

Classifier ensemble creation via false labelling

Bálint Antal

*Department of Genetics, University of Cambridge, Downing site, Cambridge, UK
Faculty of Informatics, University of Debrecen, 4010 Debrecen, POB. 12, Hungary
E-mail: antal.balint@inf.unideb.hu.*

Abstract

In this paper, a novel approach to classifier ensemble creation is presented. While other ensemble creation techniques are based on careful selection of existing classifiers or preprocessing of the data, the presented approach automatically creates an optimal labelling for a number of classifiers, which are then assigned to the original data instances and fed to classifiers. The approach has been evaluated on high-dimensional biomedical datasets. The results show that the approach outperformed individual approaches in all cases.

Keywords: Ensemble learning, Diversity, Hidden Markov Random Fields, Simulated annealing, Bioinformatics

1. Introduction

Classification is a fundamental task in machine learning. In numerous application fields very complex data needs to be classified which is often a difficult task for a single machine learning classifier [1] [2]. There are tremendous amount of research on improving the classification performance in such cases. One highly investigated field for this problem is ensemble learning [3],

where multiple prediction are fused the produce a more efficient classification approach. One fundamental requirement for the creation of classifier ensembles is diversity among them [4], that is, the classifiers included in the ensemble need to complement each other to provide more generalization capabilities than a single learner. Bagging [5] uses randomly selected training subsets with possible overlap (bootstrapping [6]) to ensure diversity among the member of the ensemble. Other diversity creation techniques may involve disjoint random sampling (random subspace methods [7], for example, some variants of Random Forest algorithms [8]), while Adaboost [9] based techniques aims to increase the accuracy of a weak learner iteratively (boosting [10]) using targeted sampling: each iteration considers the misclassified instances of the training data to be more important, and drives the iteration process to include them in the current training set. Another approach to create diverse ensembles is ensemble selection [11], where diversity of classifiers trained on the same dataset is measured and an optimal subset is selected.

A more comprehensive review on the above described techniques can be found in [12]. The relationship of classifier diversity and ensemble accuracy is highly investigated in the ensemble learning community. Although the definite connection between diversity measures and ensemble accuracy is an open question [13], a decomposition of majority voting error into good and bad diversity is proposed in [14].

In this paper, a novel approach for ensemble creation based on this theoretical result is presented. The proposed approach takes the predictions of a single classifier on a training set. Then, an optimal labelling complementing the predictions of the classifiers are created. Thus, an optimal but

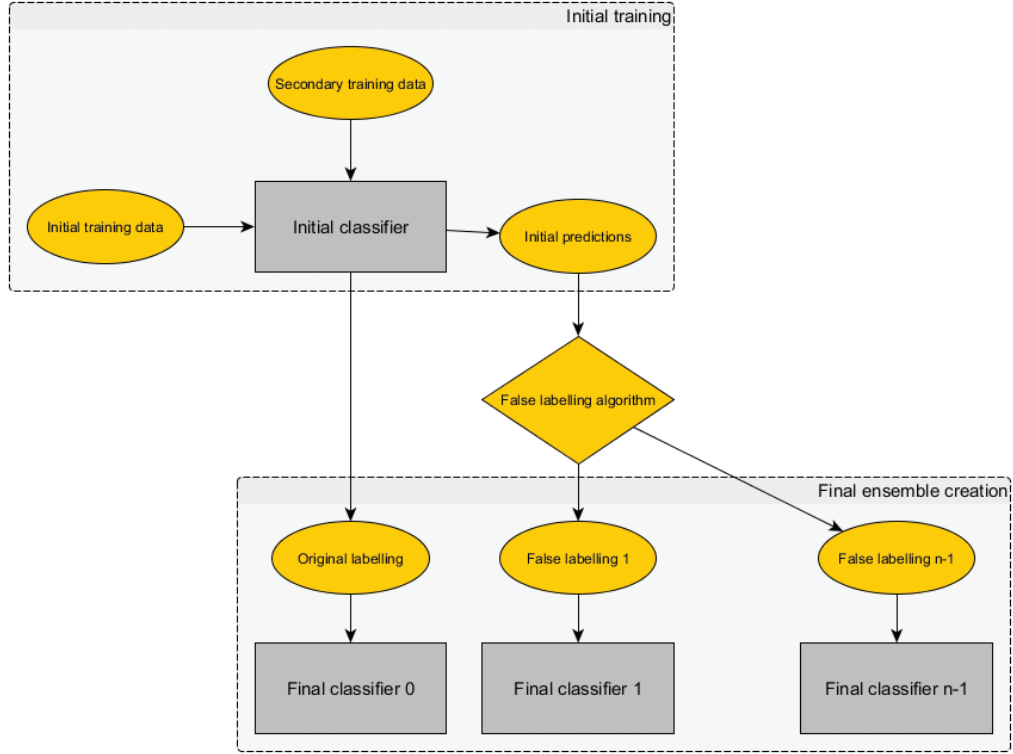


Figure 1: Flowchart of ensemble creation via false labelling

false labelling set is created for a number of classifiers. The data with each false labelling is trained to a classifier thus forming an ensemble. We define a Markov Random Field problem to create an optimal ensemble with this method. The approach has been tested on high-dimensional biomedical datasets where a large improvement over a single learner is achieved. Other aspects of the algorithm including its performance comparison with different number of ensemble members is also discussed. The outline of the proposed algorithm can be seen in Figure 1.

The rest of the paper is organized as follows: section 2 contains the math-

ematical background behind the proposed method, while section 3 defines an optimization problem to solve it and proposes an implementation for it. Section 4 contains our experimental details, while the results are presented and discussed in section 5. Finally, conclusions are drawn in section 6.

2. Ensemble creation via false labelling

The presented false labelling based ensemble creation are presented is restricted to binary classification problems. In this section, the mathematical background behind the algorithm is presented. Moreover, an optimization problem is defined to provide an efficient solution for the false labelling problem. For the basic machine learning and ensemble definitions, we relied on the classic literature [3] and [14].

Let $\Omega = \{-1, +1\}$ be a set of class labels. Then, a function

$$D : \mathbb{R}^n \rightarrow \Omega \quad (1)$$

is called a classifier, while a vector $\vec{\chi} = (\chi_1, \chi_2, \dots, \chi_n) \in \mathbb{R}^n$ is called a feature vector. A dataset $T \in \{\mathbb{R}^n \times \Omega\}^l$ can be defined as follows:

$$T = \{\langle \vec{\chi}_0, \omega_0 \rangle, \langle \vec{\chi}_1, \omega_1 \rangle, \dots, \langle \vec{\chi}_k, \omega_k \rangle\}, \quad (2)$$

where $\vec{\chi}_i \in \mathbb{R}^n, \omega_k \in \Omega, i = 1, \dots, k$ are feature vectors and labels, respectively.

Let D_1, D_2, \dots, D_L be classifiers and $d_t(\vec{\chi}) \in \Omega, t = 1, \dots, L$ their output on the feature vector $\vec{\chi}$. Then, the output of the majority voting ensemble classifier $\mathcal{D}_{maj} : \mathbb{R}^n \rightarrow \Omega$ can be defined as follows:

$$d_{maj}(\vec{\chi}) = \text{sign} \left(\frac{1}{L} \sum_{t=1}^L d_t(\vec{\chi}) \right). \quad (3)$$

The creation of an ensemble \mathcal{D}_{maj} of L classifiers (equation 1) starts by training a base classifier on the half of the training dataset (equation 2) T (T_0). We take the output C_{orig} of the classifier D_{orig} on the other half of the training set (T_1) and create $L - 1$ optimal labellings for a the remaining base classifiers $D_i, i = 2, \dots, L$. Then, we train these classifiers on T_1 with their respective false labellings \mathcal{C}_{false}^i .

