Evolutionary weighting of image features for diagnosing of CNS tumors

Maciej Komosinski   Krzysztof Krawiec

Institute of Computing Science, Poznan University of Technology
Piotrowo 2, 60-965 Poznan, Poland

{maciej.komosinski, krzysztof.krawiec}@cs.put.poznan.pl

Abstract. This paper concerns an application of evolutionary feature weighting for diagnosis support in neuropathology. The original data in the classification task are the microscopic images of ten classes of Central Nervous System (CNS) neuroepithelial tumors. These images are segmented and described by the features characterizing regions resulting from the segmentation process. The final features are in part irrelevant. Thus, we employ an evolutionary algorithm to reduce the number of irrelevant attributes, using the predictive accuracy of a classifier ("wrapper" approach) as an individual's fitness measure. The novelty of our approach consists in application of evolutionary algorithm for feature weighting, not only for feature selection. The obtained weights give quantitative information about the relative importance of features. The results of computational experiments show a significant improvement of predictive accuracy of the evolutionarily found feature sets with respect to the original feature set.

1. Introduction

Any application of computer support in medicine, regardless of methodology and techniques used, requires some domain knowledge. Artificial Intelligence (AI) offers a wide scope of various knowledge representation languages and different techniques for knowledge acquisition, processing, updating, and retrieval. Specifically, as the volume of knowledge related to particular topics grows at enormous speed with the progress in medicine, there is a vital need for methods, which would be able to acquire domain knowledge in an automated or semi-automated way [14].

One of the branches of AI which tries to cope with this “knowledge acquisition bottleneck” problem is Machine Learning (ML). For medical diagnosing purposes, the most natural ML setting is *classification*, where the decision classes correspond to different diagnoses. Such an approach is popular in medical applications, for instance [27], [10], [6]. In what is probably the most often applied paradigm of ML called *learning from examples*, the goal of the learning system is to acquire knowledge directly form a set of solved instances of the given problem, called *training set* (e.g., patient records with assigned diagnosis). That process is often referred to as *induction*, because the learner has to go beyond the training data set and generalize the knowledge being acquired to perform well when faced with new, previously unseen problems [32].

The main problem concerning usage of classification systems is the definition of *features* describing examples, which allow obtaining good classification accuracy. In ML applications for medical diagnosing support, the original examples are usually described by clinical data and/or other forms of information, such as images, time series, etc. Especially in the latter case, when the data is represented in a non-vector form, it is easy to suggest (or even automatically generate) many features. However, only part of them is usually relevant for the given classification task. The remaining features are usually *redundant* (i.e., their values depend on the values of relevant features) or *irrelevant*, being noise from the viewpoint of the diagnostic problem in question [8]. Such features can significantly deteriorate the predictive accuracy of the classifier, especially when it does not perform an inherent feature evaluation (like, for instance, decision tree inducers; see [36]). Thus, there is a need for an additional pre-processing step of *feature selection*, which would get rid of the redundant and irrelevant features.

Feature selection became a popular topic in Machine Learning. There are various approaches described in the literature, which may be grouped with respect to the search strategy and to the function used for feature set evaluation [8]. In this paper, we would like to present a variant of an evolutionary approach to feature selection and, specifically, to introduce the idea of *evolutionary weighting of features*. We tested the proposed method on

The final version of this paper appeared in *Artificial Intelligence in Medicine*, volume 19, issue 1, 2000, pp. 25–38, https://doi.org/10.1016/S0933-3657(99)00048-2.

a complex medical task of diagnosing the neuroepithelial subgroup of CNS tumors based on the microscopic images of histological sections. The results of computational experiments prove the usefulness of the proposed method and show that our idea of feature weighting significantly outperforms the ordinary genetic feature selection as far as the classification accuracy of the selected description is concerned.

The paper is organized as follows. In the next section we describe briefly basic notions and problems related to feature selection in a machine learning setting. Section 3 presents the evolutionary approach to feature selection and introduces a novel method of feature weighting. In Section 4, we briefly describe the considered data sets of histological images, which are subject to feature extraction described in Section 5. Section 6 concerns the framework of computational experiments, and Section 7 presents the final results and groups conclusions.

2. Feature selection for classification

As mentioned in the introduction, the goal of feature selection (FS) is to get rid of redundant and irrelevant features. In the classification setting, this task may be defined more precisely as follows: given a training set (labeled instances of the problem) described by a set of features *F*, find a subset of features $F' \subseteq F$, which gives the highest classification accuracy on the testing set (unlabeled instances).

