UNIVERSITÉ DE BLIDA 1
Faculté des Sciences
Département d'Informatique

DOCTORAL THESIS

Option : Génie des Systèmes Informatique

OPTIMAL ENSEMBLE PRUNING IN THE CONTEXT OF GAME THEORY-BASED LEARNING

by
Hadjer YKHLEF

In front of a jury composed of:

Nadjia BENBLIDIA	Professor, U. Blida 1	President
Saliha KHOUAS-OUKID	Associate Professor, U. Blida 1	Examiner
Narhimene BOUSTIA	Associate Professor, U. Blida 1	Examiner
Méziane AÏDER	Professor, USTHB, Algiers	Examiner
Djamel BOUCHAFFRA	Research Director, CDTA, Algiers	Supervisor

Blida, May 2017

To my mother.

ABSTRACT

Ensemble methodology combines multiple learning schemes in order to boost the generalization performance of a single classifier. An ensemble made of a large number of classifiers entails an increase in the computational cost, memory storage, and even a reduction in the prediction performance. Ensemble pruning has become an important task that lives up to these challenges. The thrust consists of constructing a subset that maintains or improves the accuracy of the original set of classifiers while reducing the number of its members. There has been a lot of attention given to the development of pruning techniques; however, most of them underestimate the contribution of learners which have strong discriminatory power as a group but are weak as individuals. To address this shortcoming, this thesis introduces an original approach to the ensemble pruning problem, which is founded on game theory principles.

First, we study the selection task from a Coalitional Game Theory perspective, in which a player corresponds to an individual learner and the benefits earned by the coalition members can be defined based on notions that characterize ensembles like accuracy, diversity, margin distance, or a combination of these metrics. A solution concept evaluates the base learners' contributions by considering the synergy that emanates from their interactions. However, most traditional solution concepts like Shapley value, Banzhaf index, and Nucleolus are computationally expensive, and hence are not practical for moderate and large ensemble sizes. To cope with the computational burden, we propose a new representation for simple coalitional games that admits, under some restrictions, a pseudo-polynomial time algorithm for computing Banzhaf power index. Moreover, we devise within this representation an optimal selection criterion which extracts sub-ensembles with *moderate diversities*.

Then, motivated by the positive role of *balancing diversity and accuracy*, we introduce an improved framework based on Shapley value that ranks the ensemble members according to their marginal contributions in achieving a fair balance between the individual accuracies and the ensemble diversity.

In order to evaluate the proposed methodologies, we performed extensive experimental comparisons and statistical tests with some major state-of-the-art methods such as semi-definite programming, genetic algorithm, and orientation ordering, based on a large set of UCI benchmark datasets. The results demonstrate the effectiveness of our approaches in terms of accuracy performance, pruning ratio, and computational cost.

Keywords: Ensemble pruning; Diversity; Coalitional game theory; Evolutionary game theory; Nash equilibrium; Shapley value; Banzhaf index.

ملخص

خلَال العشرِية الأَخِيرة، عرفت أنظمة المصنفَات المتعدّة تطورَاً ملحوظَاً. تهدف هذه الأَنظمة إِلَى تعزِيز أدَاء المصنف الوَاحد. يجدر بَالذكر بَأن المجموعة المكونة من عدّد هَائِل من المصنفَات تسبّب إِرتفَاع فِي التكلفة الحسَابِية، ذَاكرة التخزِين، وحتَى إِنخفَاض ملحوظ فِي الأَدَاء التعمِيمِي لـلنظَام. هذه الأَسبَاب أدت إِلَى إِقترَاح و تصمِيم عدة منَاهج و تقنِيَات لتقلِيم المجموعة (إِختِيَار فرقة فرعِية). أثبتـت الدرَاسَات أنه من المـمكن تكوِين فرقة فرعِية التِي تتمِيز بقدرة تعمِيمِية تفوق أدَاء المجموعة الأَصلِية. تهدف هذه الأَطروحة إِلَى تصمِيم عدة تقنّيَات لتقلِيم المجموعَات مُؤسّسة علَى مبَاديء نظرِية الأَلعَاب.

نتطرق أولَا إِلَى درَاسة مشكل تقلِيم المجموعَات فِي إِيطَار نظرِية الأَلعَاب التعَاونِية. نقترح منهجِيين: الأَوّل مبنِي علَى أسّس الأَلعَاب البسِيطة و مُؤشر بنزَاف؛ فِي حِين الحّل الثَانِي يقدر مكَانة أعضَاء المجموعة وفق قِيمة شَاپلِي التِي تهدف إِلَى تحقِيق توَازن عَادل بِين مفهومِين: أدَاء الأَفرَاد و تنوّع الفرقة.

من أجل تقِيِم المنهجِيَات المقترحة، أجرِينَا تجَارب و مقَارنَات مع عدة تقنِيَات شهِيرة، مستندِين فِي ذلك علَى عدّد كبِير من قوَاعد البِيَانَات القِيَاسِية و الإِختبَارَات الإِحصَائِية. أكدت النتَائِج فعَالِية التقنِيَات المقترحة من حِيث دقة الأَدَاء ، نسبة التقلِيم و التكلفة الحسَابِية.

كلمَات المفَاتِيح: تقلِيم المجموعَات؛ التنوع؛ نظرِية الأَلعَاب التعَاونِية؛ نظرِية الأَلعَاب التطورِية؛ توَازن نَاش؛ قِيمة شَاپلِي؛ مُؤشر بنزَاف.

RÉSUMÉ

Les méthodes d'ensemble combinent plusieurs apprenants afin de produire des prédictions plus précises. Un ensemble constitué d'un grand nombre de classifieurs entraîne une augmentation des coûts de calcul, de l'espace de stockage et même une réduction de la qualité de généralisation. L'élagage de l'ensemble est devenu une tâche très importante qui répond à ces défis. Ces méthodes visent à construire un sous-ensemble qui maintient ou améliore la performance de la collection initial de classifieurs tout en réduisant le nombre de membres qui le constituent. De nombreuses techniques d'élagage ont été proposées dans la littérature; cependant, la plupart d'entre eux sous-estiment la contribution des apprenants qui sont caractérisés par une capacité discriminatoire en tant que groupe, mais ils sont faibles en tant qu'individus. Afin de pallier cette faiblesse, nous introduisons une nouvelle approche d'élagage fondée sur des principes de la théorie des jeux.

Dans un premier temps, nous étudions la tâche d'élagage dans le contexte de la théorie des jeux coopératifs: un classifieur de base correspond à un joueur et les gains acquis par les membres de la coalition sont définis en fonction des notions qui caractérisent des ensembles telles que l'erreur, la diversité, ou une combinaison de ces mesures. Un concept de solution évalue les contributions des apprenants en considérant toutes les interactions possibles qui existent entre eux. Cependant, la plupart de ces solutions traditionnelles comme la valeur Shapley, l'indice Banzhaf et le Nucleolus sont très coûteux; par conséquent, elles ne sont pas pratiques pour traiter des ensembles composés d'un nombre élevé d'apprenants. Pour faire face à cette problématique, nous proposons une nouvelle représentation des jeux coopératifs qui admet, sous certaines restrictions, un algorithme pseudo-polynomial pour calculer l'indice de Banzhaf. En outre, nous concevons un critère d'élagage optimal fondé sur cette représentation qui extrait des sous-ensembles caractérisés par une *diversité modérée*.

Ensuite, motivés par le rôle positif *d'équilibrer la diversité et les performances individuelles* des apprenants, nous introduisons un framework amélioré basé sur la valeur de Shapley. Le modèle proposé évalue les utilités des classifieurs de base en fonction de leurs contributions marginales à la réalisation d'un équilibre adéquat entre les performances individuelles des apprenants et la diversité de l'ensemble.

Afin d'évaluer les modèles proposés, nous avons effectué des comparaisons expérimentales et des tests statistiques avec plusieurs méthodes d'élagage connues dans la littérature telles que Semi-Definite Programming, Genetic Algorithm, et Orientation Ordering, sur un grand nombre des bases d'apprentissage. Les résultats démontrent l'efficacité de nos approches en termes de performance en généralisation, de pourcentage d'élagage et de coût de calcul.

Mots clés: Elagage de l'ensemble ; Diversité ; Théorie des jeux coopératifs ; Théorie des jeux évolutionnaire ; Equilibre de Nash ; Valeur de Shapley ; Indice de Banzhaf.

ACKNOWLEDGEMENTS

I express my deep gratitude and sincere thanks to my supervisor Pr. Djamel Bouchaffra. I was extremely fortunate to receive his guidance, his inspiration and his endless encouragement throughout these years. His ideas, critical reviews, interpretations and suggestions have helped considerably in the fruition of my objectives.

Thanks to the reviewers Pr. Nadjia BENBLIDIA, Pr. Méziane AÏDER, Dr. Saliha KHOUAS-OUKID and Dr. Narhimene BOUSTIA for the time they spent on carefully reading the thesis, for their constructive comments and valuable suggestions which have helped me to improve the quality of this manuscript.

Last but not least, special thanks goes to Pr. Djamel Bennouar. In the end, it is thanks to him if I have had the chance to do my PhD. I would not have done it without his support. Thanks for believing in me.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

In this chapter, we present the topic of ensemble pruning, providing a short introduction to the relevant concepts and explaining the importance of the problem. Then, we motivate the choice of game theory, and state the research questions which we will tackle in the remaining chapters. Next, we summarize the primary contributions achieved during the doctoral research. Finally, we conclude with the outline of this thesis and the list of publications which have resulted from our work.

1.1 Context and problem statement

Classification is a fundamental and popular research area in machine learning and pattern recognition [3]. It is concerned with the development of learning algorithm i.e. machines that are able to learn from an exemplary set of labeled data and to generalize their behavior to new unseen instances. The resulting model, known as *classifier* and *learner*, enables us to predict the class label of an unseen sample. Decision trees, neural networks, and support vector machines are but a few examples of learning algorithms that have been successfully applied to many real-world classification problems [4, 5].

The generalization ability of a classifier is perhaps the most fundamental concepts in machine learning. Classifiers are prone to make errors and might perform poorly when tested on unseen data [6]. Several reasons can lead to such a behavior:

- A learner is provided with a finite set of training data which might not represent in full the classification problem.
- The set of the training samples is affected by noise; hence, the learned model might be biased toward a wrong representation of the problem.
- The learned model fits the training data perfectly which can sometimes lead to poor generalization performance. This phenomenon is known as *overfitting*.

- The invoked inducer is too simple and could not learn complex decision boundaries. As highlighted by *no free lunch theorem*, there exists no learning algorithm that is best suited to solve all existing classification problems [5].

An efficient strategy to address the above problems would be to test several classifiers and to select the one which makes fewer mistakes on a separate set of samples. This procedure, known as *model selection*, relies mainly on the estimation of the real generalization ability of a classifier. This estimation is a quite challenging task as it is affected by many factors such as: the nature of the problem and the size of the training set [7, 6].

Ensemble learning adopts an alternative strategy to address the above concerns [8]. A large body of literature has shown that a combination of multiple classifiers is a powerful decision making tool, and usually generalizes better than a single classifier [8, 9, 10]. Ensemble learning builds a classification model in two steps. The first step concerns the generation of the ensemble members (also called team, committee, and pool). To this end, continuous efforts have been put into the development of effective ensemble models. Particularly, BAGGING [11] and ADABOOST [12] have received the most attention from the research community, and many variations have been developed for different learning scenarios. Other popular methods such as random subspace [13, 14] and random forest [15] have also been introduced in the literature. In the second step, the predictions of the individual members are merged together to give the final decision of the ensemble using a combiner function. Major combining strategies include: majority voting [8], performance weighting [9], stacking [8], and local within-class accuracies [16].

Ensemble methods have some desirable features that have boosted the rapid growth of related research. First, every single classifier has limitations and might perform differently due to insufficient data. Amalgamating several learners can reduce the risk of choosing the wrong model and therefore making a poor prediction. Besides, some classification problems are just too complex and beyond the learning ability of a single classifier. Second, many learning algorithms adopt a search strategy to train a classifier. The lack of a large dataset reduces the actual search space and can lead to overfitting the training set. Under such circumstances, the aggregation of multiple learners expends the space considered for the problem and hence achieves better prediction performance than a single classifier. Finally, an ensemble method enables large companies that store data at hundreds of different locations to train learning models locally, and then to combine them for future predictions.

With the above advantages, ensemble learning has made great contributions to numerous real-world applications such: remote sensing [17], face recognition [18], intrusion detection [19], and information retrieval [20].

It is well-accepted that the generalization performance of an ensemble cannot be improved by amalgamating multiple identical learning models. However, an ensemble whose members make errors on different samples reaches higher prediction performance [9, 8]. This concept refers to the notion of "*diversity*" among the individual classifiers. According to Rokach [21], diversified members induce uncorrelated errors which boost the group performance globally. Although the benefits of diversity have been recognized by the ensemble learning community, no consensus on: *what diversity means?* and *how differences among component learners' predictions contribute to the overall classification accuracy?* have been established yet [22, 23, 24, 25]. Generally, diversity can be perceived as the degree of disagreement or complementarity within an ensemble [26][10]. As suggested by many authors [9, 27, 28], an ensemble composed of highly diversified members may result in a better or worse performance. In other words, diversity can be either harmful or beneficial and therefore requires an adequate quantification. Despite the lack of a formal definition of diversity, the research community has put continuous efforts on incorporating diversity in the design of ensemble methods [29, 30, 31, 27, 32]. It can be achieved implicitly by manipulating the training data or using different parameters for each base learner [33]. For instance, ADABOOST changes the distribution of the training samples; random subspace trains an ensemble of learners on different projections of the training data i.e. different feature subsets; and BAGGING bootstraps different training sets to create diversity.

Despite their remarkable success, ensemble methods can negatively affect both the *predictive performance* and the *efficiency* of the committee. First, several experimental and theoretical studies have shown that large ensembles do not always guarantee better predictive performance [31, 34, 35]. Specifically, most techniques for growing ensembles tend to generate an unnecessarily large number of classifiers in order to guarantee that the training error rate reaches its minimal value. This necessity may result in overfitting the training set, which in turn causes a reduction in the generalization ability of the ensemble. Second, a committee made of a large number of classifiers incurs an increase in memory requirement and computational cost. These costs may appear to be trivial for toy datasets; nevertheless, they can become critical for real-world applications such as learning from data stream.

All the above reasons motivate the appearance of *ensemble pruning* approaches (also called ensemble selection). Ensemble pruning aims at finding a compact and effective subset of component learners. The challenge consists of reducing the number of base learners that constitute the ensemble while maintaining or even improving the generalization power of the entire committee. Given an ensemble composed of n learners, one straightforward and naïve strategy consists of searching for a subset that best optimizes a criterion indicative of its generalization accuracy. This task involves evaluating 2^{n-2} subsets (excluding the empty set and the entire ensemble set), which becomes intractable for moderate and large ensemble sizes. This problem has been demonstrated to be NP-complete [9]. To cope with the computational burden, numerous approaches have been developed in the literature which can be categorized into two primary classes: search-based and ordering-based methods. A search-based technique performs a heuristic search in the space of all possible subsets of classifiers while measuring the importance of a candidate subset, examples of this category include: genetic algorithm [34] and semi definite programming [31]. An ordering-based approach assigns a rank to every ensemble member according to a certain criterion; then, the selection is conducted by aggregating the ensemble members whose ranks are above a predefined threshold. Kappa pruning [36] and orientation ordering [37] are two well-known ordering-based techniques. It is widely acknowledged that search-based methods provide better predictive performance than ordering-based techniques but usually require higher computational cost and memory storage [29]. Recent studies have reported that, in spite of their simplicity, ordering-based methods are competitive with search-based techniques and sometimes generalize very well [10, 38].

The criterion invoked for assessing the generalization ability of an ensemble lies at the core of any pruning methodology. It expresses either the utility of a candidate sub-ensemble or the contribution of a classifier to the overall performance. Usually, it is defined based on typical concepts that characterize ensembles such as accuracy, diversity, or even a combination of both. For instance, Meynet and Thiran [2] proposed a utility function designed to balance the ensemble accuracy and diversity based on information theory concepts. Unfortunately, many evaluation criteria are hand-designed and might sometimes require computation of large multivariate densities.

1.2 Pruning by playing a game

This thesis introduces novel selection criteria for classifier ranking, which are distinguished from other methodologies in the literature by being founded on game theory. We have pursued the following line of research:

Many existing selection criteria score the utility of the base learners according to their individual contributions to the ensemble performance. However, *this approach neglects the interactions that might exist among the ensemble members; and it therefore underestimates the contribution of learners which have strong discriminatory power as a group but are weak as individuals*. We refer to these ensemble members as *interactive classifiers*. The unintentional removal of interactive base learners can yield poor predictive performance. This consideration gives raise to the following research question: *can we derive a selection criterion which promotes interactive learners?* It is easy to notice that extracting interactive members is computationally intractable for moderate and large ensemble sizes. The next quest is therefore: *how could we overcome this intractability issue?* Once addressed, we wish to investigate several evaluation measures for ensembles such as diversity and relevancy. We also intend to examine *how such a selection criterion affects the accuracy performance, the pruning ratio, and the computational cost?*

In our endeavor to provide answers to the above questions, we have found that *coalitional game theory* offers an elegant mathematical framework that addresses our purposes very well. Coalitional game theory [39] models situations that involve interactions among decision-makers, called *players*. The focus is on the outcomes achieved by groups rather than by individuals. We call each group of players a *coalition*. A coalitional game associates to each subset of players a *payoff* which indicates the benefit earned by the coalition members if they chose to cooperate. The main assumption made in coalitional game theory is that players bind agreements on how to distribute the profits of these coalitions. Coalitional game theory further addresses the question of estimating the players' contributions by introducing a set of *solution concepts* such as: Core [40], Shapley value [41], Banzhaf power index [42], Nucleolus, and Bargaining set. The notion of a solution concept can be illustrated by the following example [43].

> *"A professor running a lab has decided to distribute the yearly bonus to his students in a manner which reflects their actual contributions to the academic success of the lab. During the year, the professor arranges the students into teams or coalitions;*

a student can join different teams. Each group publishes a paper summarizing its work. Every published paper is associated with a payoff defined based on the journal's impact factor. Given this annual data of the students' coalitions and their associated payoffs, a solution concept, in this case Shapley value, provides a fair manner to distribute the bonus to the students according to their contribution over the year."

Putting these notions into the context of multiple classifier systems, a player in the game corresponds to an individual learner and the benefits earned by a coalition of classifiers can be defined based on notions that characterize ensembles like accuracy, diversity, margin distance, or a combination of these metrics. A solution concept estimates the base learners' contributions by considering the synergy that emanates from their interactions. In this way, component learners receive ranks that reflect their real contributions to the ensemble performance.

1.3 Thesis contributions

This thesis is about developing selection criteria, specifically those which score the ensemble members by taking into consideration the synergy between them. Our main contribution is an interpretation of ensemble pruning within the context of game theory. We present in what follows a summary of the contributions; a thorough description will be provided in the Conclusion chapter (Chapter 7).

- We propose a new representation for non-monotone simple coalitional games and provide, under some restrictions, a pseudo-polynomial time algorithm for computing Banzhaf power index (Chapter 5).

- We derive an optimal selection criterion within the proposed representation, which extracts sub-ensembles with moderate diversities. We first rank the individual learners based on Banzhaf index; then, we define the pruned ensemble as the minimal winning coalition made only of the highly ranked members (Chapter 5).

- A thorough theoretical analysis of successful ensemble methods reveals an area of improvement: a committee which adequately balances accuracy and diversity yields better generalization performance. Consequently, we introduce an improved framework based on Shapley value that assigns to each classifier a rank which corresponds to its marginal contribution in achieving a fair balance between the individual accuracies and the ensemble diversity (Chapter 6).

- To test the efficacy of our approaches, we have conducted extensive experiments on a large number of problem sets. We have supported our analysis with numerous statistical comparisons.

1.4 Thesis structure

Chapter 2 introduces some relevant ensemble learning concepts that are necessary for understanding the ideas developed in this thesis. We review the supervised classification problem and present the primary ingredients required for devising successful ensemble methods.

Chapter 3 presents the literature surrounding ensemble pruning. We motivate this task and describe some major pruning methodologies introduced in the literature.

Chapter 4 briefly surveys some concepts from coalitional and evolutionary game theory. We also highlight some well-known applications of game theory to computer science in general and machine learning in particular.

Chapter 5 presents a novel representation for non-monotone simple games that admits a pseudo-polynomial time algorithm for computing Banzhaf power index. We formulate the ensemble selection problem within this framework and map the pruned ensemble to the notion of the minimal winning coalition.

Chapter 6 introduces an induced subgraph game for devising a selection criterion. We propose a novel framework that captures two intrinsic properties that affect the ensemble generalization performance namely diversity and accuracy. We weigh the base learners according to their contribution in keeping a fair balance between the individual accuracies and the ensemble overall diversity using Shapley value.

Chapter 7 concludes the thesis by summarizing our contributions and presents lines of future work.