The outline of the ensemble creation method is summarized in algorithm 1, while the mathematical formulation is presented in the rest of the section.

Algorithm 1 Outline of ensemble creation via false labelling

Require: a dataset $T \neq \emptyset$, a label set $\mathcal{C} \neq \emptyset$, a classifier D_{orig} , the number of ensemble members $L > 2$ (L is odd).

Ensure: an ensemble of trained classifiers \mathcal{D}_{maj} .

- 1: Split T into T_0 and T_1 randomly.
 - 2: Train D_{orig} on T_0 .
 - 3: $C_{orig} \leftarrow D_{orig}(T_1)$
 - 4: $C_{cl} \leftarrow F(C_{orig}) = \{\mathcal{C}_{false}^2, \mathcal{C}_{false}^3, \dots, \mathcal{C}_{false}^L\}$
 - 5: **for** $i \leftarrow 2, \dots, L$ **do**
 - 6: Train a classifier D_i on $LC(T_1, \mathcal{C}_{false}^i), \mathcal{C}_{false}^i \in C_{cl}$.
 - 7: **end for**
 - 8: **return** $\{D_{orig}, D_2, \dots, D_L\}$
-

2.1. Ensemble creation

The proposed ensemble creation depends on the output of one classifier D_{orig} for a given training dataset T .

First, we split T into two equal parts $T^{(0)}$ and $T^{(1)}$ randomly. We train D_{orig} on $T^{(1)}$ and classify all $\vec{\chi}_j^1 \in T^{(0)}, j = 1, \dots, k/2$ element of $T^{(1)}$:

$$\mathcal{C}_{orig}^1 = \{\omega_j | \omega_j = D_{orig}(\vec{\chi}_j^1), \vec{\chi}_j^1 \in T^{(1)}, j = 1, \dots, k/2\}. \quad (4)$$

Then, we create a majority voting classifier ensemble of L members:

$$\mathcal{D}_{maj} = \{D_1 = D_{orig}, D_2, \dots, D_L\}. \quad (5)$$

To train D_2, \dots, D_L , we will define a false labelling function $F : \Omega^{k/2} \rightarrow \Omega^{k/2 \cdot (L-1)}$. That is

$$F(\mathcal{C}_{orig}^1) = \{\mathcal{C}_{false}^2, \mathcal{C}_{false}^3, \dots, \mathcal{C}_{false}^L\}, \quad (6)$$

where $\mathcal{C}_{false}^i = \{\omega_{i,j}^f | \omega_{i,j}^f \in \Omega, i = 2, \dots, L, j = 1, \dots, k/2\}$. To apply the new labels to the existing dataset, we define the label changing operation $LC : \{R^n \times \Omega \times \Omega\}^l \rightarrow \{R^n \times \Omega\}^l$ in the following way:

$$LC(T, \mathcal{C}) = \{\langle \vec{\chi}_j, \omega_j^f \rangle | \langle \vec{\chi}_j, \omega_j \rangle \in T, \omega_j^f \in \mathcal{C}\}, \quad (7)$$

where T is a dataset and \mathcal{C} is a label set. Finally, we train $D_i, i = 1, \dots, L$ on $LC(T, \mathcal{C}_{false}^i)$, where $\mathcal{C}_{false}^i \in F(\mathcal{C}_{orig}^1)$. Then, the false labelling ensemble is created.

2.2. Selection of the false labelling function

To define an optimal false labelling function F (see equation 6), we recite the decomposition of the majority voting error described in [14]. The majority voting error can be split into three terms: the individual error of the classifiers, the disagreement of the classifiers when they classified the input correctly ("good diversity") and the disagreement of the classifiers when they

classified the input incorrectly ("bad diversity"). The majority voting error decomposition is the basis for defining the energy function for our method.

Let $y(\vec{\chi})$ be the true class label for the feature vector $\vec{\chi}$. Then, the zero-one loss for $d_t(\vec{\chi})$ is defined as follows [14]:

$$e_t(\vec{\chi}) = \frac{1}{2} (1 - y(\vec{\chi}) d_t(\vec{\chi})) \quad (8)$$

Then, the average individual zero-one loss is [14]

$$e_{ind}(\vec{\chi}) = \frac{1}{L} \sum_{t=1}^L e_t(\vec{\chi}) \quad (9)$$

and the ensemble zero-one loss is:

$$e_{maj}(\vec{\chi}) = \frac{1}{2} (1 - y(\vec{\chi}) d_{maj}(\vec{\chi})) \quad (10)$$

The disagreement between d_t and the ensemble is the following [14]:

$$\delta_t(\vec{\chi}) = \frac{1}{2} (1 - d_t(\vec{\chi}) d_{maj}(\vec{\chi})). \quad (11)$$

The classification error of an ensemble is defined [14] as follows:

$$E_{maj} = \int_{\vec{\chi}} e_{ind} - \int_{\vec{\chi}^+} \frac{1}{L} \sum_{t=1}^L \delta_t(\vec{\chi}) + \int_{\vec{\chi}^-} \frac{1}{L} \sum_{t=1}^L \delta_t(\vec{\chi}) \quad (12)$$

Based on equations 10-12, an optimization problem can be defined to find such an optimal labelling.

3. Optimization via Hidden Markov Random Fields

To solve the optimization problem, an approach based on Hidden Markov Random Fields (HMRF) is presented. HMRF is a powerful framework for

solving large-scale optimization problems, since there are multiple methods for solving HMRF problems near optimally in normal time, which would be a challenging task to find exact false labellings for real-life applications.

In this section, we briefly summarize the basis for Hidden Markov Random Field (HMRF) optimization based on [15]. Let

$$A_{k/2, L-1} = a_{i,j} = \begin{pmatrix} \omega_{1,1}^f & \omega_{1,2}^f & \cdots & \omega_{1,k/2}^f \\ \omega_{2,1}^f & \omega_{2,2}^f & \cdots & \omega_{2,k/2}^f \\ \vdots & \vdots & \ddots & \vdots \\ \omega_{L-1,1}^f & \omega_{L-1,2}^f & \cdots & \omega_{L-1,k/2}^f \end{pmatrix}$$

be a matrix containing a false labelling setup and $\mathcal{C}_{orig} = b_{i,j}$ a vector containing the labellings of D_{orig} and $\mathcal{C}_{training} = c_{i,j}$ the labels assigned originally the training instances. All $a_{i,j}$ is a variable which can contain a possible label and at the end of the optimization process, each row contain a false labelling for a classifier D_i .

Let $\Lambda = \{0, 1\}$ be a set of labels. Then, we assign each $a_{i,j}, i = 1, \dots, k/2j = 1, \dots, L-1$ a label $\omega_{i,j}$. Let X be a labelling field. X is a Markov Random Field if $P(X = \omega)$, for all $\omega \in \Lambda$ and $P(\omega_{a_{i,j}} | \omega_{a_{k,l}}, a_{i,j} \neq i_k) = P(\omega_{a_{i,j}} | \omega_{a_{k,l}}, a_{k,l} \in N_{a_{i,j}})$, where $N_{a_{i,j}}$ is a neighbourhood of $a_{i,j}$.