Following the typical view of FS algorithms [3], it is convenient to treat FS as a search problem, where each state in the search space represents a subset of features. Then, one can define the basic components of a general FS method: a *search algorithm* – which looks through the space of feature subsets; an *evaluation function* – used to evaluate examined subsets of features; and a *classifier* – which is constructed based on a final subset of features. These elements can be integrated in two basic ways forming *filter* or *wrapper* model [22]. In the filter model, features are selected as a pre-processing step before a classifier is used. Features are selected (i.e., filtered) depending on properties of the data itself, independently from the learning algorithm used in a classifier. In the wrapper model, the search algorithm conducts a search for a good subset of features using a classifier itself as the evaluation function. Evaluation is usually done by estimating predictive accuracy in a *cross-validation* test. In the literature, a broad range of feature selection algorithms have been described [3], [22], [34], [18]. For an extensive review, see [8].

There are at least two non-trivial problems which emerge when considering the FS task formulated in the classification setting. The first one is computational complexity. As there are $2^{|F|}$ subsets of the feature set *F*, an exact algorithm searching through all subsets has an exponential complexity. This is why various heuristics are often used to reduce the number of considered subsets. The second problem is a consequence of the classification setting. The feature selection algorithm has to take into account not only the descriptive properties of a particular feature set (related to the training set), but also its inductive (predictive) ability, which is usually verified on the testing set. In consequence, the above-mentioned wrapper approach is often reported to outperform other approaches, as it incorporates the *inductive bias* of the classifier into the evaluation function [24].

It is worth mentioning that feature selection can be treated as the simplest method belonging to a *constructive induction* methodology – regarded as an approach to supporting automatic, problem-oriented transformation of representation space to facilitate learning [31], [28], [29], [40]. Usually, an improvement of predictive accuracy in constructive induction is obtained by construction of new features, modification of existing ones and reduction of irrelevant ones. Most methods use a specific technique within one basic computational method. Basic methods are classified as data-driven, hypothesis-driven and knowledge-driven [40].

3. Evolutionary selection and weighting of features

3.1. Evolutionary techniques in optimization

Evolutionary techniques [11], [30] are one of many existing methods used for solving difficult (NP-hard) optimization tasks. An example of evolutionary optimization techniques is a genetic algorithm (GA), which has been successfully applied to solving many various problems [9]. This search method has proved efficient and robust. GAs simulate natural mechanisms, like selection and inheritance, while processing a *population* of *individuals* – each individual representing one solution. The algorithm is iterative – in each *generation* the best individuals are promoted (*selection*), information between them is exchanged (*crossover*) and *mutation* takes place. These operations are conducted in order to improve fitness of future populations, and are usually non-deterministic – they occur with certain probabilities. Despite this non-determinism, GAs manage to find the neighborhood of optimal solutions where other search algorithms cannot be applied or do not work effectively.

One of the inconveniences in GAs is a great number of parameters, which have to be adjusted, and a need to choose between many possible techniques used within the GA. There are many interesting ideas and modifications and it is hard to create universal rules, which would determine the optimal architecture and parameters of a GA system for any task. However, the growing number of theoretical studies and applications of GAs makes them clearer; some aspects have already been analyzed and described [11]. Further research on these topics will make GA systems easier to design and use for the non-professionals.

3.2. Evolutionary algorithms in feature selection

Among many feature selection techniques [39], [8], evolutionary techniques are not especially popular ones. There have been some experiments with GAs [39], [5], [35], [38], [41], but more research is conducted on the improvement of other existing heuristic methods than on devising new ones. Most of the existing feature selection techniques seem to be sophisticated, and often carry a large number of assumptions and parameters.

The genetic algorithm is a good tool for feature selection problems, as it can search the space of solutions in a global, non-greedy manner, discovering important features and passing them to future generations. Each individual represents a subset of features. The genotypes are binary strings, where “1” means choosing a feature, and 0 – omission of a feature. Usually the evolutionary process is aimed at finding individuals, which yield the highest classification accuracy in the given problem.

When the only evaluation criterion is the predictive accuracy of individuals, there is no pressure to reduce the number of irrelevant features. This implies that only features which decrease the accuracy will be ignored (the corresponding genes will be rewarded for being zero). The features which are irrelevant and do not affect the accuracy will be selected or not on a random basis.

The disadvantage of using GAs in feature selection tasks is the computational complexity of this optimization method. When the “wrapper” model is used, in each generation a population of individuals (subsets of features) is to be evaluated by calling the classifier and obtaining the classification accuracy. This can be a very time-consuming process, and therefore standard GAs seem to be appropriate for *off-line* feature selection. However, it is usually possible to shorten the evolutionary run, diminish the size of the population and thus obtain similar results faster, but it is never known if the final solution is not a local optimum (this problem is present in many feature selection algorithms).

