Evolutionary Feature Subset Selection with Compression–based Entropy Estimation

Pavel Krömer
VŠB - Technical University of Ostrava
17.listopadu 15
708 33 Ostrava-Poruba, Czech Republic
pavel.kromer@vsb.cz

Jan Platoš
VŠB - Technical University of Ostrava
17.listopadu 15
708 33 Ostrava-Poruba, Czech Republic
jan.platos@vsb.cz

ABSTRACT

Modern massive data sets often comprise of millions of records and thousands of features. Their efficient processing by traditional methods represents an increasing challenge. Feature selection methods form a family of traditional instruments for data dimensionality reduction. They aim at selecting subsets of data features so that the loss of information, contained in the full data set, is minimized. Evolutionary feature selection methods have shown good ability to identify feature subsets in very-high-dimensional data sets. Their efficiency depends, among others, on a particular optimization algorithm, feature subset representation, and objective function definition. In this paper, two evolutionary methods for fixed-length subset selection are employed to find feature subsets on the basis of their entropy, estimated by a fast data compression algorithm. The reasonability of the fitness criterion, ability of the investigated methods to find good feature subsets, and the usefulness of selected feature subsets for practical data mining, is evaluated using two well-known data sets and several widely-used classification algorithms.

Keywords

Genetic algorithms; differential evolution; feature subset selection; entropy estimation

1. INTRODUCTION

Mining data and extracting knowledge from high-dimensional data is a challenging problem. Many traditional data mining and machine learning methods struggle with the volume and dimension of data generated nowadays by information and communication technology, industrial applications, and modern cyber-physical systems including sensor and actuator networks, Internet of Things, and e.g. security and surveillance applications. The challenges, faced by traditional data processing methods in very-high-dimensional spaces, are many-faceted. They span from the curse of dimensionality [25] and outlier detection [2] to e.g. noise identification and removal [28]. From a computational point of view, the analysis of high-dimensional data represents for many widely-used data processing methods a performance bottleneck.

Together with dimensionality reduction, clustering, and feature extraction, *feature subset selection* is a traditional data pre-processing technique [19]. It aims at reducing the dimensionality of an investigated problem, represented by a complex data set, by selecting a subset of features (attributes, columns) that represents the original data with high fidelity.

There are several high-level feature subset selection approaches based on statistical [19], geometric, and e.g. information theoretic [4] properties of the underlying data. Nature-inspired algorithms have shown good ability to find feature subsets in the past [9]. They apply different high-level population or trajectory-based metaheuristic strategies to represent and iteratively improve a candidate feature subset or subsets. The feature subsets, explored in course of the metaheuristic search, are ranked using selected evaluation criteria. Evolutionary methods have demonstrated the ability to find good feature subsets with respect to used evaluation criteria [9]. Their accuracy and performance is, however, tightly linked to the feature subset representation and to the fitness function they employ.

In this work, we propose a new feature subset evaluation function and compare the ability of two recent evolutionary methods for fixed-length subset selection to discover high-quality feature subsets when they use a fitness function based on this measure. The proposed evaluation criterion is studied in detail and its usefulness in context of practical data mining is assessed.

An experimental evaluation of the proposed approach is performed on two well-know classification data sets from the UCI machine learning repository [18]. Two different evolutionary methods are employed to find small subsets of all features in these data sets and several traditional machine learning algorithms are applied to classify the data on the basis of selected feature subsets only. The data sets, used in the experimental part of this research, have rather small size. However, their modest dimensions enable a thorough analysis of the solution space that needs to be explored by the evolutionary methods. The fitness function is computed for all possible feature subsets of given size and its relationship to the accuracy of the classification algorithms is assessed.

The rest of this paper is structured in the following way: first, the feature subset selection problem is introduced and

GECCO'16, July 20–24, 2016, Denver, Colorado, USA.

ISBN 978-1-4503-4206-3/16/07…$15.00

DOI: http://dx.doi.org/10.1145/2908812.2908853

formally defined in section 2. Moreover, recent evolutionary feature subset selection methods are reviewed in section 2.1. The notion of entropy and its role in the area of feature subset selection is summarized in section 3. Entropy-based feature subset selection algorithms are briefly discussed in section 3.1. A new fitness function, genetic algorithm, and differential evolution for entropy-based feature subset selection are detailed in section 4 and extensive computational experiments are presented in section 5. Finally, the work is summarized and major conclusions are drawn in section 6.

2. FEATURE SUBSET SELECTION

Feature subset selection is a high-level procedure that seeks an optimum subset of data features selected according to a particular criterion (set of criteria) [19]. In a general data set, $Y = \{A \cup Z\}$, comprising of a set of input features, $A = \{a_i, \ldots, a_N\}$, and a set of decision (target) features, Z, it looks for a subset, $B \subset A$, that has the highest evaluation score, $f_{\text{eval}}(B)$ [9]. This process can be defined as a general search [19] or optimization [9] problem.

Feature subset evaluation criteria usually depend on the specific purpose of feature selection. In data mining and classification, it often aims at removing redundant and irrelevant features that can mislead some data processing algorithms [19]. Two general classes of feature subset selection criteria are used most often [19, 9]. *Wrapper*-based approaches look for subsets of features for which particular classification algorithm reaches the highest accuracy. In contrast, *filter*-based approaches are classifier independent. They utilize various indirect feature subset evaluation measures based on e.g. statistical, geometric, and information-theoretic measures.

