210–218.
- Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press.
- Duch, W., Winiarski, T., Biesiada, J., & Kachel, A. (2003). Feature selection and ranking filters. In *International conference on artificial neural networks (ICANN) and International conference on neural information processing (ICONIP)*, Citeseer (Vol. 251, p. 254).
- El Akadi, A., El Ouardighi, A., & Aboutajdine, D. (2008). A powerful feature selection approach based on mutual information. *International Journal of Computer Science and Network Security*, 8(4), 116.
- Fayyad, U. M., & Irani, K. B. (1993). Multi-interval discretization of continuous-valued attributes for classification learning. In *13th International joint conference on artificial intelligence*.
- Fleuret, F. (2004). Fast binary feature selection with conditional mutual information. *The Journal of Machine Learning Research*, 5, 1531–1555.
- Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning* Vol. 1. Berlin: Springer series in statistics Springer.
- Hall, M. A., & Smith, L. A. (1999). Feature selection for machine learning: Comparing a correlation-based filter approach to the wrapper. In *FLAIRS conference* (pp. 235–239).
- Jalalirad, A., & Tjalkens, T. (2015). An efficient method for computing a bayesian mixture of feature-based models. *International Journal of Pattern Recognition and Artificial Intelligence*, 30(1).
- Jalalirad, A., & Tjalkens, T. (2016). A feature-based model selection method based on complexity penalization. *Computational Statistics and Data Analysis* (under review).
- Joachims, T. (1999). Transductive inference for text classification using support vector machines. In *ICML* (vol. 99, pp. 200–209).
- John, G. H., Kohavi, R., Pfleger, K., et al. (1994). Irrelevant features and the subset selection problem. In *Machine learning: proceedings of the eleventh international conference* (pp. 121–129).
- Kira, K. (1992). A practical approach to feature selection. In *Proceedings of international conference on machine learning*.
- Koller, D., & Sahami, M. (1996). *Toward optimal feature selection*. Stanford InfoLab.

17. Lichman, M. (2013). UCI machine learning repository. <http://archive.ics.uci.edu/ml>.
18. Meng, H., Appiah, K., Hunter, A., & Dickinson, P. (2011). FPGA implementation of naive bayes classifier for visual object recognition. In *2011 IEEE Computer society conference on computer vision and pattern recognition workshops (CVPRW)* (pp. 123128). IEEE.
19. Mitchell, T.M. (1997). Machine learning. wcb.
20. Peng, H., Long, F., & Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8), 1226–1238.
21. Robnik-Šikonja, M., & Kononenko, I. (2003). Theoretical and empirical analysis of relief and rrelief. *Machine Learning*, 53(1–2), 23–69.
22. Willems, F. M., Shtarkov, Y. M., & Tjalkens, T. (1995). The context-tree weighting method: basic properties. *IEEE Transactions on Information Theory*, 41(3), 653–664.
23. Yu, L., & Liu, H. (2004). Efficient feature selection via analysis of relevance and redundancy. *The Journal of Machine Learning Research*, 5, 1205–1224.

Amir Jalalirad received his B.S. and M.S. degree in Electrical Engineering from Tehran Polytechnic in 2008 and 2011 respectively.

He is now a PhD candidate in Signal Processing Systems (SPS) group at the Department of Electrical Engineering in Eindhoven University of Technology, the Netherlands. His research interests include machine learning, data mining and artificial intelligence.

Tjalling Tjalkens received the M.S. degree in electrical engineering from the Eindhoven University of Technology, Eindhoven, the Netherlands in 1983 and the Ph.D. degree from the same university in 1987.

From 1983 on he is a scientific staff member at the Eindhoven University of Technology. His research interest is in (universal) source coding theory and model selection problems in probabilistic

machine learning.

In 1989 he was a visiting scientist at the Technical University of Lund, Sweden where he cooperated with Yuri Shtarkov on the analysis of the Lempel-Ziv compression algorithms.

He is a co-recipient, with F.M.J. Willems and Yu.M. Shtarkov, of the 1996 IEEE Information Theory Society Paper Award and is co-inventor of two (PCT) patents on data compression systems.

Dr. Tjalkens has been a consultant of KPN (The Dutch Telephone Company) on general data compression issues.

In 1999 he was a visiting professor at the Technical University of Lund, Sweden.

Dr. Tjalkens is recently involved in several machine learning projects, such as the European EnLight project in which a distributed rule based decision engine for lighting applications is developed and the Dutch HearScan project which involves automatic audio algorithm design.