The optimal labelling for the A variables with the HMRF optimization, one can use the the Hammersley-Clifford Theorem [16] to calculate the global energy for a labelling by summarizing the local energies for each variable. That is, during the optimization process, the global energy would be a function of the changes in the states of the $a_{i,j}$ variables.

We define the following three neighbourhoods for the optimization pro-

cess:

$$N_{a_{i,j}}^1 = \{a_{m,j} | m \in \{1, k/2\}, m \neq i\} \cup \{b_i\}, \quad (13)$$

and a neighbourhood of a single variable containing the labelling for all of the feature vectors for the same classifier

$$N_{a_{i,j}}^2 = \{a_{i,l} | l \in \{1, L-1\}, l \neq j\} \cup \{b_i\}, \quad (14)$$

which is a neighbourhood of a single variable containing the labelling of the other classifiers for the same feature vector, and

$$N_{a_{i,j}}^3 = \{a_{k,l} | k \in \{i-q, i+q\}, l \in \{j-q, j+q\}\}, \quad (15)$$

which is a neighbourhood of a variable containing labelling of its close classifiers for inputs in a $q \cdot q$ part of A . First, we consider the individual classification error the individual classifiers:

$$U_{ind}(a_{i,j}) = \frac{\sum \{a_{k,l} | a_{k,l} \in N_{a_{i,j}}^1 \wedge a_{k,l} = \omega_i\}}{k/2}, \quad (16)$$

where ω_i is the actual label assigned to the feature vector in the training set. Out next criteria for the optimization process is to give a labelling, where the number of correct votes is exactly 50%+1 in all cases. Let

$$o = L/2 + 1.$$

Then, we define the function E_{votes} in the following way:

$$U_{good}(a_{i,j}) = \frac{\sum \{a_{k,l} | a_{k,l} \in N_{a_{i,j}}^2 \wedge a_{k,l} = b_i\} - o}{o}. \quad (17)$$

That is, we sum the correct labellings for a given input and subtracting the optimal number of votes from it. In this way, the E_{votes} will be minimal if

the number of correct votes is less than or equal to the number of optimal votes. Thus, we maximize the disagreement for bad diversity and minimize to good diversity [14]. To ensure classification accuracy (and avoid having lower numbers of votes resulting from negative values of E_{votes}), we also define

$$U_{bad}(a_{i,j}) = -\frac{\sum\{a_{k,l} | a_{k,l} \in N_{a_{i,j}}^3\} \wedge \{a_{k,l} \neq b_i\} - o}{o}, \quad (18)$$

which is the disagreement term for bad diversity.

Finally, we must ensure that the votes are unevenly distributed among the classifiers to have less correlation between variables:

$$U_{smoothness}(a_{i,j}) = \begin{cases} \beta & \text{if } a_{i,j} = a_{k,l} \\ -\beta & \text{otherwise.} \end{cases}, \quad (19)$$

for all $a_{k,l} \in N_{a_{i,j}}^2$. In this way we ensure low correlation between the label sets assigned to the classifiers. In summary, the global energy U is the following:

$$U = \sum_{i=0}^{k/2} \sum_{j=0}^{L-1} E_{ind}(a_{i,j}) + E_{good}(a_{i,j}) + E_{bad}(a_{i,j}) + E_{smoothness}(a_{i,j}). \quad (20)$$

The optimization of the HMRF configuration can be done by optimizing U . Since simulated annealing [17], an efficient algorithm for finding approximate global solutions for large state-spaces.

In summary, simulated annealing measures energy values from different states of the variables. Each state is accepted as a better solution if provided a more optimal energy value or accepted by a function, which uses a random number to decide it. This step is important in avoiding stuck in local optima, as do other stochastic approaches like stochastic gradient search. The algorithm for simulated annealing can be found in algorithm 2.

Algorithm 2 Solving the optimization problem with simulated annealing.

Require: An initial temperature T , a minimal temperature T_{min} and a temperature change quotient q .

Require: A function *changeState* changing variable values from their current state.

Require: An acceptance function *accept*. E.g.

$$accept(u, u_{best}, T) = \begin{cases} true, & \exp\left(\frac{e - e_i}{T}\right) > r, \\ false, & \text{otherwise,} \end{cases} \quad (21)$$

where r is a random number.

Ensure: An approximation of the optimal false labelling.

```
1:  $A = a_{m,n} \leftarrow \{0\}$ .  
2:  $u \leftarrow U(A)$   
3:  $l_{best} \leftarrow A$   
4:  $u_{best} \leftarrow u$   
5:  $s \leftarrow 0$   
6: while  $T \geq T_{min}$  do  
7:    $A \leftarrow changeState(A)$   
8:    $u \leftarrow U(A)$   
9:   if  $u \geq u_{best}$  or  $accept(u, u_{best}, T)$  then  
10:     $l_{best} \leftarrow A$   
11:     $u_{best} \leftarrow u$   
12:   end if  
13:    $T \leftarrow T \cdot q$   
14: end while  
15: return  $l_{best}$ 
```

After the optimization process, the l_{best} state of A is the optimal false labelling, which can be used to train the classifiers.

4. Methodology

The proposed approach has been evaluated on high-dimensional biomedical datasets containing gene expressions or proteomics data downloaded from the the Keng Ridge repository [18]. The description of the datasets including the number of instances, the number of features per instance and the status of the patient by disease is summarized in Table 1. As it can be seen, the datasets contain a large number of features for a small number of instances thus making it challenging classification problems. Thus, the datasets are bootstrapped for training to ensure the number of instances per class are similar for better comparison of the methods.

The datasets were splitted into two equal partitions randomly 10 times to have a fair comparison. The false-labelling ensembles are tested with 3, 5, 7, 9, 11, 13, 15 members with Naive Bayes [19] base classifiers for each problem. The implementation of the classifiers was done using Weka [20]. To measure the accuracy of the ensembles, the classification accuracy of each cross-validation round is measured and their mean and standard deviation is calculated. For a comparison, we also show the results for a Naive Bayes classifier, which serves base classifiers in the ensembles, and three state-of-the-art ensemble approaches, namely Adaboost, Bagging and Random Forest.

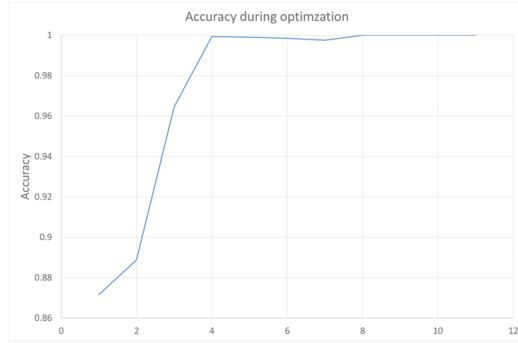
Table 1: Description of the datasets

Dataset	Number of Instances	Number of Features	Disease
breastCancer-train	73	24481	Breast cancer.
breastCancer-test	19	24481	
centralNervousSystem	60	7129	Central nervous system embryonal tumor.
colonTumor	62	2000	Colon tumor.
DLBCL-Stanford	47	4026	Diffuse large B-cell lymphoma
DLBCLOutcome	58	6817	
DLBCLTumor	77	6817	
DLBCL-NIH-train	160	7399	
DLBCL-NIH-test	80	7399	
OC0-9	216	37340	Ovarian cancer
prostate-tumorVSNormal-train	102	12600	Prostate cancer
prostate-tumorVSNormal-test	33	12600	
prostate-outcome	21	12600	

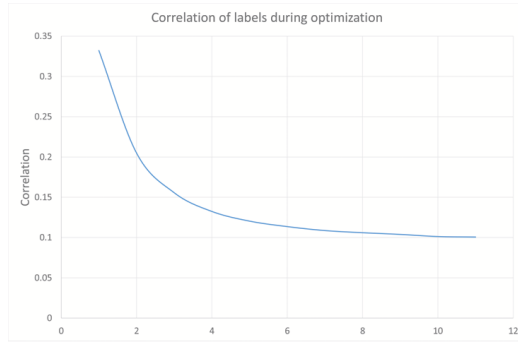
5. Results and discussion

The validity of the optimization technique can be seen in Figures 2(a) and 2(b). As it can be seen in this example, the accuracy of the ensemble has increased steadily through iteration converging to an accuracy of 1, while the correlation of the labels of the ensemble members has been decreased at the same time. Figure 2(c) shows the optimization time through iterations. As it can be seen, in earlier iterations, the optimization procedure increases the energy function with less changes in the labelling spending less time, while in later iterations most of the combinations needs to be tested to increase energy, which require more time.

