DE GRUYTER OPEN

Information Technology and Management Science doi: 10.1515/itms-2015-0018

2015 / 18

The Impact of Feature Selection on the Information Held in Bioinformatics Data

Madara Gasparovica-Asite[1], Inese Polaka[2], Ludmila Alekseyeva[3]
[1, 2, 3] *Riga Technical University*

***Abstract* – The present research examines a wide range of attribute selection methods – 86 methods that include both ranking and subset evaluation approaches. The efficacy evaluation of these methods is carried out using bioinformatics data sets provided by the Latvian Biomedical Research and Study Centre. The data sets are intended for diagnostic task purposes and incorporate values of more than 1000 proteomics features as well as diagnosis (specific cancer or healthy) determined by a golden standard method (biopsy and histological analysis). The diagnostic task is solved using classification algorithms FURIA, RIPPER, C4.5, CART, KNN, SVM, FB+ and GARF in the initial and various sets with reduced dimensionality. The research paper finalises with conclusions about the most effective methods of attribute subset selection for classification task in diagnostic proteomics data.**

***Keywords* – Bioinformatics, classification, data mining, diagnostics, feature selection.**

I. Introduction

This article presents the study of attribute subset selection effects on the accuracy of classifiers. The present research is designed based on the bioinformatics task, when biomedical experts are searching for biomarker sets among a thousand or several thousands of proteins or genes. The potential diagnostic test is modelled by a classifier, which is built using a reduced data set. The attributes that are selected for the reduced data set are the potential biomarkers.

Research aimed to determine which attribute selection methods are more suitable for omics data with large dimensionality examines the following choices and trade-offs when a researcher has to make the choice of the method for attribute selection:

- Subset searches vs. individual attribute evaluators,
- Statistics vs. information evaluation,
- Attribute set size differences.

The feature selection and evaluation of the resulting data subsets (and therefore the methods that were used in feature selection) were carried out using five data sets provided by the Latvian Biomedical Research and Study Centre. The data sets were obtained in proteomics study, where a number of cancer patients and healthy controls were tested for 1229 antibody presence (presence of each antibody was also quantified) in order to find an antibody panel for disease detection. The cancers in question are breast cancer (abbreviated in figures and tables as BrCa to save space), gastric cancer (GaCa), melanoma (Mel) and prostate cancer (PrCa) as well as a group of patients with gastrointestinal diseases other than cancer (also paired with a set of healthy controls; GIS).

II. Methods and Approaches

This study uses feature selection methods that evaluate full subsets (search algorithm and subset evaluation techniques) as well as those based on ranking to test various feature subset (gene or protein panel) sizes by selecting the top 10, 20, 50, 100 or 200 features. The feature subsets (evaluation metrics and subset sizes) were tested using a group of classification algorithms that were applied with the same parameters to all data subsets. Classification was performed using different classification approaches to reduce the preference of one method that would be more appropriate for one data set and perform badly in other data sets.

A. Ranking Methods

Ranking-based feature evaluation and subset selection methods evaluate single features using various metrics and assign a rank to each feature based on the performance of the feature. Ranking methods can filter the top features based on the metric and a predefined subset size. The evaluation metrics are usually based on statistical properties of features or the predictive potential of a feature.

One of the metrics used in ranking is *Chi-Square Statistic* (abbreviated as Chi in graphs and tables) that is calculated with respect to the class [1]. It also works with discrete data types. The statistic for a problem with k classes and N instances is calculated as shown in (1):

$$\chi^2 = \sum_{i=1}^{2}\sum_{j=1}^{k}\frac{(A_{ij}-E_{ij})^2}{E_{ij}}, \quad (1)$$

where A_{ij} is the number of instances in the i-th interval (with i-th value), j-th class,

E_{ij} is the expected frequency of A_{ij}, which is calculated as shown in (2):

$$E_{ij} = \frac{R_i \cdot C_j}{N}, \quad (2)$$

where R_i is the number of instances in the i-th interval,

C_j is the number of instances in the j-th class.