Appendix A presents some practical guidelines for the design and the analysis of ensemble learning experiments.

1.5 Publications

The contributions presented in this thesis have resulted in several publications:

[44] Ykhlef, H., Bouchaffra, D. and Ykhlef, F., “Coalitional game-based adaboost”, IEEE International Conference on Systems, Man and Cybernetics, (2014), 194-199.

[45] Ykhlef, H. and Bouchaffra, D., “Induced subgraph game for ensemble selection”, IEEE International Conference on Tools with Artificial Intelligence, (2015), 636-643. ***Best Student Paper Award***.

[46] Ykhlef, H. and Bouchaffra, D., “An efficient ensemble pruning approach based on simple coalitional games”, Information Fusion, vol. 34, (2017), 28-42.

[47] Ykhlef, H. and Bouchaffra, D., “An induced subgraph game for ensemble selection”, International Journal on Artificial Intelligence Tools, vol. 26, no. 1,(2017), 1-20.

CHAPTER 2
ENSEMBLE LEARNING

2.1 Introduction

In the previous chapter, we briefly discussed the notion of a committee and the problem of ensemble pruning. We now introduce some relevant ensemble learning concepts that are necessary for understanding the ideas developed in this thesis. We also present much of the common notation used throughout this manuscript. We start off with a short introduction to the classification problem in Section 2.2, providing the formal definitions and describing several techniques used for model evaluation and comparison. Then, in Section 2.3, we motivate ensemble learning and give some basic concepts. We finish by presenting and discussing the main ingredients for devising a successful ensemble method in Sections 2.4-2.7.

2.2 Fundamentals of classification

Classification is considered as the most common task in machine learning and pattern recognition [3]. It is concerned with the problem of attributing class labels to unseen objects. An object (also called pattern, sample, and instance) is characterized by a *feature vector* $x \in \mathcal{X}$ and by its *class label* $y \in \mathcal{Y} = \{c_1, c_2, ..., c_k\}$. We can formally express a classification problem as a mapping from the feature space $\mathcal{X}$ to the space of class labels $\mathcal{Y}$ [48]. In supervised learning, the role of any given *classification algorithm* is to learn a predictive model from a set of m data samples $\Gamma = \{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\}$ which have been labeled beforehand, where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$. This thesis takes a statistical approach to classification by assuming that the data samples are *independently identically distributed* (i.i.d) i.e. each training example is drawn independently from the same underlying distribution. A *classification model* (also called classifier, learner, and hypothesis) is the estimated mapping function f which takes in a feature vector $x \in \mathcal{X}$, some *parameters* τ and produces an *output* $\hat{y}$.

$$\hat{y} = f(x, \tau). \tag{2.1}$$

We can distinguish between three main types of outputs [49]:

- **Class label - crisp label**: $\hat{y} \in \mathcal{Y}$.
- **Probability distribution**: The classifier returns a probability vector over the k class labels $\mu = [\mu_1, \mu_2, ..., \mu_k]^T \in [0, 1]^k$.
- **Oracle output**: It is defined as a Boolean vector $Z = [z_1, ..., z_m]^T$, where m is the size of the training set Γ, with $z_i = 1$ if the learner correctly classifies instance i, and 0 otherwise.

For the remainder of this thesis, we will focus only on learning models that outputs class labels. Usually, a classifier is seen as a two-step algorithm: training phase and testing phase. The first step concerns the task of learning a hypothesis from the training data. In the second step, the produced model is used to predict the class label of unseen objects drawn from a testing set. Neural networks, decision trees, BAGGING, ADABOOST, random subspaces, and support vector machines are but a few examples of learning algorithms, and they are all based on different paradigms. The next sections provide an extended treatment on *ensemble learning* approaches like ADABOOST and BAGGING.

The aim of any learning algorithm is to find the model parameters τ (Equation 2.1) that give the best predictive performance. We can measure the quality of the predictions in multiple ways with the most common being the *error rate* i.e. the ratio between the number of misclassified samples to the total number of samples. We will discuss model evaluation and comparison later in subsection 2.2.1.

It is assumed that the *training data* is representative of the unknown distribution; hence, a classifier that accurately predicts the training samples is expected to perform well on *testing examples*. However, a model that fits the training data perfectly, i.e. a complex decision rule, can have worse performance than a simple model with higher training error. This paradox is known as *overfitting* [48]. According to *Occam's razor principal*, the simplest model that explains most of the data is expected to perform well on unseen examples [5]. As an illustrative example, Figure 2.1 compares the decision boundaries produced by k-Nearest Neighbor (kNN) and Linear Discriminant Analysis (LDA) on a toy dataset. We set the number of nearest neighbors k to 1 and invoked LDA with the default parameters. We conducted this experiment on Iris dataset (description is provided in Appendix A) using only two attributes (Sepal width, Petal length) and two classes (Versicolor, Virginica). The linear boundary has fewer parameters and does not perfectly separate the two classes on the training data, whereas the nonlinear boundary has relatively many parameters and separates all training samples very well. However, it does perform poorly on unseen patterns, which is consistent with Occam's razor principal.

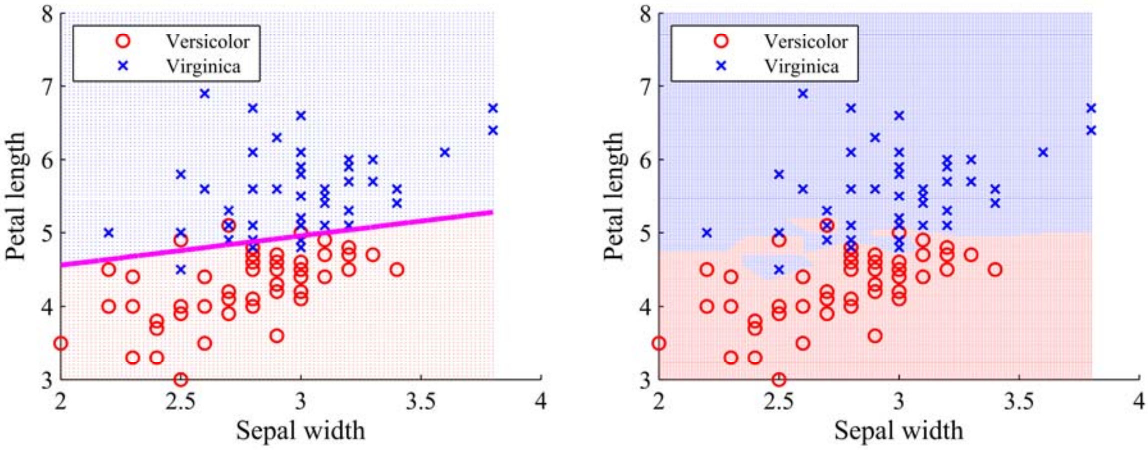


Figure 2.1: Comparison of the decision boundaries from kNN (right) and LDA (left) on Iris dataset.

Choosing the simplest model often alleviates overfitting. However, when the complexity of the classification task is not known a priori, we risk selecting a model that is too simple that can lead to poor performance i.e. *model mismatch*. Ensemble learning adopts an alternative strategy to address overfitting by *amalgamating multiple simple learners*. The combination can reduce overfitting, while providing sufficient expressive power to learn complex hypothesis [49, 9, 8].

2.2.1 Model evaluation and comparison

The goal of classification is to make use of the prior knowledge of a problem to learn a model that has the best generalization ability. According to the *no free lunch theorem* [50], there is no single learning algorithm that induces the most accurate classifier. The naturel approach is to try many learners and select the one with the best performance on a separate sample set. This task is known as *model selection*. For this purpose, we need to measure the performance of a classifier. Error rate, area under the ROC curve, precision/recall, and F-measure are examples of performance measures that are widely invoked in machine learning experiments [6]. Given a set of m labeled samples $\Gamma = \{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\}$, the error rate is defined as:

$$err(\Gamma) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{I}(\hat{y}_i = y_i), \tag{2.2}$$

where $\hat{y}_i = h(x_i)$ denotes the estimate of instance i's class label returned by classifier h. A common approach for model evaluation is to learn a hypothesis from a training set and to measure its generalization error on a test set. It is worth underscoring that the training and testing data should not overlap, otherwise the estimated performance can be overoptimistic. In addition, this approach requires a large amount of data in order to obtain a reliable estimate of the generalization error, which is rare in most situations. A possible alternative consists of invoking a *resampling technique* such as k-fold cross validation, leave-one-out, 5×2 cross validation, and 10×10 cross validation. Please refer to [51, 6] for an extended and comprehensive treatment on the subject.

Given multiple learning algorithms and datasets from various domains, model evaluation aims at identifying *which algorithm produces the most accurate classifiers when trained on samples from other domains*. This concern is one among the fundamental issues in machine learning. In order to address it, Dietterich [51], Demšar [52], García et al. [53, 54], and Japkowicz et al. [6] introduced several statistical tests such as McNemar, Friedman, Nemenyi, Bonferroni-Dunn, Wilcoxon, and ANOVA for performance comparison. In the following subsections, we briefly review the statistical tests we invoked in our experiments. Further details can be found in [52, 6].

2.2.2 Friedman test

The Friedman test is useful for comparing several algorithms over multiple domains. It first ranks the techniques for each dataset separately according to the generalization accuracy in descending order. The best performing technique gets the rank 1, the second best gets rank 2... etc. In case of ties, average ranks are assigned. Let r_i^j be the rank attributed to the j^{th} algorithm on the i^{th} dataset; and let $R_j = \frac{1}{N}\sum_{i=1}^{N} r_i^j$ denote the *average rank* of algorithm $j \in \{1, ..., t\}$ over N datasets. Under the null hypothesis, it is assumed that all techniques are equivalent; hence, their average ranks should be equal. The statistic

$$\chi_F^2 = \frac{12N}{t(t+1)} \left[\sum_{j=1}^{k} R_j^2 - \frac{t(t+1)^2}{4} \right] \tag{2.3}$$

follows chi-squared distribution with $t-1$ degrees of freedom for sufficiently large N and t (usually $N > 10$ and $t > 5$). In their study, Iman and Davenport reported that χ_F^2 is conservative and derived a new statistic:

$$F_F = \frac{(N-1)\chi_F^2}{N(t-1) - \chi_F^2}, \tag{2.4}$$

which is distributed according to the F-distribution with $t-1$ and $(t-1)(N-1)$ degrees of freedom.

This test provides only an assessment whether the observed differences in the performances are statistically significant. In order to have a zoomed-in view of what these differences correspond to precisely i.e. identify pairs of techniques with significant different performances, usually we perform a post hoc test when Friedman test rejects the null hypothesis. Nemenyi, Bonfferoni-Dunn, and Holm are examples of post hoc tests that are widely used in conjunction with Friedman test.

2.2.3 Nemenyi test

This test is invoked when all techniques are compared with each other. The performance of two methods is significantly different if their corresponding average ranks differ by at least the critical difference

$$CD = q_\alpha \sqrt{\frac{t(t+1)}{6N}}, \tag{2.5}$$

where the critical value q_α is defined based on the Studentized range statistic divided by $\sqrt{2}$.

2.2.4 Bonferroni-Dunn test

Generally, Bonferroni-Dunn test is undesirably conservative and has little power; nevertheless, this test is useful when we are only interested in comparing all techniques with a control algorithm. In this specific case, Bonferroni-Dunn test is more powerful than Namenyi test because this latter adjusts the critical value for making $t(t-1)$ comparisons, whereas when comparing with a control method we make only $t-1$ comparisons. This test is basically defined similarly to Nemenyi test except that we estimate the critical value for $\alpha/(t-1)$ significance level.

2.2.5 Wilcoxon signed-ranks test

Wilcoxon signed-ranks test is a non-parametric alternative to the paired *t-test* and is considered the best strategy to compare two algorithms over multiple domains. The formulation of this test is the following. We designate by d_i the differences between the performance scores of two techniques on N datasets, $i \in \{1, ..., N\}$. We first rank these differences according to their absolute values; in case of ties average ranks are attributed. Then, we compute the sum of ranks for the positive and the negative differences, which are denoted as R^+ and R^-, respectively. Their formal definitions are given by:

$$R^+ = \sum_{d_i>0} rank(d_i) + \frac{1}{2}\sum_{d_i=0} rank(d_i) \qquad R^- = \sum_{d_i<0} rank(d_i) + \frac{1}{2}\sum_{d_i=0} rank(d_i). \tag{2.6}$$

Notice that the ranks of $d_i = 0$ are split evenly between R^+ and R^-. Finally, the statistics T_w is computed as $T_w = min(R^+, R^-)$. For small N, the critical values for T_w can be found in any textbook on general statistics [6], whereas for larger N, the statistics

$$z = \frac{T - \frac{1}{4}N(N+1)}{\sqrt{\frac{1}{24}N(N+1)(2N+1)}} \tag{2.7}$$

follows the normal distribution with 1 mean and 0 variance. For instance, the hypothesis which states that two approaches perform equally is rejected if $z \leq -1.96$ at a 5% significance level.

2.3 Ensemble learning

Ensemble methodology imitates our second nature to seek several opinions before making a crucial decision [55]. It refers to the process of creating a *collection* (also called team, committee, ensemble, and pool) of learning models whose predictions are merged together to produce the final decision. It is also known as learning **multiple classifier systems** and **committee-based learning**. Numerous experimental and theoretical studies have demonstrated that a combination of multiple learning models reaches higher prediction performance and usually generalizes better than a single classifier [1, 34, 8]. Instead of looking for the most appropriate learning algorithm, the aggregation of several classifiers avoids the risk of choosing the wrong model. In this way, ensemble learning provides a solution to address the model mismatch problem by making best use of the strength of the individual learners and making up their weaknesses. In the following subsections, we will motivate the ensemble methodology and will provide an overview of its main concepts.

2.3.1 Motivations

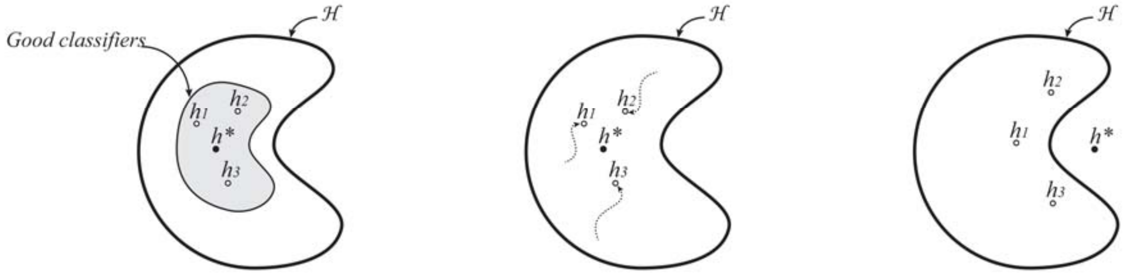
Ensemble learning has grown into an active area of research because of several theoretical and practical reasons. In the seminar paper [1], Dietterich has set three theoretical grounds for ensemble methodology (Figure 2.2[1]).

Statistical reason: Suppose that we are given a labeled sample set Z and a number of different classifiers with identical predictive performance (Figure 2.2 (a)). Although these learners are indistinguishable with respect to their error rates, they can generalize differently. Without any prior knowledge on the problem, there is no basis for picking one classifier over another. This scenario occurs when the experimental data is not sufficient to reach any clear-cut decision. A safer option would be to aggregate all classifiers predictions instead of selecting just one learner. Therefore, the combination reduces the risk of choosing an inadequate single classifier.

Computational reason: Some learning algorithms perform a local search to train a classifier, which might converge to a local optimum instead of the global optimum $h^\star$ (Figure 2.2 (b)). By suitably combining these models, we might escape the local optima and moreover get a better approximation of $h^\star$ than any of the individual learners.

Representational reason: It is possible that the true function $h^\star$ cannot be represented by an individual learning model (Figure 2.2 (c)). For instance, a single neural network can learn complex boundaries very well. However, the task of tuning its parameters requires a large set of samples. The lack of a large dataset reduces the actual search space and can lead to overfitting the training set. Under such circumstances, the aggregation of simple learners expands the search space considered for the problem and can achieve a better approximation of the true unknown function $h^\star$ than a single classifier of high complexity.

[1] The notation in the figures were changed so as to be consistent with the thesis.



(a) Statistical reason (b) Computational reason (c) Representational reason

Figure 2.2: Ensemble methodology motivations [1]. $\mathscr{H}$ denotes the space of all possible hypothesis, and $h^{\star}$ represents the optimal classifier for the task.

For practical reasons, real-world applications often require learning from large datasets. A large set of samples can be partitioned into several smaller subsets. An individual learner is trained for each subset; then, the final decision is given by aggregating the predictions of these models. Ensemble learning has also been applied to data fusion problems [56], in which data come from different sources with heterogeneous features. Each ensemble member is specialized on a portion of the feature space. As a consequence, the combination covers the whole feature space.

2.3.2 Basic notions

In this thesis, we adopt the ensemble architecture depicted by Figure 2.3. An ensemble Ω is composed of a number of classifiers $h_1, h_2, ..., h_n$ called *base learners* that are generated from training data $\Gamma = \{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\}$ using a learning algorithm. Decision trees have widely been used due to their remarkable success reported in the literature [57, 33]. Other learning algorithms such as neural network, support vector machine, and naïve bayes have also been investigated [58, 34]. Most techniques invoke a single learning algorithm for growing ensembles leading to *homogenous base learners*; there are also methods that invoke multiple learning algorithms to produce *heterogeneous base learners*. We refer to base learners as component learners, individual learners, and weak learners. Given a feature vector x, the ensemble Ω combines the predictions of its members $h_1(x), ..., h_n(x)$ following a *combination strategy*, which is responsible for turning the classifiers' private judgments into a collective decision.

2.3.3 Taxonomy of ensemble methods

An ensemble method is usually characterized by four elements [33]:

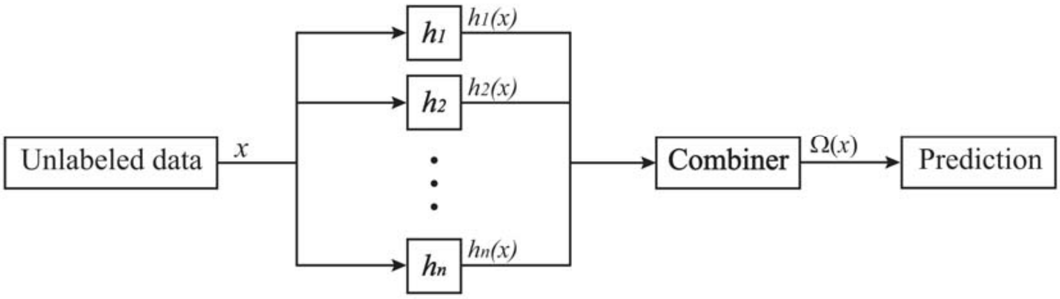


Figure 2.3: A common ensemble architecture.

Combination method: This component is responsible for combining the predictions of the ensemble members.

Ensemble generator: This element specifies the process of training the ensemble members.

Diversity: The notion of diversity is deemed to be of paramount importance in ensemble learning. Diverse members induce *uncorrelated errors* when tested on the same sample set. According to Dietterich [1], the key success of an ensemble technique is to construct a committee of individual learners that are diverse and accurate. However, measuring, defining, and properly incorporating diversity into the ensemble learning process are still matters of ongoing research [25, 22, 23].

Ensemble size: An important aspect of an ensemble approach is to define the number of component learners that should be generated. This parameter can be set by the user, determined during the training process or after the generation of the ensemble by removing the undesirable members.

The following sections provide additional details on these four elements.

2.4 Combiner method

Many fusion methods have been studied for the purpose of ensemble learning [21]. The choice of a combiner is strongly affected by the level of information provided by the ensemble members' predictions [49]. There exist two main methods for combining classifiers' outputs: *weighting* and *meta-learning* methods. The weighting methods work well on problems in which all component learners perform the same task and have comparable predictive accuracy; for instance, simple majority vote, weighted majority vote, Dempster-Shafer, and naïve bayes have been frequently used in the literature

[33]. Meta-learning methods are best suited for problems in which some members consistently correctly classify or misclassify certain samples. Examples of this strategy include stacking, grading, and mathematical programming. In our study, we focus on classifiers that directly estimate the class label of an unseen sample (crisp label) and on weighting fusion methods.

We now recall the supervised classification scenario. A sample is a pair (x, y) consisting of a feature vector $x \in \mathcal{X}$ and the true class label $y \in \mathcal{Y} = \{c_1, c_2, ..., c_k\}$. We denote with $\Omega = \{h_1, h_2, ..., h_n\}$ a collection of n base learners. Given an unseen sample x, a fusion rule combines the individual members' outputs to produce the ensemble joint decision $\Omega(x)$.