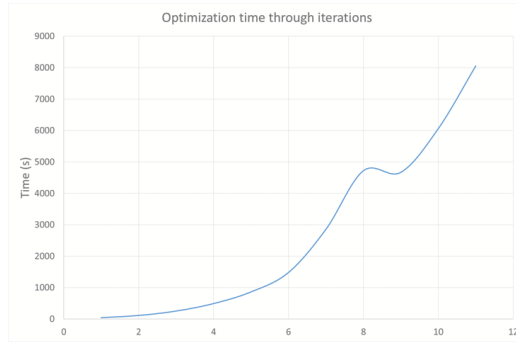
The mean accuracy and their standard deviations on the datasets for the ensembles can be found in table 2. Each column contains the classification accuracy of the respective ensembles $D_i, i \in \{3, 5, 7, 9, 11, 13, 15\}$. The results for the Naive Bayes, Adaboost, Bagging and Random Forest classifiers can be found in table 3. The values in bold for each dataset contain the best performing method. As it can be seen, for each dataset, the proposed approach provides the best values. However, the number of ensembles members varies among the best results. To have a deeper insight on the choice of optimal ensemble size, each investigated ensemble is compared to the best performing among the Naive Bayes, Adaboost, Bagging and Random Forest classifiers. Figures 4-10 show the difference between the respective ensemble and the best performing other method, where each positive value means that the respective ensemble performed better than the best among the other classifiers, while a negative value shows otherwise. As it can be seen, only the ensembles with 5 and 9 members remain above the other methods all



(a) Accuracy



(b) Correlation



(c) Time

Figure 2: Accuracy, correlation of the ensemble member labels and execution time through iterations of the optimization procedure

the time. From table 4 we can see that the sum of the all differences are the highest for the D_5 ensemble. That is, based on these experiments, a false labelling ensemble with 5 member can be recommended to generate.

Statistical analysis of the classifiers is also performed. First, Friedman-test [21] was performed to check whether the results of the proposed ensemble based classifiers, the Naive Bayes, Adaboost, Bagging and Random Forest are from the same distribution. This hypothesis was rejected with $p = 3.8499\text{e-}026$. Then, we applied post-hoc analysis to reveal the differences among the investigated classifiers. To recognize these differences, Tukey’s multiple comparison test [22] is also performed. The test revealed that the proposed ensembles consisting of 5-15 member (D_5, \dots, D_{15}) were all significantly different from the four classifiers they were compared to, while D_3 were significantly different from all but Adaboost. The Friedman ranking also revealed D_5 to be the best performing classifier among the investigated ones. For a visual representation of the Tukey test, see Figure 3, where a confidence interval for the sample mean differences are shown.

6. Conclusion

In this paper, a novel classifier ensemble creation approach is presented. The presented approach automatically creates an2 optimal labelling for a number of classifiers based on the output of a classifier, which are then assigned to the original data instances and fed to classifiers. The approach has been evaluated on high-dimensional biomedical datasets and compared to state-of-the-art classifiers. The results shown improvement in classification accuracy. The possible ensemble size is also investigated, with having

Table 2: Mean and standard deviation of the accuracies of the ensembles on the respective datasets.

Dataset	D_3	D_5	D_7	D_9	D_{11}	D_{13}	D_{15}
breastCancer-train	0.88 ± 0.20	0.98 ± 0.03	0.95 ± 0.10	0.89 ± 0.14	0.88 ± 0.14	0.95 ± 0.08	0.91 ± 0.16
breastCancer-test	0.98 ± 0.07	1.00 ± 0.04	0.98 ± 0.12	0.91 ± 0.25	0.92 ± 0.24	0.91 ± 0.26	0.92 ± 0.24
centralNervousSystem	0.98 ± 0.06	0.98 ± 0.15	0.98 ± 0.11	0.96 ± 0.14	0.93 ± 0.23	0.90 ± 0.23	0.94 ± 0.20
colonTumor	0.96 ± 0.09	1.00 ± 0.04	0.99 ± 0.06	0.94 ± 0.18	0.95 ± 0.16	0.95 ± 0.16	0.94 ± 0.17
DLBCL-Stanford	0.97 ± 0.08	0.99 ± 0.10	0.98 ± 0.14	0.97 ± 0.17	0.96 ± 0.20	0.94 ± 0.24	0.94 ± 0.24
DLBCLOutcome	0.99 ± 0.04	0.98 ± 0.14	0.99 ± 0.10	0.98 ± 0.14	0.95 ± 0.22	0.92 ± 0.27	0.93 ± 0.26
DLBCLTumor	0.98 ± 0.04	0.99 ± 0.10	1.00 ± 0.00	0.98 ± 0.14	0.98 ± 0.14	0.95 ± 0.22	1.00 ± 0.00
DLBCL-NIH-train	0.84 ± 0.05	0.87 ± 0.08	0.92 ± 0.07	0.98 ± 0.03	0.98 ± 0.07	0.92 ± 0.13	0.94 ± 0.11
DLBCL-NIH-test	0.96 ± 0.07	1.00 ± 0.00	0.99 ± 0.10	0.97 ± 0.17	0.96 ± 0.20	0.95 ± 0.22	0.94 ± 0.24
OC0	0.91 ± 0.08	0.97 ± 0.05	0.96 ± 0.05	0.98 ± 0.03	0.97 ± 0.07	0.96 ± 0.12	0.98 ± 0.03
OC1	0.88 ± 0.06	0.93 ± 0.05	0.95 ± 0.04	0.94 ± 0.07	0.91 ± 0.12	0.98 ± 0.02	0.95 ± 0.12
OC2	0.88 ± 0.05	0.95 ± 0.05	0.94 ± 0.06	0.92 ± 0.11	0.95 ± 0.05	0.92 ± 0.11	0.87 ± 0.13
OC3	0.90 ± 0.05	0.92 ± 0.06	0.88 ± 0.11	0.93 ± 0.11	0.93 ± 0.09	0.93 ± 0.10	0.96 ± 0.04
OC4	0.92 ± 0.07	0.95 ± 0.08	0.91 ± 0.09	0.93 ± 0.10	0.95 ± 0.08	0.92 ± 0.10	0.98 ± 0.04
OC5	0.95 ± 0.05	0.96 ± 0.06	0.95 ± 0.08	0.98 ± 0.02	0.96 ± 0.09	0.94 ± 0.09	0.94 ± 0.08
OC6	0.92 ± 0.06	0.95 ± 0.03	0.95 ± 0.06	0.92 ± 0.08	0.92 ± 0.10	0.92 ± 0.08	0.89 ± 0.15
OC7	0.93 ± 0.05	0.95 ± 0.07	0.98 ± 0.04	0.97 ± 0.03	0.96 ± 0.07	0.99 ± 0.03	0.98 ± 0.04
OC8	0.86 ± 0.09	0.93 ± 0.03	0.91 ± 0.10	0.96 ± 0.05	0.93 ± 0.07	0.97 ± 0.07	0.92 ± 0.09
OC9	0.89 ± 0.05	0.93 ± 0.05	0.92 ± 0.05	0.94 ± 0.07	0.93 ± 0.10	0.91 ± 0.08	0.90 ± 0.06
prostate-tumorVSNormal-train	0.91 ± 0.14	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.95 ± 0.20	0.99 ± 0.07	0.95 ± 0.18
prostate-tumorVSNormal-test	1.00 ± 0.02	0.98 ± 0.09	0.98 ± 0.10	0.96 ± 0.16	0.93 ± 0.17	0.90 ± 0.21	0.92 ± 0.17
prostate-outcome	1.00 ± 0.05	0.98 ± 0.09	0.91 ± 0.27	0.90 ± 0.26	0.92 ± 0.22	0.91 ± 0.24	0.93 ± 0.22

