Available online at www.sciencedirect.com
SCIENCE DIRECT®
Information Fusion 6 (2005) 83–98
INFORMATION FUSION
www.elsevier.com/locate/inffus

Diversity in search strategies for ensemble feature selection

Alexey Tsymbal [a,*], Mykola Pechenizkiy [b], Pádraig Cunningham [a]

[a] Department of Computer Science, Trinity College Dublin, College Green, Dublin 2, Ireland
[b] Department of Computer Science and Information Systems, University of Jyväskylä, Jyväskylä 40351, Finland

Received 3 November 2003; received in revised form 2 April 2004; accepted 4 April 2004
Available online 14 May 2004

Abstract

Ensembles of learnt models constitute one of the main current directions in machine learning and data mining. Ensembles allow us to achieve higher accuracy, which is often not achievable with single models. It was shown theoretically and experimentally that in order for an ensemble to be effective, it should consist of base classifiers that have diversity in their predictions. One technique, which proved to be effective for constructing an ensemble of diverse base classifiers, is the use of different feature subsets, or so-called ensemble feature selection. Many ensemble feature selection strategies incorporate diversity as an objective in the search for the best collection of feature subsets. A number of ways are known to quantify diversity in ensembles of classifiers, and little research has been done about their appropriateness to ensemble feature selection. In this paper, we compare five measures of diversity with regard to their possible use in ensemble feature selection. We conduct experiments on 21 data sets from the UCI machine learning repository, comparing the ensemble accuracy and other characteristics for the ensembles built with ensemble feature selection based on the considered measures of diversity. We consider four search strategies for ensemble feature selection together with the simple random subspacing: genetic search, hill-climbing, and ensemble forward and backward sequential selection. In the experiments, we show that, in some cases, the ensemble feature selection process can be sensitive to the choice of the diversity measure, and that the question of the superiority of a particular measure depends on the context of the use of diversity and on the data being processed. In many cases and on average, the plain disagreement measure is the best. Genetic search, kappa, and dynamic voting with selection form the best combination of a search strategy, diversity measure and integration method.

Keywords: Ensemble of classifiers; Ensemble diversity; Feature selection; Search strategy; Dynamic integration of classifiers

1. Introduction

A popular method for creating an accurate classifier from a set of training data is to construct several different classifiers, and then to combine their predictions. It was shown in many domains that an ensemble is often more accurate than any of the single classifiers in the ensemble. The integration of multiple classifiers, to improve classification results, is currently an active research area in the machine learning and neural networks communities. Dietterich [10] has presented the integration of multiple classifiers as one of the four most important directions in machine learning research. Sharkey [29] gives a good introduction to the area of ensembles and presents a survey of relevant work. While the focus of her paper is on neural networks, most remarks are relevant to any ensemble in general.

Both theoretical and empirical research have demonstrated that a good ensemble is one where the base classifiers in the ensemble are both accurate and tend to err in different parts of the instance space (that is have diversity in their predictions). Some studies such as boosting [3,9,21] and the random subspacing [14] show that integration of low-accuracy (also called "weak") classifiers can be effective as well. It was shown that the

* Corresponding author. Tel.: +353-1-6083837/+353-876352188; fax: +353-1-6772204.
E-mail addresses: tsymbalo@tcd.ie (A. Tsymbal), mpechen@cs.jyu.fi (M. Pechenizkiy), padraig.cunningham@cs.tcd.ie (P. Cunningham).

1566-2535/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.inffus.2004.04.003

low-accuracy of base classifiers in such ensembles is compensated for by the ensemble diversity.

Another important issue in creating an effective ensemble is the choice of the function for combining the predictions of the base classifiers. It was shown that increasing coverage of an ensemble through diversity is not enough to ensure increased prediction accuracy—if the integration method does not properly utilize the ensemble diversity, then no benefit arises from integrating multiple models [5].

One effective approach for generating an ensemble of accurate and diverse base classifiers is the use of different feature subsets, or so-called *ensemble feature selection* [23]. By varying the feature subsets used to generate the base classifiers, it is possible to promote diversity and produce base classifiers that tend to err in different subareas of the instance space. While traditional feature selection algorithms have the goal of finding the best feature subset that is relevant to both the learning task and the selected inductive learning algorithm, the task of ensemble feature selection has the additional goal of finding a set of feature subsets that will promote disagreement among the base classifiers [23].

Feature selection algorithms, including ensemble feature selection, are typically composed of the following components [1,23]: (1) *search strategy*, that searches the space of feature subsets; and (2) *fitness function*, that inputs a feature subset and outputs a numeric evaluation. The search strategy's goal is to find a feature subset maximizing this function.

It is reasonable to include in the fitness functions, explicitly or implicitly, both accuracy and diversity. One measure of fitness, which was proposed by Opitz [23], defines the fitness Fitness_i of a classifier i corresponding to a feature subset i to be proportional to the classification accuracy acc_i and the diversity div_i of the classifier:

$$\text{Fitness}_i = \text{acc}_i + \alpha \cdot \text{div}_i, \quad (1)$$

where α reflects the influence of diversity. Diversity div_i is the contribution of classifier i to the total ensemble diversity, which can be measured as the average pairwise diversity for all the pairs of classifiers including i. This fitness function was also used in experiments in [33], and we use it in our experiments in this paper.

A common measure of classification accuracy is the percentage of correct classifications on the test data set (also called apparent accuracy). If the class distributions are significantly uneven we may choose to use the average within-class accuracy, which is the average percentage of correct classifications within each class. The simple "percentage of correct classifications" measure is successfully used in the vast majority of cases. Measuring diversity is not that straightforward – there are a number of ways to measure diversity in ensembles of classifiers, and not much research has been done about the appropriateness and superiority of one measure over another.

In this paper, we consider different measures of the ensemble diversity, which could be used as a component of the fitness function (1), and which could also be used to measure the total ensemble diversity, as a general characteristic of ensemble goodness. The goal of this paper is to compare the considered measures of diversity in the context of ensemble feature selection for these two particular tasks (measuring the total ensemble diversity and measuring the contribution of each individual classifier to the ensemble diversity). To our knowledge, in the existing literature, comparing different measures of ensemble diversity is done by analyzing their correlation with various other ensemble characteristics. Such characteristics are the ensemble accuracy, the difference between the ensemble accuracy and the average base classifier accuracy, and the difference between the ensemble accuracy and the maximal base classifier accuracy [2,20]. In contrast to that approach, in this paper we compare the measures of diversity using the wrapper approach. We apply them as a component of the fitness function guiding the search in ensemble feature selection, and compare the resulting accuracies and other ensemble characteristics.

The paper is organized as follows. In Section 2 we consider the general task of constructing an effective ensemble, review different techniques for generating the base classifiers, and especially ensemble feature selection, and present four strategies for ensemble feature selection. In Section 3 we consider the question of integration of an ensemble of classifiers and review different integration methods. In Section 4 we present five different measures for diversity in classification ensembles. In Section 5 we present our experiments with these measures and conclude in the next section with a summary and assessment of further research topics.

2. Ensemble feature selection and search strategies in it

In this section, we consider ensemble feature selection with a focus on the random subsampling as an effective technique for generating the base classifiers in ensembles, and search strategies for ensemble feature selection, which we shall use in our experiments.

2.1. Ensemble feature selection

The task of using an ensemble of models can be broken down into two basic questions: (1) what set of learned models should be generated?; and (2) how should the predictions of the learned models be integrated? [10,29]. To generate a set of accurate and diverse learned models, several approaches have been tried including the use of different learning algorithms with heterogeneous representations and search biases, the use of randomization in the model search, the use of different data distributions etc.

One effective approach is to use models with homogeneous representations that differ in the data on which they are trained. For example, two well-known ensemble methods of this type are bagging and boosting, which generate models on different distributions of the original data set [3,9,21,31].

Another way for building models with homogeneous representations, which proved to be effective, is the use of different subsets of features for each model. For example, in [24] base classifiers are built on different feature subsets, where each feature subset includes features relevant for distinguishing one class label from the others (the number of base classifiers is equal to the number of classes). Finding a set of feature subsets for constructing an ensemble of accurate and diverse base models is also known as *ensemble feature selection* [23]. While traditional feature selection algorithms have the goal of finding the best feature subset that is germane to both the learning task and the selected inductive learning algorithm, the task of ensemble feature selection has the additional goal of finding a set of feature subsets that will promote disagreement among the base classifiers [23].

Ho [14] has shown that simple random selection of feature subsets may be an effective technique for ensemble feature selection because the lack of accuracy in the ensemble members is compensated for by their diversity. This technique is called the random subspace method or simply the Random Subspacing (RS). In the RS, one randomly selects F^*, F features from the F-dimensional training set. By this, one obtains the F^*-dimensional random subspace of the original F-dimensional feature space. This is repeated S times to build S feature subsets which are then used to construct S base classifiers [14].