Another popular metric to evaluate features is *Information Gain* (IG in graphs and tables) that measures information content with respect to the class. *Information Gain* is used in decision tree induction and was introduced by J. R. Quinlan [1]. Prior to feature evaluation, the numeric attribute values have to be discretised because this approach works with categorical data. This metric is based on the change of information entropy that would occur if the state of the information changes (some information is given) and can be

calculated by subtracting conditional entropy of the class from its entropy. The entropy of feature *C* is calculated as shown in (3). Conditional entropy of feature *C* if the state of feature *A* is given is calculated as shown in (4). *Information Gain* is calculated by subtracting conditional entropy from the entropy of *C* as demonstrated in (5). This shows the decrease in entropy of *C* if the value of *A* was given.

$$\mathrm{H(C)} = -\sum_{i=1}^{n} P(C = c_i) log_2(P(C = c_i)), \quad (3)$$

$$\mathrm{H(C|A)} = -\sum_{j=1}^{k} P\big(A = a_j\big) H(C|A = a_j), \quad (4)$$

$$\mathrm{IG(C, A)} = \mathrm{H(C)} - \mathrm{H(C|A)}, \quad (5)$$

where $P(C = c_i)$ is a relative appearance frequency of value c_i in feature *C* in the data set,

$H(C|A = a_j)$ is the entropy of feature *C* in the data subset where the value of attribute *A* is a_j.

Gain Ratio (GR in graphs and tables) is another metric used to evaluate features in decision tree induction [1]. It is based on *Information Gain* metric and eliminates its weakness that occur in data sets that have features with large numbers of unique values, which are given preference over other possibly better features with fewer values. Therefore, *Gain Ratio* divides *Information Gain* by entropy of the considered feature as shown in (6):

$$GR(C, A) = \frac{\mathrm{H(C)} - \mathrm{H(C|A)}}{H(A)}. \quad (6)$$

Also simple classification methods can be used as a basis for feature selection, one of them is the rule induction algorithm *OneR* [3]. It also discretises numeric features (using minimum bucket size as the criteria) and evaluates each feature using error rate that would be observed using the rule that *OneR* generates. One rule is constructed for each feature and its error evaluates how this rule classifies the data. This classification error is also used to rank features in this feature selection approach.

Relief algorithm [4] evaluates a feature by randomly sampling instances and analysing two neighbouring instances of the same and different classes. This algorithm was not able to work with missing data and data sets that included three or more classes; therefore, it was improved resulting in *Relief-F* algorithm [4]. It is adapted to work with multi-class problems by finding one or more (*k*) neighbouring instances *MI* from each different class *C* and averages their contribution for upgrading estimates *W*[*A*] weighting it with the prior probability of each class. The estimation of weight *W* of feature *A* [4] when the sampled instance is *R* (which is sampled *m* times) and the nearest instance of the same class *H* is conducted as shown in (7):

$$W[A] \coloneqq W[A] - \sum_{j=1}^{k} \frac{diff(A,R,H_j)}{m \cdot k} + \sum_{C \neq class(R)} \frac{P(C)}{1 - P(class(R_i))} \sum_{j=1}^{k} \frac{diff\big(A,R,M_j(C)\big)}{m \cdot k}. \quad (7)$$

The number of the checked neighbouring instances is determined by either predefining a number or the maximum distance. The difference *diff*(A, I_1, I_2) for discrete features is 1 if the values of instances are equal and 0 if the values are different. The difference of numeric features is calculated as shown in (8):

$$diff(A, I_1, I_2) = \frac{|value(A,I_1) - value(A,I_2)|}{\max(A) - \min(A)}. \quad (8)$$

B. Subset Evaluation Methods

Correlation-based Feature Selector (CFS) is a filter algorithm that ranks feature subsets according to a correlation-based heuristic evaluation function that selects features highly correlated with the class feature and uncorrelated with each other [5]. It allows distinguishing features with a high predictive accuracy in the instance space that is not already covered by other selected features (the low inter-correlation of the selected features). The heuristic evaluation merit *M* for a subset *S* containing *k* features is calculated as shown in (9):

$$M_s = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}}, \quad (9)$$

where r_{cf} is the mean correlation between features and the class attribute,

r_{ff} is the average correlation between features.

Consistency Subset Evaluator (CSE) evaluates feature subsets by the degree of consistency in class values when the training instances are projected onto the set, i.e., the prevalence of one class in subsets that the data set is divided into by attribute values. This also means that feature values have to be discretised [6]. Consistency of a subset can never surpass that of the full set, so the algorithm searches for the smallest subset, which has the same consistency as the full set.

The consistency of a feature subset *S* in a data set with *N* instances is calculated using the equation presented by Liu [7] and given in (10):

$$C_s = 1 - \frac{\sum_{i=0}^{J} |D_i| - |M_i|}{N}, \quad (10)$$

where *J* is the number of distinct attribute value combinations,

$|D_i|$ is the number of occurrences of the *i*-th attribute value combination,

$|M_i|$ is the cardinality of the majority class for the *i*-th attribute value combination.