Table 3: Mean and standard deviation of the accuracies of other state-of-the-art classifiers on the respective datasets.

Dataset	Naive Bayes	Adaboost	Bagging	Random Forest
breastCancer-train	0.83 ± 0.10	0.87 ± 0.10	0.74 ± 0.17	0.87 ± 0.08
breastCancer-test	0.84 ± 0.15	0.78 ± 0.08	0.74 ± 0.15	0.80 ± 0.12
centralNervousSystem	0.83 ± 0.07	0.84 ± 0.10	0.78 ± 0.16	0.81 ± 0.11
colonTumor	0.84 ± 0.09	0.86 ± 0.09	0.88 ± 0.14	0.86 ± 0.09
DLBCL-Stanford	0.92 ± 0.07	0.82 ± 0.09	0.78 ± 0.12	0.88 ± 0.08
DLBCLOutcome	0.77 ± 0.19	0.86 ± 0.10	0.85 ± 0.17	0.88 ± 0.13
DLBCLTumor	0.92 ± 0.08	0.93 ± 0.05	0.89 ± 0.13	0.93 ± 0.08
DLBCL-NIH-train	0.79 ± 0.08	0.83 ± 0.10	0.78 ± 0.11	0.84 ± 0.10
DLBCL-NIH-test	0.82 ± 0.13	0.77 ± 0.10	0.76 ± 0.10	0.80 ± 0.17
OC0	0.82 ± 0.08	0.92 ± 0.07	0.91 ± 0.04	0.89 ± 0.01
OC1	0.81 ± 0.09	0.87 ± 0.05	0.89 ± 0.04	0.86 ± 0.08
OC2	0.81 ± 0.02	0.86 ± 0.08	0.87 ± 0.05	0.84 ± 0.12
OC3	0.85 ± 0.08	0.88 ± 0.09	0.84 ± 0.07	0.85 ± 0.09
OC4	0.83 ± 0.07	0.92 ± 0.06	0.91 ± 0.05	0.91 ± 0.08
OC5	0.85 ± 0.06	0.86 ± 0.07	0.86 ± 0.07	0.87 ± 0.06
OC6	0.88 ± 0.03	0.92 ± 0.03	0.90 ± 0.02	0.86 ± 0.07
OC7	0.81 ± 0.08	0.94 ± 0.06	0.92 ± 0.04	0.92 ± 0.05
OC8	0.85 ± 0.06	0.89 ± 0.08	0.90 ± 0.06	0.93 ± 0.05
OC9	0.84 ± 0.03	0.88 ± 0.06	0.88 ± 0.08	0.87 ± 0.06
prostate-tumorVSNormal-train	0.62 ± 0.05	0.94 ± 0.03	0.88 ± 0.05	0.87 ± 0.03
prostate-tumorVSNormal-test	0.92 ± 0.07	0.94 ± 0.06	0.87 ± 0.11	0.94 ± 0.04
prostate-outcome	0.87 ± 0.10	0.89 ± 0.12	0.78 ± 0.23	0.78 ± 0.10