Instead of selecting a fixed number F^* of features as in [14] (she used approximately half of the features for each base classifier) we use probabilistic feature selection in our implementation of the RS in this paper. We consider all the F features as having equal probability of being selected to the feature subset. This probability is selected randomly from the interval (0,1) before defining each feature subset. Thus, the initial feature subsets include different numbers of features. It was shown in experiments in [33] that this implementation of the RS provides ensembles with higher diversity, and consequently, higher accuracy.

In the case, when the number of training instances is relatively small compared with the data dimensionality, by constructing classifiers in random subspaces one may solve the small sample size problem, because the training sample size relatively increases in random subspaces. Ho [14] shows that while most other classification methods suffer from the curse of dimensionality, this method can take advantage of high dimensionality.

The RS has much in common with bagging [31], but instead of sampling instances, one samples features. Like bagging, the RS is a parallel learning algorithm, that is, generation of each base classifier is independent. This makes it suitable for parallel implementation for fast learning that is desirable in some practical applications. It was shown that, like in bagging, the ensemble accuracy could be only increased with the addition of new members, even when the ensemble complexity grew [14]. Skurichina and Duin [31] have found that the RS performs relatively better when the classification ability (discrimination power and also the redundancy) is spread over many informative features than when the classification ability is condensed in few features (as with many completely redundant noisy features).

The RS is used as a base in a number of ensemble feature selection strategies, e.g. GEFS (Genetic Ensemble Feature Selection) [23] and HC (Hill Climbing) [8]. Opitz [23] mentioned that the initial population was surprisingly good and produced better ensembles on average than the popular and powerful ensemble approaches of bagging and boosting. Cunningham and Carney [8] also have made an attempt to improve the accuracy of the RS base classifiers using a hill-climbing procedure, including or deleting one feature at a time from a given feature subset if this raised the classification accuracy. However, their results were mainly negative, as this kind of hill-climbing often led to overly small diversity of the base classifiers (in some cases the feature subsets became even identical), and hence, to bad ensemble accuracy.

Ensemble feature selection is the focus of this paper, and in the next section we consider search strategies for it used in our experiments.

2.2. Search strategies for ensemble feature selection

In this section, we consider four different search strategies for ensemble feature selection: (1) Hill Climbing (HC); (2) a Genetic Algorithm for ensemble feature selection (GA); (3) Ensemble Forward Sequential Selection (EFSS); and (4) Ensemble Backward Sequential Selection (EBSS).

Techniques for ensemble feature selection are often based on plain feature selection strategies that select a single feature subset. All the four ensemble feature selection strategies presented in this section are extensions of the corresponding plain feature selection techniques. A thorough overview of search strategies for a single feature subset selection including their experimental evaluation is presented in [17]. However, the experiments in this paper have one problem that the benefits of the feature subsets obtained are validated with a test set that is shown to the search algorithm during the search as reported by Reunanen [26] (no separate validation set was used).

The use of a hill-climbing search as a local-search wrapper-based approach has been shown to be effective for a single feature subset selection [15]. The Hill Climbing (HC) ensemble feature selection strategy, which we use in this research, presented in [8], is composed of two major phases: (1) construction of the initial ensemble by the random subspacing; and (2) iterative refinement of the ensemble members with sequential-mutation hill climbing. Initial feature subsets are constructed using the random subspace method. Then, the initial ensemble is formed. Further, iterative refinement of the ensemble members is used to improve the accuracy and diversity of the base classifiers. The iterative refinement is based on a hill-climbing search. For all the feature subsets, an attempt is made to switch (include or delete) each feature sequentially from the first feature to the last one. For each switch, if the resulting feature subset produces better performance on the validation set for the base classifier on its fitness (1), that change is kept. A sequence of attempts to switch each feature of a base classifier is called a pass. A number of passes are done until no further improvements are possible for a particular base classifier. Normally, no more than 4 passes through each feature subset are necessary. The base classifiers are processed sequentially according to their position in the ensemble using the same procedure. See [33] for the detailed algorithm of HC.

It is important to note that in the case when $\alpha = 0$ in the fitness function (1), only accuracy is taken into account. It was shown in [33,35] that this often leads to poor degenerated ensembles, often with identical feature subsets in each base classifier, significantly decreasing accuracy of the random subspacing. The case when $\alpha = 0$ was never optimal for the considered data sets. On the other hand, it was shown [33,35] that the ensembles based on diversity ($\alpha \neq 0$) have higher accuracy, and the base classifiers produced focusing on diversity have less features on average than those based on accuracy only. Another important conclusion made in [33] is that the importance of diversity α is different for different data sets.

The use of genetic search has also been an important direction in the feature selection research. Genetic algorithms have been shown to be effective global optimization techniques in feature subset selection. The use of genetic algorithms for ensemble feature selection was first proposed by Kuncheva [18] and further elaborated in [19]. As the fitness function in [18,19] the accuracy of a combination of classifiers is first proposed and not that of the individual classifiers. However, such a fitness function is biased towards some particular integration method (often simple majority voting). Besides, as it was shown e.g. in [18], such a design is prone to overfitting, and some additional preventive measures are needed to be taken to avoid this (as including in the fitness function penalty terms accounting for the number of features used). This is true for the use of the ensemble accuracy in the fitness function for ensemble feature selection in general, and not only for genetic algorithms. The use of individual accuracy and diversity as in (1) is an alternative solution to this problem.

Our Genetic Algorithm for ensemble feature selection (GA) strategy is based on the GEFS algorithm of Opitz [23]. GEFS is the first genetic algorithm for ensemble feature selection that explicitly uses diversity in the fitness function. GA begins, as HC, with creating an initial population of classifiers where each classifier is generated by randomly selecting a different subset of features. Then, new candidate classifiers are continually produced by using the genetic operators of crossover and mutation on the feature subsets. After producing a certain number of individuals the process continues with selecting a new subset of candidates by selecting the members randomly with a probability proportional to fitness (it is known as the roulette-wheel selection). The process of producing new classifiers and selecting a subset of them (a generation) continues a number of times, known as the number of generations. After a predefined number of generations, the fittest individuals make up the population, which comprises the ensemble [23]. In our implementation, the representation of each individual (a feature subset) is simply a constant-length string of bits, where each bit corresponds to a particular feature. GEFS used a different representation, more convenient for use in neural networks. The representation of each individual in GEFS is a dynamic length string of integers, where each integer indexes a particular feature [23]. The crossover operator uses uniform crossover, in which each feature of the two children takes randomly a value from one of the parents. The feature subsets of two individuals in the current population are chosen randomly with a probability proportional to $\log(1 + \text{fitness})$. The mutation operator randomly toggles a percentage of bits in an individual.

EFSS and EBSS are sequential feature selection strategies, which add or delete features using a hill-climbing procedure, and have polynomial complexity. The most frequently studied variants of plain sequential feature selections algorithms (which select a single feature subset) are forward and backward sequential selection, FSS and BSS [1]. FSS begins with zero attributes, evaluates all feature subsets with exactly one feature, and selects the one with the best performance. It then adds to this subset the feature that yields the best performance for subsets of the next larger size. The cycle repeats until no improvement is obtained from extending the current subset. BSS instead begins with all features and repeatedly removes a feature whose removal yields the maximal performance improvement [1]. EFSS and EBSS apply FSS or BSS sequentially to form each of the base classifiers using a predefined fitness function. The difference among the base classifiers is provided in

this case by the fitness function, as it includes not only individual accuracy, but also diversity with already formed base classifiers.

EFSS and EBSS have polynomial complexity with regard to the number of features: $O(S \cdot F \cdot F')$, where S is the number of base classifiers, F is the total number of features, and F' is the number of features included or deleted on average in an FSS or BSS search. HC has similar polynomial complexity $O(S \cdot F \cdot N_{\text{passes}})$, where N_{passes} is the average number of passes through the feature subsets in HC until there is some improvement (usually no more than 4). The complexity of GA does not depend on the number of features, and is $O(S' \cdot N_{\text{gen}})$, where S' is the number of individuals (feature subsets) in one generation, and N_{gen} is the number of generations.

In our experiments, on average, each of the strategies looks through about 1000 feature subsets (given that the number of base classifiers is 25, the average number of features is 10, the average percentage of included or deleted features in EFSS and EBSS is 40%, the number of passes in HC is 4, the number of individuals in a population in the GA is 100, and the number of generations is 10).

Comparative experiments with these four strategies on a collection of data sets from the medical field of acute abdominal pain classification were considered in [32]. The best search strategy in that context was EFSS (it was the best for every data set considered), generating more diverse ensembles with more compact base classifiers. EFSS generated extremely compact base classifiers, including from 9% to 13% features on average (less than 3 features). The results with genetic search were found to be disappointing, not noticeably improving with generations.