2.1 Evolutionary feature subset selection

Nature-inspired metaheuristics have been extensively utilized for feature subset selection. A recent review [9] shows many examples of evolutionary and swarm-based feature selection methods. Genetic and memetic algorithms, simulated annealing, particle swarm optimization, ant colony optimization and e.g. artificial bee colonies are only the major nature-inspired algorithms used for feature subset selection in the past.

They all apply different metaheuristic operations and employ various types of feature subset models. Genetic algorithms work with binary or other discrete feature subset representations whereas real-parameter optimization methods such as the particle swarm optimization algorithm translate this combinatorial optimization problem into the continuous domain. The fitness (objective) functions, utilized by these methods, are plentiful and can include both, filter and wrapper-based feature subset evaluation measures [9].

In this work, we compare the ability of a recent genetic algorithm and differential evolution for fixed-length subset selection to find good feature subsets. The investigated evolutionary methods adopt a filter-based approach to feature subset ranking that uses a compression-based entropy estimation algorithm to evaluate the quality of selected feature subsets. The concept of entropy and the rationale of the proposed approach is discussed in the next section.

3. INFORMATION ENTROPY

Entropy is a general concept defined by Shannon [24] that expresses the average amount of information contained in a message. Entropy of a random variable, X, consisting of a sequence of values, $x_1, x_2, \ldots, x_n$, is defined by

$$H(X) = -\sum_i P(x_i) \log_2 P(x_i), \tag{1}$$

where $P(x)$ is the probability of the symbol, x, appearing in the sequence, X. However, entropy of a single variable is not enough for selection of useful features. Therefore, several derived entropy-based measures were defined. Conditional entropy $H(X|Y)$, for example, defines the amount of randomness in a variable, X, with respect to another random variable, Y. Information gain [22] measures how the entropy of X decreases with the knowledge Y. This measure reflects the amount of added information and, therefore, is very useful for an efficient comparison of data features. It is a key part of the C4.5 classification and decision tree algorithm [22]. Nevertheless, a wide variety of sophisticated entropy-based feature selection methods has been devised in the past.

3.1 Entropy-based feature selection

Various forms of entropy calculation and estimation are frequently used in many application areas, especially in computational linguistics, natural language processing, and information processing. Berger et al. [3] proposed an iterative feature selection (IFS) algorithm that uses a maximum entropy approach. Largeron et al. [16] compared four different entropy-based measures for text classification. The authors also proposed a new measure that combined the entropy of a word with its distribution among documents in a collection. Mutual information was used as a feature subset evaluation measure in [30]. Under certain circumstances, it utilized an algebraic combination of the pairwise mutual information between data features to approximate their mutual information in a high-dimensional space. The feature subset selection process required in that approach an exhaustive search to find optimum feature subsets. Jaganathan and Kuppuchamy [11] proposed a fuzzy entropy-based measure for medical data classification. They computed entropy using fuzzy membership values obtained by the Fuzzy C-means clustering algorithm applied to all features. The best feature subsets were then selected using one of three distinct strategies.

A comparison of six feature selection algorithms based on entropy is provided in [4]. After a thorough analysis, the authors concluded that there is no single best feature selection algorithm and each surveyed method had its pros and cons. Zeng et al. [29] investigated in another work the risk that some features can be removed even though they have a high classification ability. Such ability can be hidden and is revealed only in combination with other data features. In order to address this problem, a method based on so called neighborhood entropy was proposed and evaluated using game theory. A non-parametric differential entropy estimator based on the nearest neighbors of a sample data set was proposed by Nilsson and Kleijn [20]. The estimator works for data that lies on a differentiable $d-$dimensional manifold and its underlying random process has a differentiable probability density function. Jiang et al. [12] suggested recently a novel, scalable algorithm for feature selection based on the rough set theory.

This overview demonstrates that different types of entropy

can be used as valid classifier-independent feature selection criteria. Practical entropy evaluation is an important part of all methods that use entropy-based concepts for feature selection. In many cases, computationally efficient entropy estimators are used in place of exact information measures.

3.2 Compression-based entropy estimation

Entropy evaluation is a difficult task. The computation of advanced measures such as conditional entropy is therefore even more complicated. Compression-based entropy estimates are often used for entropy approximation in practical settings [17, 26]. Data compression involves, in fact, a direct computation of the theoretical entropy contained in a block of data. The link between data compression and entropy is possible due to the dual relationship between Shannon entropy, H, [24] and Kolmogorov complexity [13]. The Kolmogorov complexity, $K(x)$, of a binary string, $x = \{0, 1\}^*$, is the length of the shortest binary program with no input that can generate x [13]. The Kolmogorov complexity of a string, x, given another string, y, is denoted $K(x|y)$. It is defined as the length of the shortest binary program for universal prefix Turing machine that, on input y, outputs x [13]. $K(x|y)$ corresponds to conditional Shannon entropy, $H(X|Y)$. Unfortunately, the Kolmogorov complexity is non-computable. Li et al. [17] and Cilibrasi [7] transformed this concept into a computable form by associating it with lossless data compression [17, 26]. The value of $K(x|y)$ is approximated by $C(x \cdot y)$, where $C(x \cdot y)$ is the compressed size of a concatenation of x and y. Li and Vitanyi [17, 26] proved that any compression algorithm that satisfies a set of constraints, including the requirement that $C(x) \approx C(x \cdot x)$, can be used to approximate Kolmogorov complexity.