3.3. Evolutionary feature weighting

Genetic algorithms have an important advantage: they process a set of solutions, and so they are a more global method, compared to other optimization techniques. This is why it may be worthwhile to employ evolutionary methods in the constructive induction process [25]. In this paper we use GAs not only for feature selection, but also for adjusting weights of attributes in order to improve the classification accuracy. Both feature selection and their weighting can be regarded as simple methods of constructive induction.

When using a k-Nearest Neighbor (k-NN) classifier [7], weights of attributes may be used to reflect their importance. The distance between two examples is usually the sum of differences on all the attributes describing these examples. Therefore, when a difference in a given attribute has a high weight (i.e., is multiplied by a high value), this attribute becomes important in the process of discriminating examples. There are some classification algorithms that are able to adjust weights in order to obtain better classification accuracy (for instance IBL4 [1], [2]). However, it may be important to employ here some global technique which adjusts all the weights simultaneously, and uses the classification accuracy itself as the evaluation criterion of the weight vector.

This idea was used in devising the method presented in this paper. Genetic algorithms process here a population of sets of weights, each weight corresponding to one attribute. Such individuals are evaluated, and their fitness is defined as the classification accuracy they yield. Thus the only feedback used in the selection phase of the evolutionary process is the accuracy of individuals. The classification accuracy is a complex, “global” evaluation criterion, because all the properties of single attributes are encompassed in this value.

Weights are not adjusted on the one-by-one basis, but the set of optimal weights emerges simultaneously, during the evolutionary run, respecting mutual dependencies and regarding relationships between attributes' weights.

The genotype representation needed for evolutionary feature weighting is quite similar to the standard GA zero-one representation. The genotype length is still constant. The difference is that each gene is a number that defines a weight of an attribute. There is a need to define appropriate genetic operators, crossover and mutation. There are a few possibilities here. The crossover may be any of the well-known operators, like one- or two-point crossover, or uniform crossover, but another reasonable choice would be the average operator. The mutation operator can be a simple "random" operator, but also a "creep" one [9], or a modified random (changing 0 to a random value or a non-zero value to zero, as used in the experiments described in this paper).

In general, numbers-weights in the genotype can be positive and negative. Positive weights mean that the larger the difference in the attribute, the larger the dissimilarity of two considered examples. Negative weight corresponds to the opposite situation: the larger the difference, the smaller the dissimilarity. Zero weight means that the corresponding attribute may be ignored (treated as irrelevant) during the computation of total similarity of examples. Similar to feature selection, when there is no penalty for the number of features used, weights of features which do not affect the accuracy (fitness of individuals) will be random, not necessarily zero.

The weight-adjusting task is more general than feature selection. In fact, feature selection is a special case of the weighting task, where weights can be zero or one only [41]. The search space for feature selection is $2^{|F|}$, and for weight adjusting it is $w^{|F|}$, where *w* is the number of possible weight values, and *F* denotes the set of features.

4. Data sets

The data sets used in our experiments represent one of the main groups of brain tumors, i.e., *tumors of neuroepithelial tissue*, which covers about 50% of the most frequent clinical cases. Specifically, we tested the proposed approach on two subgroups of these tumors [23], which have different clinical origin. The first data set, denoted hereafter by *A*, concerns the classification of astrocytic tumors, and the second (*B*) covers the most often occurring glial tumors. In both subgroups, there are six types of tumors represented. The decision classes represented in the data sets are shown in Table 1.

Each class in our data set is represented by 5 clinical cases, and for each case 10 images have been acquired from microscopic section. Thus, the considered data sets contain 6×5×10 = 300 images each. Each image has 512×512 pixels, 256 levels of intensity for each base color component (RGB) and shows a fragment of particular microscopic section magnified 200 times. The images were collected and classified by Janusz Szymaś, medical senior expert from the Department of Pathology, University School of Medicine in Poznań.

A	*B*
Astrocytoma anaplasticum	Astrocytoma fibrillare
Astrocytoma fibrillare	Oligodendroglioma isomorphum
Astrocytoma gemistocyticum	Ependymoma
Astrocytoma pilocyticum	Choroid plexus papilloma
Astrocytoma protoplasmaticum	Glioblastoma multiforme
Glioblastoma multiforme	Medulloblastoma

Table 1. Two groups of considered neuroepithelial tumors.

5. Feature extraction

In our previous experiments [17], [19], we used mainly various conventional global image features, such as RGB histograms and co-occurrence matrix [13], to discriminate the classes of tumors described in the previous section. However, none of the applied conventional techniques gave a satisfactory accuracy of classification in the cross-validation test. We performed also some experiments with other more domain-specific image features [20], [21]. Unfortunately, no significant improvement was observed in comparison to the former features.