In comparison with [32] we have made a number of changes in the genetic search-based strategy. In our new version considered in this paper no full feature sets are allowed in the random subspacing nor may the crossover operator produce a full feature subset. Individuals for crossover are selected randomly proportional to $\log(1 + \text{fitness})$ instead of just fitness, which adds more diversity into the new population (each feature subset has more chances for being selected, no matter what is its fitness). The generation of children identical to their parents is prohibited in the crossover operator—if a child is the same as one of its parents, the mutation operator is applied to it until it is different. To provide a better diversity in the length of the feature subsets and the population in general, we use two different mutation operators, one of which always adds features randomly with a given probability, and the other—deletes features. Each operator is applied to exactly a half (25 of 50) of the individuals being mutated. Our pilot studies have shown that these minor changes help us to significantly improve the work of GA so that it does not converge after a single generation already to a local extremum (as it was in [23,32]). These changes support a finding made in [19] that a genetic algorithm for ensemble feature selection should be closer to a random search, and should be given a chance to hit a good solution rather than elaborate one.

Parameter settings for our implementation of the genetic search in GA include a mutation rate of 50% (as proposed in [23]), a population size of 25, a search length of 100 feature subsets (the number of new individuals produced by crossover and mutation), of which 50 are offsprings of the current population of 25 classifiers generated with the crossover operator, and 50 are mutated offsprings. 10 generations of individuals were produced, as our pilot studies have shown that in most cases, with this configuration, the ensemble accuracy does not improve after 10 generations, due to overfitting the training data.

3. Techniques for integration of an ensemble of models

Brodley and Lane [5] have shown that simply increasing coverage of an ensemble through diversity is not enough to insure increased prediction accuracy (coverage is defined there as the percentage of instances on which at least one base classifier is correct). If the integration method does not utilize the coverage, then no benefit arises from integrating multiple classifiers. Thus, the diversity and coverage of an ensemble are not in themselves sufficient conditions for the ensemble accuracy. It is also important for the ensemble accuracy to have a good integration method that will utilize the diversity of the base models.

The challenging problem of integration is to decide which one(s) of the classifiers to choose or how to combine the results produced by the base classifiers. Techniques using two basic approaches have been suggested as a solution to the integration problem: (1) a *combination approach*, where the base classifiers produce their classifications and the final classification is composed using them; and (2) a *selection approach*, where one of the classifiers is selected and the final classification is the result produced by it.

Several effective techniques for the *combination* of classifiers have been proposed. One of the most popular and simplest techniques used to combine the results of the base classifiers, is simple voting (also called majority voting and select all majority (SAM)) [3]. In the voting technique, the classification of each base classifier is considered as an equally weighted vote for that particular class value. The class value that receives the biggest number of votes is selected as the final classification (ties are solved arbitrarily). Often, weighted voting is used: each vote receives a weight, which is usually proportional to the estimated generalization performance of the corresponding classifier. Weighted Voting (WV)

works usually much better than simple majority voting (or at least, in some cases, they were reported to work comparably) [3].

A number of *selection* techniques have also been proposed to solve the integration problem. One of the most popular and simplest selection techniques is Cross-Validation Majority (CVM, also called Single Best, we call it simply Static Selection, SS, in our experiments) [28]. In CVM, the cross-validation accuracy for each base classifier is estimated using the training set, and then the classifier with the highest accuracy is selected (ties are solved using voting).

The approaches to classifier selection can be divided into two subsets: *static* and *dynamic* selection. The static approaches propose one "best" method for the whole data space, while the dynamic approaches take into account each new instance to be classified and its neighbourhood only. The CVM is an example of the static approach. Techniques for combining classifiers can be static or dynamic as well. For example, widely used Weighted Voting [3] is a static approach. The weights for each base classifier's vote do not depend on the instance to be classified. Usually, better results can be achieved if the classifier integration is done dynamically taking into account the characteristics of each new instance (if there is enough data to reliably estimate the needed parameters).

We consider in our experiments three dynamic integration techniques based on the local accuracy estimates: Dynamic Selection (DS) [25], Dynamic Voting (DV) [25], and Dynamic Voting with Selection (DVS) [34]. All these are based on the same local accuracy estimates. The three dynamic integration techniques contain two main phases [25,34]. First, at the learning phase, the training set is partitioned into folds. During the cross validation run, we estimate the local classification errors of each base classifier for each instance of the training set according to the 1/0 loss function. The learning phase finishes with training the base classifiers on the whole training set. The application phase begins with determining the k-nearest neighbourhood for a new instance using a distance metric based on the values of the features. Then, the weighted nearest neighbour procedure is used to predict the local classification errors of each base classifier for the new instance.

Then, DS simply selects a classifier with the least predicted local classification error, as was also proposed in [13]. In DV, each base classifier receives a weight that is proportional to the estimated local accuracy of the base classifier, and the final classification is produced by combining the votes of each classifier with their weights. The reliability-based weighted voting (RBWV) introduced in [7] is another example of dynamic voting. It uses a model-dependent estimation of the reliability of predictions for each particular instance instead of local accuracy as the weights. In DVS, the base classifiers with highest local classification errors are discarded (the classifiers with errors that fall into the upper half of the error interval of the base classifiers) and locally weighted voting (DV) is applied to the remaining base classifiers.

In our experiments with ensemble feature selection (Section 5), we compare two commonly used static integration techniques: static selection with cross validation (SS) and weighted voting (WV) with three dynamic integration techniques: dynamic selection (DS), dynamic voting (DV), and dynamic voting with selection (DVS).

4. Measures of the ensemble diversity

There are a number of ways to quantify the ensemble diversity. In this section we consider five different measures of the ensemble diversity: plain disagreement, fail/non-fail disagreement, the Q statistic, the correlation coefficient, and the kappa statistic. They all are pairwise as they are able to measure diversity in predictions of a pair of classifiers. The total ensemble diversity is the average of all the classifier pairs in the ensemble. In our experiments in this paper, we compare the use of the pairwise measures as the guiding diversity in the ensemble training process (as a component of the fitness function (1)), and consider the correlation of the total average ensemble diversity with the difference between the ensemble accuracy and the average base classifier accuracy.

4.1. The plain disagreement measure

The plain disagreement measure is probably the most commonly used measure for diversity in the ensembles of classifiers with crisp predictions. For example, in [14] it was used for measuring the diversity of decision forests, and its correlation with the forests' accuracy. In [33,35] it was used as a component of the fitness function guiding the process of ensemble construction.

For two classifiers i and j, the plain disagreement is equal to the proportion of the instances on which the classifiers make different predictions:

$$\text{div\_plain}_{i,j} = \frac{1}{N}\sum_{k=1}^{N} \text{Diff}(C_i(\mathbf{x}_k), C_j(\mathbf{x}_k)), \tag{2}$$

where N is the number of instances in the data set, $C_i(\mathbf{x}_k)$ is the class assigned by classifier i to instance k, and $\text{Diff}(a,b) = 0$, if $a = b$, otherwise $\text{Diff}(a,b) = 1$.

The plain disagreement varies from 0 to 1. This measure is equal to 0, when the classifiers return the same classes for each instance, and it is equal to 1 when the predictions are always different.

4.2. The fail/non-fail disagreement measure

The fail/non-fail disagreement was defined in [30] as the percentage of test instances for which the classifiers make different predictions but for which one of them is correct:

$$\text{div\_dis}_{i,j} = \frac{N^{01} + N^{10}}{N^{11} + N^{10} + N^{01} + N^{00}}, \quad (3)$$

where N^{ab} is the number of instances in the data set, classified correctly ($a = 1$) or incorrectly ($a = 0$) by the classifier i, and correctly ($b = 1$) or incorrectly ($b = 0$) by the classifier j. The denominator in (3) is equal to the total number of instances N. (3) is equal to (2) for binary classification problems, where the number of classes is 2. It can be also shown that $\text{div\_dis}_{i,j} \leqslant \text{div\_plain}_{i,j}$, as the instances contributing to this disagreement measure form a subset of instances contributing to the plain disagreement.

The fail/non-fail disagreement varies from 0 to 1. This measure is equal to 0, when the classifiers return the same classes for each instance, or different but incorrect classes, and it is equal to 1 when the predictions are always different and one of them is correct.

4.3. The Q statistic

This measure is based on Yule's Q statistic used to assess the similarity of two classifiers' outputs [20]:

$$\text{div\_}Q_{i,j} = \frac{N^{11}N^{00} - N^{01}N^{10}}{N^{11}N^{00} + N^{01}N^{10}}, \quad (4)$$

where N^{ab} has the same meaning as in (3). For statistically independent classifiers, the expected value of Q is 0. Q varies between -1 and 1. Classifiers that tend to recognize the same objects correctly will have positive values of Q, and those which commit errors on different objects will render Q negative [20]. In the case of undefined value with division by zero, we assume the diversity is minimal, equal to 1.