The compression of data features is a non-trivial problem. Real-world data is usually represented by floating-point values. Compressing sequences of floating-point numbers is due to their specific representation a difficult problem. However, several efficient floating-point data compression algorithms exist. One of the best methods for the compression of double-precision values is due to Burtscher and Ratanaworabhan [5]. The algorithm combines the results from two value predictors and stores the difference using a prefixed code. According to our experiments, it satisfies the requirements for a compression-based approximation of Kolmogorov complexity. In this work, it is used to evaluate the quality of feature subsets.

4. ENTROPY-BASED FEATURE SUBSET SELECTION BY GENETIC ALGORITHMS AND DIFFERENTIAL EVOLUTION

Over the last decades, evolutionary methods [1] have been successfully used to solve a number of hard optimization problems. Two widely-used evolutionary metahuristics are in this study used to find subsets of features in real-world data sets. In order to do that, a novel features subset evaluation measure, based on a fast compression-based approximation of data entropy, is proposed and tested in context of the evolutionary methods.

4.1 Genetic Algorithms and Differential Evolution

Genetic algorithms (GAs) and Differential Evolution (DE) are two popular population-based evolutionary metaheuristics [1, 21]. They both solve complex optimization problems by a programmatic evolution of an encoded population of candidate solutions that are evaluated using a domain-specific fitness function. The artificial evolution is implemented by an iterative application of various operators that trigger different kinds of problem space exploration strategies.

GAs operate on a discrete representation of candidate solutions and problem encoding is an important part of their problem solving strategy. The encoding translates candidate solutions from a problem space (phenotype) to an encoded search space (genotype) that is explored by the algorithm. It is defined by chromosome data structure and by an encoding (decoding) function [8]. The data structure specifies the actual search space, its size, and shape.

GAs employ two main operators, crossover and mutation. Crossover is the principal operator of genetic algorithms distinguishing them from other population-based stochastic search methods [1]. It is the primary creative force behind the GA process that is intended to re-combine parent chromosomes in a stochastic manner and to propagate building blocks (low-order, low-defining-length schemata with above average fitness) from one generation to another. Crossover creates new (higher-order) building blocks by combining low-order ones and introduces to the population large changes with only small disruption of existing building blocks [27]. In contrast, mutation is expected to discover new genetic material by random perturbations of chromosome structure. As a result, new building blocks can be created or old ones destroyed [27].

The DE, on the other hand, is an evolutionary algorithm for real-parameter optimization [21]. It evolves a population of real-encoded candidate vectors by differential mutation and crossover [21]. During the optimization, the DE generates new vectors that are scaled perturbations of existing population vectors. The algorithm perturbs selected base vectors with the scaled difference of two (or more) other population vectors in order to produce the trial vectors. The trial vectors compete with members of the current population with the same index called the target vectors. If a trial vector represents a better solution than the corresponding target vector, it takes its place in the population [21].

GAs and DE are successful optimization methods with many applications. They both are highly parallel population based stochastic search metaheuristics. The traditional GAs use a discrete representation (encoding) of candidate solutions while the DE operates on real-encoded candidate vectors. Each algorithm uses different high-level operations to evolve the population. It results in different search strategies and different directions found by the algorithms when solving a particular problem. Both methods have been recently used to solve problems involving fixed-length subset selection [14, 15]. In the next sections, we define a novel fitness function for feature subset evaluation and summarize the basic principles of a GA and a DE for feature subset selection.

4.2 Compression-based feature subset evaluation

Fitness function is the only domain-specific element of the GA and DE for feature subset selection. It uses a fast lossless compression algorithm, FPC, [5] to estimate the entropy of selected feature subsets. The evaluation of a candidate

solution, $\boldsymbol{c}$, representing a subset of k features, $(c_1, \dots c_k)$, is defined by

$$f_{\text{eval}}(\boldsymbol{c}) = \text{FPC}(c_1 \cdot c_2 \cdots c_k), \tag{2}$$

where FPC is the compressed length of the sequence of concatenated features selected by $\boldsymbol{c}$. The compressed size of a fixed-length feature subset is assumed to be proportional to its entropy and information content. Feature subsets with large entropy and rich information content are expected to have larger compressed sizes than subsets of highly correlated features. Compressed feature subset size is in the investigated evolutionary metaheuristics used as fitness function. However, because the traditional DE is defined as a minimization algorithm, the fitness function is transformed into

$$f_{\text{fit}}(\boldsymbol{c}) = \frac{f_{\text{eval}}(\mathbf{c})}{\text{FPC}(f_1 \cdots f_N)}, \tag{3}$$

where $f_1, \dots, f_N$ is the concatenation of all input features of a data set. The fitness function is minimized by both investigated algorithms.

4.3 GA for feature subset selection

The GA for feature subset selection, employed in this work, uses a compact chromosome encoding and special genetic operators that enable the use of a full-flavoured GA with crossover and mutation [15]. The encoding is based on the ordering of genes in chromosomes and prevents the creation of invalid individuals in course of the evolution. A subset of k features from N is in this approach represented by a chromosome, $\mathbf{c}$, defined by

$$\begin{aligned} &\boldsymbol{c} = (c_1, \dots c_k), \\ &\forall (i,j) \in \{0, \dots N\} : c_i \neq c_j,\ i < j \implies c_i < c_j, \end{aligned} \tag{4}$$

where c_i and c_j are indices of selected features. An index-based subset encoding is invariant to the ordering of genes. However, the ordering must be maintained during the genetic search process to avoid the inception of invalid individuals. Because of that, order-preserving crossover and mutation operators are used. Special genetic operators are required because this encoding differs from that used for permutation-based combinatorial optimization problems and the traditional order-type crossover operators (e.g. order crossover, partially matched crossover, uniform partially matched crossover, etc. [6]) cannot be used.