These unsatisfactory results were mainly due to the high complexity of the problem, reflected by high intra-class and low inter-class diversification of images. Our experience led us to consider more abstract image features. Specifically, we decided to perform *image segmentation* to obtain more qualitative description of the original image. In general, the goal of image segmentation is to decompose an image into *regions* (sets of adjacent pixels) of uniform characteristics, which may have some interpretation in terms of the domain of application (e.g., cell nuclei in microscopic histological section). From the scope of various approaches to image segmentation described in the literature [33], [37], we chose the *region growing* technique. Though it has relatively high computational complexity, it usually outperforms other methods as far as the fidelity of the resulting segmented image compared with the original one is concerned. Region growing starts with each image pixel being a separate region. Then, the algorithm performs bottom-up joining of adjacent regions. If a result of a merger of a pair of adjacent regions does not fulfill the *region uniformity condition,* a backtracking is performed and the considered pair of regions is restored. That process is repeated until no pair of regions can be merged.

The main problem with the segmentation is the choice of an appropriate condition for region uniformity. Usually, some *uniformity index u* is introduced together with an experimentally adjusted threshold *T* imposed on it. Then, a region *r* is uniform if $u(r) \le T$. The uniformity index *u* reflects usually the intensity profile of a region, being for instance the variance of the intensity values. Specifically, the uniformity index based exclusively on the color characteristics of the region is used. However, as we focus on stained microscopic images, we would like to make the results independent from the staining and lighting variations, which are inherent to the section preparation and image acquisition. Thus, instead of using the conventional Red-Green-Blue color representation space, we switched to the Hue-Saturation-Intensity (HSI) space [12] and discarded Saturation and Intensity, taking into account the Hue component only. Some preliminary experiments showed that such an approach ensures better fidelity of the segmented image with respect to the original one when compared to the RGB color representation.

The uniformity condition for a given region *r* is formulated as

$$\max_{p\in r}\left[Hue(p) \div \overline{Hue(r)}\right] \le T \quad , \tag{1}$$

where $Hue(p)$ denotes the value of the Hue for the pixel *p*, $\overline{Hue(r)}$ is the mean value of $Hue(p)$ for all pixels *p* in region *r*, and ÷ is the cyclic symmetric difference operator (since *Hue* is defined on a cyclic scale [0,255]). The process of region merging is carried out with gradually increasing value of $T = [1, T_{\max}]$, where $T_{\max}$ is a user-adjusted threshold. For each value of *T*, all pairs of adjacent regions fulfilling the uniformity condition are merged. As a consequence, only very similar adjacent regions are joined in the first iterations of the algorithm, whereas more dissimilarity is allowed when merging regions at further stages. In the experiments reported below, $T_{\max} = 12$ has been used.

The application of the segmentation algorithm to microscopic images of CNS tumors gave in average ca. 13,000 regions per image, which is a significant reduction when compared to over a quarter of million (512×512) pixels in the original image. However, as we are interested in a standard machine learning setting, which requires a constant length vector of relatively few features, there is still a need for further compression of the description obtained in such a way. For this purpose, we decided to classify image regions into categories, according to their mutual similarity.

The definition of region prototypes is based on the knowledge resulting from the image segmentation process. For this purpose, five images were randomly selected from each decision class. Their descriptions resulting from segmentation process have been collected, building a database of over 340,000 region records, each described by four features: region area and mean values of hue, saturation, and intensity. Then we performed a *cluster analysis* in the space spanned over these features to find the centroids of the most characteristic regions. As we intended to obtain univocal and stable region prototypes, we did not use the popular k-means technique [26], because it is known to depend heavily on the initial cluster assignment. Instead of this, a technique was applied, which is exact in the sense that it does not depend on the starting placement of cluster centroids. However, due to a large number of objects being analyzed, it was inconvenient to use the simple exact method of bottom-up hierarchical analysis, as it requires computation of all the pairwise object distances.

Instead of this, we developed a complementary top-down approach of hierarchical analysis. The proposed technique starts from one cluster containing all considered objects. Then, it recursively divides each cluster into two sub-clusters using the Principal Component Analysis [15], building in such a way a binary *hierarchy* (tree) *of clusters*. The method uses the best-first strategy, splitting always the tree nodes with the highest intra-node variance at the moment. The process of partitioning continues until no node with variance greater than a user-defined threshold can be found.