Symmetrical Uncertainty attribute set evaluation [8] is another method that is based on information theoretical concept *entropy* (see (3)) and *Information Gain* (see (5)). This evaluator compares attribute informativity (data set X and attribute Y) based on symmetrical uncertainty, see (11) for calculation:

$$SU(X, Y) = 2\frac{IG(X|Y)}{H(X) + H(Y)}. \quad (11)$$

The algorithm works with *Fast Correlation-Based Filter Search Method* (FCBFS), which first evaluates, if the feature has SU above a specified threshold, to add it to the selected subset, and then analyses all attributes in the subset for redundancy.

C. Subset Search Methods

Feature subset selection algorithms perform a search over the feature space to select the optimal subset. To perform the search they have to address four basic issues [9]:

- Starting point: starting with no features in the initial subset (*forward selection*) or starting with the full set of features (*backward elimination*);
- Search organisation: considering each possible subset (*exhaustive search*) or locally changing the subset without returning to reconsider the change (*greedy search*); another possible approach is based on adding and removing a feature from the subset in each step to make the search more flexible (*stepwise selection*);
- Evaluation strategy: testing each feature of the subset individually (*filters*) against an evaluation merit or testing the whole subset (*wrappers*).

Stopping criterion: lack of improvement on change, reaching the other end of the feature space or a particular subset size.

Re-ranking search ranks all attributes according to an evaluation metric and then processes this obtained list using *ASSearch* approach to re-rank the features in the list.

D. Classification Methods

Classification was used to evaluate feature subsets. Since feature selection was carried out in order to improve classification results, the obtained data subsets with reduced dimensionality were used for classification applying different classification algorithms and the feature selection methods that were used in order to obtain the reduced data sets, which were evaluated by overall classifier accuracy. The data subsets were evaluated using single classifiers as well as average classification accuracies of groups of classifiers to avoid influence of single classification algorithms.

RIPPER (*Repeated Incremental Pruning to Produce Error Reduction*) algorithm was proposed by J. Cohen in 1995 [10] as an improvement to *IREP* (*Incremental Reduced Error Pruning*) [11] algorithm. *IREP* was created from a combination of *FOIL* (*First Order Inductive Learner*) [12] (to build the rules on the training set) and *REP* [13] (*Reduced Error Pruning*, to prune the rules on the test set). *FOIL* learns first-order rules that cover the positive rules by joining antedescents that improve gain from the rule that covers *n* features until gain of the rule does not improve, see (12):

$$I(c_i) = -log_2 \frac{n_i^{positive}}{n_i^{positive} - n_i^{negative}}. \quad (12)$$

The overall rule improvement is calculated according to (13):

$$WIG(c_{i+1}, c_i) = n_i^{positive+}\big(I(c_i) - I(c_{i-1})\big), \quad (13)$$

where $n_i^{positive+}$ is the number of instances covered by the c_i and at least one tuple of c_{i+1}.

Then the found rules are pruned using reduced error pruning (*REP*), validating them on a training set. This combination was improved in *IREP* by introducing immediate rule pre- and post-pruning. *RIPPER* takes improvements one step further by improving pruning metrics and rule optimisation. *RIPPER* algorithm is used in this study as its *Java* implementation *JRip* provided in *Weka* library [14].

FURIA (*Fuzzy Unordered Rule Induction Algorithm*) algorithm [15] extends the *RIPPER* algorithm [10], keeping its advantage – a simple and understandable rule base. One of the improvements, introduced in *FURIA*, is learning fuzzy rules instead of crisp rules induced by *RIPPER*. *FURIA* also induces an unordered rule set instead or an ordered list of rules used in *RIPPER*. *FURIA* learns rules for each class separately using one vs. all classes approach. This leads to a situation that there is not one major rule and the order of classes to learn rules is not significant. But this approach has its shortcomings – if a record is covered by rules of two classes equally, a confidence factor should be calculated. The main improvements to *RIPPER* are related to branching. But the main advantage of this algorithm is a rule stretching method, which solved the pressing problem that new records that should be classified using the induced rules might not be covered by the previously induced rules. The representation of the fuzzy rules is also different – the intervals are replaced by fuzzy intervals, named fuzzy sets, with trapeze-type membership functions [15].