2.4.1 Majority vote

Majority vote decides for class $c_i \in \mathcal{Y}$ that obtains the highest number of votes among the individual learners. It is given by:

$$\Omega(x) = \operatorname*{argmax}_{c_i \in \mathcal{Y}} \sum_{j=1}^{n} \mathbb{I}(h_j(x) = c_i). \tag{2.8}$$

2.4.2 Weighted majority vote

Weighted majority vote rule assigns to each ensemble member h_i a weight w_i; the higher the weight, the stronger the classifier's output will affect the ensemble final decision. It is defined as:

$$\Omega(x) = \operatorname*{argmax}_{c_i \in \mathcal{Y}} \sum_{j=1}^{n} w_j \times \mathbb{I}(h_j(x) = c_i). \tag{2.9}$$

In theory, weighted majority vote can be more accurate than simple majority vote [8]. However, its performance relies considerably on the actual weights: a bad choice can cause a sharp drop in the predictive performance. For example, *performance weighting* strategy weighs the base learners according to their individual accuracies estimated on a separate set of samples [33].

2.5 Ensemble generator

There are two main strategies for building multiple base learners. We can either train the same classification model on different training sets — homogenous base learners, or we can train different classification models on the same training set — heterogeneous base learners.

The first strategy is the foundation of most successful ensemble methods like BAGGING, ADABOOST, and random subspace [49, 8]. These methods can be distinguished according to which extent each member affects the other members [9]. Some techniques generate a collection of *dependent base learners*: the outcome of a member affects the training of the next member. Therefore, it is possible to use the knowledge acquired in the previous iterations to train the next base learner. Some of established methods for growing dependent classifiers are ADABOOST and ARC-X4 [49]. Alternatively, in an *independent framework*, the dataset is partitioned into several subsets from which multiple base learners are trained. The obtained subsets can be disjointed (resampling without replacement) or overlapping (resampling with replacement). Methods that implement this methodology include BAGGING [11], random forest [15], and random subspace [32, 13, 14].

The second strategy can be achieved by amalgamating different types of learning algorithms, or by changing the parameters of the individual learners. As an illustrative example, we can combine neural networks, decision trees, support vector machines, and naïve bayes, or simply combine support vector machines with different kernel functions or different cost parameters [58].

2.5.1 ADABOOST

ADABOOST, short for "ADAPTIVE BOOSTING", was initially proposed by Freund and Schapire [12] as an ensemble method for improving the performance of a weak learner i.e. a classifier that performs better than random guessing such as decision trees. ADABOOST is a sequential algorithm in which each new inducer is built by taking into account the performance of the previously trained ensemble members. At stage t, every training sample x_i receives a weight $W_i^{(t)}$ that indicates its probability of being selected to train a new weak classifier. The first classifier is built by setting these weights to $1/m$, where m denotes the number of training samples, i.e. all samples initially have the same importance. If a training sample is correctly classified, then its chance of being reused in the next stage is decreased; conversely, if a sample is misclassified, then its chance of being reselected is increased. In this way, the subsequent classifiers focus on examples that are difficult to classify. ADABOOST assigns to the new trained classifier a weighting coefficient α_t: accurate members receive higher weights. This process continues until the desired number of base learners or the overall accuracy has been reached. The final classification decision of a test sample is based on the weighted linear combination of these weak classifiers.

ADABOOST was initially proposed for binary classification problems and then extended for multiple classes. Figure 2.4 shows ADABOOST.M1, which is the most straightforward multiclass extension of ADABOOST [49]. Several other ADABOOST extensions such as ADABOOST.M2, Real ADABOOST, Float ADABOOST, and SABOOST have been proposed in the literature. Real ADABOOST combines the class probability estimates predicted by the weak learners by fitting an additive logistic regression model in a forward stepwise manner [59]. Tsao and Chang considered ADABOOST as a stochastic approximation procedure —SABOOST [60]. They introduced a new weighting method for estimating the individual learners' contributions to the final decision.

2.5.2 BAGGING

The most well-known ensemble method for generating independent classifiers is BOOTSTRAP AGGREGATING or BAGGING for short [11]. The main idea of BAGGING is simple yet effective: First, the ensemble members are built on bootstrap replicates of the training set; then, their predictions are combined following simple majority vote strategy. Specifically, each component classifier h_i is learned from a set of instances S_i taken with replacement from the training set Γ. It is worth underscoring that the size of a bootstrap sample S_i is equal to the number of instances of the initial training set Γ. Therefore, some entries of Γ may appear more than once or may not be considered at all during the training process. To make best use of the variations of the training set, the base learners have to be *unstable* i.e. small changes in the training samples lead to large changes in the classifiers' predictions [49]. Otherwise, the trained ensemble will be composed of almost identical members. Examples of unstable classifiers are neural network and decision trees, whereas k-nearest neighbor is an example of stable learners. One advantage of BAGGING is that it can be easily implemented in a parallel mode by training the individual members on different processors. Figure 2.5 depicts the pseudocode of BAGGING.

2.6 Diversity

It is well-accepted that the generalization performance of an ensemble cannot be improved by amalgamating multiple identical learning models. However, an ensemble whose members make errors on different samples reaches higher predictive performance [8]. This concept refers to the notion of "*diversity*" among the individual classifiers. According to Rokach [9], diversified members induce uncorrelated errors which boost the group performance globally. Although the benefits of diversity have been recognized by the ensemble learning community, no consensus on: (1)*what does diversity mean?* and (2) *how differences among component learners' predictions contribute to the overall classification accuracy?* have been established yet [22, 23, 24, 25].

```
Training phase
Input:       $I$: a weak learner.
             $T$: number of iterations.
             $\Gamma$: a set of $m$ labeled training samples.
Initialize:    $t = 1$;
               $W_i^{(1)} = 1/m,\ i = 1, ..., m$;
               $\Omega = \emptyset$;
    Repeat
       —Learn a hypothesis $h_t$ from $\Gamma$ using $I$;
       $\varepsilon_t = \sum_{(x_i,y_i)\in\Gamma} W_i^{(t)} \times \mathbb{I}(h_t(x_i) \neq y_i)$;

        If $\varepsilon_t > 0.5$
           $T = t - 1$;
           Break;
        End if

       $\beta_t = \varepsilon_t/(1 - \varepsilon_t)$;

       $$W_i^{(t+1)} = \frac{W_i^{(t)}}{Z_t} \times \begin{cases} \beta_t & if \quad h_t(x_i) = y_i \\ 1 & Otherwise \end{cases},$$

       where $Z_t$ denotes a normalization constant which enables $W^{(t+1)}$ to be a distribution i.e.
    $\sum_{i=1}^{m} W_i^{(t+1)} = 1$;

       $\Omega = \Omega \cup \{h_t\}$;
       $\alpha_t = \log 1/\beta_t$;
       $t = t + 1$;
    Until $t \geq T$

Output:     The ensemble members $h_1, ..., h_T$ and their voting weights $\alpha_1, ..., \alpha_T$.

Classification phase
Input:       $x$: a feature vector characterizing a pattern.
Output:

       $$\Omega(x) = \underset{c_i \in \mathcal{Y}}{\operatorname{argmax}} \sum_{j=1}^{T} \alpha_j \times \mathbb{I}(h_j(x) = c_i).$$
```

Figure 2.4: The ADABOOST algorithm.

```
Training phase
Input:      $I$: a weak learner.
            $T$: number of iterations.
            $\Gamma$: a set of $m$ labeled training samples.
Initialize:   $t = 1$;
              $\Omega = \emptyset$;
    Repeat
        —Take a bootstrap sample $S_t$ from $\Gamma$;
        —Learn a hypothesis $h_t$ from $S_t$ using $I$;
        $\Omega = \Omega \cup \{h_t\}$;
        $t = t + 1$;
    Until $t \geq T$

Output:     $\Omega$: BAGGING ensemble.

Classification phase
Input:      $x$: a feature vector characterizing a pattern.
Output:
        $\Omega(x) = \underset{c_i \in \mathcal{Y}}{\operatorname{argmax}} \sum_{j=1}^{T} \mathbb{I}(h_j(x) = c_i).$
```

Figure 2.5: The BAGGING algorithm.

Despite the lack of a formal definition of diversity, the research community has put continuous efforts on incorporating diversity in the design of ensemble methods [29, 30, 31, 27, 27]. Rokach identified five different strategies to inject randomness into an ensemble [33]:

Manipulating the base learner: A simple strategy for creating diversity consists of changing the learning algorithm's parameters used for training the ensemble members. For instance, Islam et al. combined multiple neural networks with different number of hidden layers [61]; Drucker varied the confidence level parameter of a C4.5 and examined its impact on the performance of an ADABOOST ensemble [62].

Manipulating the training samples: This approach incorporates diversity by training each ensemble member on a different sample of the training set. ADABOOST and BAGGING are two examples that adopt this strategy. ADABOOST changes the distribution of the training samples, whereas BAGGING bootstraps different training sets to create diversity.

Manipulating the output's representation: Techniques that adopt this strategy alter the representation of the target attribute (class label). For instance, we can divide a multiclass problem into a series of binary classification problems, where each problem considers the discrimination of one class to the other classes; then, we train multiple inducers to learn these binary classification problems. Sivalingam et al. transformed a k-class problem into a minimal two-class problem using the minimal classification method along with error correcting code [63].

Partitioning the feature space: In this method, each ensemble member is provided with a different projection of the training data. For example, Ho randomly selected subsets of features to create a forest of decision trees [14].

Multi-base learning algorithms: This strategy generates diversity by aggregating different learning algorithms. Langdon et al. introduced genetic programming to learn a suitable rule for combining neural networks with decision trees [64].

2.6.1 Diversity measures

The assessment of diversity among the ensemble components is a matter of paramount importance because it would: (a) improve the understanding of how different base learners cooperate to reduce the generalization error; and (b) point out practical guidelines for the design of successful ensemble methods. However, the partial understanding of diversity in classification problems had led to the proposal of numerous measures [8, 26, 49]. Kuncheva and Whitaker have studied and analyzed ten statistics that can be classified into two categories: pairwise and non-pairwise measures [26]. Q-statstics, Cohen's kappa, and double fault are examples of pairwise measures. The ensemble overall diversity is defined as the average over all possible pairwise interactions. A non-pairwise measure can be defined, for instance, based on the correlation of each base learner with the averaged output.

In this thesis, we focus only on pairwise diversity measures. For an ensemble made of n component learners, the total diversity is defined as:

$$Div_{av} = \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} f_{i,j}, \tag{2.10}$$

where $f_{i,j}$ denotes a diversity measure between two classifiers' outputs h_i and h_j.

Let N_{ij}^{00}, N_{ij}^{10}, N_{ij}^{01}, and N_{ij}^{11} designate the number of correct/incorrect predictions made by two ensemble members h_i and h_j (Table 2.1). In what follows, we define the metrics that we find useful for the subsequent investigations.

Table 2.1: The number of correct/incorrect predictions made by a pair of classifiers.

	h_j **correct**	h_j **wrong**
h_i **correct**	N_{ij}^{11}	N_{ij}^{10}
h_i **wrong**	N_{ij}^{01}	N_{ij}^{00}

The disagreement measure is defined as the ratio between the number of samples on which one classifier is correct and its counterpart is incorrect to the total number of samples [9].

$$Dis_{i,j} = \frac{N_{ij}^{01} + N_{ij}^{10}}{N_{ij}^{00} + N_{ij}^{11} + N_{ij}^{10} + N_{ij}^{01}}. \tag{2.11}$$

The double fault measure is defined as the proportion of instances that have been misclassified by the two classifiers h_i and h_j [26].

$$DF_{i,j} = \frac{N_{ij}^{00}}{N_{ij}^{00} + N_{ij}^{11} + N_{ij}^{10} + N_{ij}^{01}}. \tag{2.12}$$

Mutual information. Brown et al. [65] used mutual information to assess the diversity between two classifiers. They proposed the following expansion: First, let X_i, X_j and Y be three discrete random variables designating the predictions of two classifiers h_i and h_j on the training set and the true class label, respectively. Then, the diversity function is given by:

$$MI_{i,j} = I(X_i; X_j|Y) - I(X_i; X_j), \tag{2.13}$$

where $I(X_i; X_j|Y)$ and $I(X_i; X_j)$ denote the conditional mutual information and the mutual information, respectively.

We can also use a similarity function to quantify diversity:

Cohen's kappa is a well-known metric to assess the agreement between two raters in statistics [66]. It was first used by Margineantu and Dietterich [36] as a measure of diversity to prune an ensemble trained by ADABOOST. Formally, Cohen's kappa between two classifiers h_i and h_j is defined as:

$$\kappa_{i,j} = \frac{\theta_{i,j} - \vartheta_{i,j}}{1 - \vartheta_{i,j}}, \tag{2.14}$$

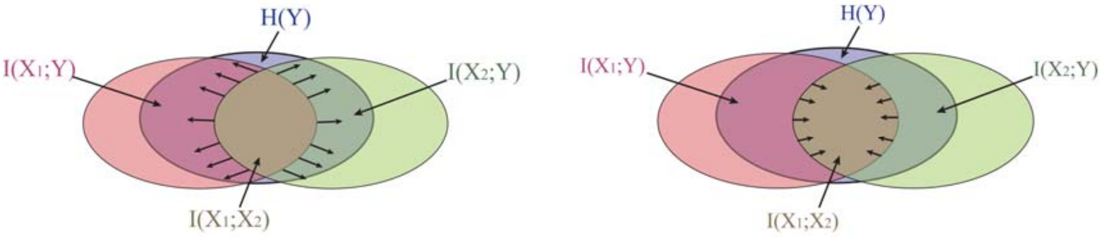
where $\theta_{i,j}$ is the proportion of samples on which both h_i and h_j make the same predictions on the training set, and $\vartheta_{i,j}$ corresponds to the probability that the two classifiers agree by chance. In this context, the diversity function can be expressed as:

$$Div - \kappa_{i,j} = \frac{1}{\kappa_{i,j} + \varepsilon}. \tag{2.15}$$

A small positive constant ε is introduced to avoid numerical difficulties when kappa statistic approaches zero.

2.6.2 Accuracy/diversity dilemma

It is a fact that two very accurate ensemble members are correlated (low diversity), whereas two weak learner, i.e. their accuracies are slightly better than random guessing, are commonly diverse [2, 38]. This phenomenon is known as *accuracy/diversity dilemma*, which can be illustrated through the following example [2]. We assume that the reader is familiar with information theory concepts; for a comprehensive treatment on this subject please refer to [67]. Let X_1, X_2, and Y be three discrete random variables designating the predictions of the two ensemble members h_1 and h_2 on the training set and the true class label, respectively. The accuracy/diversity dilemma can be summarized graphically by the Venn diagrams shown in Figure 2.6.



(a) Maximizing individual accuracies

(b) Maximizing diversity

Figure 2.6: Accuracy/diversity dilemma [2].

Figure 2.6 (a) shows that maximizing both individual learners' accuracies ($I(X_1;Y)$ and $I(X_2;Y)$) expends the intersection between the two marginal entropies; hence, high similarity between h_1 and h_2 (low diversity). Inversely, Figure 2.6 (b) reveals that minimizing similarity between the base learners decreases the individual accuracies. This paradox indicates that diversity cannot be increased without negatively affecting the ensemble performance. According to various studies [45, 38, 68, 31], a committee that adequately balances the individual accuracies and the ensemble diversity can achieve better generalization performance.

2.7 Ensemble size

Setting the number of base learners that compose the ensemble is of great importance and considerably affects the generalization performance. For instance, Opitz and Maclin have observed a decrease in the error rate of large ADABOOST ensembles made of decision trees [69]. Determining the ensemble size depends on many factors such as the desired accuracy, the computational cost, the nature of the ensemble method, and the number of available processors. Rokach described three different strategies for defining the ensemble size [33]:

- **Selection by the end user:** A straightforward and simple approach considers the ensemble size as an input parameter provided by the user. Ensemble methods such as BAGGING implement this strategy.

- **Selection while training:** Methods that adopt this strategy determine the ensemble size based on a stopping criterion. For instance, ADABOOST iteratively tests whether the contribution of the last base learner to the group performance is significant; if it is not the case, the algorithm stops.

- **Post selection (ensemble pruning):** It is sometimes useful to let the committee grow freely and then to invoke a *pruning approach* in order to extract effective and compact sub-ensembles. Ensemble pruning aims at finding a subset of component learners that maintains or improves the accuracy of the entire set of classifiers, while reducing the number of members that constitutes the committee [10]. Numerous experimental studies have demonstrated that large ensembles do not always guarantee better predictive performance [31, 70, 37]. The investigation carried out by Zhou et al. revealed that extracting a subset of learners from an ensemble composed of neural networks could improve the generalization ability [34]. Ensemble pruning task is the main concern of this thesis. In the next chapter, we will define this problem properly and will discuss some major selection methods introduced in the literature.

2.8 Summary

We reviewed the background knowledge of classification and ensemble learning that are relevant to the enquiries perused in this thesis. Four lessons can be learned from the materials covered in this chapter:

- The difficulty of assessing the generalization ability of a classifier has led to the introduction of multiple classifier systems, an approach that is based on amalgamating many diverse and accurate base learners.

- Many factors such as diversity, feature space, the nature and the number of base learners affect the generalization performance of an ensemble method.

- Diversity is deemed to be of paramount importance for the design of successful ensemble methods. However, there is no consensus on how diverse members cooperate to increase the generalization accuracy.

- A pruning approach extracts sub-ensembles that are compact and effective. It is used not only for determining the appropriate ensemble size but has other benefits as well. The literature surrounding ensemble selection will be the subject of the next chapter.

CHAPTER 3
ENSEMBLE PRUNING

3.1 Introduction

In the previous chapter, we presented the main ingredients for devising successful ensemble methods. We pointed out that the generalization ability of a committee depends on many factors, for instance: diversity, feature space, the nature and the number of base learners. Several theoretical and experimental studies have shown that the ensemble size considerably affects the generalization ability of a committee [23, 22, 69]. Furthermore, an ensemble made of a large number of classifiers entails an increase in memory storage and computational cost [25]. Ensemble pruning addresses these shortcomings by extracting a fraction of individual classifiers that maintains of even improves the predictive performance of the entire committee.

This chapter is dedicated to review the literature surrounding ensemble pruning. Section 3.2 provides a short introduction to this topic. Section 3.3 highlights the main reasons that have led to the introduction of ensemble pruning approaches. Finally, Section 3.4 explores the four principal paradigms for selecting effective sub-ensembles and describes some major state-of-the-art pruning methods.

3.2 What is ensemble pruning?

Given a set of individual learners, rather than combining all of them, ensemble pruning (also called ensemble shrinking, ensemble thinning, and ensemble selection) extracts a subset of classifiers to comprise the ensemble. The challenge consists of reducing the number of base learners composing the committee while maintaining or even improving the generalization power of the ensemble. Caruana et al. showed that pruning ensembles made of different types of base learners is more effective than taking the entire ensemble [71]. Liu et al. conducted an empirical study in order to understand how accuracy and diversity are affected by the ensemble size [38]. They showed that a smaller ensemble can be constructed while maintaining the accuracy and the diversity of the full ensemble.

Given an ensemble composed of n classifiers, finding a subset that yields the best generalization performance requires searching the space of $2^n - 2$ subsets (excluding the empty set and the entire ensemble set), which is unfeasible for large and moderate ensemble sizes. This problem has been demonstrated to be NP-complete [9], and therefore it is not practical to produce a globally optimal solution. One straightforward and naïve pruning method is to order the ensemble members according to their individual accuracies estimated on a separate sample set, and then pick the best ones [8]. Although this method may sometimes work well, it neglects other desired properties of ensembles and evaluates the utility of the individual classifiers based solely on their accuracies. As an illustrative example, Zhang et al. have shown that an ensemble composed of three identical members with 95% accuracy is worse than an ensemble of three classifiers with 67% accuracy and least pairwise correlated errors [31].

Several evaluation functions (or criteria) have been introduced in the literature in order to score the utility of classifiers. Margineantu and Dietterich ordered the ensemble members according to a diversity measure estimated based on Cohen's kappa [36]. Lu et al. [38] and Meynet et al. [2] proposed to measure each classifier's contribution by considering the accuracy/diversity trade-off. Rokach et al. [72], Arbel and Rokach [73], Quinlan [74] ranked the base learners according to their ROC performance. Windeatt and Ardeshir compared several subset evaluation functions namely Minimum Error Pruning (MEP), Error-based Pruning (EBP), Reduced-Error Pruning (REP), Critical Value Pruning (CVP) and Cost-Complexity Pruning (CCP) that were applied to ADABOOST and BAGGING ensembles [75]. The results indicate that on average EBP outperforms the other criteria.

An important issue concerns the choice of the number of classifiers to include in the pruned ensemble. Properly setting this parameter is of vital importance and considerably affects the success of a pruning method. This parameter could be determined based on numerous scoring functions such as error rate, area under ROC curve, or a diversity measure evaluated on a separate sample set [45, 76]. In [77, 37, 70, 31, 38, 68], the pruned ensemble size was considered as an input parameter provided by the user.

3.3 Why ensemble pruning?

Ensemble pruning approaches address two issues:

Efficiency: It is easy to notice that both the processing time needed to produce a prediction and the memory required for storage increase linearly with the number of classifiers in the ensemble. These costs may appear to be trivial for toy datasets; nevertheless, they can become critical for real-world applications. In fact, a large scale implementation of ensemble learning can easily generate a committee made of thousands of learning models [71]. For example, ensemble-based distributed data-mining techniques enable large companies that store data at hundreds of different locations to build learning models locally and then combine them for future prediction and knowledge discovery. Under such circumstances, the memory and computation costs are no longer trivial.