After preliminary experiments with different parameter adjustments, we decided to use a tree composed of 69 nodes, 35 of which were leaves. This tree has been used in the last stage of feature extraction, which consisted in describing the segmentation result for a particular image in terms of these 69 categories of regions. The approach used is somehow similar to the querying of decision trees [36]. Each region of the image "passes" through the tree, starting from the root and choosing always the more similar (in terms of the four features mentioned above) child node. When processing regions from a particular image, each tree node counts the regions passing through it. After processing all the regions from an image, a final feature vector is created based on the values of region counters of particular nodes. Specifically, each tree node yields two attributes: the absolute and the relative number of regions (the second one being the fraction of the counter and the total number of regions in a particular image). Thus, the final feature vector used in our experiments consists of 69×2=138 attributes.

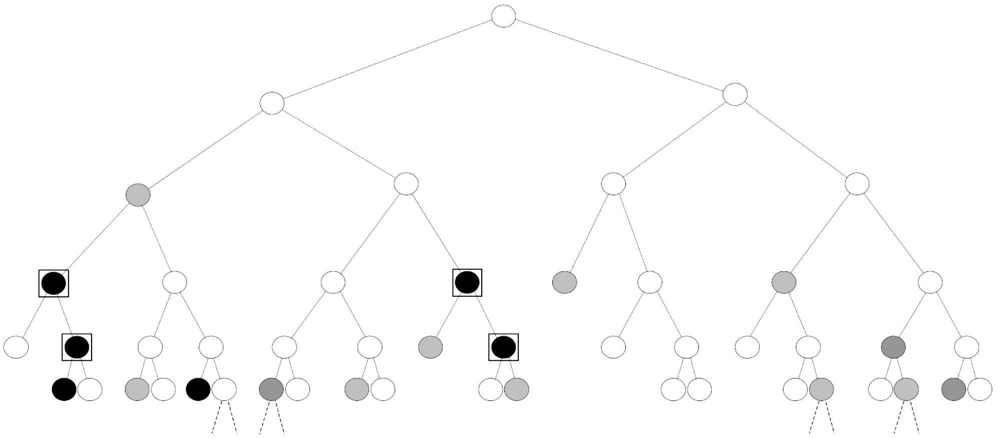

Fig. 1. Hierarchy of region types used for region classification (fragment). Node filling reflects the final weight assigned by GA weighting (○ ≡ 0, ● ≡ 9).

Figure 1 shows the main part of the region classification tree used in the computational experiments described in the next section. Note that due to the best-first strategy the tree is not balanced.

6. Computational experiments

In the wrapper model the classifier is used to evaluate each individual (feature subset). As in GA a population of individuals is evaluated in each generation, the classifier should be quick and simple. Furthermore, as we want the selection and weighting to be performed by a GA, the classifier itself should not remove redundant and irrelevant features nor adjust their importance. This is why a 1-Nearest Neighbor classifier [7] was used.

We carried out separate experiments with feature selection and feature weighting for the two six-class data sets (denoted in Section 4 as *A* and *B*). For each example (image), a feature vector was computed by means of the procedure described in Section 5. The data sets were normalized, as required by the k-NN classifier. To achieve more stable and certain evaluation of individuals, cross-validation tests in the wrapper method were repeated 10 times and the mean value was computed.

In our genetic algorithm, each individual is represented by a genotype string of digits. In the feature selection task, the genes are binary: '1' gene means that the corresponding attribute is chosen, '0' means that it is discarded. In a weight-adjusting task, each gene can be a digit '0' to '9', which is the integer (non-negative) weight of the corresponding attribute. During the run of GA, the features and their weights that best distinguish all the classes are searched for.

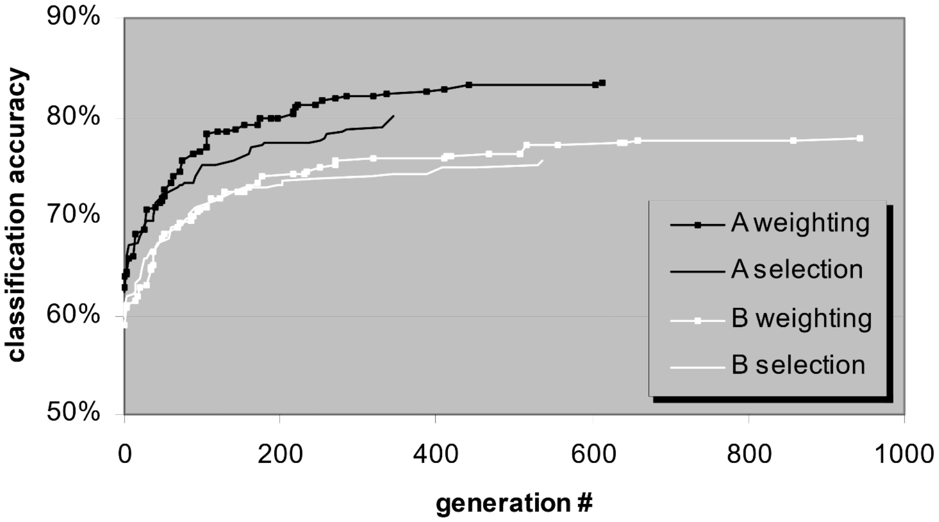

Fig. 2. Evolution of subsets of features and their weights (best-so-far accuracy is shown).