In our experiments with diversity as a component of the fitness function we normalize this measure to vary from 0 to 1, where 1 corresponds to the maximum of diversity:

$$\text{div\_}Q^* = \frac{1 - \text{div\_}Q}{2}. \quad (5)$$

In [20], after comparative experiments on the UCI Breast cancer Wisconsin data set, the Phoneme recognition and the Cone-torus data sets, and two experiments with emulated ensembles (artificially generating possible cases of the base classifiers' outputs), Q was recommended as the best measure for the purposes of developing committees that minimize error, taking into account the experimental results, and especially its simplicity and comprehensibility (or the ease of interpretation). In our experiments, we show that the question of the superiority of a particular measure depends on the context of the use of diversity, and on the data being processed.

One problem, which we have noticed with this measure in our pilot studies, was its insensitivity on small data sets. For a small number of instances N^{00} is often equal to 0. Q in this case is equal to -1 (maximal diversity) no matter how big the values of N^{01} and N^{10} are, which is not a good reflection of the true differences in classifiers' outputs.

4.4. The correlation coefficient

The correlation between the outputs of two classifiers i and j can be measured as [20]

$$\text{div\_corr}_{i,j} = \frac{N^{11}N^{00} - N^{01}N^{10}}{\sqrt{(N^{11} + N^{10})(N^{01} + N^{00})(N^{11} + N^{01})(N^{10} + N^{00})}}, \quad (6)$$

where N^{ab} have the same meaning as in (3) and (4).

The numerator in (6) is the same as in (4), and for any two classifiers i and j, $\text{div\_corr}_{i,j}$ and $\text{div\_}Q_{i,j}$ have the same sign, and it can be proven that $|\text{div\_corr}_{i,j}| \leqslant |\text{div\_}Q_{i,j}|$ [20]. We normalize this measure to vary from 0 to 1 in the same way we do it for Q (4).

This measure, as well as the fail/non-fail disagreement and the Q statistic were considered among the group of 10 measures in the comparative experiments in [20].

4.5. The kappa degree-of-agreement statistic

The kappa statistic was first introduced by Cohen [6]. Let N_{ij} be the number of instances in the data set, recognized as class i by the first classifier and as class j by the second one, N_{i*} is the number of instances recognized as i by the first classifier, and N_{*i} is the number of instances recognized as i by the second classifier. Define then Θ_1 and Θ_2 as

$$\Theta_1 = \frac{\sum_{i=1}^{l} N_{ii}}{N} \quad \text{and} \quad \Theta_2 = \sum_{i=1}^{l} \left(\frac{N_{i*}}{N} \cdot \frac{N_{*i}}{N} \right), \quad (7)$$

where l is the number of classes and N is the total number of instances. Θ_1 estimates the probability that the two classifiers agree, and Θ_2 is a correction term for Θ_1, which estimates the probability that the two classifiers agree simply by chance (in the case where each classifier chooses to assign a class label randomly). The pairwise diversity $\text{div\_kappa}_{i,j}$ is defined as follows:

$$\text{div\_kappa}_{i,j} = \frac{\Theta_1 - \Theta_2}{1 - \Theta_2}. \quad (8)$$

'Kappa' is equal to 0 when the agreement of the two classifiers equals to that expected by chance, and 'kappa'

is equal to 1 when the two classifiers agree on every example. Negative values occur when agreement is less than expected by chance—that is, there is systematic disagreement between the classifiers [9]. 'Kappa' is able to track negative correlations in a similar manner to Q and correlation. We normalize this measure to vary from 0 to 1 in the same way we do for Q and correlation (4) and (6).

Dietterich [9] used this measure in scatter plots called "κ-error diagrams", where kappa was plotted against the mean accuracy of the classifier pair. κ-error diagrams, introduced in [21] are a useful tool for visualising ensembles.

5. Experimental investigations

In this section, our experiments with the five measures of diversity in ensemble feature selection are presented. First, the experimental setting is described, then the results of the experiments are presented. The experiments are conducted on 21 data sets taken from the UCI machine learning repository [4]. These data sets include real-world and synthetic problems, vary in characteristics, and were previously investigated by other researchers.

The main characteristics of the 21 data sets are presented in Table 1. The table includes the name of a data set, the number of instances included in the data set, the number of different classes of instances in the data set, and the numbers of different kinds of features included in the instances of the data set.

In [27] the use of the data sets from the UCI repository was strongly criticized. Salzberg [27] warns that any new experiments on the UCI data sets run the risk of finding "significant" results that are no more than statistical accidents. However, we do believe that nevertheless, the UCI data sets provide useful benchmark estimates that can be of great help in experimental analysis and comparisons of learning methods, even though these results must be looked at carefully before final conclusions are made.

5.1. Experimental setting

For our experiments, we used an updated version of the experimental setting presented in [33] to test the EFS_SBC algorithm (Ensemble Feature Selection with the Simple Bayesian Classification). We extended the EFS_SBC setting with an implementation of three new search strategies besides the existing HC: GA, EFSS, and EBSS (Section 2.2), and with an implementation of four new measures of diversity besides the existing plain disagreement: the fail/non-fail disagreement, the Q statistic, the correlation coefficient, and the *kappa* statistic (Section 4).

We used the simple Bayesian classification as the base classifiers in the ensembles. It has been recently shown experimentally and theoretically that the simple Bayes can be optimal even when the "naïve" feature-independence assumption is violated by a wide margin [11]. Second, when the simple Bayes is applied to the subproblems of lower dimensionalities as in the random subspacing, the error bias of the Bayesian probability

Table 1
Data sets and their characteristics

Data set	Instances	Classes	Features	
			Categorical	Numerical
Balance	625	3	0	4
Breast Cancer Ljubljana	286	2	9	0
Car	1728	4	6	0
Pima Indians Diabetes	768	2	0	8
Glass Recognition	214	6	0	9
Heart Disease	270	2	0	13
Ionosphere	351	2	0	34
Iris Plants	150	3	0	4
LED	300	10	7	0
LED17	300	10	24	0
Liver Disorders	345	2	0	6
Lymphography	148	4	15	3
MONK-1	432	2	6	0
MONK-2	432	2	6	0
MONK-3	432	2	6	0
Soybean	47	4	0	35
Thyroid	215	3	0	5
Tic-Tac-Toe Endgame	958	2	9	0
Vehicle	846	4	0	18
Voting	435	2	16	0
Zoo	101	7	16	0

estimates caused by the feature-independence assumption becomes smaller. It also can easily handle missing feature values of a learning instance allowing the other feature values still to contribute. Besides, it has advantages in terms of simplicity, learning speed, classification speed, and storage space, which made it possible to conduct all the experiments within reasonable time. It was shown [33] that only one "global" table of Bayesian probabilities is needed for the whole ensemble when the simple Bayes is employed in ensemble feature selection (for each feature of the base classifiers the corresponding probabilities from this table are simply taken). We believe that most of the findings and conclusions presented in this paper do not depend on the learning algorithm used and would be similar for most known learning algorithms.

To estimate the ensemble accuracy after ensemble feature selection, we have used random-sampling cross validation. 70 test runs of EFS_SBC are made for each search strategy, for each diversity measure, and on each data set. In each test run the data set is first split into the training set TrS, the validation set VS, and the test set TS by the stratified random sampling. In the stratified random sampling (sometimes called the proportional random sampling), a simple random sample from instances of each class is taken, so that the class distributions of instances in the resulting sets are approximately the same as in the initial data set. It was shown that such a sampling often gives better accuracy estimates than the simple random sampling [15]. Each time 60% of instances are assigned to the training set. The remaining 40% of instances of the data set are divided into two sets of approximately equal size (VS and TS). The validation set VS is used in the ensemble refinement to estimate the accuracy and diversity guiding the search process. The test set TS is used for the final estimation of the ensemble accuracy. We have used the same division of the data into the training, validation and test sets for each search strategy and guiding diversity to avoid unnecessary variance and provide better comparison.

The ensemble size S was selected to be 25. It has been shown that for many ensembles, the biggest gain in accuracy is achieved already with this number of base classifiers [3].