Predictive performance: Several experimental and theoretical studies have shown that large ensembles do not always guarantee better predictive performance [31, 34, 78]. Zhou et al. proved the many-could-do-better-than-all theorem which states that aggregating a subset of classifiers could achieve better generalization performance than the entire committee [8]. Additionally, most techniques for growing ensembles tend to generate an unnecessarily large number of classifiers in order to guarantee that the training error rate reaches its minimal value. This necessity may result in overfitting the training set, which in turn causes a reduction in the generalization performance of the ensemble. For instance, boosting-based technique iteratively trains base learners until the error rate becomes close to zero. In noisy settings, the generated ensemble usually overfits the training set, hence poor generalization ability.

3.4 Categorization of pruning approaches

To cope with the computational burden discussed in the previous section, numerous approaches have been introduced in the literature. The existing efforts fall into four categories [76]:

3.4.1 Ordering-based pruning

Methods of this category first assign a rank to every classifier according to an evaluation measure (or criterion); then, the selection is conducted by aggregating the ensemble members whose ranks are above a predefined threshold. The main challenge an ordering-based method faces, consists of adequately setting the criterion used for scoring every member's contribution to the ensemble performance.

Margineantu and Dietterich were the first to address the ensemble pruning problem [36]. They evaluated the utility of a classifier based on a diversity measure. Specifically, their approach first estimates the agreement between all pairs of classifiers using *kappa statistic*; then, it selects the pairs of classifiers starting with the one which has the lowest kappa statistic (high diversity), and it considers them in ascending order of their agreement until the desired number of classifiers is reached.

Accuracy ordering ranks classifiers based on their individual accuracies on a separate sample set and chooses the first N members, where N is an input parameter provided by the end user [9].

Martínez-Muñoz et al. introduced the concepts of *signature vector* associated to an individual member and the *ensemble signature vector* [77, 77]. Given a sample set Z, the signature vector $c^{(i)} \in \{-1, 1\}^{|Z|}$ of classifier h_i is defined as:

$$c_j^{(i)} = 2 \times \mathbb{I}(h_i(x_j) = y_j) - 1, \tag{3.1}$$

where $(x_j, y_j) \in Z$. The j^{th} component of $c^{(i)}$ equals 1 if h_i correctly classifies the instance x_j, and -1 otherwise. The average ensemble signature vector $\bar{c}$ is given by:

$$\bar{c} = \frac{1}{n} \sum_{i=1}^{n} c^{(i)}. \tag{3.2}$$

A sample x_j is correctly classified by the ensemble if $\bar{c}_j$ is positive. Consequently, a sub-ensemble whose average ensemble signature vector lies in the first quadrant of the $|Z|$-dimensional hyperspace, correctly classifies all the instances in Z. *Margin distance pruning* approach extracts a subset of classifiers which minimizes the distance between its average signature vector and an *objective position* p placed in the first quadrant [77]. It is easy to notice that the success of this method depends on setting the value of the vector p. Exploratory experiments indicated that low values are preferable ($p_i \sim 0.075$, for all $i \in \{1, ..., |Z|\}$).

Another technique that uses similar ideas as margin distance pruning is *orientation ordering* [37]. This method ranks the individual members according to the angle between their signature vectors and a *reference vector* c_{ref}. The reference vector is defined as the projection of the first quadrant onto the hyperplane defined by the average ensemble signature vector, which corresponds to the direction of a perfect classification performance estimated based on the sample set Z. Only classifiers whose angles are less than $\pi/2$ are chosen to compose the pruned ensemble.

3.4.2 Search-based pruning

A technique which belongs to this category performs a heuristic search in the space of all possible subsets of classifiers while optimizing an evaluation function. This function expresses the utility of a candidate sub-ensemble, and it is usually defined based on typical criteria in machine learning such as accuracy, diversity, or a combination of both. A well-known search-based pruning approach is GENETIC ALGORITHM BASED SELECTIVE ENSEMBLE (GASEN) [34]. This technique assigns a weight to each classifier: a low value indicates that the associated individual member should be excluded. Given an ensemble made of n base learners, these weights are organized as a n-dimensional vector, which corresponds to a subset in the solution space. A weight vector is initialized randomly, and then evolved toward an optimal solution following *genetic algorithm*. The fitness function is computed based on the corresponding ensemble performance on a separate sample set. Finally, pruning is conducted by discarding members whose weights are below a predefined *threshold*. A revised version of GASEN, called GASEN-B has been introduced by Zhou and Tang [79]. Instead of assigning a weight to each classifier, GASEN-B uses a bit coding scheme which directly takes $0-1$ weights and avoids the problem of setting the pruning threshold.

Zhang et al. formulated ensemble pruning as a *quadratic integer programming* problem that considers both diversity and accuracy [31]. Their approach is defined in terms of a matrix G, whose element G_{ij} represents the number of common errors between classifier h_i and classifier h_j. The diagonal term G_{ii} corresponds to the number of errors made by h_i. Normalization is applied so that the elements of the matrix are on the same scale:

$$\begin{aligned} \tilde{G}_{ii} &= \frac{G_{ii}}{m} \\ \tilde{G}_{ij,i\neq j} &= \frac{1}{2}\left(\frac{G_{ij}}{G_{ii}} + \frac{G_{ji}}{G_{jj}}\right), \end{aligned} \tag{3.3}$$

where m is the number of training instances. Thus, $\tilde{G}_{ii}$ is the error rate of classifier h_i, and $\tilde{G}_{ij}$ measures the pairwise diversity between h_i and h_j. Consequently, ensemble pruning can be formulated as a quadratic integer programming problem:

$$\min_{x} \;\; x^T \tilde{G} x, \quad s.t. \sum_{i=1}^{n} x_i = k, \quad x_i \in \{0, 1\}. \tag{3.4}$$

The binary variable x_i indicates whether classifier h_i should be selected, and k denotes the size of the pruned ensemble. Equation 3.4 is a standard binary optimization problem, which is NP-hard. In spite of the computational difficulty, Zhang et al. proposed the following relaxation: let $v_i = 2x_i - 1 \in \{-1, 1\}$,

$$V = vv^T, \; H = \begin{pmatrix} \mathbf{1}^T \tilde{G} \mathbf{1} & \mathbf{1}^T \tilde{G} \\ \tilde{G} \mathbf{1} & \tilde{G} \end{pmatrix}, \; and \; D = \begin{pmatrix} n & \mathbf{1}^T \\ \mathbf{1} & \mathbf{I} \end{pmatrix}, \tag{3.5}$$

where $\mathbf{1}$ is all-one column vector of size n and $\mathbf{I}$ denotes a $n \times n$-identity matrix, then Equation 3.4 can be rewritten as:

$$\begin{gathered} \min_{V} \;\; H \otimes V \\ s.t. \; D \otimes V = 4k, \; diag(V) = \mathbf{1}, V \succcurlyeq 0 \end{gathered} . \tag{3.6}$$

The new formulation corresponds to a convex *Semi Definite Programming* (SDP) problem which can be solved to any preset precision in polynomial time. However, this technique is still costly. Furthermore, its success considerably depends on setting the appropriate size of the pruned ensemble k.

In the same context as SDP, Xu et al. formulated ensemble selection as a combinatorial optimization problem with the goal of maximizing both accuracy and diversity [68]. Despite the fact that the original problem is computationally expensive, they derived a relaxation of the original problem into *constrained eigen-optimization*, which can be solved efficiently. Although eigen-optimization technique yields better computational costs than SDP, it still requires setting the size of the pruned ensemble, which can affect both the classification accuracy and the running time.

Rokach introduced *Collective Agreement-based ensemble Pruning* (CAP), a criterion for measuring the goodness of a candidate ensemble [29]. CAP is defined based on two terms: member-class and member-member agreement. The first term indicates how much a classifier's predictions agree with the true class label, whereas the second term measures the agreement level between two ensemble members. This metric promotes sub-ensembles whose members highly agree with the class and have low inter-agreement among each other. Note that CAP provides only a criterion for measuring the goodness of a candidate ensemble in the solution space, and hence requires defining a search strategy like best-first or directed hill climbing [76, 8].

3.4.3 Clustering-based pruning

The key idea behind this category consists of invoking a clustering technique, which allows identifying a set of representative *prototype* classifiers that compose the pruned ensemble. A clustering-based method involves two main steps. In the first step, the ensemble is partitioned into clusters, where individual members in the same cluster make similar predictions (strong correlation), while classifiers from different clusters have large diversity. For this purpose, several clustering techniques such as k-means [80], hierarchical agglomerative clustering [81], and deterministic annealing [82] have been proposed. In the second step, each cluster is separately pruned in order to increase the diversity of the ensemble. For example, Bakker and Heskes selected the individual members at the centroid of each cluster to compose the pruned ensemble [82].

An important issue concerns the choice of the number of clusters. A straightforward way consists of setting this parameter based on the classification performance of the method evaluated on a separate sample set [83]. Lazarevic and Obradovic increased the number of clusters until the disagreement among the centroids began to deteriorate [80].

3.4.4 Other methods

This category comprises the pruning approaches that do not belong to any of the above categories. Martínez-Muñoz et al. used ADABOOST to prune an ensemble trained by BAGGING [70]. Similarly to ADABOOST, boosting-based pruning is a multistage technique. At each iteration, instead of training a base learner, it selects from the pool of classifiers the member with the lowest weighted training error. If no individual learner has a weighted error less than 0.5, this approach restarts the boosting process and resets all instances' weights. Note that the weights associated to the training samples are initialized and updated similarly to the ADABOOST algorithm.

Tsoumakas et al. proposed *statistical tests* to prune an ensemble made of heterogeneous members [84]. First, their approach uses statistical procedures like Turkey and Hsu tests with the goal of identifying pairs of classifiers with significant differences; then, only the individual learners that achieve significantly better performance constitute the pruned ensemble.

Partlas et al. considered the ensemble pruning problem from a *reinforcement learning perspective* [35]. They first defined an episodic task in which an agent takes T sequential actions, each one corresponds to either the exclusion or the inclusion of an individual learner. Then, they invoked Q-learning algorithm to approximate the optimal policy for the ensemble selection task.

3.5 Summary

In this chapter, we introduced the literature surrounding ensemble pruning. In summary:

- We presented the main reasons that have motivated the appearance of pruning approaches: predictive performance, storage and computational costs.
- We reviewed the main categories of pruning techniques: ordering-based, search-based, and clustering-based. We pointed out that in spite of their simplicity, ordering-based techniques are competitive with search-based methods that are known to be effective but have high computational cost.
- We described some major state-of-the-art pruning approaches.

The main objective of this thesis is to provide a game theory-based framework for the design of powerful evaluation criteria that can be embedded into the pruning process. Before we proceed to this, we will review in the next chapter the notions from coalitional and evolutionary game theory that are necessary to understand the contributions provided in the remainder of this manuscript.

CHAPTER 4
GAME THEORY

4.1 Introduction

In the previous chapter, we reviewed several ensemble pruning approaches. Some methods invoke complex mathematical paradigms like semi-definite programming and eigen optimization, whereas others are defined based on simple concepts like Cohen's kappa. This thesis undertakes a game theory perspective for pruning a committee of learners. Game theory provides a flexible mathematical framework that can elegantly capture key criteria of the pruning task. In this chapter, we describe some game theory principles that are required for understanding the contributions discussed in the remainder of this thesis. We begin by introducing some basic notions from game theory in Section 4.2. Then, we provide a brief review of the literature surrounding coalitional and evolutionary game theory in Sections 4.3 and 4.4, respectively. Finally, we highlight in Section 4.5 some well-known applications of game theory to computer science in general and machine learning in particular.

4.2 Preliminaries

Game theory is concerned with the theory of decision making in situations of conflict and cooperation among several parties also known as players or *decision makers* [85]. It provides mathematical models (formally *games*) in order to capture key attributes of scenarios in which two or more *rational* individuals make decisions that influence one another's *welfare* either negatively or positively. A rational player has her own description of which outcomes or states of the world she prefers (it can include positive and/or negative impact on the other players), and she acts in attempt to maximize her benefits. It is worth underscoring that the term "game" is used in a technical sense of game theory: it does not refer to games like poker or chess; nevertheless, the term and some of the associated theory originate from recreational games.

A key concern in game theory consists of understanding what counts as a *rational outcome*. For this purpose, numerous *solution concepts* have been introduced in the literature [41, 42, 86]. A solution concept identifies a subset of possible outcomes of a given game while capturing some notion of rationality. Generally, solution concepts do not guarantee the existence or the uniqueness of a rational outcome. Such problems have led to the development of different solution concepts, which define distinct notions of rationality.

Game theory can be divided into two broad classes: *cooperative* and *non-cooperative*. The term “non-cooperative” could be misleading, since it suggests that the theory applies only to situations of competition between the involved parties, which is not the case. The main difference between these two classes is that in non-cooperative game theory the basic modeling unit is the individual player and there is no way to bind agreements prior to decision-making, whereas in cooperative game theory the modeling unit is the group (or coalition) and binding agreements are possible. To illustrate this difference, let us consider the Prisoner’s Dilemma game described as follows [86]:

> *“Two men A and B are arrested for a crime and held in separate cells to prevent them from communicating or binding any sort of agreement. The police lack sufficient evidence to convict either suspect and consequently require them to testify against each other. The police tell the suspects that:*
>
> - *If one confesses and the other does not, then the confessor will be released and the other will be sentenced for **three years** in prison.*
> - *If both confess, then each one will be jailed for **two years**.*
> - *If neither confesses, then each one will go to jail for **one year**.”*

The suspects have to decide whether to cooperate (not confess) or not to cooperate with each other (confess). Although the Prisoner’s Dilemma is an example from *non-cooperative* game theory, it seems like it should be *cooperative*.

Consider the following line of reasoning from suspect A’s point of view: Suppose that suspect B testifies against A; if A does not confess, then his prison term would be three years, but if he confesses, it would be only two years. Therefore, the best choice in this case is to confess. Conversely, suppose that suspect B does not testify against A; if A does not confess, then he would spend one year in prison; otherwise, he would walk free. Again, the best choice is to confess. Therefore, *no matter what B chooses, the best choice is to confess*.

Since the roles of players A and B are interchangeable, suspect B will reason in the same way about A and concludes that the best play is also to confess. In this way, the rational outcome is that both suspects are sentenced to two years in prison. However, this outcome is not the best that could be done. If both players cooperated by not confessing against each other, then they would serve one year in jail as opposed to two years. Consequently, mutual cooperation (not confessing) is strictly preferred over mutual confession by both suspects. In this case, the rational outcome is suboptimal for both suspects.

Following the above line of reasoning, *why do not both suspects cooperate by keeping quiet i.e. not confessing?* The cooperation cannot occur in the Prisoner's Dilemma because we assumed that binding agreements are not possible; hence, the suspects cannot trust each other, and they must choose the strategy that maximizes their own benefits (minimize the amount of time spent in prison) based solely on the information they have about the game.

4.3 Coalitional game theory

4.3.1 Definitions

Coalitional Game Theory (CGT) [39] models situations that involve interactions among decision-makers, called *players*. The focus is on the outcomes achieved by groups rather than by individuals. We call each group of players a *coalition*, where $\emptyset$ corresponds to the *empty coalition*, and the set of all the players is the *grand coalition*. A coalitional game associates to each subset of players a *payoff* which indicates the benefit earned by the coalition members if they chose to cooperate. The main assumptions made in CGT are that players form coalitions and bind agreements on how to distribute the profits of these coalitions. Furthermore, players receive more benefit by working together than by working individually.

Definition 4.1. A coalitional n-player game with transferable utility (TU-game) G is a pair (N, v) consisting of a finite set of players $N = \{1, 2, ..., n\}$, and a characteristic function (a.k.a payoff function) $v : 2^N \mapsto \mathbb{R}$, where 2^N denotes the set of all possible coalitions that can be formed i.e. power set $\mathscr{P}(N)$. Given a coalition $S \subseteq N$, $v(S)$ indicates its worth or the benefit that can be distributed among the coalition members.

Definition 4.2. A simple game [85] is a coalitional game where the characteristic function only assigns the values 0 or 1, i.e. $v : 2^N \mapsto \{0, 1\}$. We say that a coalition $S \subseteq N$

wins if $v(S) = 1$ and loses if $v(S) = 0$. If in a simple game $v(T) = 1 \Rightarrow v(S) = 1$ for all $T \subseteq S \subseteq N$, then the characteristic function v is said to be *monotone* [87]. In the literature, some authors refer to simple games as being strictly monotone; however, we use the term “simple games” to designate both monotone and non-monotone games.

4.3.2 Solution concepts

An outcome of a coalitional game [88] is a pair $(\mathcal{CS}, x)$ consisting of: (i) a coalition structure $\mathcal{CS} = \{S^1, S^2, ..., S^\ell\}$, such that $\bigcup_{j=1}^{\ell} S^j = N$ and $S^i \cap S^j = \emptyset$ for all $i, j \in \{1, 2, ..., \ell\}$, $i \neq j$; and (ii) a payoff vector $x = (x_i)_{i \in N}$, where x_i measures the total utility assigned to player i. A solution concept defines for each coalitional game a set of feasible outcomes. It aims at capturing two appealing properties: *fairness* and *stability*. A payoff allocation x satisfies the fairness criteria if every player receives a value that corresponds to her real contribution in the game, while stability guarantees that no subset of players has an incentive to deviate from the current coalition structure and form a coalition on their own. Famous solution concepts for characteristic function games include: Core, Shapley value, Banzhaf power index, Nucleolus, and Bargaining set. In this work, we assume that the grand coalition will form $\mathcal{CS} = \{N\}$ and focus on solution concepts that divide its total worth among its members such as Shapley value and Banzhaf index.

Shapley value

Shapley value, which was axiomatically established by Lloyd Shapley, is a well-known solution concept that defines a *fair* way of dividing the grand coalition’s payoff among its members [41]. It assigns to each player her average marginal contribution in the game, such that players who have important contributions receive greater payoff allocations.

Definition 4.3. Given a coalitional game $G = (N, v)$, Shapley value of player i, denoted $\varphi_i(G)$, is formulated as:

$$\varphi_i(G) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} [v(S \cup \{i\}) - v(S)].$$

The payoff allocated to player i corresponds to her average marginal contribution in the game. Specifically, suppose that a coalition S is formed, starting with an empty set and adding one player at a time. Within any such sequence of additions, we first compute player i’s marginal contribution ($v(S \cup \{i\}) - v(S)$); then, we multiply this quantity by $|S|!$ (the number of different ways the coalition S could have been assembled) and by $(|N| - |S| - 1)!$ (the number of different ways the remaining players could join S). Finally, we calculate *the average of these marginal contributions* by summing over all possible coalitions and by dividing by $|N|!$ i.e. the number of all possible permutations of n players.

Core

The core is the best-known solution concept for addressing the stability criterion. An outcome is stable if no coalition can obtain a payoff that exceeds the sum of its members' current payoffs [40]. As an illustrative example, let us consider a characteristic function game $G = (N, v)$ and an outcome $(\mathcal{CS}, x)$ of this game, where $\mathcal{CS} = \{S, \bar{S}\}$. In addition, suppose that $\sum_{i \in S} x_i < v(S)$. In this case, the players in S could do better by abandoning the current coalition structure $\mathcal{CS}$ and forming other coalitions of their own. Therefore, the outcome $(\mathcal{CS}, x)$ is unstable. The set of payoff allocations in which no group of players can jointly deviate to improve their payoffs, i.e. stable outcomes, forms the core of a coalitional game. Note that Shapley value assigns to every characteristic function game a unique payoff allocation, whereas the core can be an empty set [88].

Definition 4.4. The core of a coalitional game $G = (N, v)$ consists of all outcomes $(\mathcal{CS}, x)$ such that: $\sum_{i \in S} x_i \geq v(S)$ for every coalition $S \subseteq N$.

Banzhaf power index

Another solution concept that is motivated by the fairness consideration is Banzhaf power index [42]. Unlike Shapley value, Banzhaf index was primarily introduced for the purpose of measuring a player's power in a voting system i.e. the probability that she can influence the outcome of the game. In spite of this, it has also been applied to any simple coalitional game.

Definition 4.5. Given a simple coalitional game $G = (N, v)$, player i's Banzhaf index, denoted $Bz_i(G)$, is defined as:

$$Bz_i(G) = \frac{1}{2^{n-1}} \sum_{S \subseteq N \setminus \{i\}} [v(S \cup \{i\}) - v(S)].$$

Banzhaf index of non-monotone simple games has an interesting interpretation, but before analyzing it, we need to introduce two concepts: positive and negative swings.