The *order-preserving mutation* operator replaces the ith gene, c_i, in a chromosome, $\boldsymbol{c}$, by a random value taken from the interval defined by its left and right neighbour, as defined in eq. (5)

$$mut(c_i) = \begin{cases} urand^*(0, c_{i+1}), & \text{if } i = 0 \\ urand(c_{i-1}, c_{i+1}), & \text{if } i \in (0, N-1) \\ urand(c_{i-1}, N), & \text{if } i = N-1 \end{cases}, \tag{5}$$

where $i \in \{0, \dots, N\}$ and $urand(a, b)$ selects a uniform pseudo-random integer from the interval (a, b) (whereas $urand^*(a, b)$ selects a uniform pseudo-random integer from the interval $[a, b)$). This mutation operator guarantees that the ordering of indices within the chromosome remains valid after the mutation. However, the order-preserving mutation of the ith gene has no effect on chromosomes for which it holds that $(c_{i-1} + 1) = c_i = (c_{i+1} - 1)$.

The *order-preserving crossover* operator is based on the traditional one-point crossover [1]. It selects a random position, i, in parent chromosomes, $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$, and checks if it can be used for crossover. A position, i, is suitable for crossover iff eq. (6) is true.

$$c_{1,i} < c_{2,i+1} \wedge c_{2,i} < c_{1,i+1} \tag{6}$$

If eq. (6) does not hold for i, the remaining positions in the chromosomes are sequentially scanned in the search for a suitable crossover location (i.e. a position for which eq. (6) holds). It should be noted that an order-preserving crossover between 2 chromosomes might not be always possible.

4.4 DE for feature subset selection

The investigated DE for feature subset selection is based on the $/DE/rand/1$ version of the algorithm [21] and uses the traditional DE crossover and mutation operators. It translates the combinatorial optimization problem into the continuous domain using an intuitive candidate representation that was previously employed to solve the $p-$Median problem [14].

A candidate solution is in this approach represented by a real-valued vector, $\boldsymbol{c}$, of the size k. Each vector, $\boldsymbol{c}$, is decoded into a set of k feature indices, B. Every floating-point coordinate of $\boldsymbol{c}$, c_i, is in this process truncated and added to B. If $trunc(c_i)$ already belongs to B, the next available feature that is not in B yet is added to the subset.

5. EXPERIMENTS

A series of computational experiments was conducted in order to evaluate the ability of the GA and the DE to find good feature subsets. Two well-known data sets, *Hepatitis* and *Spambase*, were downloaded from the UCI machine-learning repository [18] and used to find subsets of 2, 3, 4, 5, 10, and 15 features, respectively. In order to validate the proposed fitness function and its usefulness in the context of practical data mining and classification, the data sets were processed by several traditional classification algorithms. The battery of classification methods, used in this research, consisted of the classification and regression tree algorithm (CART), Naïve Bayes classifier (NB), and two variants of the k-Nearest Neighbour algorithm with $k = 1$ (kNN(1)) and $k = 3$ (kNN(3)), respectively [10]. These classification algorithms were selected due to their wide use in the field of data mining and because similar types of classifiers (C4.5 and Naïve Bayes classifier) were employed to evaluate various nature-inspired feature selection methods in a recent survey [9].

Properties of the test data sets and the number of classification errors, obtained by each classifier for each data set, are summarized in table 1. Apparently, CART and kNN(1) were able to classify both full data sets with the least error.

Table 1: Data set properties and the number of classification errors for full data sets.

			Classification errors			
Dataset	Attrs.	Records	CART	NB	kNN(1)	kNN(3)
Hepatitis	20	80	0	11	8	13
Spambase	58	4601	3	513	3	216

5.1 FPC as a feature subset evaluation criterion

All possible subsets of 2, 3, and 4 features, respectively, were analyzed for the test data sets. The value of the fitness function, FPC, and the classification error of all utilized classifiers were computed for each feature subset in order to discover the relationship between FPC and classification error. This relationship was also validated by Spearman's rank correlation [23], as summarized in table 2.

Table 2: Rank correlation (Spearman's ρ and $p-$value) between FPC and the number of classification errors on the test data sets. The $p-$value is shown in parentheses.

	Classifier			
Dataset	CART	NB	kNN(1)	kNN(3)
Hepatitis	-0.786 ($3.9E^{-7}$)	-0.039 (0.6)	-0.781 ($2.2E^{-36}$)	-0.688 ($2.5E^{-25}$)
Spambase	-0.840 (0.0)	-0.300 ($1.2E^{-34}$)	-0.534 ($1.7E^{-118}$)	-0.530 ($4.6E^{-116}$)

The table illustrates that the correspondence between FPC and the classification error depends on particular classifier and data set. There is a high negative correlation between FPC and CART errors on both test data sets. The number of NB classification errors exhibits the lowest correlation with FPC. Still, there is a low negative correlation between FPC and NB classification errors on the *Spambase* data set. The number of classification errors of both k-Nearest Neighbour classifiers exhibit moderate negative correlation with FPC. The results of this analysis are graphically illustrated in fig. 1 and fig. 2, respectively. The figures clearly show that high FPC values correspond to feature subsets with lower classification error. However, as shown especially in fig. 1, good feature subsets can be associated also with low FPC values. This trend is less pronounced on the larger *Spambase* data set (fig. 2). The figures also document that the Naïve Bayes classifier performs poorly on both test data sets (second columns of fig. 1 and fig. 2, respectively). These results show that the FPC is a reasonable feature subset evaluation measure. Evolutionary search for feature subsets with high FPC value will lead to feature subsets (i.e. reduced data sets) that model the original data with low error and high accuracy.