We experimented with the five different values of the diversity coefficient α: 0, 0.25, 0.5, 1, 2, 4 and 8, for the fitness function (1). At each run of the algorithm, we collect accuracies for the five types of integration of the base classifiers (Section 3): Static Selection (SS), Weighted Voting (WV), Dynamic Selection (DS), Dynamic Voting (DV), and Dynamic Voting with Selection (DVS). In the dynamic integration strategies DS, DV and DVS, the number of nearest neighbors (k) for the local accuracy estimates was pre-selected from the set of seven values: 1, 3, 7, 15, 31, 63, 127 ($2^n - 1$, $n = 1, \ldots, 7$), for each data set separately, if the number of instances in the training set permitted. Such values were chosen, as the nearest-neighbor procedure in the dynamic integration is distance-weighted. The distances from the test instance to the neighboring training instances were computed and used to calculate the estimates of local accuracies. As it was shown in [13], this makes the dynamic integration less sensitive to the choice of the number of neighbors, but still the local accuracy estimates are more sensitive to the small numbers of neighbors (for example, there are more chances to get different results for $k = 1$ and $k = 3$ than for $k = 7$ and $k = 9$). Heterogeneous Euclidean-Overlap Metric (HEOM) [25] was used for calculation of the distances (for numeric features, the distance is calculated using the Euclidean metric, and for categorical features the simple 0/1 overlap metric is used).

Thus, we varied the following parameters in our experiments with EFS_SBC on the 21 data sets presented in Table 1: search strategy, α, diversity, integration method, and k in dynamic integration. All the possible values of these parameters are summarized in Table 2.

To reduce the number of possible combinations of parameters and consequently the resources needed for our experiments, we conducted a separate series of preliminary experiments using the wrapper approach based on cross validation for combinations of a search strategy, guiding diversity, integration method, and data set to select the best α and k. From the experimental results we could see the best value of k depends mostly only on the integration method used and on the data set. The best α's varied with the search strategy, integration method, and data set used. After, the experiments were repeated with the pre-selected values of α and k, for each combination of search strategy and diversity (resulting in 20 different ensembles for each data set). All the five integration methods were used for those 20 ensembles resulting in 100 ensemble accuracies for each data set. Although the same data were used for the pre-selection of α and k and for the later experiments, we believe that this did not lead to overfitting due to the small number of possible values for α and k.

Besides the test-set classification accuracies of the base classifiers and the ensembles, in our final experiments, we collected other characteristics as the

Table 2
Parameters of the experiments with EFS_SBC

Parameter	Values
Search strategy	EBSS, EFSS, GA, HC
α	0, 0.25, 0.5, 1, 2, 4, 8
Diversity div_i	div_plain, div_dis, div_Q, div_corr, div_kappa
Integration method	SS, WV, DS, DV, DVS
k in DS, DV and DVS	1, 3, 7, 15, 31, 63, 127

ensemble accuracies on the validation set (to measure overfitting), the total ensemble diversity (using the five measures presented in Section 4), the ensemble coverage, and the average relative number of features in the base classifiers. For the GA strategy all the ensemble characteristics were collected after 1, 5 and 10 generations.

The test environment was implemented within the MLC++ framework (the machine learning library in C++) [16]. A multiplicative factor of 1 was used for the Laplace correction in simple Bayes as in [11]. Numeric features were discretized into ten equal-length intervals (or one per observed value, whichever was less), as it was done in [11]. Although this approach was found to be slightly less accurate than more sophisticated ones, it has the advantage of simplicity, and is sufficient for comparing different ensembles of simple Bayesian classifiers with each other, and with the "global" simple Bayesian classifier. The use of more sophisticated discretization approaches could lead to better classification accuracies for the base classifiers and ensembles, but should not influence the main findings and conclusions presented in the paper.

5.2. Correlation between the total ensemble diversity and the difference between the ensemble accuracy and the average base classifier accuracy

First, we measured correlation between the total ensemble diversity and the difference between the ensemble accuracy and the average base classifier accuracy (we shall call the later expression "ensemble improvement" thereafter). To measure the correlation we used the commonly used Pearson's linear correlation coefficient r and a non-parametric counterpart to it—Spearman's rank correlation coefficient (RCC).

We have measured the correlations for the four search strategies, EBSS, EFSS, GA and HC, and also for the simple random subspacing (RS). In the search strategies we have used all the five pairwise measures of diversity considered in Section 4.

Our first finding was that the correlation depends greatly on the data set (varying from −0.374 to 0.997) and does not depend on the search strategy used (the difference in the corresponding correlation coefficients was not more than 0.017 for every data set), so we continue our analysis only for the ensembles built with the random subspacing in this section.

In Table 3, Pearson's correlation coefficient (r) is presented that quantifies the correlation between the total ensemble diversity and ensemble improvement for the ensembles built with the random subspacing. All the five diversity measures (Section 4) are considered, corresponding to different columns of the table: (1) the plain disagreement, div_plain; (2) the fail/non-fail disagreement, div_dis; (3) the Q statistic, div_Q; (4) the correlation coefficient, div_corr; and (5) the kappa statistic, div_kappa. We measured the correlations for three major integration methods (DVS, which is the strongest of the dynamic integration methods, Weighted Voting, WV, and plain non-weighted Voting), presented in the lines of the table. Each cell corresponding to a measure of diversity and an integration method contains three values: average, maximal and minimal values of the correlation coefficients, calculated over the 21 data sets.

From our experiments, it could be seen that the r and RCC coefficients uncovered the same dependencies for the diversities in this context. In fact, for the vast majority of data sets, the difference between r and RCC was at most only 0.05 (for all the diversities), and more than 0.05 (but no more than 0.11)—only for data sets with small negative correlations. So, we present the results with r only in this paper (the same conclusions hold true for RCC as well).

The correlations change greatly with the data sets; this can be seen from the minimal and maximal values presented. Naturally, for some data sets, where ensembles are of little use, the correlations are low or even negative. For the DVS integration method (the strongest integration method), the best correlations are shown by div_plain and div_dis. Div_kappa is very close to the

Table 3
Pearson's correlation coefficient (r) for the total ensemble diversity and ensemble improvement (average, maximal and minimal values) averaged over the 21 data sets

r	div_plain	div_dis	div_Q	div_corr	div_kappa
DVS	0.534	0.534	0.383	0.436	0.492
	0.997	0.994	0.702	0.810	0.991
	−0.021	−0.011	−0.121	0.007	−0.163
WV	0.459	0.477	0.380	0.435	0.398
	0.997	0.994	0.791	0.910	0.991
	0.074	0.108	−0.019	0.040	−0.206
Voting	0.363	0.404	0.346	0.382	0.292
	0.997	0.994	0.791	0.910	0.991
	−0.312	−0.312	−0.257	−0.360	−0.369

best. Div_Q and div_corr behave in a similar way, which reflects the mentioned similarity in their formulae. Div_Q has surprisingly the worst average correlation with DVS. However, this correlation does not significantly decrease with the change of the integration method to WV and Voting, as it happens with all the other diversities (!) besides div_corr. The decrease in the correlations with the change of the integration method from DVS to WV, and from WV to Voting can be explained by the fact that DVS better utilizes the ensemble diversity in comparison with WV (as it was shown, for example, in [33]), and WV better utilizes the ensemble diversity than Voting, and, naturally, the correlation coefficients change as well. The fact that div_Q and div_corr do not decrease with the change of integration method is surprising, and, probably, can be explained by the different way of calculation of the diversity, but this needs further research. Div_dis is the best measure of diversity on average in this context.

The line with the maximal correlations corresponds to the Soybean data set. The dependency is almost perfectly linear for this data set for all the diversities besides div_Q and div_corr. This behaviour, probably, can be explained by the characteristics of the data set. Soybean is the smallest data set including 47 instances only, and is easy to learn.

5.3. Comparison of the test set accuracy for different strategies, diversity metrics, and integration methods

In Fig. 1, the ensemble accuracy is shown for different search strategies including the simple random subspacing (RS), and for the five pairwise measures of diversity used within the fitness function (1). The best integration method was selected for each data set, and the results were averaged over all the 21 data sets.

From Fig. 1, one can see that the GA was the best on average, and the HC strategy was the second best. These two strategies are based on the random subspacing (RS), which shows very good results on these data sets in itself. The RS was better than EFSS for all the diversities, and better than or equal to EBSS. Surprisingly, after it was the best for the Acute Abdominal Pain data sets in [32], EFSS was significantly worse than all the other strategies for this collection of data sets. This is more in line with the results of Aha and Bankert [1], which show that, for simple feature selection, backward sequential selection is often better than forward sequential selection as forward selection is not able to include groups of correlated features that do not have high relevance as separate features, only as a group.

There is not many significant differences in the use of the guiding diversities that can be seen on these averaged results. The only two significant dependencies seen are: (1) div_Q is significantly worse than the other diversities for EBSS, EFSS, and the GA; and (2) div_kappa is better than the other diversities for the GA. The first dependency is expected and goes in line with the results presented in Table 3. The second dependency is unexpected and can be explained, probably, by the peculiarities of the genetic search. HC is not that sensitive to the choice of the guiding diversity, as it starts with an already accurate ensemble (generated with the RS), and the search is not global as in the GA.

A similar behaviour of Q was reported by Ferri et al. [12]. In their paper, diversity was used as a measure of dissimilarity of ensemble and a base classifier, so that to select the best base classifier (called archetype) with predictions closest to those of the ensemble. Q was found to be the worst diversity measure in that context, resulting even in lower accuracy than the first single hypothesis.