C4.5 is a decision tree construction algorithm that was proposed in 1993 by J. R. Quinlan [1]. If a data set *S* is given, algorithm *C4.5* first constructs an initial decision tree using the 'divide and conquer' strategy iteratively dividing *S* into subsets *S**. If all records from *S** belong to the same class or *S** is smaller than a previously defined threshold, the subset is used to define a leaf (end node) with a class $c = \mathrm{mode}(c(S))$. If neither of these applies, a test is chosen to split the subset according to an attribute with two or more unique values. This test is represented as a root node of a subtree with branches according to all values of the test, splitting the subset into further subsets accordingly. *C4.5* uses two heuristic criteria in test evaluation:

1) *Information Gain* that decreases the total entropy of a data subset *S**, but this mostly applies to continuous attributes;
2) *Gain Ratio* that divides the *Information Gain* according to the values of the attribute (discrete, nominal attributes).

The algorithm works with both continuous and discrete attributes, and the tests are chosen accordingly. When the decision tree classifier is built using the described iterative process, the tree is pruned to avoid overfitting and classifiers that are very complex gaining little additional accuracy. The pruning is carried out according to a pessimistic pruning approach that does not require a separate validation set for pruning. A subtree is pruned if the resulting change in accuracy does not exceed one standard deviation of the error obtained with the reference tree. This process is carried out in a top-down manner and if a node is pruned, none of its descendants is tested, which results in a relatively fast pruning.

CART (*Classification and Regression Tree*) is another decision tree classifier construction method. It is based on binary recursive splits and works with continuous and nominal data. *CART* was proposed in 1984 by L. Breiman et al. [16]. The data is processed without any transformation inside the algorithm. The test *CART* uses to select optimal splits is Gini impurity.

The trees are constructed to their full size and then pruned down to a root node using *Cost-Complexity Pruning*. The pruning process analyses pruning of every internal node and their combinations. A subtree is pruned if it has a low increase in error rate, see (14):

$$\alpha = \frac{\varepsilon(pruned(T,t),S) - \varepsilon(T,S)}{|leaves(T)| - |leaves\left(pruned(T,t)\right)|}, \qquad (14)$$

where $\varepsilon(T,S)$ is the error of tree *T* in data set *S*,

$pruned(T,t)$ is the tree pruned by replacing node t with a leaf node,

$leaves$(T) is the number of leaves in tree *T*.

The *Naïve Bayes* classifier (NB in graphs and tables) uses probabilistic knowledge to assign class values [17]. It assumes that features are conditionally independent (hence the naïve approach) and predicts the most probable class according to class probabilities that are calculated for class set *C* with value *c* and feature value vector *X* with values *x* as shown in (15):

$$P(C = c|X = x) = \frac{p(C=c)p(X=x|C=c)}{p(X=x)}. \qquad (15)$$

K-nearest neighbour (KNN) classification algorithm is one of the most simple and trivial classification algorithms and it is based on instance learning [18]. This approach consists of three key elements: set with record labels, i.e., set of stored records, distance or similarity metric to calculate distance/similarity between two records and *k* value – the number of neighbours in the selected neighbourhood. *K-nearest neighbour* algorithm classifies a new record using distance/similarity metric to evaluate how close a vector *z* (the new record) is to each of the stored records. Then it uses *k* closest records to assign a class value to *z*.

If there is a data set *D* and a new record *z*, the algorithm calculates distance $d(z, x_i)$ where $x_i \in D$ and selects *k* nearest records into *Dz* to calculate the class value based on the majority vote of its neighbours, see (16):

$$y` = \underset{v}{argmax} \sum_{(x_i, y_i) \in D_z} I(v = y_i), \qquad (16)$$

where v is a class label,

y_i is a class label of the *i*-th neighbour,

I() is an induction function that returns 1 if true and 0 otherwise.

One of the most recent and perspective approaches in classification nowadays is *Support Vector Machines* (SVM) [19]. *SVM* has a strong mathematical theoretical base and needs only a dozen records for training; it does not have dimensionality-related limitations and there are many novel training approaches being developed. *Support Vector Machines* were proposed by Vapnik in 1963 [20], but the currently used version was developed by Vapnik and Cortes in 1995 [21]. *SVM* was built for binary classification where a hyperplane is used to divide the two classes as clearly as possible. This hyperplane is searched for by transforming data according to mathematical kernels and maximising the distance between groups of objects with different class labels and the hyperplane that divides them. The points on the margin of classes are called support vectors and between these margins there is the hyperplane that divides objects into class-specific groups. If there are objects belonging to a different class than the group label, these objects have smaller weights in order to give them less influence on the end result. In order to find the hyperplane with the maximum distance, the algorithm maximises the function shown in (17) based on weight vector $\vec{w}$ of record value vector x_i and constant b:

$$L_P = \frac{1}{2}\|\vec{w}\| - \sum_{i=1}^{t} \propto_i y_i(\vec{w} \cdot \vec{x_i} + b) + \\ + \sum_{i=1}^{t} \propto_i, \qquad (17)$$

where *t* is the number of records in a data set,

α is Karush-Kuhn-Tucker multiplier,

y is a class identifier (1 or −1),

L_P is a Lagrangian.