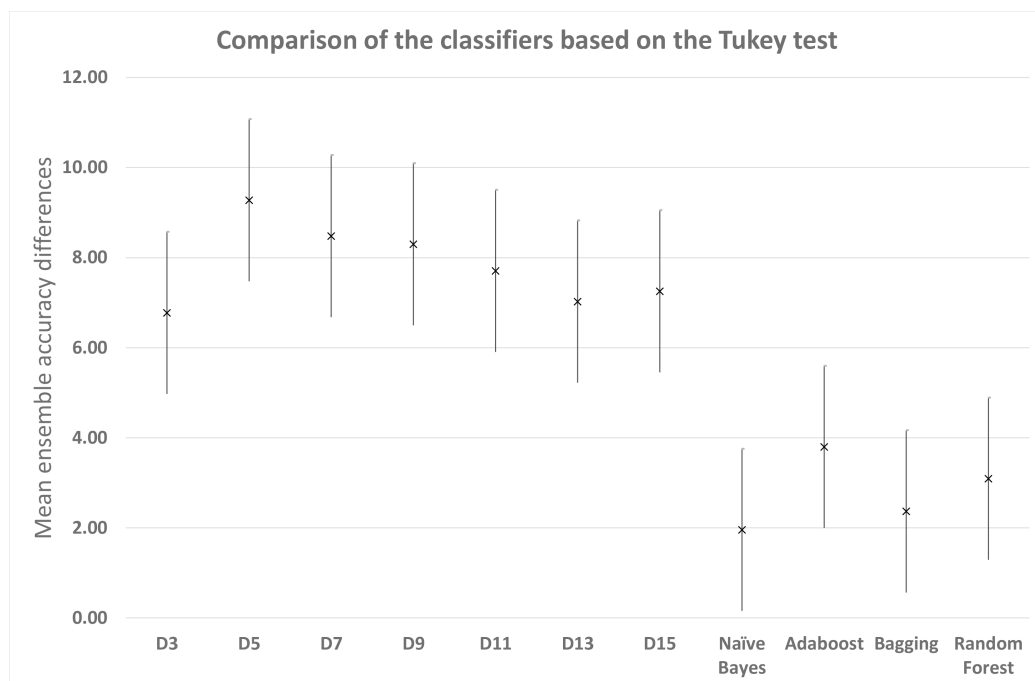


Figure 3: Multiple comparison test

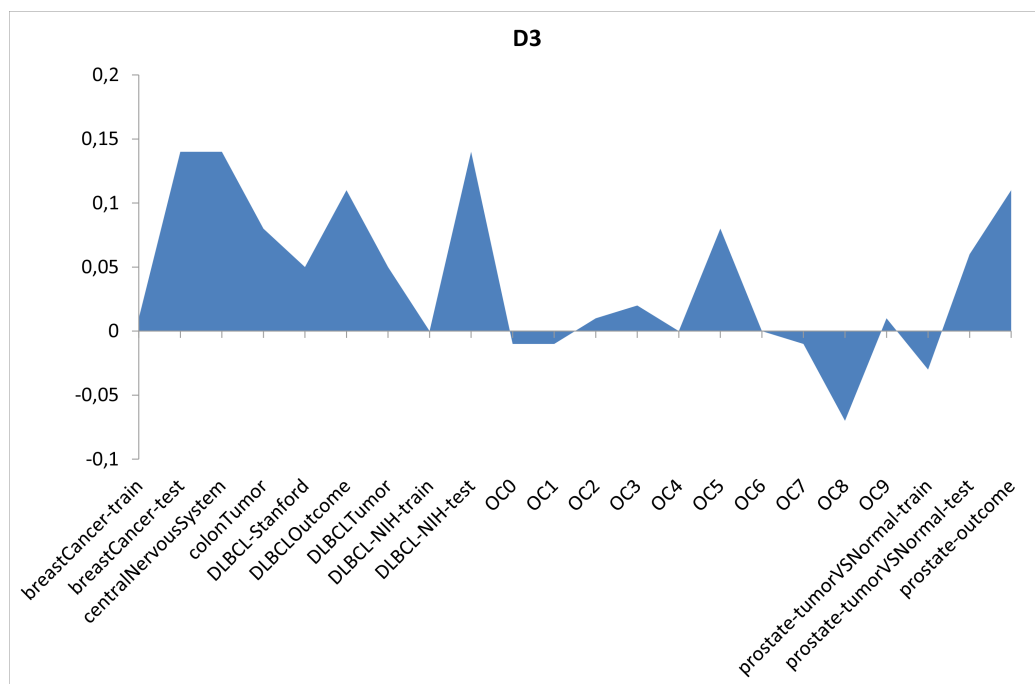


Figure 4: Comparison of the D3 ensemble and the best performing classifiers from Naive Bayes, Adaboost, Bagging and Random Forest.

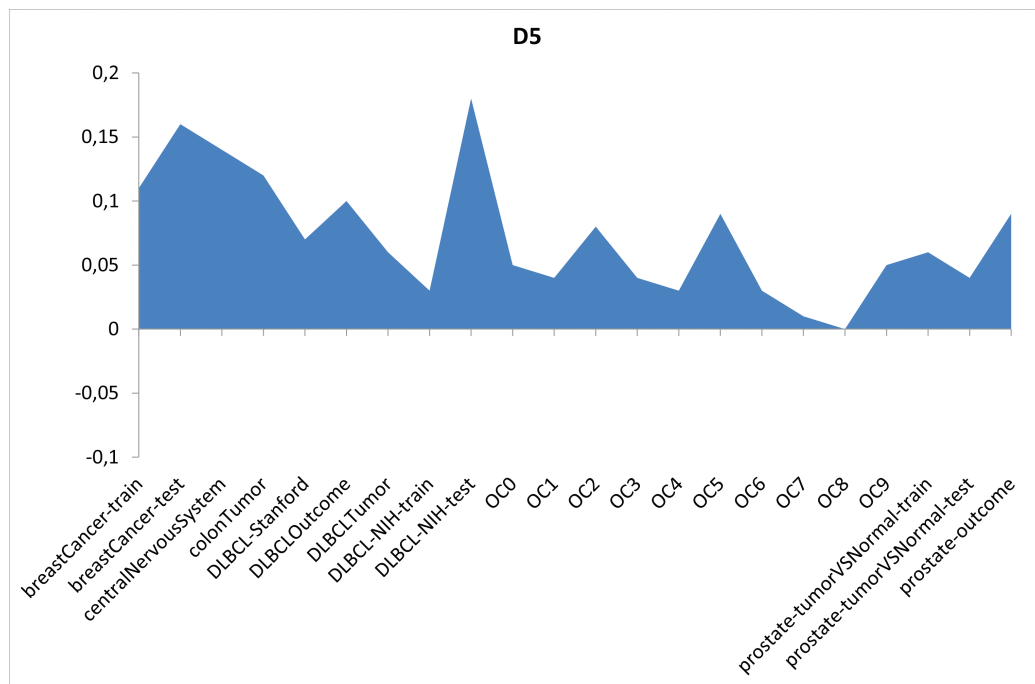


Figure 5: Comparison of the D5 ensemble and the best performing classifiers from Naive Bayes, Adaboost, Bagging and Random Forest.

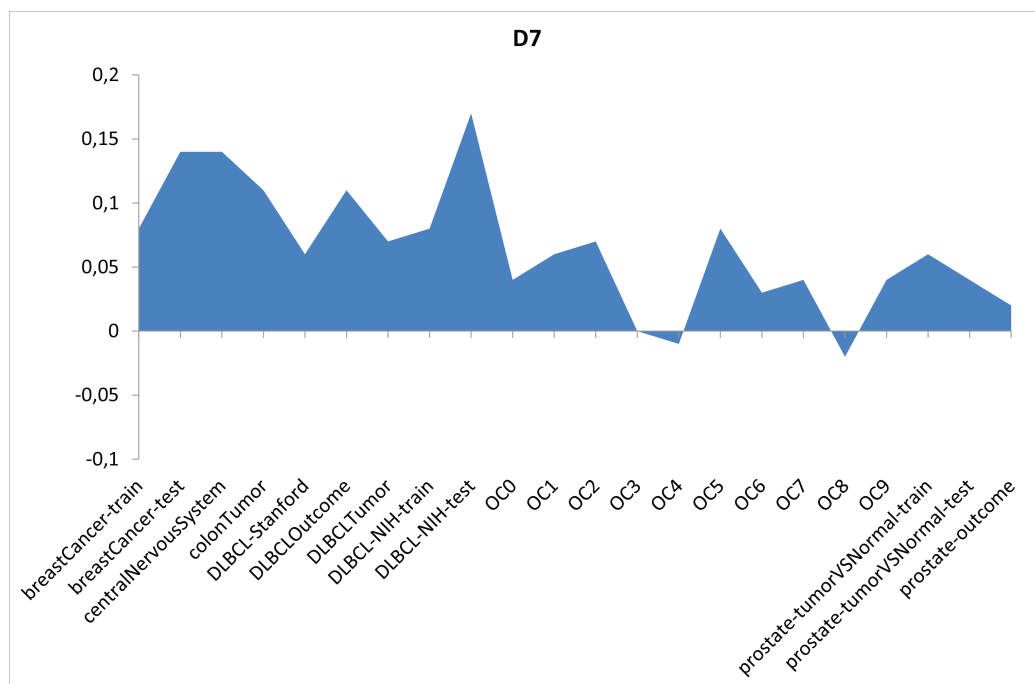


Figure 6: Comparison of the D7 ensemble and the best performing classifiers from Naive Bayes, Adaboost, Bagging and Random Forest.