Definition 4.6. A coalition $S \subseteq N$ is a *positive swing* for player i if $S \cup \{i\}$ wins $(v(S \cup \{i\}) = 1)$ and S loses $(v(S) = 0)$. Conversely, the coalition S is considered as a *negative swing* for player i if $v(S \cup \{i\}) = 0$ and $v(S) = 1$. Let $swing_i^+(G)$ and $swing_i^-(G)$ denote, respectively, the set of positive and negative swing coalitions for player i. They are defined as:

$$swing_i^+(G) = \{S \subseteq N \setminus \{i\} | v(S \cup \{i\}) = 1 \wedge v(S) = 0\}.$$

$$swing_i^-(G) = \{S \subseteq N \setminus \{i\} | v(S \cup \{i\}) = 0 \wedge v(S) = 1\}.$$

Since the characteristic function of a simple game is Boolean, the computation of Banzhaf power index is reduced to a counting problem. It suffices to identify all possible values of the formula $v(S \cup \{i\}) - v(S)$, count and sum them. Due to the non-monotonicity property, $v(S \cup \{i\}) - v(S)$ has three possible values: -1,$+1$, and 0. We are only interested in counting the number of ones θ_1 and negative ones θ_{-1}. Notice that θ_1 and θ_{-1} correspond to the number of positive and negative swing coalitions, respectively. Therefore, **Banzhaf power index is proportional to the difference between the number of *positive* and *negative swing* coalitions**. Formally, Banzhaf index of player i can be given by:

$$Bz_i(G) = \frac{1}{2^{n-1}} \times (|swing_i^+(G)| - |swing_i^-(G)|). \tag{4.1}$$

4.3.3 Representations of coalitional games

A straightforward representation of a coalitional game consists of enumerating the payoffs for all coalitions $S \subseteq N$. However, this naïve representation requires space exponential in the number of players $|N| = n$, which is impractical for most problems. To alleviate this tractability issue, several representation schemes for coalitional games such as marginal contribution nets [89], network flow games [90], Induced Subgraph Games (ISGs) [91], synergy coalition groups [92], and Weighted Voting Games (WVGs) [88] have been proposed in the literature. In this thesis, we consider only WVG and ISG representations.

Induced subgraph games

This representation considers a coalitional game to be played on an undirected weighted graph $\mathcal{G} = (N, E)$, in which every edge $(i, j) \in E$ is associated with a weight

$\rho_{i,j}$, we write $\boldsymbol{\rho} = (\rho_{i,j})_{(i,j)\in E}$. In the induced subgraph game $G = (\mathcal{G}, \boldsymbol{\rho})$, a node $i \in N$ corresponds to a player and the worth of a coalition $S \subseteq N$ is defined as:

$$v(S) = \sum_{\substack{(i,j)\in E \\ \{i,j\}\subseteq S}} \rho_{i,j}. \tag{4.2}$$

This formulation is concise because it is sufficient to use a $|N| \times |N|$ matrix to represent a coalitional game. Interestingly, induced subgraph games admit an efficient algorithm for computing Shapley value. Formally, given an induced subgraph game $G = (\mathcal{G}, \boldsymbol{\rho})$, player i's Shapley value is defined as:

$$\varphi_i(G) = \rho_{i,i} + \frac{1}{2} \times \sum_{\substack{(i,j)\in E \\ i\neq j}} \rho_{i,j}. \tag{4.3}$$

The proof of the above formulation can be found in [88]. In addition, when all edge weights are positive $\rho_{i,j} \geq 0 \ \forall (i,j) \in E$, induced subgraph games are guaranteed to have a non-empty core, and moreover, Shapley value belongs to the core [91]; hence, in this particular case, Shapley value satisfies both the fairness and the stability criteria.

Weighted voting games

Weighted voting games form one of the simplest useful representations of simple coalitional games. These games can be used to model settings in which each player has a certain amount of a given resource, for instance time, money, or manpower; and there is a goal that can be reached by a coalition that possesses a sufficient amount of this resource.

Definition 4.7. A weighted voting game G is defined by a set of players $N = \{1, ..., n\}$, a list of weights $\mathbf{w} = (w_1, w_2, ..., w_n) \in \mathbb{R}^n$, and a threshold $q \in R$ also known as *quota*; we write $G = (N, [\mathbf{w}, q])$. The payoff function v is given by:

$$v(S) = \begin{cases} 1 & if \ \sum_{i\in S} w_i \geq q \\ 0 & otherwise \end{cases}.$$

Usually, we assume that all the weights are positive integers. In addition, we suppose that $0 < q \leq \sum_{i\in N} w_i$, in this way the empty coalition loses and the grand coalition always wins.

Several techniques have been proposed for computing power indices for WVGs. The main three methods are: generating functions [93, 94], binary decision diagrams [95, 96, 97], and dynamic programming [98, 99]. This thesis is only concerned with the problem of estimating Banzhaf indices; however, the aforementioned approaches can be adapted to address other solution concepts like Shapley-Shubik and Deegan-Packel indices. The generating function technique formulates Banzhaf index in terms of the coefficients of a polynomial. Binary decision diagrams provide a powerful formalism for representing and analyzing the characteristic function of a WVG. For instance, S. Bolus suggested building a quasi-reduced and ordered binary decision diagram (QOBDD) of the set of winning coalitions, and devised algorithms for computing Banzhaf, Shapley-Shubik, Holler-Packel and Deegan-Packel indices of the players [95]. The dynamic programming approach computes the number of size-k coalitions that have weight w, and stores these values in a $n \times q$ matrix. Then, it uses this matrix's entries to exactly calculate a player's Banzhaf index.

4.4 Evolutionary game theory

Evolutionary Game Theory (EGT) originated from the work of the biologists Maynard Smith and Price [100, 101]. Initially, it was introduced to explain the evolutionary process related to competition over resources in nature. It deals with a large population of individuals hardwired (programmed) to play a certain type of behavior (formally a strategy in the game). First, pairs of individuals are repeatedly drawn at random from this population to initiate a two-player game; then, these individuals update their behavior according to the fitness that emanates from their repeated pairwise interactions. Strategies with high fitness values spread quickly within the population, which can be achieved either by learning, copying, inheriting, or even by infection. EGT provides a paradigmatic framework to: (1) simulate how the frequencies of strategies evolve over time, (2) determine the nature of the long-run aggregate behavior, and (3) establish connections with dynamical systems and with game theory concepts such as Nash equilibrium. In the next section, we will briefly describe the main ingredient of an evolutionary game that is symmetric two-player games. For an extended and comprehensive treatment on the subject, please refer to [39, 86].

4.4.1 Symmetric two-player game

A symmetric two-player game [86] consists of two elements: a set of n pure strategies $S = \{1, ..., n\}$ available to both players, and a payoff matrix $\mathcal{A} \in \mathbb{R}^{n \times n}$, where each entry a_{ij} corresponds to the utility obtained when one player chooses strategy i and the other plays strategy j. For instance, the payoff matrix of the Prisoner's Dilemma game discussed earlier (Section 4.2) can be described by Figure 4.1.

	NC	**C**
NC	-1	-3
C	0	-2

Figure 4.1: Prisoner's Dilemma game.

In this game, player I and II have two pure strategies each: **NC**, **C** denote **N**ot **C**onfess and **C**onfess, respectively. Each entry in this matrix a_{ij} specifies the payoff of a player when she chooses the row strategy i and her opponent picks the column strategy j (the opponent obtains the payoff a_{ji}).

A mixed strategy $\mathfrak{x} = [\mathfrak{x}_1, \mathfrak{x}_2, ..., \mathfrak{x}_n]^T$ is a probability distribution over the set S. The vector $\mathfrak{x}$ belongs to the *standard simplex* Δ in n-space, defined as $\Delta = \{\mathfrak{x} \in \mathbb{R}^n_+ : \sum_{i=1}^n \mathfrak{x}_i = 1\}$. We denote with $e^i : i \in S$ the vertices of the simplex Δ, where e^i assigns probability 1 to the i^{th} pure strategy and 0 elsewhere. The expected payoff from playing strategy $\mathfrak{x} \in \Delta$ against $\boldsymbol{y} \in \Delta$ is given by:

$$\mathbb{E}(\mathfrak{x}, \boldsymbol{y}) = \mathfrak{x}^T A \boldsymbol{y} = \sum_{i=1}^{n} \sum_{j=1}^{n} \mathfrak{x}_i a_{ij} \boldsymbol{y}_j. \tag{4.4}$$

The strategy $\mathfrak{x} \in \Delta$ is said to be a best response to $\boldsymbol{y} \in \Delta$ if $\mathfrak{z}^T \mathcal{A} \boldsymbol{y} \leq \mathfrak{x}^T \mathcal{A} \boldsymbol{y}$ for all $\mathfrak{z} \in \Delta$. One of the cornerstones of non-cooperative game theory is *Nash Equilibrium* [86] (NE). A strategy is a symmetric NE if it is a best reply to itself.

Definition 4.8. A strategy $\mathfrak{x} \in \Delta$ is a symmetric NE if and only if:

$$\mathbb{E}(\mathfrak{x}, \mathfrak{x}) \geq \mathbb{E}(e^i, \mathfrak{x}), \forall i \in S. \tag{4.5}$$

Consequently, if her opponent chooses a symmetric NE strategy $\mathfrak{x}$, a player does not have an incentive to deviate so to improve her benefits i.e. she cannot do better than to choose $\mathfrak{x}$ herself. When $\mathfrak{x} \in S$ then $\mathfrak{x}$ is known as symmetric NE in pure strategies.

Consider again the Prisoner's Dilemma game (Figure 4.1): Observe that $a_{C,NC} > a_{NC,NC}$ and $a_{C,C} > a_{NC,C}$; hence playing strategy **C** is strictly preferred than playing **NC** regardless of the opponent's strategy. We say that **C** is best reply to **NC**. In this case, both players cannot improve their payoffs by deviating from **C** and playing **NC**. As a result, strategy **C** is a symmetric NE.

4.4.2 Evolutionary stable strategies and replicator dynamics

By undertaking the evolutionary settings, we consider a large population of individuals hardwired (programmed) to play a certain strategy. Pairs of individuals are randomly engaged in repeated strategic interactions modeled as a symmetric two-player game. The payoffs under the evolutionary settings represent the *Darwinian fitness* or the *reproductive value* (the number of offspring) that results from the pairwise interactions. This matching takes place continuously over time until the population reaches an equilibrium. Let $\mathfrak{w}^{(t)} = [\mathfrak{w}_1^{(t)}, \mathfrak{w}_2^{(t)}, ..., \mathfrak{w}_n^{(t)}]^T \in \Delta$ denote the *population state* at time t, where each component $\mathfrak{w}_i^{(t)}$ corresponds to the frequency of individuals programmed to play the pure strategy $i \in S$. It is noteworthy that the concept of mixed strategy is formally identical to the population state.

The two fundamental notions of EGT are the Evolutionary Stable Strategies (ESS) and the replicator dynamics. A strategy (or state) $\mathfrak{w}^\star \in \Delta$ is ESS if *it has resistance against any small mutant invasion*. Specifically, suppose that a population at state $\mathfrak{w}^\star$ is invaded by a small fraction ε of *mutant*, whose distribution of strategies is $\boldsymbol{y} \in \Delta$. The new population state is given by $\mathfrak{z} = \varepsilon\boldsymbol{y} + (1-\varepsilon)\mathfrak{w}^\star \in \Delta$. The strategy $\mathfrak{w}^\star$ is ESS if the average fitness of the mutants $\boldsymbol{y}$ in the new population $\mathfrak{z}$ is lower than that of the incumbents $\mathfrak{w}^\star$. Since an incumbent $\mathfrak{w}^\star$ will meet another incumbent with probability $1-\varepsilon$ and it will meet a mutant $\boldsymbol{y}$ with probability ε, we can calculate the expected fitness of an incumbent, which is simply $(1-\varepsilon)\mathbb{E}(\mathfrak{w}^\star, \mathfrak{w}^\star) + \varepsilon\mathbb{E}(\mathfrak{w}^\star, \boldsymbol{y})$. Similarly, the expected fitness of a mutant is $(1-\varepsilon)\mathbb{E}(\boldsymbol{y}, \mathfrak{w}^\star) + \varepsilon\mathbb{E}(\boldsymbol{y}, \boldsymbol{y})$. Therefore, the formal definition of ESS is given by:

Definition 4.9. A strategy $\mathfrak{w}^\star$ is an ESS if there exists $\varepsilon_{\boldsymbol{y}} \in (0,1)$ for all $\boldsymbol{y} \in \Delta$, $\boldsymbol{y} \neq \mathfrak{w}^\star$:

$$(1-\varepsilon)\mathbb{E}(\mathfrak{w}^\star, \mathfrak{w}^\star) + \varepsilon\mathbb{E}(\mathfrak{w}^\star, \boldsymbol{y}) > (1-\varepsilon)\mathbb{E}(\boldsymbol{y}, \mathfrak{w}^\star) + \varepsilon\mathbb{E}(\boldsymbol{y}, \boldsymbol{y}), \forall\varepsilon \in (0, \varepsilon_{\boldsymbol{y}}).$$

The ESSs are those where the incumbents reproduce more frequently than the mutants. Consequently, the mutants will have fewer offspring, and the fraction of the mutants in the population will eventually vanish over time (the new state $\mathfrak{z}$ will get back to the initial state $\mathfrak{w}^\star$).

On the other hand, the replicator dynamics provides a simple yet efficient framework for simulating how the frequencies of strategies evolve over time, as they are involved in strategic interactions. Most importantly, these dynamics are used to demonstrate the convergence toward a stable or an equilibrium state.

Definition 4.10. The discrete-time replicator equations are defined as follows:

$$\mathfrak{w}_i^{(t+1)} = \frac{\gamma + \mathbb{E}(e^i, \mathfrak{w}^{(t)})}{\gamma + \mathbb{E}(\mathfrak{w}^{(t)}, \mathfrak{w}^{(t)})} \mathfrak{w}_i^{(t)}, \tag{4.6}$$

where $\gamma \geq 0$ denotes the *birthrate*. This parameter regulates the rate of change along a solution trajectory: high values imply slower convergence, while low rates lead to faster convergence but less stable outcomes.

Definition 4.11. A population state $\mathfrak{w}^\star \in \Delta$ is stationary in Equation 4.6 if and only if:

$$\mathbb{E}(e^i, \mathfrak{w}^\star) = \mathbb{E}(\mathfrak{w}^\star, \mathfrak{w}^\star) \text{ for all } i \text{ such that } \mathfrak{w}_i^\star > 0.$$

The notion of ESS has a stronger condition than the one of NE. It was demonstrated that any ESS is a refinement of NE [100]. In addition, a NE satisfies the stationary state condition. Thus, we come to the following propositions (the proofs can be found in [102]).

Proposition 4.1. Let Δ^{ESS}, Δ^{NE}, and Δ^0 denote the set of ESS, NE, and stationary states, respectively. We have $\Delta^{ESS} \subseteq \Delta^{NE} \subseteq \Delta^0$.

Proposition 4.2. (convergence) The fundamental theorem of natural selection states: *every non-stationary solution trajectory to Equation 4.6 in any symmetric game with positive payoff matrix converges to:*

- a stationary state $\mathfrak{w}^\star \in \Delta^0$.
- a NE $\mathfrak{w}^\star \in \Delta^{NE}$ if for all $i \in \{1, ..., n\} : \mathfrak{w}_i^{(0)} > 0$.
- an ESS $\mathfrak{w}^\star \in \Delta^{ESS}$ if $\mathfrak{w}^\star$ is a local maximizer of $\mathbb{E}(\mathfrak{w}^{(t)}, \mathfrak{w}^{(t)})$.

4.5 Game theory in machine learning

The audience for game theory has known a dramatic increase in recent years, inspiring many applications in myriad disciplines like communication networks [103], social networks [104], electricity grid [105], smart grid [106], neuroscience [107], and indeed machine learning [108, 109] among many others. Game theory provides a promising new paradigm that can elegantly capture key features for addressing machine learning tasks like feature selection [109, 43, 110]), clustering [108, 111], and ensemble pruning [46]. The research community has put a great deal of effort into the development of novel approaches within CGT and EGT. Cohen et al. were the first to address feature selection from a CGT perspective [43]. They ranked the features according to their contributions to the overall accuracy performance using Shapley value. Similarly, Sun et al. viewed the problem of feature selection as a voting game [109]. They proposed a framework to assess the power of each feature by considering Banzhaf index in conjunction with well-known evaluation criteria such as and minimum-Redundancy Maximum-Relevance (mRMR) [112]. Their technique selects the features starting with one which has the highest power, and it considers them in descending order of their importance until the desired number of features is reached. Garg et al. formulated clustering as a coalitional game with transferable utility among the data points [111]. Their methodology is defined based on two crucial properties for clustering: potential (distance between a data point to its closest centroid) and scatter (intra-cluster point-to-point distance). Moreover, they demonstrated that Shapley value satisfies these properties, and hence captures key criteria of the clustering task. Bulò et al. derived a new formulation of the hypergraph clustering problem [108]. They showed that the notion of a cluster is equivalent to an ESS, which can incorporate both the internal coherency and the external incoherency of a cluster.

4.6 Summary

This chapter provided a short survey on game theory. We presented two powerful paradigms: CGT and EGT. We also described several well-known solution concepts like Banzhaf power index, Shapley value, and evolutionary stable strategies.

In this thesis, we introduce novel frameworks for addressing the ensemble pruning problem, which are distinguished from the other approaches in the literature by being founded on game theory. Our work provides new insights toward the understanding, the analysis, and the design of novel classifier evaluation and ordering criteria from CGT and EGT perspectives. The next chapter presents the key idea behind this work, a derivation of a diversity-based selection criterion in the context of simple games.

CHAPTER 5
SELECTION OF OF SUB-ENSEMBLES WITH MODERATE DIVERSITIES THROUGH SIMPLE GAMES

5.1 Introduction

We reviewed in Chapter 2 the main elements that characterize an ensemble method. The reader may recall that *diversity* is recognized to play a key role in constructing a successful ensemble methodology. However, its impact on the ensemble generalization power has not been established yet. As a matter of fact, it has been demonstrated that an ensemble composed of highly diversified members may result in a better or worse performance [28, 9, 27]. In other words, diversity can be either harmful or beneficial and therefore requires an adequate quantification. Based on these insights, we propose a powerful criterion that scores the utility of a base learner according to its contribution to the overall ensemble diversity based on Banzhaf index. It is worth underscoring that the original definition of this solution concept is intractable for moderate and large ensembles. To overcome the computational burden, we introduce a new formulation of Banzhaf power index and show that its time complexity is pseudo-polynomial in the number of classifiers (size of the initial committee).

5.2 Extraction of sub-ensembles with moderate diversities

The concept of “diversity” is considered as the key success in constructing a committee of classifiers [8]. According to Rokach [9], the action of creating an ensemble of diversified learners leads to uncorrelated errors that boost the group performance globally. Unfortunately, efficiently measuring diversity and understanding its relationship with the classification generalization power of the committee remains an open problem [24, 65, 22]. Several experimental studies have shown that large diversity within an ensemble causes a sharp drop in its performance [27]. Furthermore, it is well-known that an ensemble composed of identical classifiers does not generalize well. To seek a tradeoff between these two extreme effects, we propose a methodology that focuses on extracting a set of classifiers with average diversity. Specifically, we cast the problem of ensemble pruning as a simple game played among the component learners. The devised model captures several levels of classifiers' disagreement and promotes average diversity over the other two extreme scenarios (correlation and high diversity). The various steps of Simple Coalitional Game-based Pruning (SCG-Pruning) are depicted by Figure 5.1.

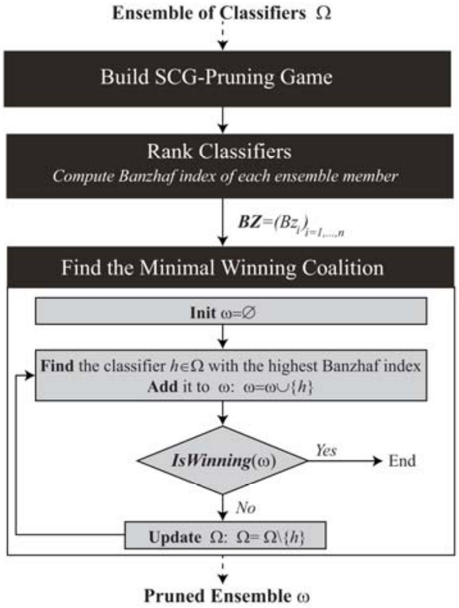

Ensemble of Classifiers Ω
Build SCG-Pruning Game
Rank Classifiers
Compute Banzhaf index of each ensemble member
BZ=(Bz_i)_{i=1,...,n}
Find the Minimal Winning Coalition
Init ω=∅
Find the classifier h∈Ω with the highest Banzhaf index
Add it to ω: ω=ω∪{h}
IsWinning(ω)
Yes
End
No
Update Ω: Ω=Ω\{h}
Pruned Ensemble ω

Figure 5.1: The SCG-Pruning process.