The study also applies two classification methods developed specifically for bioinformatics tasks – *FuzzyBEXA+* and *Genetic Algorithm Generated Random Forests*. These methods were evaluated and compared to classical classification methods.

The structure of fuzzy data classification algorithm *FuzzyBEXA* is based on crisp data classification algorithm BEXA [22]. *FuzzyBEXA* algorithm expands the use of definitions described in *BEXA* algorithm to their application to fuzzy data. In the case of the algorithm of classical data classification *BEXA*, the set of conjunction covered instances is considered to be all records that fit the given conjunction. In this case, a clearly defined value of a specific attribute either fits or does not fit the conjunction. In the case of fuzzy data classification algorithm *FuzzyBEXA*, the value of an attribute fits the conjunction in the scale from 0 to 1, and therefore a record can fit the conjunction with a very small membership indicator. Such a situation may be undesirable; therefore, new variables are used with "*alpha-cut*" and "*alpha-class cut*".

Variable "*alpha-cut*" (or *alpha-levelling*) (α_a) determines that all membership values of a record that are below the level of this variable value are considered 0 [23]. Thus, the instance set covered by a conjunction $X_s(c)$ can be defined as follows (see (18)):

$$X_s(c) = \{s \in S | \mu_c(S) \geq \alpha_a, \qquad (18)$$

where s is a record belonging to record set S,

c is the conjunction,

a is an attribute identifier,

$\mu_a(s)$ is a membership function,

α_a is the alpha-cut variable for attribute a.

For *BEXA* tree algorithms to function correctly, there is a necessity to divide the data into positive and negative class records. The problem is that such a division in the case of fuzzy data is not directly possible. It is explained by the fact that values of each record, which are similar to attributes, and the class of a record are not one value but rather a membership to all possible classes with a specific membership level. To solve this problem, another user-defined variable is introduced – the "*alpha-class cut*" (α_c) [23]. This variable points to the value that has to be reached by a class membership value of a record for this record to be considered a positive class instance. By using the variable "*alpha-class cut*" (α_c), it is possible to define positive (see (19), left part) and negative (see (19), right part) record sets:

$$P = \{s \in T | \mu_c(s) \geq \alpha_c\}, N = \{s \in T | \mu_c(s) < \alpha_c\}, \quad (19)$$

where s is a record belonging to test set T,

c is the corresponding class,

$\mu_c(s)$ is a membership function for record s,

α_c is the alpha-class cut value.

Before inspecting the real *BEXA* conditions in the context of *FuzzyBEXA*, it is important to note the fact that *FuzzyBEXA* algorithm does not contain the use of a specific membership function – in its place there is fuzzy data analysis. Since *FuzzyBEXA* algorithm uses fuzzy data, this algorithm does not differentiate between processing of categorical and continuous data [23].

The *FuzzyBEXA* modification (*FuzzyBEXA+*, abbreviated as FB+) used in this study was proposed by Gasparovica-Asite in [24]. This modification introduced adaptation for membership function construction using one of the three methods – *k-means*, *x-means* based clustering algorithm or *triangle* membership functions. This modification also added rule stretching and rule fuzzification approaches in order to make more universal rule bases, which are particularly useful in bioinformatics problems.

The *Genetic Algorithm based Random Forest* algorithm was proposed by Polaka [25] in order to improve accuracy of tree-based classifiers while keeping the transparent and comprehensive model representations. The algorithm generates fixed size (in levels) binary decision trees based on *Gain Ratio* attribute and cut-off point evaluation metric.

III. Results

The experimental evaluation was carried out using five data sets, eight different attribute selection and evaluation method combinations. Several attribute subset sizes were chosen for attribute ranking approaches – the attribute sets were reduced to 200, 100, 50, 20 and 10 attributes. The initial data sets with full attribute sets were used as a benchmark. Data set evaluation was carried out using eight classification algorithms with different approaches – rule based *FURIA* and *JRIP*, decision tree induction using *C4.5* and *CART*, as well as *KNN*, *SVM* and two methods previously proposed by the authors for bioinformatics data analysis – *FB+* and *GARF*. This resulted in an experiment series with 1160 experiment runs, each of which was executed using 10-fold cross-validation.