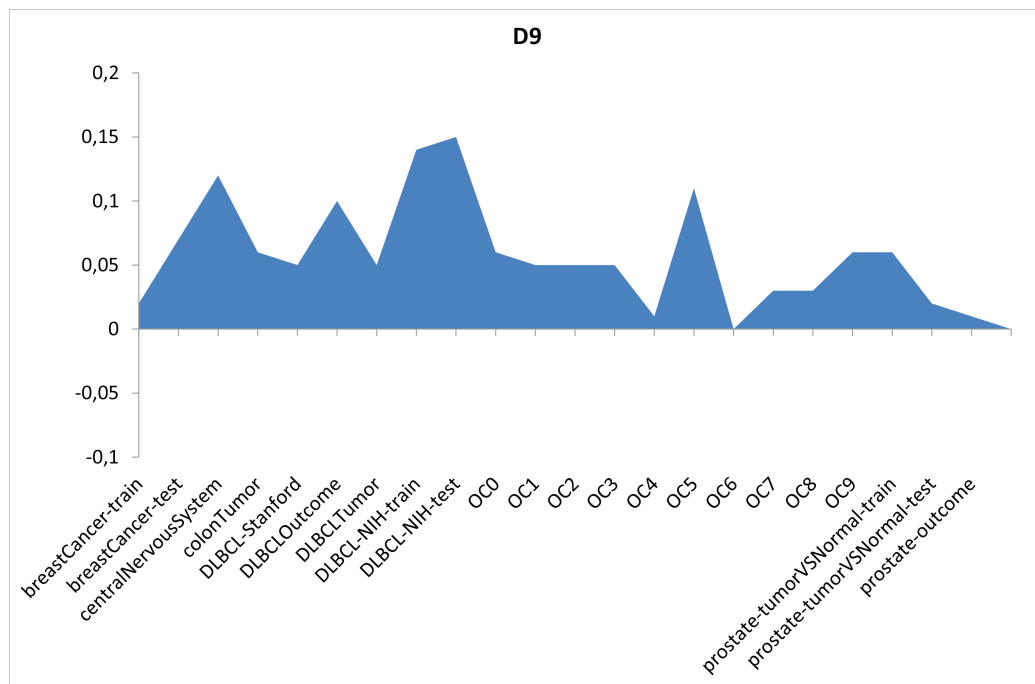


Figure 7: Comparison of the D9 ensemble and the best performing classifiers from Naive Bayes, Adaboost, Bagging and Random Forest.

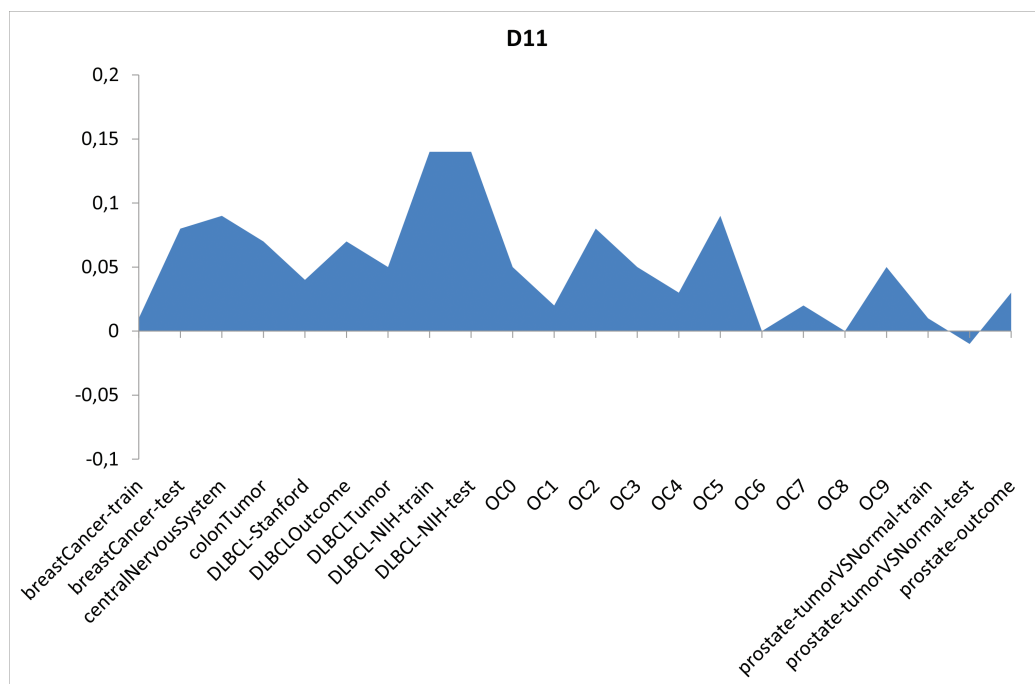


Figure 8: Comparison of the D11 ensemble and the best performing classifiers from Naive Bayes, Adaboost, Bagging and Random Forest.

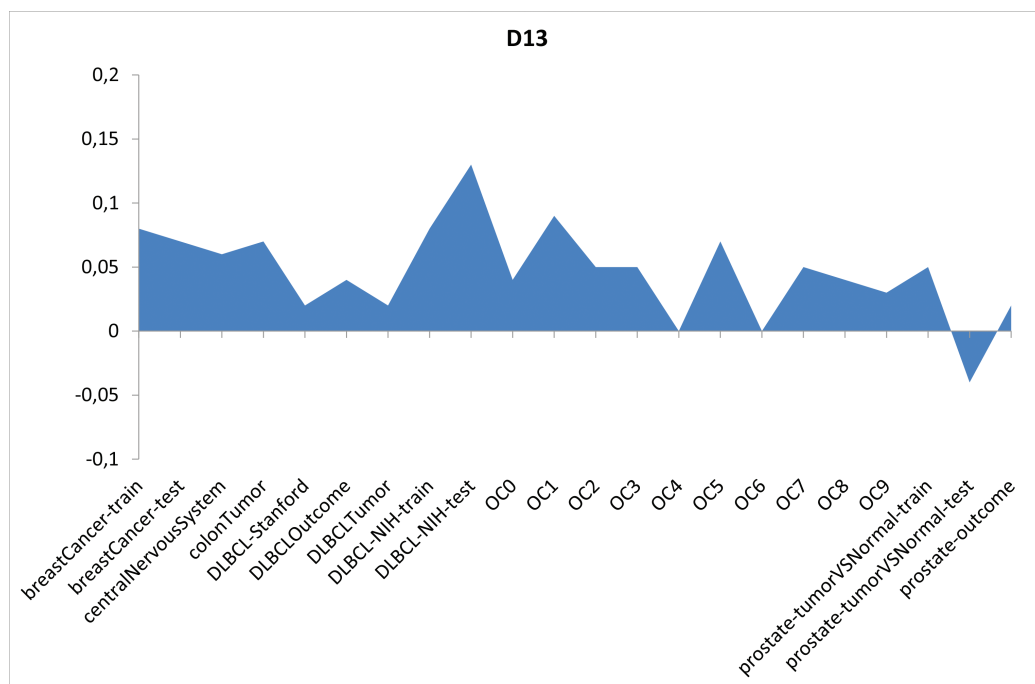


Figure 9: Comparison of the D13 ensemble and the best performing classifiers from Naive Bayes, Adaboost, Bagging and Random Forest.