In Fig. 2, the average ensemble accuracy is shown for the GA strategy with the five guiding diversities

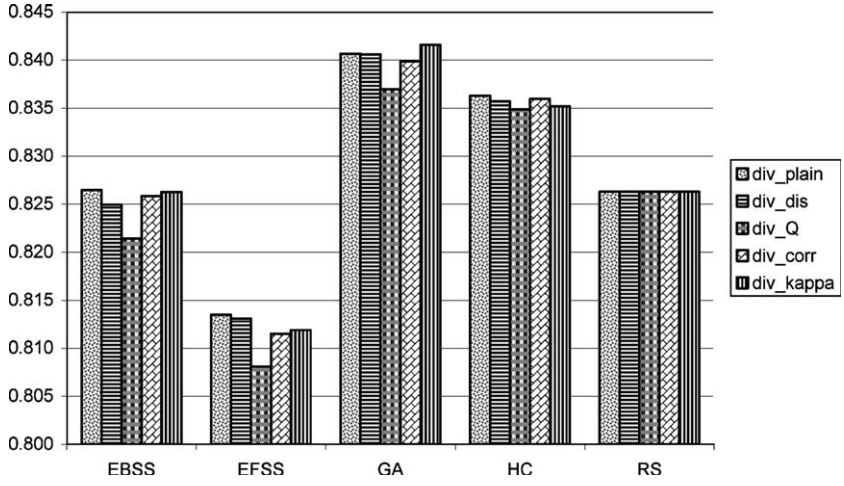

Fig. 1. Average test set accuracy for different search strategies and guiding diversities.

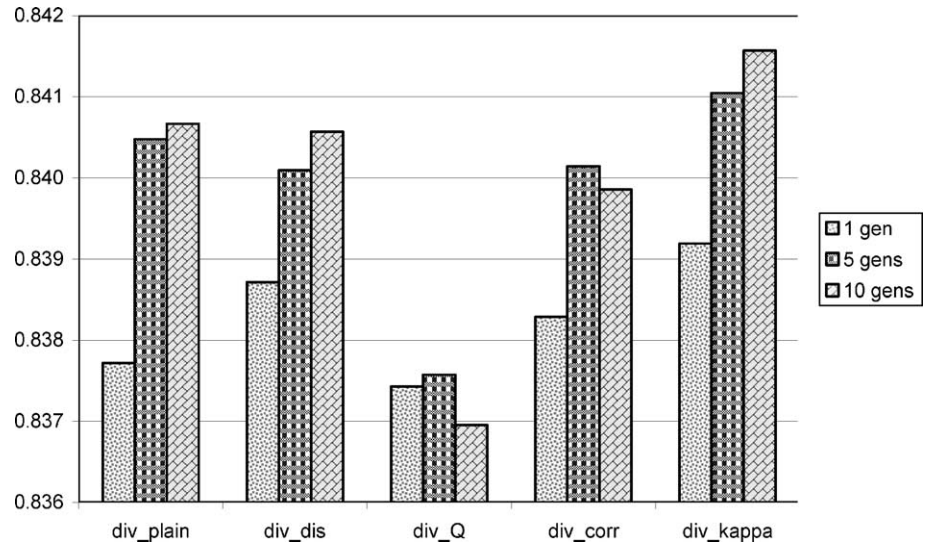

Fig. 2. Average test set accuracy for different numbers of generations in the GA search strategy with different guiding diversities.

after 1, 5 and 10 generations. From this figure, one can see that ensembles for all the diversities show improvement as the search progresses for the number of generations from 1 to 5. Then, ensembles built with div_*Q* and div_corr begin to degenerate, while the other ensembles still improve the average accuracy, though this improvement starts to tail off. This corresponds to the results presented in Table 3, where these two measures of diversity have shown the worst average correlation with the difference between the ensemble accuracy and the average base classifier accuracy for DVS. As it was also shown in the previous figure, the best diversity for the GA is div_kappa.

For each guiding diversity, for each search strategy, and for each data set we have compared the ensemble accuracy of the 70 cross-validation runs using Student's *t*-test for significance of the difference in two proportions with the level of significance 0.01. The results of this are presented in Table 4. It was shown that the resampled *t*-test and other commonly used significance tests have an unacceptably high probability of detecting a difference in generalization performance when no difference exists (Type 1 error) [27]. However, this is still the most commonly used procedure for comparing learning methods, and it does provide approximate confidence intervals that can be of great help in interpreting experimental comparisons of learning methods.

Each cell of Table 4 compares the diversity metric corresponding to the line against the metric corresponding to the column, and contains win/tie/loss information over the 21 data sets for the four search strategies correspondingly (EBSS, EFSS, GA and HC), and in total (these numbers are given in gray, bold and italic). For each line and for each column we have also calculated the total numbers, which compare the corresponding diversity against all the others, summing up the corresponding cells.

This table supports our conclusions from Fig. 1. Div_*Q* is significantly worse than the others for EBSS (10 losses and no wins of the 84 comparisons), for EFSS (11 losses and 2 wins), and for GA (8 losses and no wins). For HC there is no difference in the use of any diversity. Div_kappa is the best diversity for the GA (6 wins and no losses) and it works well for EBSS (12 wins against 3 losses), however, it is very unstable for EFSS (10 wins and 9 losses). The other differences are insignificant.

As the rest of the dependencies and conclusions presented in this section and Section 5.4 do not depend on what guiding diversity is used, we consider them only for div_plain as the guiding diversity. In Fig. 3, the average ensemble accuracies for the five integration methods (SS, WV, DS, DV and DVS) are shown in combination with the four search strategies (EBSS, EFSS, GA and HC) and the RS.

From Fig. 3, one can see that the dynamic methods always work better on average than the static methods, and DVS is the best dynamic method, that supports the conclusions presented in [32,33]. DVS for the GA shows the best accuracy, and DVS with HC is the second best. For each of the dynamic methods, the ranking of the search strategies is the same: GA, HC, EBSS, RS and EFSS.

5.4. Further interesting experimental findings

In this section we discuss briefly our further interesting findings and dependencies uncovered from the experimental results, and the results of the check of presented conclusions with division of the data sets into two groups with different numbers of features.

Table 4
Student's *t*-test results comparing guiding diversities for different search strategies on 70 cross-validation runs

Win/tie/loss EBSS EFSS GA HC total	div_plain	div_dis	div_corr	div_*Q*	div_kappa	total
div_plain		1/20/0	0/21/0	2/19/0	1/19/1	4/79/1
		0/20/1	1/19/1	3/18/0	2/17/2	6/74/4
	X	0/21/0	0/21/0	3/18/0	0/21/0	3/81/0
		0/21/0	0/21/0	0/21/0	2/19/0	2/82/0
		1/82/1	1/82/1	8/76/0	5/76/3	15/316/5
div_dis	0/20/1		0/21/0	1/20/0	1/17/3	2/78/4
	1/20/0		1/20/0	3/18/0	2/17/2	7/75/2
	0/21/0	X	0/21/0	1/20/0	0/20/1	1/82/1
	0/21/0		0/21/0	0/21/0	2/19/0	2/82/0
	1/82/1		0/83/0	5/79/0	5/73/6	12/317/7
div_corr	0/21/0	0/21/0		2/19/0	1/17/3	3/78/3
	1/19/1	0/20/1		2/19/0	3/15/3	6/73/5
	0/21/0	0/21/0	X	1/20/0	0/19/2	1/81/2
	0/21/0	0/21/0		0/21/0	2/19/0	2/82/0
	1/82/1	0/83/1		5/79/0	6/70/8	13/314/10
div_*Q*	0/19/2	0/20/1	0/19/2		0/16/5	0/74/10
	0/18/3	0/18/3	0/19/2		2/16/3	2/71/11
	0/18/3	0/20/1	0/20/1	X	0/18/3	0/76/8
	0/21/0	0/21/0	0/21/0		1/19/1	1/82/1
	0/76/8	0/79/5	0/79/5		3/69/12	3/303/30
div_kappa	1/19/1	3/17/1	3/17/1	5/16/0		12/69/3
	2/17/2	2/17/2	3/15/3	3/16/2		10/65/9
	0/21/0	1/20/0	2/19/0	3/18/0	X	6/78/0
	0/19/2	0/19/2	0/19/2	1/19/1		1/76/7
	3/76/5	6/73/5	8/70/6	12/69/3		29/288/19
Total	1/79/4	4/78/2	3/78/3	10/74/0	3/69/12	
	4/74/6	2/75/7	5/73/6	11/71/2	9/65/10	
	0/81/3	1/82/1	2/81/1	8/76/0	0/78/6	X
	0/82/2	0/82/2	0/82/2	1/82/1	7/76/1	
	5/316/15	7/317/12	10/314/13	30/303/3	19/288/29	