5.2.1 Ensemble pruning game

Let $\Omega = \{h_1, h_2, ..., h_n\}$ be an ensemble of n classifiers. Each learner is provided with the same training set $\Gamma = \{(x_i, y_i), i = 1...m\}$, where $x_i \in \mathcal{X}$ is a feature vector characterizing the i^{th} instance, and $y_i \in \mathcal{Y}$ denotes its true class label. We assume that every ensemble member is trained separately using the same training set Γ by invoking a learning algorithm. The final decision of the committee Ω combines the predictions of all members following majority vote.

We define a simple game G built on the initial ensemble of classifiers Ω, where a classifier h_i is considered as a player and is associated with a weight w_i, $i \in \{1, ..., n\}$. These weights are computed as follows. We define the *diversity contribution* of a classifier h_i, with respect to the entire ensemble Ω, as the average diversity between h_i and the rest of classifiers, which we denote by $Div_\Omega(h_i)$. In order to approximate high-order-diversity induced by a candidate classifier, we consider that the ensemble members exhibit only pairwise interactions.

Definition 5.1. The diversity contribution of a classifier $h_i \in \Omega$ is defined as:

$$Div_{\Omega}(h_i) = \frac{1}{n-1} \sum_{h_j \in \Omega \setminus \{h_i\}} f_{i,j},$$

where $f : \Omega \times \Omega \mapsto \mathbb{R}$ assigns to a pair of classifiers (h_i, h_j) a real number that corresponds to the diversity between the decisions of h_i and h_j, with $f_{i,i} = 0$ and $f_{i,j} = f_{j,i}$.

Definition 5.2. The weight w_i assigned to a classifier $h_i \in \Omega$ is given by:

$$w_i = \sum_{h_j \in \Omega \setminus \{h_i\}} \mathbb{I}(Div_{\Omega}(h_i) \geq Div_{\Omega}(h_j)).$$

It is noteworthy that each voting weight w_i can be thought as a level of diversity induced by h_i, in which highly diversified members receive higher weights. In addition to the list of weights, we introduce two thresholds q_1 and q_2 in order to define the payoff function of the pruning game, such that $q_2 - q_1 > \max_{h_i} w_i$ and $q_1 > \max_{h_i} w_i$.

Definition 5.3. Given two thresholds q_1 and q_2, the payoff function of the proposed game, denoted $G = (\Omega, [\mathbf{w}, q_1, q_2])$, is defined as:

$$v(S) = \begin{cases} 1 \quad if \quad q_1 \leq \sum_{h_i \in S} w_i \leq q_2 \\ 0 \qquad\qquad otherwise \end{cases}.$$

Under this payoff function, a coalition S of classifiers wins if the sum of its members' weights falls between q_1 and q_2. The term $\sum_{h_i \in S} w_i$ measures the *amount of diversity* present in S: a low value corresponds to strong correlations between the ensemble members, whereas a large value indicates that the coalition is composed mainly of diversified classifiers. Furthermore, the interval $[q_1, q_2]$ corresponds to the width of *permitted diversity*, in which the lower bound q_1 controls the degree of correlation present in S, and the upper bound q_2 serves as barrier for highly diverse ensembles. Both extreme cases can decrease the generalization performance of the group [24]. When q_1 and q_2 are set properly, this payoff function ignores coalitions made of correlated classifiers (lower bound) and those highly diverse (upper bound). As a result, the focus will only be on groups with moderate diversities that can lead to better generalization performance [27].

Correctly setting the values of q_1 and q_2 is of vital importance for the success of the proposed methodology. We can distinguish two extreme cases: (i) *low values for* q_1 *and* q_2: in this case, the proposed technique focuses mainly on correlated ensembles; and

(ii) *high values for* q_1 *and* q_2: this choice considers only ensembles composed of the most diverse members. One should avoid the configurations indicated by (i) and (ii), and set the values of q_1 and q_2 between these two extreme cases. The choice of q_1 and q_2 will be further discussed in the experiments section (Section 5.3.2).

5.2.2 Classifier evaluation based on Banzhaf power index

The next step consists of ranking each classifier according to Banzhaf power index. Under the SCG-Pruning game, the formulation of this solution concept (provided by Equation 4.1) has an interesting interpretation that is summarized as follows. Let us consider a coalition made of correlated classifiers S, where $v(S) = 0$. If a classifier h_i induces the proper amount of diversity into a losing coalition S and turns it into a winning coalition ($v(S \cup \{h_i\}) = 1$), then h_i is *pivotal* for S and the coalition S is a *positive swing* for h_i. Conversely, the set of negative swing for a classifier h_i is defined as the ones in which h_i introduces large diversity into winning coalitions and changes their status into losing coalitions. Therefore, Banzhaf power index assigns high ranks to members that induce diversity into correlated ensembles while penalizing members that exhibit strong disagreement with the group.

Computational complexity reduction

It is commonly acknowledged that the exact and direct computation of Banzhaf index (Definition 4.5) for non-monotone simple games requires summing over all possible coalitions, which is exponential in the size of the initial committee, and is therefore intractable for large ensembles [88]. To cope with the computational burden, we have investigated the relationship between the proposed game and other representations of simple games. As a result, we have expressed Banzhaf power index within the proposed framework in terms of Banzhaf indices of two weighted voting games (Theorem 5.2).

Theorem 5.1. *Consider the weighted voting game* $G_1 = (\Omega, [\mathbf{w}, q_1])$*,* $Bz_i(G_1)$ *player* h_i*'s Banzhaf power index of* G_1*, and* $|swing_i^+(G)|$ *the number of positive swing coalitions for* h_i *under the SCG-Pruning game* G*, then:*

$$|swing_i^+(G)| = 2^{n-1} \times Bz_i(G_1), \quad \forall i \in \{1, ..., n\}.$$

Proof. *Banzhaf power index of weighted voting games can be written as [98]:*

$$\begin{aligned} Bz_i(G_1) &= \frac{1}{2^{n-1}} \times |\{S \subseteq \Omega \setminus \{h_i\} | v_1(S \cup \{h_i\}) = 1 \wedge v_1(S) = 0\}|. \\ &= \frac{1}{2^{n-1}} \times |\{S \subseteq \Omega \setminus \{h_i\} | \mathcal{W}(S) + w_i \geq q_1 \wedge \mathcal{W}(S) < q_1\}|. \end{aligned}$$

where $\mathcal{W}(S) = \sum_{h_j \in S} w_j$.

Since all weights are positive integers, we can write:

$$Bz_i(G_1) = \frac{1}{2^{n-1}} \times |\{S \subseteq \Omega \setminus \{h_i\} | q_1 - w_i \leq \mathcal{W}(S) < q_1\}|. \tag{5.1}$$

On the other hand, the set of positive swing coalitions for player h_i *under* G *is given by:*

$$\begin{aligned} swing_i^+(G) &= \{S \subseteq \Omega \setminus \{h_i\} | v(S \cup \{h_i\}) = 1 \wedge v(S) = 0\}. \\ &= \{S \subseteq \Omega \setminus \{h_i\} | q_1 \leq \mathcal{W}(S) + w_i \leq q_2 \wedge \mathcal{W}(S) < q_1\}. \\ &= \{S \subseteq \Omega \setminus \{h_i\} | q_1 - w_i \leq \mathcal{W}(S) \leq q_2 - w_i \wedge \mathcal{W}(S) < q_1\}. \end{aligned}$$

Recall that $q_2 - q_1 > max_{h_i} w_i$. *This consideration implies* $q_1 < q_2 - w_i$ *for all* $i \in \{1, ..., n\}$. *Consequently,* $swing_i^+(G)$ *can be further simplified as:*

$$swing_i^+(G) = \{S \subseteq \Omega \setminus \{h_i\} | q_1 - w_i \leq \mathcal{W}(S) < q_1\}.$$

Using Banzhaf power index formulation given by Equation 5.1, one can write:

$$|swing_i^+(G)| = 2^{n-1} \times Bz_i(G_1)\square.$$

Corollary 5.1.1. *Given the weighted voting game* $G_2 = (\Omega, [\mathbf{w}, q_2 + 1])$ *and player* h_i*'s Banzhaf index* $Bz_i(G_2)$*, then the number of negative swing coalitions for* h_i *under the SCG-Pruning game* G *can be expressed as:*

$$|swing_i^-(G)| = 2^{n-1} \times Bz_i(G_2), \quad \forall i \in \{1, ..., n\}.$$

Theorem 5.2. *Consider the two weighted voting games* $G_1 = (\Omega, [\mathbf{w}, q_1])$ *and* $G_2 = (\Omega, [\mathbf{w}, q_2 + 1])$*, then* $Bz_i(G)$*, player* h_i*'s Banzhaf power index of the SCG-Pruning game* G*, can be simplified as:*

$$Bz_i(G) = Bz_i(G_1) - Bz_i(G_2), \quad \forall i \in \{1, ..., n\}.$$

Proof. *From Equation 4.1, we have:*

$$Bz_i(G) = \frac{1}{2^{n-1}} \times (|swing_i^+(G)| - |swing_i^-(G)|).$$

Using Theorem 5.1 and Corollary 5.1.1, one obtains:

$$Bz_i(G) = Bz_i(G_1) - Bz_i(G_2)\square.$$

5.2.3 A general ensemble pruning scheme

The last step of the SCG-Pruning methodology is to determine the pruned ensemble size L. For this purpose, we propose to map the pruned ensemble to the *minimal winning coalition* composed only of highly ranked classifiers. In CGT, the definition of the minimal winning coalition is outlined by Riker [113]:

> *"If a coalition is large enough to win, then it should avoid taking in any superfluous members, because the new members will demand a share in the payoffs. Therefore, one of the minimal winning coalitions should form. The ejection of the superfluous members allows the payoff to be divided among fewer players, and this is bound to be advantage of the remaining coalition members".*

Notice that this concept does not predict the coalition structure of the game, but it provides strong evidence that one of the minimal winning coalitions will form. Putting this notion into the context of SCG-Pruning, the minimal winning coalition corresponds to the smallest sub-ensemble of classifiers that *together exhibit moderate diversity*.

The pseudocode of the proposed approach is depicted by Figure 5.2. The SCG-Pruning method takes as input an initial ensemble of classifiers Ω, two thresholds (q_1, q_2) , and a training set Γ. In addition, SCG-Pruning requires defining a function for estimating the classifiers' voting weights. For instance, pairwise diversity can be estimated using statistical measures [9, 36] like: Cohen's kappa, disagreement measure, Q-statistic, etc., or even information theory concepts [65, 22, 2]. The algorithm first computes the classifiers' predictions of every training sample (lines [3-7]), and uses them to estimate the voting weights of the ensemble members (lines [8-10]). Then, it ranks every individual learner based on Banzhaf power index (lines [11-13]). Finally, it sets the pruned ensemble as the minimal winning coalition made of the top ranked learners (lines [14-18]). More specifically, the algorithm iteratively chooses, from among the classifiers not yet selected, the classifier with the highest rank, and adds it to the selected set ω until ω wins.

```
Input:   $\Gamma$: Training set.
         $\Omega$: Ensemble of classifiers.
         $q_1, q_2$: Two thresholds.
Initialize:   $\omega = \emptyset$;
                                        /*Getting classifiers' predictions*/
    For each $h_i \in \Omega$
        For each $(x_j, y_j) \in \Gamma$
            $Preds_j^i = h_i(x_j)$;
        End for each $(x_j, y_j)$
    End for each $h_i$
                                        /*Estimating classifiers' weights using $Preds$*/
    For each $h_i \in \Omega$
        Compute $w_i$ provided by Definition 5.2;
    End for each $h_i$
                                        /*Computing classifiers' Banzhaf indices*/
    For each $h_i \in \Omega$
        $Bz_i(G) = Bz_i(G_1) - Bz_i(G_2)$;
    End for each $h_i$
                                        /*Searching for the minimal winning coalition*/
    Repeat
        $h = argmax_{h_i} Bz_i(G)$;
        $\omega = \omega \cup \{h\}$;
        $\Omega = \Omega \setminus \{h\}$;
    Until $v(\omega) = 1$

Output:   $\omega$: Pruned ensemble.
```

Figure 5.2: The SCG-Pruning algorithm.

5.3 Experimental analysis

5.3.1 Experimental setup

To demonstrate the validity and the effectiveness of the proposed methodology, we carried out extensive experiments on 58 datasets selected from the UCI Machine Learning Repository [114]. An overview of the datasets properties is shown in Appendix B, Table B.1.

We resampled each dataset following Dietterich's 5×2 cross validation (cv) to produce ten training and ten testing folds, denoted $train_i$, $test_i$, $i = 1, ..., 10$, respectively. The ensemble members were trained using $train_i$ and tested on $test_i$. We obtained ten trained committees and ten performance estimates of each ensemble technique. We reported only the mean of these ten measurements. It is noteworthy that we estimated the base learners' weights on the training fold.

As indicated in the previous section (Definition 5.2), the weights assigned to the ensemble members are computed based on a pairwise diversity measure. In our experiments, we used the three metrics: disagreement measure (SCG-DIS), Cohen's kappa (SCG-κ), and mutual information (SCG-MI) defined by Equations 2.11, 2.15, and 2.13, respectively. We invoked MITOOLBOX [115] to compute the information theory concepts.

5.3.2 First set of experiments

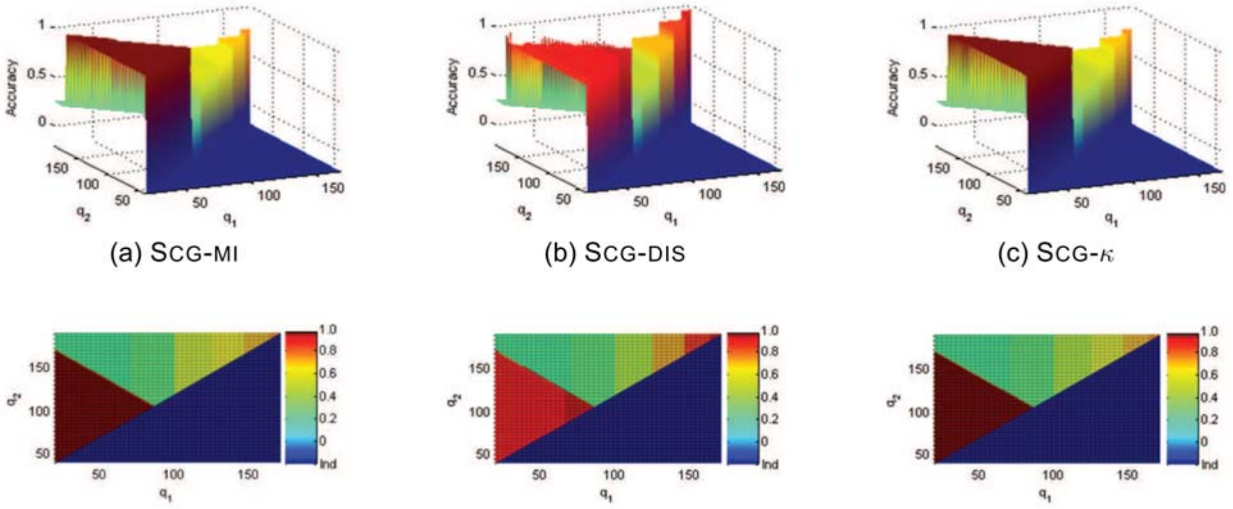
We used 20 classifiers taken from WEKA 3.6 [4], PRTOOLS 5.0.2 [116], and LIBSVM 3.18 [117] to generate the initial ensemble. A summary of these learning algorithms is given in Table 5.1. Additional description is provided in Appendix B.

Table 5.1: List of classifiers used in the experiments.

No.	Algorithm	Platform	Description
1	J48	WEKA	C4.5 decision tree.
2	CART	WEKA	Decision tree learner using CART's minimal cost complexity pruning.
3	Logistic	WEKA	Multinomial logistic regression.
4-6	IBk	WEKA	K-nearest neighbors classifier using linear search with the Euclidean distance, and 3 values for $k = 1, 3, 5$.
7	OneR	WEKA	1R rule-based learning algorithm.
8	NaïveBayes	WEKA	Standard probabilistic naïve Bayes classifier.
9	Multilayer Perceptron	WEKA	Multilayer perceptron classifier.
10-11	Decision Table	WEKA	Simple decision table majority classifier using (10) BestFirst and (11) Genetic search methods.
12	JRip	WEKA	RIPPER (Repeated Incremental Pruning to Produce Error Reduction) algorithm for rule induction.
13	PART	WEKA	PART decision list built using J48.
14	Fisherc	PRTOOLS	Fisher's least square linear classifier.
15	Ldc	PRTOOLS	Linear Bayes normal classifier.
16	Qdc	PRTOOLS	Quadratic Bayes normal classifier.
17	Parzendc	PRTOOLS	Parzen density based classifier.
18-20	SVM	LIBSVM	Support vector machines using (18) a radial (Gaussian) kernel; (19) a polynomial kernel; and (20) a linear kernel.

Influence of the thresholds q_1 and q_2

In this experiment, we study the impact of the thresholds q_1 and q_2 on the performance of our approach. We present a 3D plot which displays the relationship between



(a) SCG-MI

(b) SCG-DIS

(c) SCG-κ

(d) SCG-MI

(e) SCG-DIS

(f) SCG-κ

Figure 5.3: The impact of (q_1, q_2) on the performance of SCG-MI, SCG-DIS, and SCG-κ for the "Audiology" dataset.

these thresholds and the accuracy of the produced ensemble by each of the SCG-Pruning variants. Figure 5.3 shows 3D plots of the SCG-Pruning variants on the Audiology dataset. Given a point (x, y, z), x and y coordinates correspond to the values of q_1 and q_2, respectively. The z-coordinate indicates the performance of SCG-Pruning on the training set. The subplots (d), (e), and (f) show 2D plots from the top view of (a), (b), and (c), respectively.

Examining Figure 5.3 (d), we can identify four main regions: The lower right half of the plot "blue surface" represents the set of impossible configurations of SCG-Pruning game. In this case, the values of q_1 and q_2 violate our initial condition, which states that $q_2 - \max_{h_i} w_i > q_1$, and therefore the game can't be defined. The points laying close to the right upper corner of the plot "yellow triangle" (large q_1 and q_2) correspond to the configurations where the pruned ensemble exhibits very large diversity. On the left upper region "green triangle", we observe a very low performance by the three SCG-Pruning variants. We believe that this behavior occurs because the proposed game is not well-defined and fails to deliver an appropriate ranking of the ensemble members. More specifically, let consider the two extreme values of the thresholds $q_1 = 20$ and $q_2 = 190$. In this case, the interval that defines if a coalition wins (width of permitted diversity) is extremely large, and hence almost any coalition wins. In addition, the number of negative swings for every player is 0 since no coalition has a weight that exceeds 190. Finally, the last region "red triangle" yields the best performance and corresponds to the set of preferable game settings. We refer to it as $\mathscr{R}$. Under these settings, SCG-pruning variants produce ensembles with moderate diversities.

Based on these observations, we set the values for these thresholds as follows. For small-sized ensembles, we picked the pair (q_1, q_2) from $\mathscr{R}$ that yields the best performance on the training set; whereas for larger ensembles, we selected their values randomly from the search region $\mathscr{R}$.

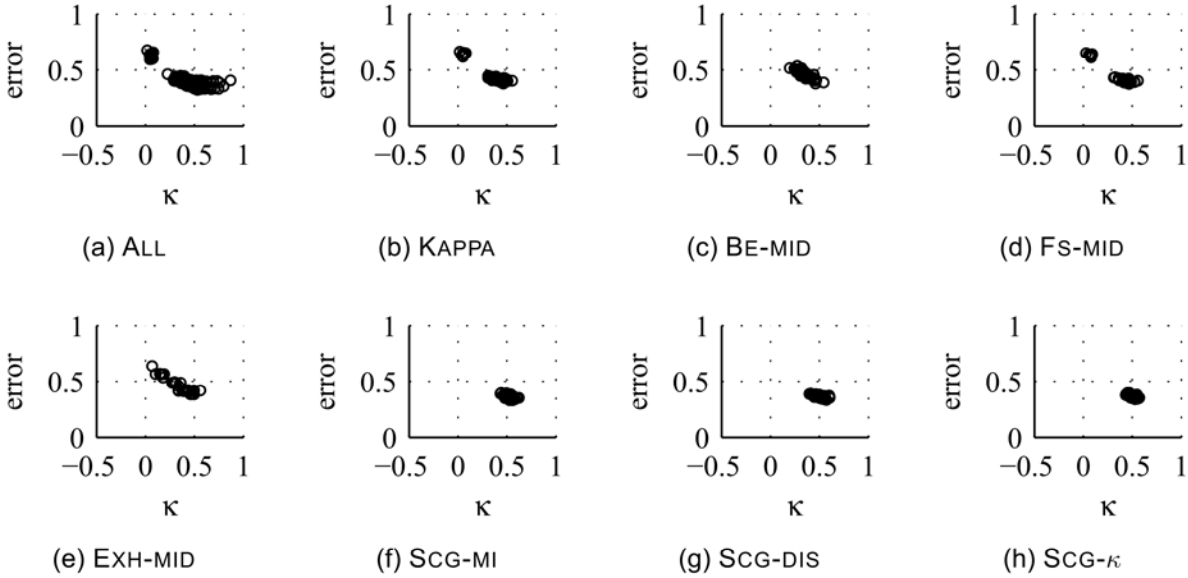
Kappa error diagrams

This section presents kappa error diagrams to gain some insight into the accuracy/diversity tradeoff. These diagrams depict an ensemble of classifiers as a scatterplot. Every pair of classifiers is represented as a point on the plot, where the x-coordinate corresponds to the value of Cohen's kappa κ between the pair, and the y-coordinate is the averaged individual error rate of the two classifiers. Following García-Pedrajas et al. [27], we estimated the error rate of every classifier on the test set. The aim of this experiment is to investigate whether the proposed methodology extracts sub-ensembles with moderate diversities.