The average classification accuracies of all methods are shown in Table I. The evaluation is based on average values to assess the impact of attribute set reduction and exclude bias that can be introduced by specific classification approaches and algorithms. Only the best result for melanoma data set (74.27 % overall classifier accuracy) was achieved using the full set, although the second best result was worse only by 0.19 % (74.08 % overall accuracy) in the data set with 100 attributes, which could be considered equal (the difference was not significant). In gastric cancer, gastro-intestinal disease and prostate cancer data sets the best results were obtained using data sets with 50–100 attributes, which was less than 10 % of the initial attribute set. The breast cancer data set with the decreased attribute set showed the best adaptation to the most strict attribute subset reduction – the best results were obtained in 10 and 20 attribute subsets. This could be explained by heavy noise caused by other attributes and/or few very definite biomarkers that pointed to breast cancer. The overall accuracies varied a lot from one data set to another. This was due to different disease specifics. The worst results were in gastro-intestinal disease data, which might be caused by mixing everal different diagnoses as the positive group, which might have different expressions in autoantibodies. The maximum difference between the best average and the worst average value was ~4 % in melanoma data set, while the minimum difference was <1 % in breast cancer data set. This shows how little impact of decreasing the attribute subset removes more than 90 % of the attributes.

TABLE I

Average Classification Accuracies in the Reduced Data Sets

#Attributes	BrCa	GaCa	GIS	Mel	PrCa
1230	92.33	58.27	57.07	74.27	80.31
10	92.99	58.07	56.09	70.89	79.11
20	93.02	58.07	57.87	72.59	80.41
50	92.90	59.19	57.06	72.38	81.05
100	92.56	58.42	58.33	74.08	79.72
200	92.26	57.82	56.16	73.65	79.03

Comparison of the two main studied algorithms, *FB+* and *GARF*, with the minimum, maximum and average accuracies of the other algorithms used in this study is shown in Table II. In 16 experiments out of 29 with breast cancer data set, gastric cancer data set and gastrointestinal disease data set the best results were achieved using *GARF* algorithm, but in experiments with melanoma and prostate cancer data sets the best results were acquired using *FB+* algorithm, which showed that algorithms specifically developed with bioinformatics data in mind were more suitable to work with this type of data. It is especially obvious in experiments with reduced data sets with 50, 100 and 200 attribute subsets.

TABLE II

CLASSIFICATION ACCURACIES OF FB+ AND GARF

	BrCa	GaCa	GIS	Mel	PrCa
Maximum of all other methods	96.43	69.51	67.97	90.96	91.30
Average of all other methods	92.74	56.74	56.00	71.37	78.91
Minimum of all other methods	83.45	46.65	44.84	44.61	71.98
FuzzyBEXA+	92.26	64.94	63.70	88.63	92.75
Genetic Algorithm generated Random Forests	95.26	78.95	78.36	87.19	93.74

If *GARF* results are analysed in detail for all five data sets, the algorithm shows better accuracy for breast cancer data set when the number of attributes is 20 or 50, for gastric cancer data set – when there are 50 attributes, for gastrointestinal disease data set – with 20 or 50 attributes, for melanoma data set – with 10 or 50 attributes and for prostate cancer data set – with 50 attributes.

The best attribute selection and evaluation methods for *GARF* algorithm are the following: *Consistency Subset Evaluator* and *Re-ranking Search*, *ReliefF* (top 20 attributes) and *Information Gain* (top 50 attributes). The most suitable methods for *FB+* algorithm are *OneR* (top 50 attributes) and *ReliefF* (top 50 attributes).

If one looks at algorithms that have one of the top 3 results in each case (each of the data sets with each attribute selection method and the full data set, totalling in 29 cases), the best classification accuracies are obtained using *FURIA* (22 cases), *GARF* (19 cases), *FB+* and *SVM* (14 cases both).

IV. CONCLUSION

Dimensionality reduction by selecting a small subset of the attribute set (decreasing the attribute set by more than 90 %) does not have a significant influence on classification accuracy and, therefore, selecting a small subset of biomarkers will not have a significant impact on diagnostic accuracy of the resulting test. The accuracies in reduced data sets were even slightly higher than in full data sets in four out of five data sets.