5.2 Evolutionary feature subset selection

Both investigated evolutionary methods for feature subset selection were implemented in C++ and used to find subsets of features in the *Hepatitis* and *Spambase* data sets. The GA used a steady-state replacement scheme [1] with generation gap 2 (offspring chromosomes immediately entered the population), population size 100, probability of mutation $m = 0.3$, probability of crossover $c = 0.8$, and maximum number of generations 5,000. The DE was a traditional $/DE/rand/1$ variant of the algorithm with scaling factor $F = 0.5$, mutation probability $C = 0.9$, population size 50 and the maximum number of generations 200. Both method used f_{fit}, defined in eq. (3), as the fitness function. The parameters of both methods were selected on the basis of best practices and initial trial-and-error runs. The maximum number of generations was set so that the total number of fitness function evaluations was in both algorithms the same (10,000).

The algorithms were used to find subsets of 2, 3, 4, 5, 10, and 15 features, respectively. To cope with their stochastic nature, all performed experiments were repeated 50 times. The results of the evolutionary search for feature subsets by both methods are shown in table 3.

The table illustrates that both methods were able to find feature subsets with good FPC values in the long run. The best found and average feature subsets, discovered by the GA and the DE, always reached high FPC values. However, the worst feature subsets, found during the 50 independent trials by the GA, reached only low FPC for all feature subset sizes on the *Hepatitis* data set and for the subsets of 2 and 3 features also on the *Spambase* data set. The DE, on the other hand, was in most cases able to find feature subsets with high FPC values also in the worst case. Lower values of the standard deviation, σ, indicate stability and good convergence of this algorithm. The experiments with the *Spambase* data set produced different results, though. The DE was in this case better than the GA only for small feature subsets. On the contrary, it was outperformed by the GA for larger feature subsets with $k \in \{4, 5, 10, 15\}$.

The results of this comparison were validated by the t-test [23] at confidence level $\alpha = 0.05$. The test showed that the DE was on the *Hepatitis* data set significantly better than the GA for all $k \in \{2, 3, 4, 15\}$. The differences between the results obtained on this data set by the GA and the DE for the subsets of 5 and 10 features, respectively, were found insignificant. The differences between the results obtained by the GA and the DE on the *Spambase* data set were statistically significant at confidence level $\alpha = 0.05$ only for the subsets of 10 and 15 features, respectively. In all other cases, the differences between the final results found by the GA and the DE were insignificant. It means that the low quality solutions, found by the GA in the worst runs of the *Spambase* experiment, can be treated as exceptional, outlying cases. The evolved subsets of 2, 3, and 4 features were checked against all feature subsets of the same size and the percent of all subsets with worse FPC (i.e. the percentile score of the evolved feature subsets) was computed. The results of this analysis are summarized in table 4. We note that the best found feature subsets reached in all cases the best possible FPC value. In case their percentile score is lower than 100, it is because multiple subsets with the same FPC value existed in the data set.

Table 4: The percent of feature subsets with FPC lower than best, average, and worst subsets found by the investigated methods.

		GA percentile			DE percentile		
Dataset	k	best	average	worst	best	average	worst
Hepatitis	2	99.42	57.89	2.34	99.42	99.42	99.42
	3	100.00	94.22	24.10	100.00	100.00	100.00
	4	99.96	97.81	33.13	99.96	99.96	99.96
Spambase	2	100.00	99.81	47.99	100.00	100.00	100.00
	3	100.00	99.99	4.97	100.00	100.00	100.00
	4	100.00	100.00	100.00	100.00	99.99	99.99

The quality of the evolved feature subsets in terms of classification error, obtained by the employed classification algorithms, is illustrated in fig. 3. The figure shows for both test data sets the number of errors obtained by CART and kNN(1) applied to all evolved 2-feature subsets. The fea-