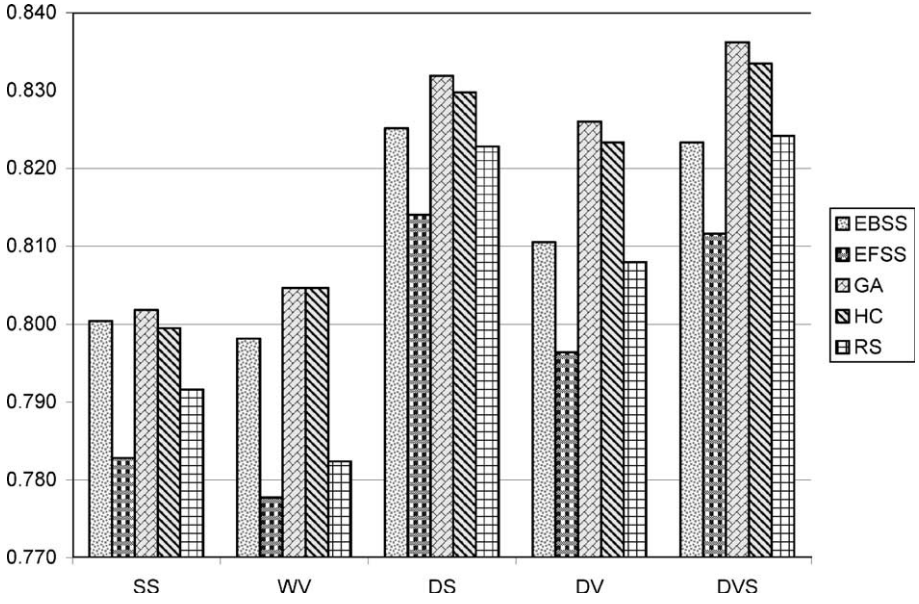

Fig. 3. Average test set accuracy for different integration methods with different search strategies.

5.4.1. Overfitting in the search strategies and in the integration methods

As a measure of overfitting, we calculated the difference in the average ensemble accuracy on the test and validation sets for the four search strategies with the DVS integration method, and for the five integration methods averaged over the search strategies. The highest difference between the test set and the validation set accuracies is with the GA and EFSS search strategies (0.038), the lowest difference in the test and validation set accuracies is with the HC search strategy (0.033), and with the EBSS search strategy the difference is equal to 0.036.

For the five integration methods, the highest average difference between the test set and validation set accuracies is with the WV integration method (0.038). Then we have DV (0.032), SS (0.031), DVS (0.029), and DS (0.027). In general, the static integration methods show more overfitting than their dynamic counterparts, probably because in dynamic integration local accuracy estimates from the nearest neighborhoods are used (many different subsets of the original data set are used for the accuracy estimation in fact), and not from one whole data set always.

5.4.2. GA and integration methods

The experimental results for the GA strategy show that the static integration methods, SS and WV, and the dynamic DS start to overfit the validation data set already after 5 generations and show lower accuracies, while the accuracies of DV and DVS continue to grow up to 10 generations. This shows the importance of selection of the appropriate integration method for the GA search strategy.

5.4.3. Average numbers of features selected for different search strategies and integration methods

The average numbers of features corresponding to the selected α values for each combination of a search strategy, integration method, and data set were calculated. Naturally, the RS selects exactly half of the features on average. Surprisingly, the GA strategy selects the biggest number of features on average, even higher than the EBSS strategy. One possible explanation for this is that, due do the global nature of its search, it is able to achieve better diversity even with classifiers including bigger feature subsets and having more features in common. As expected, the EFSS strategy selects the lowest number of features on average. As a rule, more features are needed in the static integration methods than in the dynamic ones to achieve better accuracy (42% vs 37% on average).

5.4.4. k-neighborhood for dynamic integration

DS needs bigger values of k. This can be explained by the fact that its prediction is based on only one classifier being selected, and thus, it is very unstable. Bigger values of k provide more stability to DS. The average selected k is equal to 32 for DS, and it is only 10 for DV. For DVS, as a hybrid strategy, it is in between at 22. The selected values of k do not change significantly with the search strategies.

5.4.5. Selected values of α for integration methods and search strategies

Selected values of α were different for different data sets, supporting our conclusion in [33]. Naturally, EBSS has the bigger values of α selected of all the search strategies, as it needs more diversity to achieve better ensemble accuracy, when all the classifiers include mostly identical features. It is not the case with the static integration methods, especially WV, as diversity is not that important for them, as they do not use diversity to the same extent the dynamic integration does. In general, α for the dynamic methods is much bigger for all the strategies than for the static ones (2.2 vs 0.8 on average). There is no significant difference seen for α in the EFSS, HC and GA strategies.

5.4.6. Two groups of data sets with different numbers of features

To validate the findings and conclusions presented in Sections 5.2 and 5.3, and in this section and to check the dependency of the results on the selection of the data sets, we divided all the data sets into two groups: (1) with less than 9 features (10 data sets), and (2) with greater or equal to 9 features (11 data sets); and checked all the dependencies for these two groups.

The results for these two groups supported our previously reported findings in this paper. The dependencies for the second group were even clearer than for the first group and for the total case. Two interesting phenomena that were noticed for these two groups concern measuring overfitting and the behaviour of the random subspacing.

The patterns with overfitting were the same as reported before for both of the groups, but the difference in accuracy on the validation and test sets was steadily higher in the second case (about 0.01), supporting previously reported in the literature conclusions that high-dimensional data sets are prone to overfitting to a greater extent than the data sets including less features.

The other interesting finding is that the RS with the dynamic methods (DS, DV and DVS) in the first group of data sets works relatively better than the RS with the dynamic methods in the second group (in the first case only the GA works always better than the RS, while for the second group the GA, HC and EBSS are always better than the RS for the dynamic methods). This shows that the RS works better with data sets having smaller numbers of features. However, surprisingly, with the static methods the RS is relatively better for the

second group, which is counterintuitive and may be due to the relatively lower diversity of ensembles, generated with the RS for the data sets with bigger numbers of features (there are more features in common). For the data sets with bigger numbers of features, the four search strategies are able to achieve better diversities than the RS, which is also supported by the difference between the accuracy of dynamic and static methods on the strategies: it is about 2% for the first group and 3.5% for the second one in favour of dynamic integration.

6. Conclusions

In our paper, we have considered five pairwise diversity metrics (the plain disagreement, div_plain; the fail/non-fail disagreement, div_dis; the Q statistic, div_Q; the correlation coefficient, div_corr; and the 'kappa' statistic, div_kappa), which can be used as a component of the fitness function to guide the process of ensemble training in ensemble feature selection or to measure the total ensemble diversity as a characteristic of ensemble goodness.

We consider four search strategies for ensemble feature selection: Hill Climbing (HC), a Genetic Algorithm for ensemble feature selection (GA), Ensemble Forward Sequential Selection (EFSS), and Ensemble Backward Sequential Selection (EBSS). In our implementation, all these search strategies employ the same fitness function, which is proportional to the accuracy and diversity of the corresponding base classifier. Diversity in this context can be one of the five pairwise diversities considered. To integrate the base classifiers generated with the search strategies, we use five integration methods: Static Selection (SS), Weighted Voting (WV), Dynamic Selection (DS), Dynamic Voting (DV), and Dynamic Voting with Selection (DVS).

In our experiments, first, we analyse the total ensemble diversity calculated with the considered metrics. To evaluate each measure of diversity we calculated its correlation with the difference between the ensemble accuracy and the average base classifier accuracy. The correlations did not depend on the search strategy used, and did depend significantly on the data set. The best correlations averaged over the 21 data sets were shown by div_plain, and div_dis. Div_Q and div_corr behaved in a similar way to each other, supported by the similarity of their formulae. Surprisingly, div_Q had the worst average correlation. All the correlations besides div_Q and div_corr changed with the change of the integration method, showing the different usage of diversity by the integration methods.

Then, we have compared the ensemble accuracies of the four search strategies for the five diversity metrics, and for the five integration methods being used. For all the diversities, the GA was the best strategy on average, and HC was the second best. The power of these strategies can be explained by the fact that they are based on the random subspacing. Surprisingly, EFSS was significantly worse than the other strategies for this collection of data sets. div_Q was significantly worse on average than the other diversities for EBSS, EFSS, and the GA; and div_kappa was better than the other diversities for the GA. HC was not sensitive to the choice of the guiding diversity. The results of the Student's t-test for significance supported these findings. The dynamic integration methods DS, DV and DVS always worked better on average than the static methods SS and WV, and DVS was the best dynamic method on average. The GA, div_kappa and DVS was the best combination of a search strategy, diversity measure and integration method.