The algorithms particularly developed to work with bioinformatics-specific data sets (*GARF* and *FB+*) showed better results than the regular methods in most cases. These methods build easily comprehensible models that can be interpreted by medical experts. Although these methods can be slower due to more computations, the task does not ask for real time computations and the speed is a small drawback.

REFERENCES

[1] H. Liu and R. Setiono, "Chi2: Feature selection and discretization of numeric attributes," in *Proc. IEEE 7th Int. Conf. on Tools with Artificial Intelligence*, pp. 338–391, 1995.

[2] J.R. Quinlan, *C4.5: Programs for Machine Learning*. – San Mateo, CA: Morgan Kaufmann Publishers, 1993, p. 302.

[3] R.C. Holte. "Very simple classification rules perform well on most commonly used datasets," *Machine Learning*, vol. 11, pp. 63–91, 1993. http://dx.doi.org/10.1023/A:1022631118932

[4] I. Kononenko, "Estimating Attributes: Analysis and Extensions of RELIEF," in *European Conf. on Machine Learning*, pp. 171–182, 1994. http://dx.doi.org/10.1007/3-540-57868-4_57

[5] M. A. Hall, "Correlation-based Feature Subset Selection for Machine Learning," Dissertation at University of Waikato (Hamilton, New Zealand), 1998. 198 p.

[6] C.P. Tan, K.S. Lim, W.K. Lai, "Multi-Dimensional Features Reduction of Consistency Subset Evaluator on Unsupervised Expectation Maximization Classifier for Imaging Surveillance Application," *Int. J. of Image Processing*, 2–1, pp. 18–26, 2008.

[7] H. Liu, R. Setiono, "A probabilistic approach to feature selection – a filter solution," in *Proc. of the 13th Int. Conf. on Machine Learning (ICML'96)*, Bari, Italy, July 3–6, *1996*. San Mateo: Morgan Kaufmann Pub., 1996, pp. 319–327.

[8] L. Yu, H. Liu, "Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution," in *Proc. of the Twentieth Int. Conf. on Machine Learning*, pp. 856–863, 2003.

[9] P. Langley, "Selection of relevant features in machine learning," in *Proc. of the AAAI Fall Symposium on Relevance*. New Orleans, Louisiana, USA, Nov. 4–6, 1994. New Orleans: AAAI Press, pp. 140–144, 1994.

[10] W. W. Cohen, "Fast Effective Rule Induction," in *Machine Learning: Proc. of the 12th Int. Conf. (ML'95)*, Morgan Kaufmann, 1995, pp. 115–123. http://dx.doi.org/10.1016/b978-1-55860-377-6.50023-2

[11] J. F¨urnkranz and G. Widmer,"Incremental reduced error pruning," in *W.W. Cohen and H. Hirsh, editors, Proc. of the 11th Int. Conference on Machine Learning*, pp. 70–77. Morgan Kaufmann, 1994.

[12] R. Quinlan R. "Learning logical definitions from relations," *Machine Learning*, vol. 5, no. 3, 1990.

[13] R. Quinlan R. "Simplifying decision trees," *International Journal of Man-Machine Studies*, vol. 27, pp. 221–234, 1987. http://dx.doi.org/10.1016/S0020-7373(87)80053-6

[14] "The WEKA Data Mining Software: An Update", M. Hall, E. Frank, G. Holmes, et al., *ACM SIGKDD explorations newsletter*, 2009, vol. 11, issue 1, pp. 10–18.

[15] J. Hühn, E. Hüllermeier E. "FURIA: An Algorithm for Unordered Fuzzy Rule Induction", *Data Mining and Knowledge Discovery*, 2009, vol. 19, no. 3, pp. 293–319. http://dx.doi.org/10.1007/s10618-009-0131-8

[16] "Classification and Regression Trees", L. Breiman, J.H. Fridman, L.A. Olshen et al. –Washington, DC: Chapman & Hall / CRC, 1984, 358 p. (*Series: Wadsworth Statistics/Probability*).

[17] *Data Mining and Knowledge Discovery Handbook* / Ed. O. Maimon, L. Rokach. Berlin Heidelberg: Springer, 2010, 1285 p.