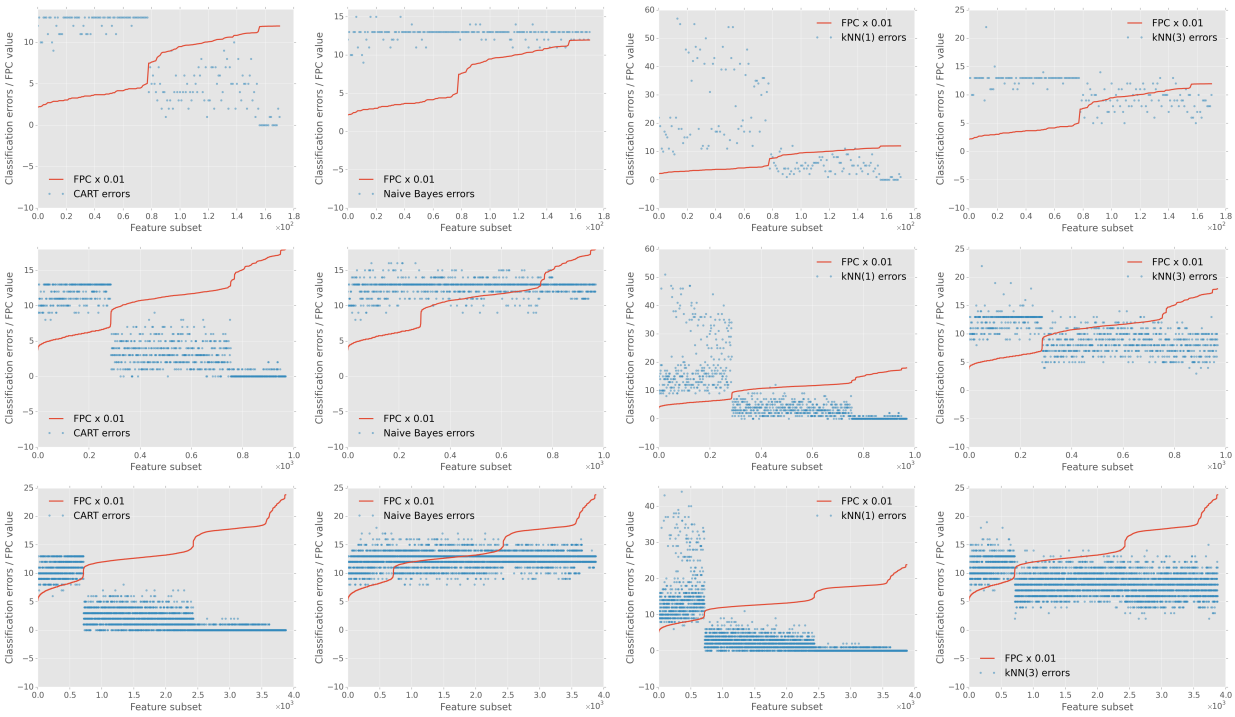

Figure 1: The relationship between the proposed fitness function, FPC, and the classification error for all subsets of 2 (1st row), 3 (2nd row), and 4 (3rd row) features on the *Hepatitis* data set. The classifiers are CART (1st column), Naïve Bayes (2nd column), and k-Nearest Neighbour classifier with one (3rd column) and three neighbours (last column), respectively. Feature subsets are ordered by FPC.

Table 3: The final FPC of feature subsets evolved by the GA and the DE.

		FPC of GA-evolved feature subsets			FPC of DE-evolved feature subsets		
Dataset	k	best	average (σ)	worst	best	average (σ)	worst
Hepatitis	2	1195	939.52 (331.43)	230	1195	1195 (0)	1195
	3	1796	1694.94 (255.15)	646	1796	1796 (0)	1796
	4	2380	2274.38 (294.86)	1238	2380	2380 (0)	2380
	5	2972	2887.30 (303.79)	1317	2972	2972 (0)	2972
	10	4728	4677.40 (277.75)	2743	4728	4727.90 (0.30)	4727
	15	5544	5261.40 (457.61)	3989	5544	5518.04 (32.31)	5452
Spambase	2	66064	63203.02 (11328.08)	16671	66064	66064 (0)	66064
	3	97466	95822.56 (11504.08)	15294	97466	97466 (0)	97466
	4	122431	122431 (0)	122431	122431	122318.92 (549.08)	119629
	5	142234	142234 (0)	142234	142234	142110.56 (604.73)	139148
	10	228155	221059.80 (5283.04)	210622	217278	206335.58 (4840.57)	198413
	15	287258	276567.86 (7387.49)	258259	274438	260328.52 (5225.25)	251003

ture subsets, found during the 50 independent runs by both methods, are in the figure represented by red crosses. In case less than 50 crosses are shown, multiple runs have found the same solution.

6. CONCLUSIONS

Two new evolutionary algorithms for feature subset selection are designed and evaluated in this work. They both utilize a novel feature subset evaluation criterion based on a fast approximation of feature subset entropy. The entropy is in this approach linked to the size of the feature subset compressed by the FPC algorithm [5]. It assumes that the compressed size of a feature subset with high entropy and high information content is higher than the compressed size of feature subsets with highly correlated, redundant content.

The FPC was selected as an entropy approximation method due to its ability to process arbitrary double-precision data including e.g. time series. Extensive computational experiments, performed on two well-known data sets, showed that the proposed fitness function negatively correlates with the number of classification errors of several traditional data mining algorithms (CART, kNN with k equal to 1 and 3). On the other hand, it did not correspond to the number of classification errors of the Naïve Bayes classifier. However, the NB classifier did not perform well even on full variants of the test data sets. Although these observations were made for a particular combination of data sets, classification algo-