To analyse overfitting in these ensembles, we have measured all the ensemble accuracies on both the validation and test sets. Among the search strategies, GA and EFSS had the biggest overfitting on average. For the integration methods, it was discovered that the static integration methods show more overfitting than their dynamic counterparts.

In [22] it was shown that besides the use of weights to combine a number of objectives in the fitness function in genetic algorithms (as the use of α in our case), another common approach that often gives better results for single feature subset selection is based on Pareto-front dominating solutions. Adaptation of this technique to ensemble feature selection is an interesting topic for further research.

In this paper we considered fixed-size ensembles. In [33] ensembles of different sizes (5, 10, 25 and 100) were analyzed for div_plain and HC search strategy. It was shown that different integration functions need different numbers of ensemble members to achieve the highest accuracy. Analysis of the optimal ensemble size for different guiding diversities and search strategies in ensemble feature selection is another important topic for future research.

Acknowledgements

This material is based upon works supported by the Science Foundation Ireland under Grant No. S.F.I.-02IN.1I111. This research is partly supported by the COMAS Graduate School of the University of Jyväskylä, Finland. We would like to thank the UCI machine learning repository of databases, domain theories and data generators for the data sets, and the MLC++ library for the source code used in this study. We are grateful to the anonymous reviewers and especially to the main editor of this special issue Dr. Ludmila I. Kuncheva for their valuable comments and constructive criticism.

References

[1] D.W. Aha, R.L. Bankert, A comparative evaluation of sequential feature selection algorithms, in: D. Fisher, H. Lenz (Eds.), Proceedings of 5th International Workshop on Artificial Intelligence and Statistics, 1995, pp. 1–7.

[2] R.E. Banfield, L.O. Hall, K.W. Bowyer, W.P. Kegelmeyer, A new ensemble diversity measure applied to thinning ensembles, in: T. Windeatt, F. Roli (Eds.), Multiple Classifier Systems, 4th International Workshop, MCS 2003, LNCS 2709, Springer, 2003, pp. 306–316.

[3] E. Bauer, R. Kohavi, An empirical comparison of voting classification algorithms: bagging, boosting, and variants, Machine Learning 36 (1,2) (1999) 105–139.

[4] C.L. Blake, E. Keogh, C.J. Merz, UCI repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html], Dept. of Information and Computer Science, University of California, Irvine, CA, 1999.

[5] C. Brodley, T. Lane, Creating and exploiting coverage and diversity, in: Proceedings of AAAI-96 Workshop on Integrating Multiple Learned Models, Portland, OR, 1996, pp. 8–14.

[6] J. Cohen, A coefficient of agreement for nominal scales, Educational and Psychological Measurement 20 (1960) 37–46.

[7] L.P. Cordella, P. Foggia, C. Sansone, F. Tortorella, M. Vento, Reliability parameters to improve combination strategies in multi-expert systems, Pattern Analysis and Applications 2 (3) (1999) 205–214.

[8] P. Cunningham, J. Carney, Diversity versus quality in classification ensembles based on feature selection, in: R.L. deMántaras, E. Plaza (Eds.), Proceedings of ECML 2000 11th European Conference on Machine Learning, Barcelona, Spain, LNCS 1810, Springer, 2000, pp. 109–116.

[9] T.G. Dietterich, An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization, Machine Learning 40 (2) (2000) 139–157.

[10] T.G. Dietterich, Machine learning research: four current directions, AI Magazine 18 (4) (1997) 97–136.

[11] P. Domingos, M. Pazzani, On the optimality of the simple Bayesian classifier under zero-one loss, Machine Learning 29 (2,3) (1997) 103–130.

[12] C. Ferri, J. Hernández-Orallo, M.J. Ramírez-Quintana, From ensemble methods to comprehensible models, in: S. Lange, K. Satoh, C.H. Smith (Eds.), Proceedings of DS 2002, 5th International Conference on Discovery Science, LNCS 2534, Springer, 2002, pp. 165–177.

[13] G. Giacinto, F. Roli, Methods for dynamic classifier selection, in: Proceedings of ICIAP '99, 10th International Conference on Image Analysis and Processing, IEEE CS Press, 1999, pp. 659–664.

[14] T.K. Ho, The random subspace method for constructing decision forests, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (8) (1998) 832–844.

[15] R. Kohavi, Wrappers for performance enhancement and oblivious decision graphs, Dept. of Computer Science, Stanford University, Stanford, USA, PhD Thesis, 1995.

[16] R. Kohavi, D. Sommerfield, J. Dougherty, Data mining using MLC++: a machine learning library in C++, Tools with Artificial Intelligence, IEEE CS Press, 1996, pp. 234–245.

[17] M. Kudo, J. Sklansky, Comparison of algorithms that select features for pattern classifiers, Pattern Recognition 33 (1) (2000) 24–41.

[18] L.I. Kuncheva, Genetic algorithm for feature selection for parallel classifiers, Information Processing Letters 46 (1993) 163–168.

[19] L.I Kuncheva, L.C. Jain, Designing classifier fusion systems by genetic algorithms, IEEE Transactions on Evolutionary Computation 4 (4) (2000) 327–336.

[20] L.I. Kuncheva, C.J. Whitaker, Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy, Machine Learning 51 (2) (2003) 181–207.

[21] D.D. Margineantu, T.G. Dietterich, Pruning adaptive boosting, in: Proc. 14th Int. Conf. on Machine Learning, Morgan Kaufmann, 1997, pp. 211–218.

[22] L.S. Oliveira, R. Sabourin, F. Bortolozzi, C.Y. Suen, A methodology for feature selection using multi-objective genetic algorithms for handwritten digit string recognition, International Journal of Pattern Recognition and Artificial Intelligence 17 (6) (2003) 903–930.

[23] D. Opitz, Feature selection for ensembles, in: Proc. 16th National Conf. on Artificial Intelligence, AAAI Press, 1999, pp. 379–384.

[24] N. Oza, K. Tumer, Dimensionality reduction through classifier ensembles, Computational Sciences Division, NASA Ames Research Center, Technical report NASA-ARC-IC-1999-126, 1999.

[25] S. Puuronen, V. Terziyan, A. Tsymbal, A dynamic integration algorithm for an ensemble of classifiers, in: Z.W. Ras, A. Skowron (Eds.), Foundations of Intelligent Systems: 11th Int. Symp. ISMIS'99, Warsaw, Poland, LNAI 1609, Springer, 1999, pp. 592–600.

[26] J. Reunanen, Overfitting in making comparisons between variable selection methods (Special Issue on Variable and Feature Selection), Journal of Machine Learning Research 3 (2003) 1371–1382.

[27] S.L. Salzberg, On comparing classifiers: a critique of current research and methods, Data Mining and Knowledge Discovery 1 (1999) 1–12.

[28] C. Schaffer, Selecting a classification method by cross-validation, Machine Learning 13 (1993) 135–143.

[29] A.J.C. Sharkey, On combining artificial neural nets, Connection Science, Special Issue on Combining Artificial Neural Networks: Ensemble Approaches 8 (3,4) (1996) 299–314.

[30] D.B. Skalak, The sources of increased accuracy for two proposed boosting algorithms, in: AAAI-96 Workshop on Integrating Multiple Models for Improving and Scaling Machine Learning Algorithms (in conjunction with AAAI-96), Portland, Oregon, USA, 1996, pp. 120–125.

[31] M. Skurichina, R.P.W. Duin, Bagging and the random subspace method for redundant feature spaces, in: J. Kittler, F. Roli (Eds.), Proceedings of 2nd International Workshop on Multiple Classifier Systems MCS 2001, Cambridge, UK, 2001, pp. 1–10.

[32] A. Tsymbal, P. Cunningham, M. Pechinizkiy, S. Puuronen, Search strategies for ensemble feature selection in medical diagnostics, in: M. Krol, S. Mitra, D.J. Lee (Eds.), Proceedings of 16th IEEE Symposium on Computer-Based Medical Systems CBMS'2003, The Mount Sinai School of Medicine, New York, NY, IEEE CS Press, 2003, pp. 124–129.

[33] A. Tsymbal, S. Puuronen, D. Patterson, Ensemble feature selection with the simple Bayesian classification, Information Fusion 4 (2) (2003) 87–100.

[34] A. Tsymbal, S. Puuronen, I. Skrypnyk, Ensemble feature selection with dynamic integration of classifiers, in: Int. ICSC Congress on Computational Intelligence Methods and Applications CIMA'2001, Bangor, Wales, UK, 2001, pp. 558–564.

[35] G. Zenobi, P. Cunningham, Using diversity in preparing ensembles of classifiers based on different feature subsets to minimize generalization error, in: L.D. Raedt, P.A. Flach (Eds.), Proc. ECML 2001 12th European Conf. On Machine Learning, LNCS 2167, Springer, 2001, pp. 576–587.