[18] D. W. Aha, D. Kibler, and M.K. Albert, "Instance-Based Learning Algorithms", *Mach. Learn*. vol. 6, issue 1, Jan. 1991, pp. 37–66. http://dx.doi.org/10.1023/A:1022689900470

[19] D. Meyer, "Support Vector Machines. The Interface to libsvm in package" e1071. Online-Documentation of the package e1071 for R. – Wien: Technische Universität Wien, 2001. pp. 1–8. Available from: http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=BCDB6D08469CF19CF416EAADC044C6B3?doi=10.1.1.151.5271&rep=rep1&type=pdf.

[20] V. Vapnik, and A. Lerner, "Pattern recognition using generalized portrait method," *Automation and Remote Control*. 24, pp. 774–780, 1963.

[21] C. Cortes, V. Vapnik, "Support-Vector Network," *Machine Learning*, 1995, vol. 20, pp. 273–297. http://dx.doi.org/10.1007/BF00994018

[22] H. Theron, I. Cloete, "BEXA: A Covering Algorithm for Learning Propositional Concept Descriptions," *Machine Learning*. 1996. Vol. 24, Issue 1. pp. 5–40. http://dx.doi.org/10.1007/BF00117830

[23] J. van Zyl, "Fuzzy Set Covering as a New Paradigm for the Induction of Fuzzy Classification Rules," PhD thesis. – Mannheim: University of Mannheim, 2007. 263 p.

[24] M. Gasparovica-Asite, "Fuzzy classification methodology for processing and analyzing bioinformatics data," PhD thesis. Riga: Riga Technical University, 2015. 160 p., in press.

[25] I. Poļaka, A. Borisovs, "Genethic Algorithm and Tree Based Classification in Bioinformatics," in *European Conference on Data Analysis 2013: Book of Abstracts*, Luxembourg, Luxembourg, July 10–12, 2013. Luxembourg: 2013, pp. 107–107. ISBN 9782879711058.

Madara Gasparovica-Asite received her diploma of *Mg. sc. ing.* in Information Technology from Riga Technical University in 2010. Now she is a Doctoral Student at the study programme "Information Technology", Riga Technical University.

Since 2008 she has worked as a Senior Laboratory Assistant at Riga Technical University, and since 2010 she has been working as a Researcher at the Department of Modelling and Simulation, the Institute of Information Technology. Previous publications: Gasparovica M., Novoselova N., Aleksejeva L. Using Fuzzy Logic to Solve Bioinformatics Tasks. Proceedings of Riga Technical University. Issue 5, Computer Science. Information Technology and Management Science, Vol. 44, 2010, pp. 99–105. Gasparoviča M., Aleksejeva L. Using Fuzzy Unordered Rule Induction Algorithm for Cancer Data Classification. Proceedings of the 17th International Conference on Soft Computing, MENDEL 2011, Czech Republic, Brno, June 15–17, 2011, pp. 141–147.

Her interests include decision support systems, data mining tasks and modular rules. She is a member of IEEE.

Address: 1 Kalku Str., LV-1010, Riga, Latvia.

E-mail: madara.gasparovica-asite@rtu.lv

Inese Polaka, *Dr. sc. ing.*, is a Lecturer at the Institute of Information Technology of Riga Technical University (Latvia) and Leading Researcher at the Faculty of Medicine, University of Latvia. Main research interests include data mining, machine learning, classifiers, evolutionary algorithms and their applications, as well as bioinformatics and biostatistics.

Address: 1 Kalku Str., Riga, LV-1658, phone: +371 67089530.

E-mail: inese.polaka@rtu.lv

Ludmila Aleksejeva received her *Dr. sc. ing.* degree from Riga Technical University in 1998. She is an Associate Professor at the Department of Modelling and Simulation, Riga Technical University. Her research interests include decision making techniques and decision support system design principles, as well as data mining methods and tasks, and especially collaboration and cooperation of the mentioned techniques.

The most important previous publications: Gasparoviča M., Novoselova N., Aleksejeva L. Using Fuzzy Logic to Solve Bioinformatics Tasks. Proceedings of Riga Technical University. Issue 5, Computer Science. Information Technology and Management Science, Vol. 44, 2010, pp. 99–105. Gasparoviča M., Aleksejeva L., Tuleiko I. Finding Membership Functions for Bioinformatics Data. Proceedings of the 17th International Conference on Soft Computing, MENDEL 2011, Czech, Brno, June 15–17, 2011. pp. 133–140. Aleksejeva L., Užga-Rebrovs O., Borisovs A. Fuzzy Classification and Clustering. Textbook. Riga: RTU Press, 2012. 248 p.

Address: 1 Kalku Str., LV-1010, Riga, Latvia.

E-mail: ludmila.aleksejeva_1@rtu.lv