Most of the genetic algorithm's parameters were identical for all runs and based on our previous experience. Population size was constant (100 individuals). Selection was carried out according to the remainder with repetitions rule [11]. The evolutionary run was terminated after 200 non-improving generations.

Crossover probability was 0.8. Uniform crossover was used. For the feature selection task, mutation was a simple bit-flipping (0 to 1 or 1 to 0). For feature weighting, mutation turned a gene to 0 when it was greater than 0, or set it randomly to a 1-9 digit when it was 0. The mutation probability was equal to 1/(population size) [11], which amounts to 0.01. As we do not expect convergence to a single best solution, we keep track of the best individual found so far during the evolution. This is a standard *off-line* process.

Standard deviation (sigma) truncation scaling was used. The scaling coefficient decreased non-linearly from 5 in the beginning to 1 at the end of the evolution. Non-linear decrease yields a shorter low-pressure period and a longer high-pressure period.

Figure 2 shows the evolutionary runs for both data sets and evolutionary feature selection and weighting. As best-so-far fitness (classification accuracy) is shown, only improvements can be seen.

7. Results and conclusions

The experiments we have performed show the ability of evolutionary algorithms to select and weight features in order to improve the predictive accuracy of a machine learning classifier. As is shown in Table 2, the accuracy of classification yielded by the reduced description for both feature selection and weighting is much better than that of the original set of features. The best achieved classification accuracy exceeded 80%, which is reported to be the minimum accuracy needed for medical diagnosing [4]. Moreover, the proposed feature weighting outperforms slightly the ordinary feature selection as far as the classification accuracy is concerned (it is statistically significant at the confidence level 0.01), while reducing the comparable number of features. In general, as feature selection may be considered a special case of feature weighting, weighting should always yield classification accuracy at least as good as feature selection. This principle was also fulfilled in all the previous experiments with various data sets [16]. We suppose that even better improvements of predictive accuracy could be achieved with continuous positive and negative weights.

Data set	Classification accuracy [%]			# of features
	All features	Selection	Weighting	All / FS / FW
A: astrocytic tumors	57.50 ± 1.15	80.00 ± 1.22	83.43 ± 1.93	138 / 45 / 53
B: glial tumors	54.73 ± 2.32	75.70 ± 1.64	77.83 ± 2.03	138 / 49 / 62

Table 2. Classification accuracy and the final number of features for evolutionary feature selection (FS) and feature weighting (FW)

The performance of the genetic algorithm could be improved in terms of design, parameters, etc. However, the point was to compare feature selection and feature weighting and in both experiments, similar genetic algorithms were used to find best-classifying solutions.

It is interesting that in all experiments the number of attributes used was significantly reduced during the evolutionary run, although there was no penalty for the number of features in the evaluation function. In most cases, only about 1/3 of the original features were employed in classification (see Table 2 for details). Such considerable reduction may imply that many features were not only irrelevant, but lowered the predictive accuracy.

Genetic feature selection and weighting seem to co-operate well with the method used for feature extraction. They constitute useful tools for acquisition of domain knowledge, as they allow interpreting the results in terms of the application domain. One can make use of region types, which have been recognized as useful by GA selection or weighting. In the hierarchy shown in Figure 1, the tree nodes corresponding to such region types have been marked with filled circles (weighting on data set *A*). Note that this weight assignment confirms a common-sense conviction that the most general region types (corresponding to the tree nodes near the root) are not as useful for discrimination between decision classes as those more specific (placed deeper in the tree). Moreover, provided the data describing region types and their weights, it is also possible to interpret the results in a pictorial way. For instance, Figure 3 shows a few representatives of well-discriminating region types on an exemplary tumor image. It would be interesting to ask medical experts to interpret these selections in terms of their diagnostic experience.