We compared the proposed variants with: Kappa pruning, greedy, and exhaustive search strategies. For the greedy search [8], we implemented two variants: Forward Selection (FS) and Backward Elimination (BE). Forward selection starts with an empty set; then, it chooses from among the classifiers not yet selected the classifier which best improves a specific evaluation criterion until the preset size of the pruned ensemble is met. Conversely, in backward elimination, the pruned ensemble is initialized as the entire ensemble; next, the algorithm proceeds by iteratively eliminating classifiers based on an evaluation criterion until the desired number of classifiers is reached. Exhaustive search (EXH) tests all possible subsets of size L classifiers (there are $\binom{20}{L}$ subsets), and selects the ensemble with the best predefined criterion. Both exhaustive and greedy search approaches require defining a criterion that indicates the ensemble generalization performance. To this end, we implemented the Mutual Information-based Diversity (MID) criterion introduced in [2]. In addition, we reported kappa error diagrams for the entire ensemble which we denote by ALL. Note that we set the size of the pruned ensemble to $L = 9$ for all compared techniques. In this case, our approach selects the top L classifiers based on their Banzhaf indices. Table 5.2 gives a summary of the compared ensemble selection techniques. Figures 5.4-5.5 show kappa error diagrams for

Table 5.2: Legend for Tables and Figures presented in the first set of experiments.

Pruning technique	**Description**
SCG-κ	SCG-Ranking with Cohen's kappa (Equation 2.15) as the diversity measure.
SCG-DIS	SCG-Ranking with disagreement measure (Equation 2.11) as the diversity metric.
SCG-MI	SCG-Ranking with mutual information (Equation 2.13) as the diversity measure.
FS-MID	Forward selection using the MID evaluation criterion.
BE-MID	Backward elimination with MID as the search criterion.
KAPPA	Kappa pruning.
EXH-MID	Exhaustive search that uses the MID criterion.
ALL	This technique combines the decisions of the ensemble members, without selection, using majority vote.



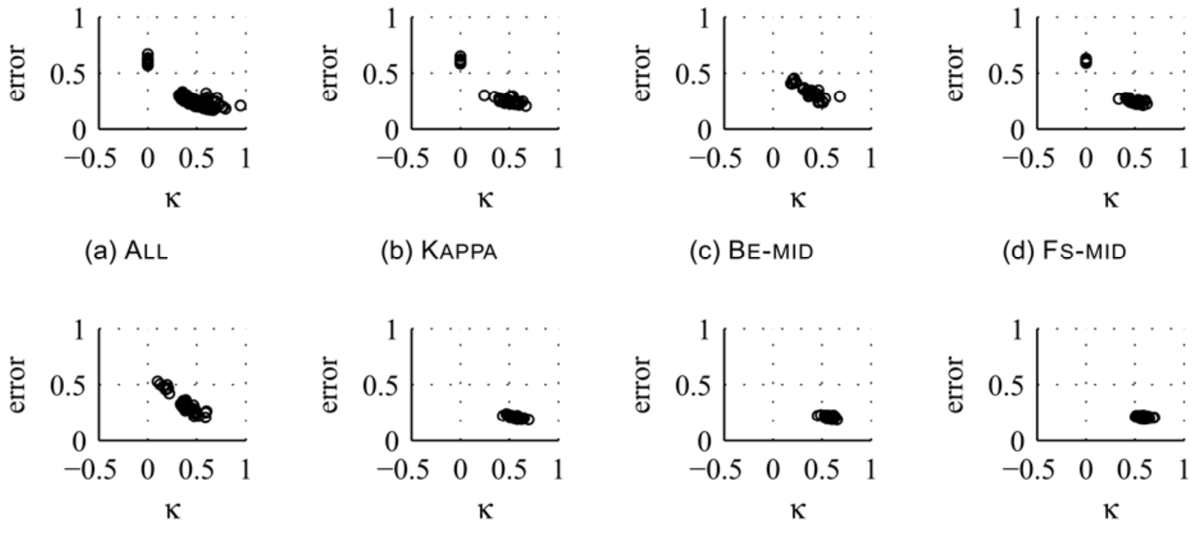
(a) ALL (b) KAPPA (c) BE-MID (d) FS-MID

(e) EXH-MID (f) SCG-MI (g) SCG-DIS (h) SCG-κ

Figure 5.4: Kappa error diagrams for the dataset "Glass identification".

numerous pruning approaches on two datasets: Glass identification and Lymphography.

The analysis Figures 5.4-5.5 is summarized as follows. First, the diagrams associated with the diversity-based pruning techniques (subplots 5.4b-5.4e) are skewed to the left side of the plot, which indicates large diversity. This behavior is expected since these techniques construct ensembles that are made of the most diverse members. On the other hand, SCG-Pruning variants provide less diversity than the aforementioned approaches. Additionally, when compared to ALL, the proposed approach does not select strongly correlated classifiers. This behavior is consistent with our initial idea, that is, the proposed methodology extracts sub-ensembles with moderate diversities.



(a) All (b) Kappa (c) Be-mid (d) Fs-mid

(e) Exh-mid (f) Scg-mi (g) Scg-dis (h) Scg-κ

Figure 5.5: Kappa error diagrams for the dataset “Lymphography”.

5.3.3 Second set of experiments

In this experiment, we trained an ensemble made of 100 Decision Stump trees using Bagging. We compared Scg-mi and Scg-κ with Reduce Error (Re) [36], Complementarity Measure (CC) [77], Margin Distance Minimization (Mdsq) [77] with a moving reference point p set to $2\sqrt{2 \times i}/n$ at the i^{th} iteration, Orientation Ordering (OO) [37], Boosting-Based (BB) [70], Genetic algorithm (Gasen), and Kappa pruning (Kappa). We set the parameters of Gasen to the following values: crossover probability$= 0.6$, mutation rate$= 0.05$, number of generations$= 100$, and population size$= 100$. It is noteworthy that the pruning approaches Re, CC, Mdsq, OO, BB, and Kappa require setting the size of the pruned ensemble L. In order to make a fair comparison, we set L to the same size obtained by Scg-mi.

Accuracy performance

Table 5.3 gives the average accuracy results of the second experiment. The last row specifies the mean rank of each method over all datasets. We statistically compared the performances of these pruning schemes using Friedman test. Under the null hypothesis, we assumed that all techniques are equivalent and the observed differences are due to chance. Friedman test rejects this hypothesis with with $F_F = 20.77 > F(9, 513) = 11.62$ for $\alpha = 1 \times 10^{-16}$ (F_F is distributed according to the F distribution with $10 - 1 = 9$ and $(10 - 1) \times (58 - 1) = 513$ degrees of freedom), and therefore confirms the existence of at least one pair of ensemble pruning techniques with significantly different performances. Because we are only interested in testing whether the pruning approaches significantly improve the initial ensemble BAGGING. Consequently, we conducted a Bonferroni-Dunn test at a 10% significance level with the critical value $q_{0.10} = 2.54$ and the critical difference $CD = 1.43$. The results of this test are depicted by Figure 5.6. On the horizontal axis, we represent the averaged rank of every pruning technique given in the last row of Table 5.3, and mark using a thick line an interval of $2 \times CD$ one on the right and the other to the left of BAGGING's mean rank.

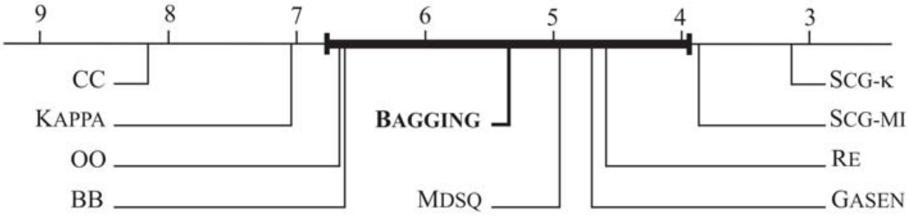


Figure 5.6: Comparison of BAGGING with 9 pruning techniques using Bonferroni-Dunn test.

The analysis of Bonferroni-Dunn test (Figure 5.6) reveals that the performances of SCG-κ and SCG-MI are in the lead followed by RE, GASEN, and MDSQ. Most importantly, we notice that both SCG-κ and SCG-MI fall outside the marked interval. Therefore, we can conclude that the proposed variants perform significantly better than BAGGING, while the experimental data cannot detect any improvement of the initial ensemble using RE, GASEN, BB, OO, or MDSQ.

Pruning time

We compared in Table 5.4 the average running time (in seconds) required by every pruning technique over all datasets.

Orientation ordering is the fastest technique followed by MDSQ, BB, and CC. Both SCG-κ and SCG-MI converge to similar pruning times. The results also indicate that GASEN and FS-MID approaches are slower than the other alternatives. The reported behavior is expected since search-based pruning methods generally tend to have high computational costs.

5.4 Summary

In this chapter, we developed a novel approach to address ensemble pruning based on non-monotone simple games. Our idea is to: (1) Devise a criterion that estimates

Table 5.3: Summary of mean accuracy results of the second experiment.

Datasets	SCG-κ	SCG-MI	GASEN	MDSQ	RE	OO	KAPPA	CC	BB	BAGGING
Anneal	**83.54**	**83.54**	82.78	82.78	82.78	79.11	78.33	82.34	78.35	82.78
Audiology	**47.17**	47.08	46.46	46.46	46.46	46.46	46.46	46.46	46.46	46.46
Australian	**85.51**	**85.51**	**85.51**	**85.51**	**85.51**	**85.51**	**85.51**	**85.51**	**85.51**	**85.51**
Balance	**80.16**	78.82	80.13	78.72	79.17	79.23	74.49	74.46	77.47	72.38
Balloons1	87.00	87.00	84.00	87.00	81.00	81.00	75.00	72.00	**94.00**	74.00
Balloons2	81.00	76.00	75.00	76.00	72.00	71.00	**82.00**	**82.00**	80.00	72.00
Balloons3	**75.00**	**75.00**	67.00	69.00	68.00	60.00	69.00	64.00	69.00	68.00
Balloons4	67.50	67.50	68.75	65.00	65.00	**70.00**	66.25	66.25	65.00	62.50
BCW	**95.57**	95.11	94.59	94.91	94.39	94.56	95.39	93.45	92.70	93.39
BC	73.71	73.92	72.73	**74.34**	73.71	73.92	70.49	71.61	72.59	71.89
Car	**70.02**	**70.02**	**70.02**	**70.02**	**70.02**	**70.02**	**70.02**	**70.02**	**70.02**	**70.02**
Chess	**66.05**	**66.05**	**66.05**	**66.05**	**66.05**	**66.05**	**66.05**	**66.05**	**66.05**	**66.05**
CVR	**95.63**	**95.63**	**95.63**	**95.63**	**95.63**	94.94	94.94	94.94	95.03	**95.63**
Credit	**85.51**	**85.51**	**85.51**	**85.51**	**85.51**	**85.51**	**85.51**	**85.51**	**85.51**	**85.51**
Cylinder	70.04	69.04	70.52	69.44	70.33	68.19	67.11	64.52	69.04	**70.56**
Dermatology	**59.13**	56.01	53.11	51.69	53.06	50.08	52.08	50.11	50.11	51.37
Ecoli	**67.44**	**67.44**	64.64	64.64	64.64	64.70	63.81	64.58	64.58	64.64
Glass	53.83	**57.38**	52.52	55.05	56.54	51.04	50.16	50.64	50.55	51.25
Hayes-Roth	**60.75**	59.50	**60.75**	56.00	59.38	54.38	54.38	50.08	50.13	56.25
Hepatitis	81.80	81.80	81.03	**83.22**	81.67	82.83	79.48	79.75	80.50	81.03
Ionosphere	83.31	82.79	82.96	83.13	82.79	82.16	83.02	81.48	83.25	**83.37**
Iris	**95.33**	**95.33**	95.07	94.27	95.20	87.60	82.47	80.00	94.67	94.53
Labor	85.97	85.20	83.17	**88.40**	81.77	84.19	88.39	78.95	88.39	82.41
Lenses	**76.67**	70.00	75.83	72.50	**76.67**	71.67	64.17	61.67	67.50	64.17
Letter	70.78	71.29	68.03	68.97	69.91	67.98	67.63	67.08	67.58	**71.94**
LRS	**51.49**	50.06	49.72	49.68	50.10	47.38	48.97	49.72	49.72	49.68
Lymph	76.22	76.08	76.35	**77.30**	75.41	75.81	72.97	72.03	70.81	74.46
Monks1	**74.64**	**74.64**	**74.64**	**74.64**	**74.64**	**74.64**	**74.64**	**74.64**	**74.64**	**74.64**
Monks2	65.19	65.19	65.16	65.39	65.52	65.03	65.39	64.43	65.39	**65.72**
Monks3	78.81	78.81	78.81	78.81	78.81	77.65	77.83	78.48	78.81	**89.89**
MFF	**68.41**	67.70	65.90	61.63	68.26	62.12	63.67	60.68	60.53	62.64
MFKL	65.04	**65.09**	62.17	61.12	63.43	60.63	63.20	60.50	60.58	64.30
MFPC	74.99	73.29	72.04	67.70	**77.89**	65.84	60.77	61.77	62.85	77.88
MFZ	66.62	**67.26**	64.40	64.38	66.60	63.71	63.29	63.39	63.43	66.02
Mushroom	**88.68**	**88.68**	**88.68**	**88.68**	**88.68**	**88.68**	**88.68**	**88.68**	**88.68**	**88.68**
Musk1	**72.27**	71.72	72.18	71.26	72.18	70.76	69.79	70.55	71.89	71.47
Musk2	**84.59**	**84.59**	**84.59**	**84.59**	**84.59**	**84.59**	**84.59**	**84.59**	**84.59**	**84.59**
Nursery	**66.25**	**66.25**	**66.25**	**66.25**	**66.25**	66.08	66.08	66.08	**66.25**	**66.25**
Optical	**65.40**	64.35	63.49	62.96	63.38	62.67	62.62	61.79	61.79	64.12
Page blocks	93.17	**93.18**	93.13	93.13	93.13	93.06	93.06	93.13	93.06	93.13
Pen	**60.66**	60.56	60.59	60.51	60.63	60.05	60.01	60.46	60.49	60.57
Pima	**74.97**	74.66	74.77	74.61	74.58	73.85	71.85	71.59	72.76	74.11
POP	64.22	62.44	68.00	65.33	70.67	62.89	65.78	61.11	64.22	**70.89**
Soybean L	68.26	**68.49**	68.38	68.41	68.43	66.38	66.21	67.44	67.47	67.50
Soybean S	**97.83**	95.80	97.39	90.62	81.49	76.54	71.45	72.84	74.09	96.21
Spambase	**83.31**	83.15	81.73	81.26	81.53	81.04	79.97	79.06	79.95	79.07
SPECT	**79.40**	**79.40**	**79.40**	**79.40**	**79.40**	**79.40**	**79.40**	**79.40**	**79.40**	**79.40**
SPECTF	78.05	77.75	77.83	78.13	78.35	78.20	79.25	76.47	77.30	**79.40**
TAE	47.39	46.71	47.39	46.46	**49.91**	49.27	45.08	44.55	44.96	46.72
Thyroid D	**95.24**	**95.24**	**95.24**	**95.24**	**95.24**	**95.24**	**95.24**	**95.24**	**95.24**	**95.24**
Thyroid G	82.69	**82.78**	81.58	80.93	82.60	81.12	79.54	80.47	80.37	79.72
Tic-Tac-Toe	**70.02**	69.79	69.48	69.94	69.94	69.06	68.85	67.16	68.81	69.94
Waveform1	60.90	60.18	60.22	60.28	59.93	60.21	60.08	57.47	58.11	**61.46**
Wine	92.70	92.02	91.35	92.13	90.79	91.46	83.71	80.85	**94.94**	89.44
WDBC	92.83	**92.94**	91.81	92.72	92.44	92.72	92.65	91.21	92.83	90.97
WPBC	72.32	74.24	74.44	73.84	75.56	72.73	76.06	70.71	73.54	**76.36**
Yeast	50.58	50.67	50.61	50.50	50.61	47.78	49.02	50.61	**50.70**	50.54
Zoo	**73.62**	64.37	61.95	62.55	60.58	59.20	65.90	59.40	56.07	61.57
Average ranks	**3.14**	**3.86**	**4.69**	**4.94**	**4.58**	**6.66**	**7.03**	**8.15**	**6.62**	**5.34**

Table 5.4: Average pruning times (in seconds) of several pruning approaches.

SCG-κ	SCG-MI	GASEN	MDSQ	RE	OO	KAPPA	CC	BB	FS-MID
0.320	0.401	36.86	0.015	0.793	0.003	0.174	0.032	0.016	3.075

the importance of each member's contribution to the overall diversity based on Banzhaf power index. (2) Map the pruned ensemble to the minimal wining coalition made of the members that together exhibit moderate diversity. Experimental comparisons with various pruning methods on 58 Benchmark datasets substantiate the efficacy of the proposed approach.

We have noticed that the thresholds q_1 and q_2 are of paramount importance to the success of our methodology. Consequently, it would be interesting to investigate the relationship between these thresholds and the generalization performance of SCG-Pruning so that they can be properly set for real world applications.

The proposed scoring function measures the power of a base learner based solely on the notion of diversity. In the next chapter, we will explore balancing diversity and accuracy in order to devise a better indicator of the ensemble performance.

CHAPTER 6
INDUCED SUBGRAPH GAME FOR CLASSIFIER ORDERING

6.1 Introduction

In the previous chapter, we presented a new criterion that evaluates the decision power of a component learner based on Banzhaf index. The proposed game promotes sub-ensembles with moderate diversities, which can yield better accuracy performance.

It is widely acknowledged that the *ensemble diversity* decreases with the increase in the *individual accuracies* i.e. “accuracy/diversity dilemma” [33]. This new consideration has gained a widespread attention from the ensemble learning community. As a matter of fact, numerous evaluation criteria that balance accuracy with diversity have been introduced in the literature [38, 2], but very few attempts addressed this problem from a CGT perspective. Motivated by the success of this principle, in the present chapter, we present an improved framework that handles the accuracy/diversity tradeoff using Shapley value. We first introduce an induced subgraph game which is defined in terms of the individual accuracies and the ensemble diversity. Then, we rank a component learner according to its contribution in keeping a proper tradeoff between these two crucial concepts.

6.2 Notation and problem statement

We use similar notation to Chapter 5 that we summarize here for clarity, with extensions to include oracle outputs. We denote an ensemble made of n classifiers with $\Omega = \{h_1, h_2, ..., h_n\}$, where every component learner is trained separately using the same training set $\Gamma = \{(x_i, y_i), i = 1...m\}$. Given a feature vector x, the ensemble Ω combines the predictions of its members $h_1(x), ..., h_n(x)$ using majority vote. We represent the oracle outputs of the ensemble members as a Boolean matrix $Z = (z_{ki})_{k,i=1}^{m,n}$, with $z_{ki} = 1$ if h_i is correct on the k^{th} sample, and 0 otherwise. The number of correct/incorrect predictions made by two classifiers h_i and h_j on the training set Γ is defined as:

$$N_{ij}^{ab} = \sum_{k=1}^{m} \mathbb{I}(z_{ki} = a \quad and \quad z_{kj} = b), \quad a, b \in \{0, 1\}. \tag{6.1}$$

6.3 Induced Subgraph Game for Classifier Ordering (ISGCO)

Classifier ordering first assigns a rank φ_i to every classifier h_i according to an evaluation measure (or criterion); then, the selection is conducted by aggregating the ensemble members whose ranks are above a predefined threshold. The main challenge consists of adequately setting the criterion used for scoring every member's contribution to the ensemble performance.

It is almost always the case that two accurate classifiers have low diversity, while two weak learners (their accuracies are slightly better than random guessing) often disagree with each other. This phenomenon is known as accuracy/diversity dilemma (for additional details, please refer to Section 2.6.2). Several studies have shown that extracting a sub-ensemble which properly balances diversity and accuracy achieves better generalization performance than the entire committee [58, 68].