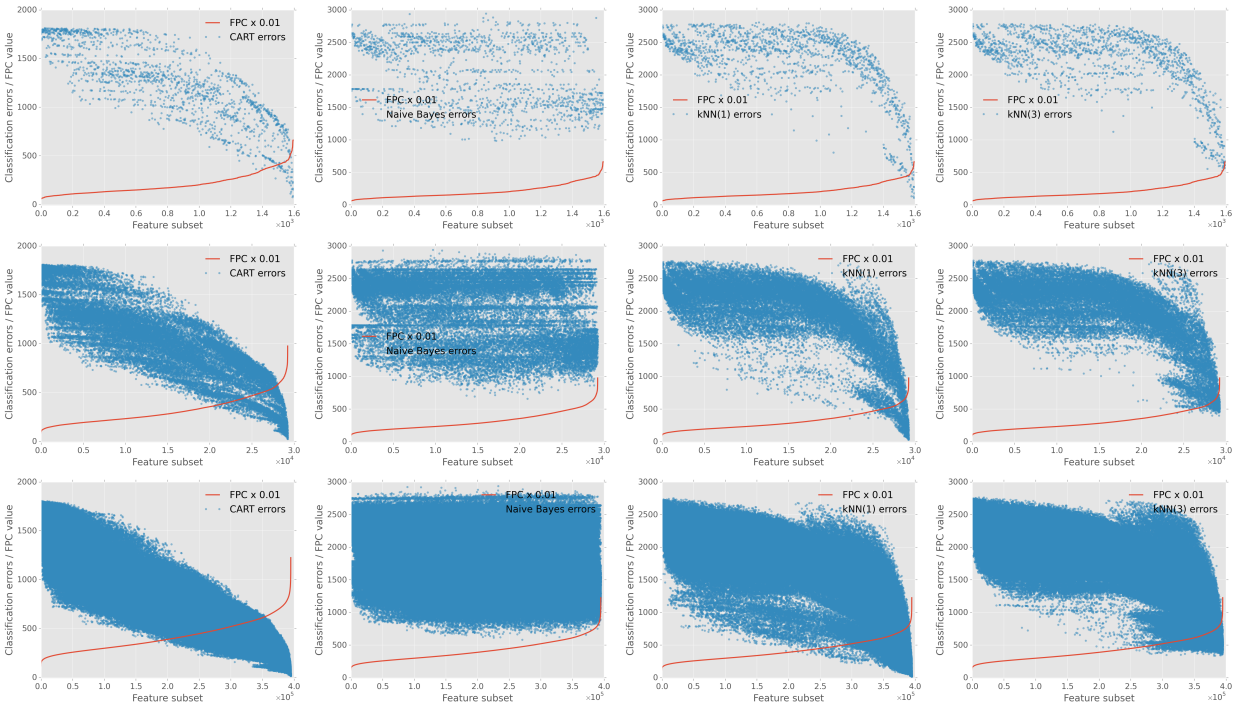

Figure 2: The relationship between the proposed fitness function, FPC, and the classification error for all subsets of 2 (1st row), 3 (2nd row), and 4 (3rd row) features on the *Spambase* data set. The classifiers are CART (1st column), Naïve Bayes (2nd column), and k-Nearest Neighbour classifier with one (3rd column) and three neighbours (last column), respectively. Feature subsets are ordered by FPC.

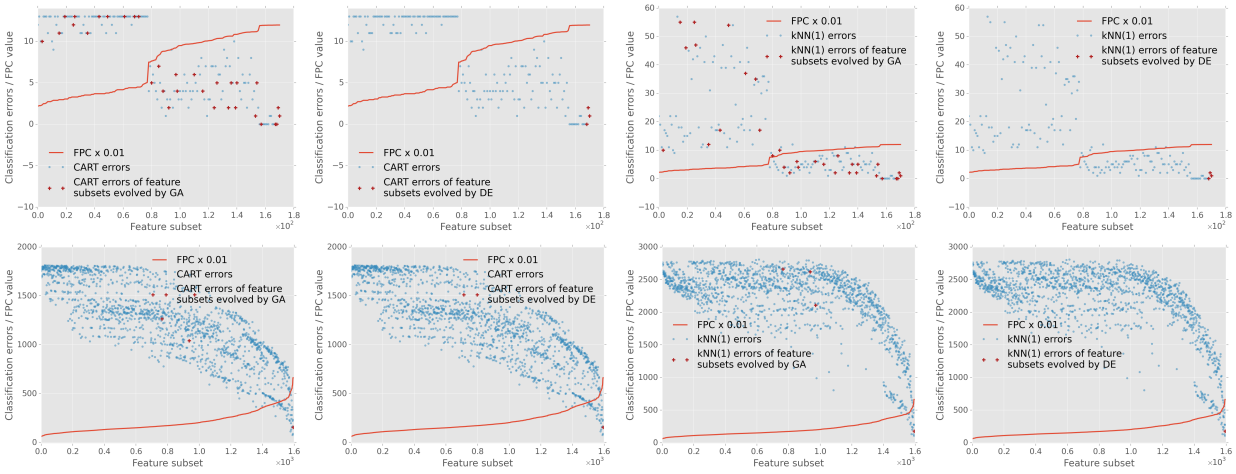

Figure 3: CART and kNN(1) classification errors of 2-feature subsets evolved by GA and DE on the *Hepatitis* (1st row) and *Spambase* data sets (2nd row).

rithms, and feature subset sizes, they strongly suggest that the proposed fitness function is a good feature subset selection criterion that can be optimized by evolutionary methods well.

The GA and the DE were used to search for feature subsets of different sizes. Conducted computational experiments showed that both algorithms were able to discover feature subsets with high FPC values. A direct comparison of the results, obtained by the GA and the DE, showed that their accuracy and stability depends on both, the data set and the target feature subset size. A statistical analysis of experimental results showed that the DE performed on par with or significantly better than the GA on the smaller data set, *Hepatitis*. However, it was outperformed by the GA in case of the larger and more complex data set. The subsets of 10 and 15 features, found by the GA in the *Spambase* data set, had significantly higher FPC than the results discovered by the DE. The differences between the results, found by the

GA and the DE in the *Spambase* data set in all other cases, were found insignificant.

The results, obtained in this research, suggest that the DE is more appropriate for feature subset selection from smaller data sets or when the target feature subset sizes are very small. The GA, on the other hand, seems to be more suitable for feature subset selection from more complex data sets and when the target feature subset sizes are larger. These findings are encouraging. The proposed fitness function is a promising feature subset selection criterion and investigated evolutionary methods have confirmed good ability to find excellent feature subsets.

7. ACKNOWLEDGEMENTS

This work was supported by the Czech Science Foundation under the grant no. GJ16-25694Y and in part by the Grant of SGS No. SP2016/68, VŠB - Technical University of Ostrava, Czech Republic.