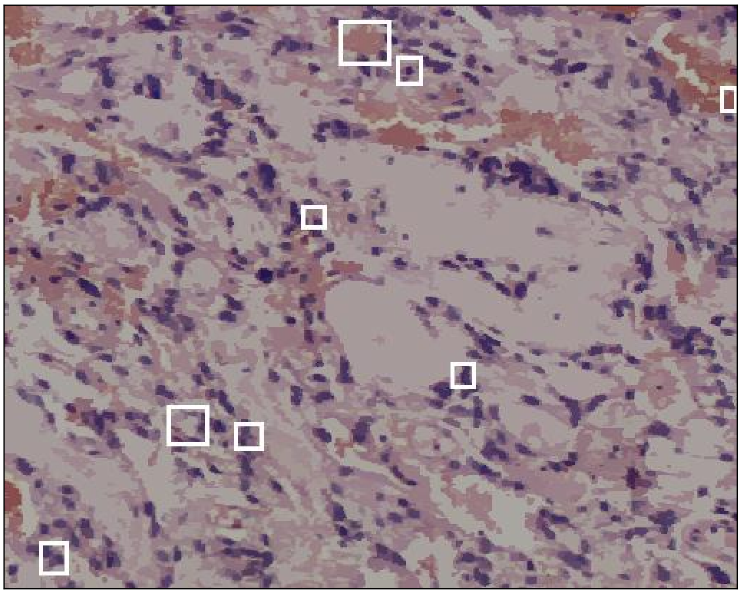

Fig. 3. An exemplary image from the considered data set with representatives of region types selected by evolutionary weighting.

Our future work will concern various improvements of the genetic algorithm (genetic operators, parameters, etc.), and comparison with another algorithms for feature selection, weight adjusting and constructive induction. We are working on implementation of a two-level cross-validation, so that evaluation of individuals during the evolution will be done by a cross-validation in a learning subset of the whole data set. Such architecture of the system may bring more objective evaluation of its performance and results, but will also be much more complex and time consuming.

It would be worthwhile to extend our approach by more “constructive” evolutionary induction (including creation of new attributes, rules etc. [25]). A tradeoff between the number of features, their easy interpretation and classification accuracy could also be introduced. This approach should also be tested on more known data sets to validate its promising utility.

Acknowledgements

We would like to thank professor Janusz Szymaś from the Department of Pathology, University School of Medicine in Poznan, for preparing the histological sections and classifying the images. This work has been supported from State Committee for Scientific Research, KBN research grant no. 8T11C 013 13 and from CRIT 2 – Esprit Project no. 20288.

References

1. Aha, D.W.: Case-based learning algorithms. In: *Proceedings. of Case-based reasoning workshop*, 1991.
2. Aha, D.W., Kibler E., Albert M.K.: Instance-based learning algorithms. *Machine Learning* 6, 1991, 37-66.
3. Aha, D.W., Bankert R.L.: Feature Selection for Case-based Classification of Cloud Types: An Empirical Comparison. *Proceedings AAAI-94 Workshop Case-Based Reasoning*, 1994, 106-112.
4. Bouckaert, A., Medical Diagnosis: Are Expert Systems Needed? *N Engl J Med* 350, 1988, 2800-2815.
5. Brill, F., Brown, D., Martin, W.: Fast genetic selection of features for neural network classifiers. *IEEE Transactions on Neural Networks* 3(2), 1992, 324-328.
6. Chan, S.W.K., Leung, K.S., Wong, W.S.F.: An expert system for the detection of cervical cancer cells using knowledge-based image analyzer. *Artificial Intelligence in Medicine* 8, 1996, 67-90.
7. Cover, T., Hart, P.: Nearest neighbor pattern classification. *IEEE Trans. on Information Theory* 13, 1967, 21-27.
8. Dash, M., Liu, H.: Feature selection for classification. *Intelligent data analysis* 1(3), 1997.
9. Davis, L.: *Handbook of genetic algorithms.* Van Nostrand Reinhold, New York, 1991.
10. Ercal, F., Chawla, A., Stoecker, W.V., Lee, H.Ch., Moss, R.H.: Neural network diagnosis of malignant melanoma from color images. *IEEE Trans. on Biomedical Engineering*, 1994.
11. Goldberg, D.E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing Company, 1989.
12. Gonzalez, R.C., Woods, R.E.: *Digital Image Processing*. Addison-Wesley Publishing Company, 1992.
13. Haralick R.M., Shanmugasm K., Dinstein I.: Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-3(6), 1973, 610-620.
14. Hayes-Roth, F., Waterman, D.A., Lenat, D.: *Building expert systems*. Addison-Wesley Publishing Company, 1983.
15. Hotelling, H.: Analysis of a complex of a statistical variables into principal components. *Journal of Educational Psychology* 24, 1933.
16. Jelonek, J., Komosinski, M.: Genetic algorithms in constructive induction. In: Raś, Z.W., Skowron, A. (eds) *Foundations of Intelligent Systems*, Proceedings of ISMIS 1999. Lecture Notes in Computer Science, vol. 1609. Springer, 1999, 665-673.
17. Jelonek, J., Krawiec, K., Słowiński, R., Stefanowski, J., Słowiński, R.: Rough Set Reduction of Features for Picture-Based Reasoning. In: Lin, T.Y. (ed.) *Soft Computing*. Simulation Councils, San Diego, 1995.
18. Jelonek, J., Stefanowski J.: Feature subset selection for classification of histological images, *Artificial Intelligence in Medicine 9*, 1997, 227-239.
19. Jelonek, J., Krawiec, K., Słowiński, R., Szymas, J.: Grizzly – An image processing and analysis system oriented towards medical images. *Journal of Decision Systems* 7(3-4), 1998.
20. Jelonek, J., Krawiec, K., Słowiński, R.: Construction of Textural Features for Classification of Histological Images. In: *Proceedings of the 7th International Symposium 'Intelligent Information Systems'*. Malbork 1998.
21. Jelonek, J., Krawiec, K., Słowiński, R., Szymas, J.: Intelligent decision support in pathomorphology. *Polish Journal of Pathology* 1999 (in press).