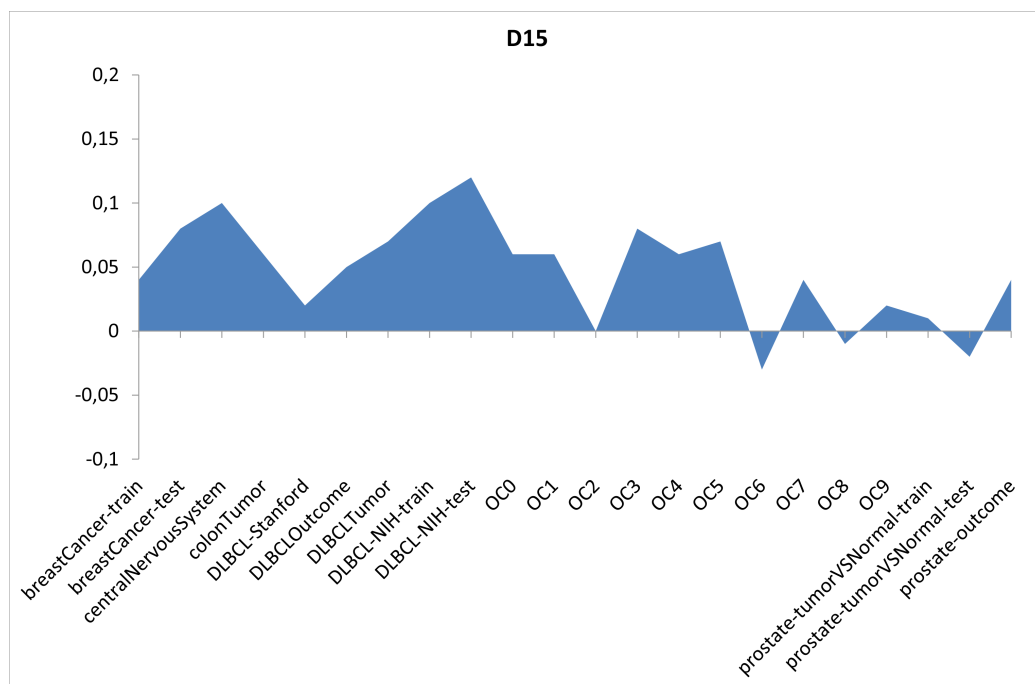


Figure 10: Comparison of the D15 ensemble and the best performing classifiers from Naive Bayes, Adaboost, Bagging and Random Forest.

Table 4: Difference of the respective ensembles and the best performing methods from table 3.

Dataset	D3	D5	D7	D9	D11	D13	D15
breastCancer-train	0.01	0.11	0.08	0.02	0.01	0.08	0.04
breastCancer-test	0.14	0.16	0.14	0.07	0.08	0.07	0.08
centralNervousSystem	0.14	0.14	0.14	0.12	0.09	0.06	0.1
colonTumor	0.08	0.12	0.11	0.06	0.07	0.07	0.06
DLBCL-Stanford	0.05	0.07	0.06	0.05	0.04	0.02	0.02
DLBCLOutcome	0.11	0.1	0.11	0.1	0.07	0.04	0.05
DLBCLTumor	0.05	0.06	0.07	0.05	0.05	0.02	0.07
DLBCL-NIH-train	0	0.03	0.08	0.14	0.14	0.08	0.1
DLBCL-NIH-test	0.14	0.18	0.17	0.15	0.14	0.13	0.12
OC0	-0.01	0.05	0.04	0.06	0.05	0.04	0.06
OC1	-0.01	0.04	0.06	0.05	0.02	0.09	0.06
OC2	0.01	0.08	0.07	0.05	0.08	0.05	0
OC3	0.02	0.04	0	0.05	0.05	0.05	0.08
OC4	0	0.03	-0.01	0.01	0.03	0	0.06
OC5	0.08	0.09	0.08	0.11	0.09	0.07	0.07
OC6	0	0.03	0.03	0	0	0	-0.03
OC7	-0.01	0.01	0.04	0.03	0.02	0.05	0.04
OC8	-0.07	0	-0.02	0.03	0	0.04	-0.01
OC9	0.01	0.05	0.04	0.06	0.05	0.03	0.02
prostate-tumorVSNormal-train	-0.03	0.06	0.06	0.06	0.01	0.05	0.01
prostate-tumorVSNormal-test	0.06	0.04	0.04	0.02	-0.01	-0.04	-0.02
prostate-outcome	0.11	0.09	0.02	0.01	0.03	0.02	0.04
sum	0.88	1.58	1.41	1.3	1.11	1.02	1.02

5 ensemble members as an accurate choice. The presented approach is the first ensemble creation algorithm which creates diversity among classifiers using an artificially created labelling, a technique which can hopefully be reused to create more robust algorithms in problems where individual classifier accuracy can be very varying. In the future, the ensemble creation method could be extended to handle unbalanced or multiclass classification problems efficiently.

Acknowledgments

The publication was supported by the TÁMOP-4.2.2.C-11/1/KONV-2012-0001 project. The project has been supported by the European Union, co-financed by the European Social Fund.

- [1] B. Antal, A. Hajdu, An ensemble-based system for microaneurysm detection and diabetic retinopathy grading, *IEEE Transactions on Biomedical Engineering* 59 (2012) 1720 – 1726.
- [2] B. Antal, A. Hajdu, An ensemble-based system for automatic screening of diabetic retinopathy, *Knowledge-Based Systems* 60 (2014) 20–27.
- [3] L. I. Kuncheva, *Combining Pattern Classifiers. Methods and Algorithms*, Wiley, 2004.
- [4] G. Brown, J. Wyatt, R. Harris, X. Yao, Diversity creation methods: a survey and categorisation, *Information Fusion* 6 (1) (2005) 5–20. doi:10.1016/j.inffus.2004.04.004.
URL <http://linkinghub.elsevier.com/retrieve/pii/S1566253504000375>

- [5] L. Breiman, Bagging predictors, *Machine Learning* 24 (1995) 123–140.
- [6] B. Efron, Nonparametric estimates of standard error: The jackknife, the bootstrap and other methods, *Biometrika* 68 68 (1981) 589–599.
- [7] R. Bryll, Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets, *Pattern Recognition* 20 (2003) 1291–1302.
- [8] T. Ho, The random subspace method for constructing decision forests, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20 (1998) 832–844.
- [9] Y. Freund, R. E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting (1995).
- [10] R. E. Schapire, The boosting approach to machine learning: An overview, in: *MSRI (Mathematical Sciences Research Institute) Workshop on Nonlinear Estimation and Classification*, 2003.
- [11] D. Ruta, B. Gabrys, Classifier selection for majority voting, *Information Fusion* 6 (1) (2005) 63 – 81.
- [12] R. Polikar, Ensemble based systems in decision making, *IEEE Circuits and Systems magazine Third Quarter* (2006) 21–45.
- [13] L. Kuncheva, C. Whitaker, , *Machine Learning* 51 (2) (2003) 181–207.
doi:10.1023/A:1022859003006.
URL <http://dx.doi.org/10.1023/A%3A1022859003006>

- [14] G. Brown, L. I. Kuncheva, GOOD and BAD diversity in majority vote ensembles, in: Proc. 9th International Workshop on Multiple Classifier Systems (MCS'10), Vol. LNCS 5997 of LNCS, Springer-Verlag, Cairo, Egypt, 2010, pp. 124–133.
- [15] M. Berthod, Z. Kato, S. Yu, J. Zerubia, Bayesian image classification using markov random fields, *Image and Vision Computing* 14 (4) (1996) 285 – 295. doi:10.1016/0262-8856(95)01072-6.
URL <http://www.sciencedirect.com/science/article/pii/0262885695010726>
- [16] J. M. Hammersley, P. Clifford, Markov field on finite graphs and lattices (1971).
- [17] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, Optimization by simulated annealing, *Science* 220 (1983) 671–680.
- [18] J. Li, H. Liu, L. Wong, Mean-entropy discretized features are effective for classifying high-dimensional biomedical data, in: Proceedings of the 3rd ACM SIGKDD Workshop on Data Mining in Bioinformatics, 2003, pp. 17–24.
- [19] P. L. George H. John, Estimating continuous distributions in bayesian classifiers, in: Eleventh Conference on Uncertainty in Artificial Intelligence, 1995, pp. 338–345.
- [20] I. H. Witten, E. Frank, Data Mining: Practical machine learning tools and techniques, 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

- [21] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, *Journal of the American Statistical Association* 32 (1937) 675–701.
- [22] J. W. Tukey, Comparing individual means in the analysis of variance., *Biometrics* 5 (1949) 99–114.