8. REFERENCES

[1] M. Affenzeller, S. Winkler, S. Wagner, and A. Beham. *Genetic Algorithms and Genetic Programming: Modern Concepts and Practical Applications*. Chapman & Hall/CRC, 2009.

[2] C. C. Aggarwal and P. S. Yu. Outlier detection for high dimensional data. *SIGMOD Rec.*, 30(2):37–46, 2001.

[3] A. L. Berger, V. J. D. Pietra, and S. A. D. Pietra. A maximum entropy approach to natural language processing. *Comput. Linguist.*, 22(1):39–71, 1996.

[4] J. Biesiada, W. Duch, A. Kachel, K. Maczka, and S. Palucha. Feature ranking methods based on information entropy with parzen windows. In *Int. Conf. on Research in Electrotechnology and Applied Informatics*, vol. 1, p. 1, 2005.

[5] M. Burtscher and P. Ratanaworabhan. Fpc: A high-speed compressor for double-precision floating-point data. *IEEE Trans. on Computers*, 58(1):18–31, 2009.

[6] V. A. Cicirello. Non-wrapping order crossover: An order preserving crossover operator that respects absolute position. In *Proc. of the 8th Annual Conf. on Genetic and Evolutionary Computation*, GECCO '06, pp. 1125–1132, New York, NY, USA, 2006. ACM.

[7] R. Cilibrasi and P. Vitanyi. Clustering by compression. *Information Theory, IEEE Trans. on*, 51(4):1523–1545, 2005.

[8] A. Czarn, C. MacNish, K. Vijayan, and B. A. Turlach. Statistical exploratory analysis of genetic algorithms: The influence of gray codes upon the difficulty of a problem. In *Australian Conf. on Art. Int.*, vol. 3339 of *LNCS*, pp. 1246–1252. Springer, 2004.

[9] R. Diao and Q. Shen. Nature inspired feature selection meta-heuristics. *Art. Int. Rev.*, 44(3):311–340, 2015.

[10] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer New York, 2013.

[11] P. Jaganathan and R. Kuppuchamy. A threshold fuzzy entropy based feature selection for medical database classification. *Computers in Biology and Medicine*, 43(12):2222 – 2229, 2013.

[12] F. Jiang, Y. Sui, and L. Zhou. A relative decision entropy-based feature selection approach. *Pattern Recognition*, 48(7):2151 – 2163, 2015.

[13] A. N. Kolmogorov. Three approaches to the quantitative definition of information. *Problems of Information Transmission*, 1(1):1–7, 1965.

[14] P. Kromer, J. Platos, and V. Snasel. Traditional and self-adaptive differential evolution for the p-median problem. In *Cybernetics (CYBCONF), 2015 IEEE 2nd Int. Conf. on*, pp. 299–304, 2015.

[15] P. Krömer and J. Platoš. Genetic algorithm for sampling from scale-free data and networks. In *Proc. of the 2014 Annual Conf. on Genetic and Evolutionary Computation*, GECCO '14, pp. 793–800, New York, NY, USA, 2014. ACM.

[16] C. Largeron, C. Moulin, and M. Géry. Entropy based feature selection for text categorization. In *Proc. of the 2011 ACM Symposium on Applied Computing*, SAC '11, pp. 924–928, New York, NY, USA, 2011. ACM.

[17] M. Li, X. Chen, X. Li, B. Ma, and P. Vitanyi. The similarity metric. *Inf. Theory, IEEE Trans. on*, 50(12):3250–3264, 2004.

[18] M. Lichman. UCI machine learning repository, 2013.

[19] H. Liu and H. Motoda. *Feature Selection for Knowledge Discovery and Data Mining*. The Springer Int. Series in Eng. and Comp. Sci. Springer US, 2013.

[20] M. Nilsson and W. Kleijn. On the estimation of differential entropy from data located on embedded manifolds. *Information Theory, IEEE Trans. on*, 53(7):2330–2341, 2007.

[21] K. V. Price, R. M. Storn, and J. A. Lampinen. *Differential Evolution A Practical Approach to Global Optimization*. Natural Comp. Series. Springer-Verlag, Berlin, Germany, 2005.

[22] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Francisco, CA, USA, 1993.

[23] N. Salkind. *Encyclopedia of Measurement and Statistics*. SAGE Publications, 2006.

[24] C. Shannon. A mathematical theory of communication. *Bell System Technical Journal, The*, 27(3):379–423, 1948.

[25] M. Verleysen and D. François. *IWANN 2005, Proc. of*, ch. The Curse of Dimensionality in Data Mining and Time Series Prediction, pp. 758–770. Springer Berlin Heidelberg, 2005.

[26] P. M. B. Vitányi. Universal similarity. In *Information Theory Workshop, 2005 IEEE*, pp. 6 pp.–, Aug 2005.

[27] A. S. Wu, R. K. Lindsay, and R. Riolo. Empirical observations on the roles of crossover and mutation. In T. Bäck, editor, *Proc. of the Seventh Int. Conf. on Genetic Algorithms*, pp. 362–369, San Francisco, CA, 1997. Morgan Kaufmann.

[28] H. Xiong, G. Pandey, M. Steinbach, and V. Kumar. Enhancing data analysis with noise removal. *IEEE Trans. on Knowl. and Data Eng.*, 18(3):304–319, 2006.

[29] K. Zeng, K. She, and X. Niu. Feature selection with neighborhood entropy-based cooperative game theory. *Comp. int. and neuroscience*, 2014:11, 2014.

[30] Y. Zheng and C. K. Kwoh. A feature subset selection method based on high-dimensional mutual information. *Entropy*, 13(4):860, 2011.