22. John, G., Kohavi R., Pfleger K.: Irrelevant features and the subset selection problem, In: *Proceedings 11th International Machine Learning Conference*, (1994), 121-129.
23. Kleihues, P., Buerger, P.C., Scheithauer, B.W.: The new WHO classification of brain tumours. *Brain Pathology* 3, 1993.
24. Kohavi, R., John, G.H. Wrappers for feature subset selection. *Artificial Intelligence* (Special issue on relevance), 1997 (in press).
25. Komosinski, M.: *ECIS – Evolutionary Constructive Induction System.* https://www.cs.put.poznan.pl/mkomosinski/ecis/
26. MacQueen, J.B.: Some Methods for Classification and Analysis of Multivariate Observations. In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1, 1967, 281-297.
27. Mango, L.J. Computer-assisted cervical cancer screening using neural networks. *Cancer Letters* 77, 1994, 155-162.
28. Matheus, Ch.J.: The need for constructive induction, *In Proceedings 8th International Workshop on Machine Learning*.
29. Matheus, C.J. and Rendell, L.: Constructive Induction on Decision Trees, *Proceedings of IJCAI-89*, Detroit, MI, 1989, 645-650.
30. Michalewicz, Z.: *Genetic Algorithms + Data Structures = Evolution Programs.* Springer-Verlag 1996.
31. Michalski, R.S.: Pattern recognition as knowledge-guided computer induction, Departament of Computer Science Reports, No. 927, University of Illinois, June 1978.
32. Michalski, R.S., Tecuci, G.: *Machine Learning. A multistrategy approach*. Volume IV. Morgan Kaufmann 1994.
33. Pal, N.R., Pal, S.K.: A review of image segmentation techniques. *Pattern Recognition* 26(9) 1993, 1277-1294.
34. Pudil, P., Novovicova J., Kittler J.: Floating Search methods in feature selection, *Pattern Recognition Letters* 15, 1994, 1119-1125.
35. Punch, W., Goodman, E., Pei, M., Chia-Shun, M., Hovland, P., Enbody, R.: Further research on feature selection and classification using genetic algorithms. In: *Proc. International Conference on Genetic Algorithms*, Springer Verlag 1993, 557-564.
36. Quinlan, J.R.: C4.5: *Programs for machine learning*. Morgan Kaufmann Publishers, Inc., San Mateo, 1992.
37. Reed, T.R., Du Buf, J.M.H.: A review of recent texture segmentation and feature extraction techniques. *CVGIP: Image Understanding* 57 (3) 1993, 359-372.
38. Richeldi, M., Lanzi, P.: Performing effective feature selection by investigating the deep structure of the data. In: *Proc. Second International Conference on Knowledge Discovery and Data Mining*, AAAI Press, 1996, 379-383.
39. Siedlecki, W., Sklansky, J.: On automatic feature selection. *International Journal of Pattern Recognition* 2, 1988, 197-220.
40. Wnek, J. and Michalski, R.S.: Hypothesis-driven constructive induction in AQ17: a method and experiments, *Proceedings of IJCAI-91, Workshop on Evaluating and Changing Representation in Machine Learning*, Sydney, 1991.
41. Yang, J., Honavar, V., Feature subset selection using a genetic algorithm. In: *Feature Extraction, Construction, and Subset Selection: A Data Mining Perspective*. Motoda, H. and Liu, H. (ed.) New York: Kluwer, 1998.