Motivated by the positive role of balancing diversity with accuracy, we propose an ordering-based pruning technique that addresses this dilemma through CGT. Our approach (ISGCO) operates in three steps:

Step 1. We formulate classifier ranking as an induced subgraph game played among the individual learners. The proposed game is defined based on two measures namely *accuracy* and *diversity*:

Definition 6.1. The accuracy of classifier h_i, denoted Acc_i, is given by:

$$Acc_i = \frac{N_{ii}^{11}}{N_{ii}^{00} + N_{ii}^{11}}.$$

Definition 6.2. The diversity between two ensemble members h_i and h_j is defined as:

$$Div_{i,j} = \frac{1}{2} \times \left(\frac{N_{ij}^{10}}{N_{ij}^{00} + N_{ij}^{10}} + \frac{N_{ij}^{01}}{N_{ij}^{00} + N_{ij}^{01}} \right).$$

Recall that N_{ij}^{00}, N_{ij}^{11}, N_{ij}^{01}, and N_{ij}^{10} denote the number of correct/incorrect predictions made by two ensemble members h_i and h_j on the training set. In the first term $N_{ij}^{10}/(N_{ij}^{00} + N_{ij}^{10})$, the nominator corresponds to the number samples on which h_i is correct and its counterpart h_j is incorrect, whereas the denominator measures the total number of errors made by h_j. Therefore, the quotient expresses the *conditional probability* that h_i correctly classifies a sample given that h_j does not. We can derive the same observation regarding the other term. In order to keep the diversity term on the same scale as the accuracy, we take the average of these two probability estimates. This definition elegantly captures the notion of diversity: *pairs of individuals that make uncorrelated errors yield higher diversity*.

Definition 6.3. Let $G = (\mathcal{G}, \boldsymbol{\rho})$ be an induced subgraph game, where each node corresponds to an ensemble member h_i and the weights $\boldsymbol{\rho} = (\rho_{i,j})$ are defined as:

$$\rho_{i,j} = \begin{cases} Acc_i & if \quad i = j \\ Div_{i,j} & otherwise \end{cases} .$$

The weight assigned to a self-loop corresponds to the accuracy of h_i, while the weight of an edge linking two ensemble members h_i and h_j expresses the diversity between them.

Step 2. We rank the component learners using Shapley value. Under our framework, this solution concept measures the contribution of each member by considering its accuracy and the ensemble diversity. Formally, the rank assigned to a classifier h_i is given by:

$$\varphi_i = Acc_i + \frac{1}{2} \times \sum_{h_j \in \Omega \setminus \{h_i\}} Div_{i,j}. \tag{6.2}$$

Since all edge weights are non-negative, the payoff allocation (rank) provided by Equation 6.2 belongs to the core, and hence in addition to fairness, Shapley value guarantees stability. Equation 6.2 consists of two terms: the individual accuracy Acc_i and the diversity contribution $\frac{1}{2} \times \sum_{h_j \in \Omega \setminus \{h_i\}} Div_{i,j}$. The analysis of this equation can be summarized by two important observations: first, when two ensemble members are *similarly accurate*, Shapley value promotes the individual classifier that induces *higher diversity*; second, when two ensemble members have *equal diversity terms*, the one that *performs better* receives higher payoff allocation (rank). Therefore, the focus is on accurate members that contribute considerably to the overall ensemble diversity i.e. a fair balance between accuracy and diversity.

Step 3. The pruned ensemble is made of the individual classifiers whose Shapley values φ_i exceed a preset selection threshold σ. Exploratory experiments indicate that a value $\sigma = \sum_{i=1}^{n} \varphi_i / n$ is appropriate.

6.4 ISGCO algorithm

The pseudocode of ISGCO is depicted by Figure 6.1. The algorithm takes as an input a training set Γ, an ensemble of classifiers Ω, and a selection threshold σ. It begins with

```
 1: Input:      $\Gamma$: Training set.
                $\Omega$: Ensemble of classifiers.
                $\sigma$: Selection threshold.
 2: Initialize:   $\omega = \emptyset$;
                                                  /*Getting classifiers' predictions*/
 3:     For each $h_i \in \Omega$
 4:         For each $(x_j, y_j) \in \Gamma$
 5:             $Preds_j^i = h_i(x_j)$;
 6:         End for each $(x_j, y_j)$
 7:     End for each $h_i$
                                   /*Constructing the adjacency matrix using $Preds$*/
 8:     For each $h_i \in \Omega$
 9:         $\rho_{i,i} = Acc_i$;
10:         For each $h_j \in \Omega \setminus \{h_i\}$
11:             $\rho_{i,j} = Div_{i,j}$;
12:         End for each $h_j$
13:     End for each $h_i$
                                             /*Computing classifiers' Shapley values*/
14:     For each $h_i \in \Omega$
15:         $\varphi_i = \rho_{i,i} + \frac{1}{2} \times \sum_{h_j \in \Omega \setminus \{h_i\}} \rho_{i,j}$;
16:     End for each $h_i$

17:     For each $h_i \in \Omega$
18:         If $\varphi_i \geq \sigma$
19:             $\omega = \omega \cup \{h_i\}$;
20:         End if
21:     End for each $h_i$

22: Output:    $\omega$: Pruned ensemble.
```

Figure 6.1: The ISGCO algorithm.

computing the ensemble members' predictions $Preds$ of the training samples (lines [3-7]), and uses them to build the adjacency matrix $\boldsymbol{\rho}$ (lines [8-13]). Then, it estimates the individual contribution of every classifier using the definition provided by Equation 6.2 (lines [14-16]). Finally, selection is conducted by aggregating the ensemble members whose ranks are above the threshold σ (lines [17-21]).

6.5 Experiments

6.5.1 Experimental setup

To evaluate the performance of ISGCO, we conducted experiments using 35 datasets selected from the UCI benchmark Repository of Machine Learning Databases [114]. A summary of the datasets properties is provided in Appendix B, Table B.1.

We invoked the same resampling technique as Chapter 5 that we recall here for clarity. A detailed description of our experimental environment is provided in Appendix B. We resampled each dataset following Dietterich's 5×2 cross-validation to generate ten training and ten testing folds, denoted $train_i$, $test_i$, $i = 1, ..., 10$, respectively. We trained the classifiers and computed the adjacency matrix $\boldsymbol{\rho}$ using $train_i$, whereas the other fold $test_i$ was employed to measure the classification accuracy, the running time, and the pruning ratio. Note that we reported only the mean of these ten measurements.

In order to generate the initial ensemble, we used BAGGING with CART trees as a base learner to train a set of 100 classifiers. We set the selection threshold σ of ISGCO to $\sum_{i=1}^{n} \varphi_i / n$, where φ_i denotes the rank assigned to h_i, and n is the number of classifiers ($n = 100$). We compared ISGCO with eight state-of-the-art techniques: Accuracy ordering (BESTN), Semi Definite Programming (SDP) [31], Genetic Algorithm (GASEN) [34], Orientation Ordering (OO) [37], Margin Distance Minimization (MDSQ) [77] with a moving reference point p set to $\sqrt{i}$ at the i^{th} iteration, Boosting-Based (BB) [70], Complementarity Measure (CC) [77], and Reduce Error (RE) [36]. Note that BESTN, SDP, MDSQ, BB, CC, and RE methods prune the initial ensemble to a preset size L. Therefore, in order to make a fair comparison, we set L to the same size obtained by ISGCO.

6.5.2 Accuracy performance

Table 6.1 displays the classification accuracy results obtained by the different approaches on each dataset. The last row specifies the averaged rank of each method.

The results given in Table 6.1 indicate that ISGCO outperforms the other methods in most cases. In order to confirm the significance of the observed differences, we compared the performances of these pruning techniques using the average ranks over 35 datasets. Friedman test rejects the null hypothesis which states that all methods have similar performances with $F_F = 20.49 > F(9, 306) = 8.06$ for $\alpha = 1 \times 10^{-10}$($F_F$ is distributed according to the F distribution with $10 - 1 = 9$ and $(10 - 1) \times (35 - 1) = 306$ degrees of freedom). Because we are only interested in comparing ISGCO with the other alternatives, we then proceeded with a Bonferroni-Dunn test while considering ISGCO as the control algorithm. Figure 6.2 shows the results of a Bonferroni-Dunn test at a 5% significance level with the critical value $q_{0.05} = 2.77$ and the critical difference $CD = 2.01$.

Table 6.1: Summary of mean accuracy results.

Datasets	Bagging	Gasen	BB	OO	BestN	CC	Mdsq	Re	Sdp	Isgco
Arrhythmia	71.99±2.56	71.59±2.43	71.90±3.01	71.90±2.94	71.64±2.37	71.95±2.60	71.68±2.41	**72.17±2.67**	**72.17±2.78**	**72.17±2.86**
Australian	85.88±1.51	85.74±1.51	85.88±1.69	85.83±1.29	85.65±1.24	85.80±1.17	85.77±1.48	85.97±1.09	**86.03±1.43**	86.00±1.33
BCW	96.11±0.65	95.99±0.52	96.11±0.69	96.02±0.77	95.88±0.88	96.02±0.75	95.82±0.99	**96.28±0.69**	96.25±0.44	**96.28±0.42**
Car	95.71±1.04	95.82±0.65	96.08±0.74	96.20±0.69	95.87±0.70	95.98±0.92	95.88±0.71	96.04±0.84	95.79±0.94	**96.23±0.84**
Glass	71.59±4.52	70.65±3.77	72.34±4.43	72.90±4.34	72.90±4.01	72.06±4.37	72.71±3.46	72.90±4.72	69.63±4.67	**73.36±3.92**
Hayes-Roth	80.25±6.34	80.88±4.00	82.00±2.78	**82.50±4.08**	81.13±3.65	82.00±4.09	81.13±3.65	80.88±4.64	81.63±4.60	**82.50±3.91**
HDC	80.72±3.08	80.86±2.85	81.85±1.84	81.85±2.47	80.99±2.94	81.25±2.68	80.99±3.06	81.52±2.82	81.38±2.73	**82.44±2.55**
HDH	79.12±2.87	78.78±2.86	79.25±2.67	78.91±2.90	78.50±3.50	78.64±3.16	78.64±3.18	79.25±3.47	79.12±2.87	**79.46±2.77**
HDS	**56.31±2.89**	52.26±3.85	53.71±5.14	54.03±5.62	54.69±4.01	52.40±5.35	54.37±4.47	54.19±4.37	55.01±5.37	55.97±4.87
Hepatitis	**81.16±4.35**	79.73±5.04	80.64±4.25	79.99±5.51	79.86±6.32	80.39±4.84	79.99±6.23	80.77±4.24	80.38±5.23	80.77±5.54
Ionosphere	91.68±1.62	91.62±1.48	**92.65±2.43**	92.42±2.07	91.96±2.53	91.56±1.77	92.08±2.57	92.31±1.89	92.37±2.33	**92.65±1.91**
Labor	85.90±5.93	84.53±3.92	85.59±5.46	85.58±5.75	84.52±3.96	82.81±3.89	83.50±4.72	85.90±6.56	84.56±5.81	**86.63±7.10**
Lenses	65.83±8.29	72.50±13.64	72.50±10.43	75.00±13.61	69.17±12.45	72.50±7.91	74.17±12.08	73.33±12.30	71.67±8.96	**80.00±8.05**
Letter	91.56±0.62	91.44±0.50	91.63±0.47	**91.82±0.47**	91.27±0.65	91.66±0.55	91.29±0.66	91.65±0.47	89.25±4.95	91.75±0.40
Lymphography	78.78±4.37	77.97±3.55	**79.73±4.08**	79.46±4.67	78.24±4.20	78.24±5.11	78.51±4.39	78.92±4.28	78.51±4.43	79.59±4.83
MFF	79.40±1.22	79.01±1.09	79.28±1.03	79.48±1.25	79.31±1.26	79.23±0.97	79.24±1.32	79.27±1.03	79.14±1.14	**79.50±1.13**
MFKL	89.36±1.48	89.26±1.39	89.87±1.37	90.14±1.26	89.36±1.23	88.88±1.03	89.35±1.15	89.59±1.11	88.89±1.28	**90.33±1.01**
MFM	**71.40±1.06**	70.94±1.24	71.35±1.26	71.22±1.14	71.36±1.26	71.30±1.21	71.23±1.22	71.04±1.33	71.26±1.29	71.35±1.17
MFPC	94.16±0.87	93.98±0.97	94.21±0.96	94.43±1.05	94.03±0.95	93.25±1.08	94.04±0.98	94.11±0.99	94.48±0.92	**94.52±1.00**
MFZ	75.69±1.60	75.63±1.53	75.71±1.66	76.27±1.47	75.78±1.83	75.42±1.62	75.85±1.72	75.81±1.70	74.64±4.06	**76.29±1.41**
Musk1	82.65±3.26	81.97±2.16	83.53±2.05	83.28±2.92	83.28±3.66	81.26±2.97	83.32±3.54	82.98±3.08	**84.08±2.56**	83.78±2.55
Musk2	96.33±0.31	96.44±0.34	96.44±0.32	96.48±0.36	96.46±0.41	96.49±0.33	96.46±0.37	96.49±0.34	96.47±0.42	**96.52±0.33**
Nursery	99.08±0.16	99.13±0.19	99.14±0.13	99.15±0.14	99.07±0.13	99.13±0.16	99.07±0.13	99.16±0.13	99.09±0.15	**99.18±0.16**
Optical	95.54±0.49	95.55±0.43	95.79±0.39	96.02±0.48	95.41±0.50	95.65±0.44	95.43±0.50	95.60±0.48	95.53±1.05	**96.03±0.55**
Pen	97.64±0.24	97.59±0.19	97.75±0.19	97.84±0.19	97.60±0.21	97.69±0.17	97.60±0.21	97.74±0.20	97.67±0.37	**97.90±0.19**
Soybean L	91.54±1.57	92.04±1.39	92.01±1.71	91.89±1.60	91.33±1.44	91.80±1.39	91.36±1.41	91.74±1.60	91.68±1.73	**92.09±1.52**
Soybean S	**98.71±2.91**	98.70±2.93	**98.71±2.91**	**98.71±2.91**	98.28±3.02	96.54±5.72	98.28±3.02	**98.71±2.91**	96.54±5.72	**98.71±2.91**
Spambase	92.96±0.69	93.11±0.51	93.23±0.67	93.18±0.63	92.96±0.66	**93.25±0.68**	92.99±0.68	93.11±0.63	93.01±0.74	**93.25±0.64**
Thyroid D	99.54±0.09	99.58±0.09	99.59±0.09	99.58±0.11	99.57±0.10	99.59±0.11	99.57±0.10	99.59±0.11	**99.61±0.10**	99.59±0.12
Waveform1	**83.65±0.68**	83.38±0.63	83.62±0.65	83.54±0.67	83.32±0.63	83.60±0.60	83.31±0.69	83.54±0.76	82.30±1.74	83.57±0.63
Waveform2	82.86±0.64	82.78±0.61	82.56±0.69	82.79±0.80	82.53±0.73	82.89±0.56	82.59±0.71	**82.90±0.58**	82.51±0.77	82.83±0.73
Wine	94.49±3.69	94.38±3.39	95.62±2.51	95.73±3.03	93.48±3.78	92.70±4.28	93.48±3.78	93.82±3.87	94.61±4.39	**95.84±3.05**
WDBC	93.71±1.83	93.85±1.87	94.13±1.64	94.13±1.69	93.88±1.83	93.92±1.86	93.88±1.93	93.74±1.79	93.71±2.02	**94.73±1.36**
WPBC	**77.17±1.97**	75.45±2.13	75.56±3.26	75.96±3.04	76.46±2.13	75.66±2.45	76.46±2.82	76.06±1.58	76.06±2.52	76.26±3.20
Yeast	**59.92±1.20**	59.57±1.47	59.54±1.48	59.35±1.79	59.29±1.45	59.69±1.39	59.33±1.51	59.51±1.55	59.39±1.41	59.80±1.67
Average ranks	**5.76**	**7.76**	**4.21**	**4.10**	**7.43**	**6.24**	**7.07**	**4.49**	**6.19**	**1.76**

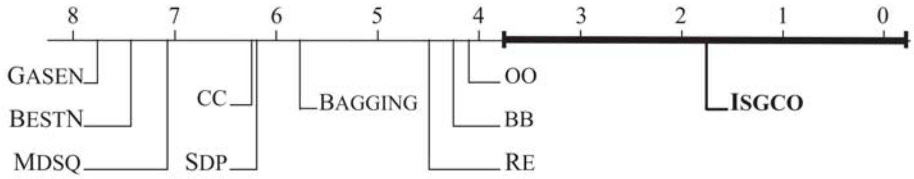


Figure 6.2: Comparison of ISGCO with the other pruning approaches using Bonferroni-Dunn test.

The analysis of Bonferroni-Dunn test results illustrated by Figure 6.2 can be summarized by two main observations: (1) We notice that ISGCO has the lowest rank and all the other pruning techniques fall outside the marked interval. Therefore, we can conclude that ISGCO significantly improves the original ensemble and outperforms the other alternatives, which is consistent with our initial observations. (2) The technique GASEN exhibits very poor performance, which is not expected since search-based approaches are slow but generally very effective and accurate than ranking-based methods. A possible cause of this behavior might be related to the size of the ensemble found by GASEN that will be investigated in Section 6.5.3.

6.5.3 Pruning ratio

Table 6.2 reports the pruning ratios obtained by GASEN, OO and ISGCO on each dataset. We specify in the last row the average pruning ratios over all datasets. We excluded BB, BESTN, CC, MDSQ, RE, and SDP from this comparison because these approaches yield the same pruning ratio results as ISGCO.

Table 6.2 indicates that GASEN achieves the best pruning ratio followed by ISGCO and OO. In addition, the reported results support our previous claim with regard to the behavior of GASEN (refer to Section 6.5.2). The analysis of both Tables 6.1 and 6.2 reveals that GASEN fails to extract the appropriate number of classifiers, which causes a drop in its performance.

6.5.4 Pruning time

Table 6.3 presents the average pruning time (in milliseconds) required by each ensemble technique over all datasets.

The ensemble techniques BESTN and OO yield the lowest pruning times, whereas the second-best result is attributed to ISGCO, BB, and MDSQ. Although ISGCO does not achieve the best running time, it succeeds in extracting very accurate sub-ensembles,

Table 6.2: Pruning ratio (%).

Datasets	Gasen	OO	Isgco
Arrhythmia	69.60±10.08	44.90±3.93	47.40±2.63
Australian	76.90±14.26	46.70±4.16	50.50±3.92
BCW	80.50±11.40	49.70±4.40	55.50±4.06
Car	69.00±10.68	47.10±2.77	48.20±2.15
Glass	78.40±10.73	47.80±2.74	50.50±2.07
Hayes-Roth	64.00±15.64	48.90±2.69	54.50±3.47
HDC	64.80±15.82	48.30±3.80	52.00±3.13
HDH	80.20±14.09	47.00±2.67	50.60±4.38
HDS	60.70±10.34	46.00±2.26	48.10±3.73
Hepatitis	79.70±10.56	48.20±3.52	54.50±6.62
Ionosphere	83.30±8.33	52.10±3.31	54.70±3.06
Labor	75.10±10.66	48.30±8.19	53.50±6.33
Lenses	77.30±12.43	58.30±11.83	64.10±4.79
Letter	51.00±3.53	45.70±2.67	48.70±3.62
Lymphography	72.00±15.30	49.00±2.62	52.50±3.72
MFF	54.90±7.98	43.10±2.33	46.50±2.95
MFKL	61.90±12.16	42.00±4.22	46.00±3.23
MFM	58.90±11.47	45.90±3.38	48.10±2.85
MFPC	56.60±9.75	45.40±3.20	47.70±3.09
MFZ	48.00±3.53	45.40±2.50	47.10±3.35
Musk1	66.90±12.69	48.30±3.68	48.80±3.08
Musk2	77.00±7.36	50.40±3.53	53.80±3.22
Nursery	70.60±10.29	47.70±2.58	50.80±2.62
Optical	54.00±8.00	50.60±2.07	50.50±4.14
Pen	56.00±10.97	50.70±2.21	51.70±5.33
Soybean L	71.50±11.57	49.90±2.85	50.20±3.74
Soybean S	83.70±8.38	46.20±5.20	59.30±4.40
Spambase	76.00±11.09	51.70±2.87	53.40±3.44
Thyroid D	81.60±15.03	49.30±4.64	55.40±2.22
Waveform1	54.20±7.07	43.60±3.24	45.70±3.62
Waveform2	52.20±4.13	43.90±2.02	49.00±1.83
Wine	73.40±9.13	49.10±3.54	52.00±5.16
WDBC	79.90±14.00	50.50±3.92	51.80±4.29
WPBC	76.60±10.01	46.60±4.81	50.50±3.10
Yeast	56.40±8.63	46.00±3.83	46.10±3.07
Mean	**68.37±10.49**	**47.84±3.66**	**51.13±3.61**

Table 6.3: Average pruning times (in milliseconds).

Gasen	BB	OO	BestN	CC	Mdsq	Re	Sdp	Isgco
56560	33.8	3.08	1.70	106	37.2	6420	223	19.5

requiring relatively low computational costs. Furthermore, as one should expect, search-based schemes Gasen and Sdp deliver the worst pruning times.

6.5.5 Effect of the ensemble size on the classification accuracy

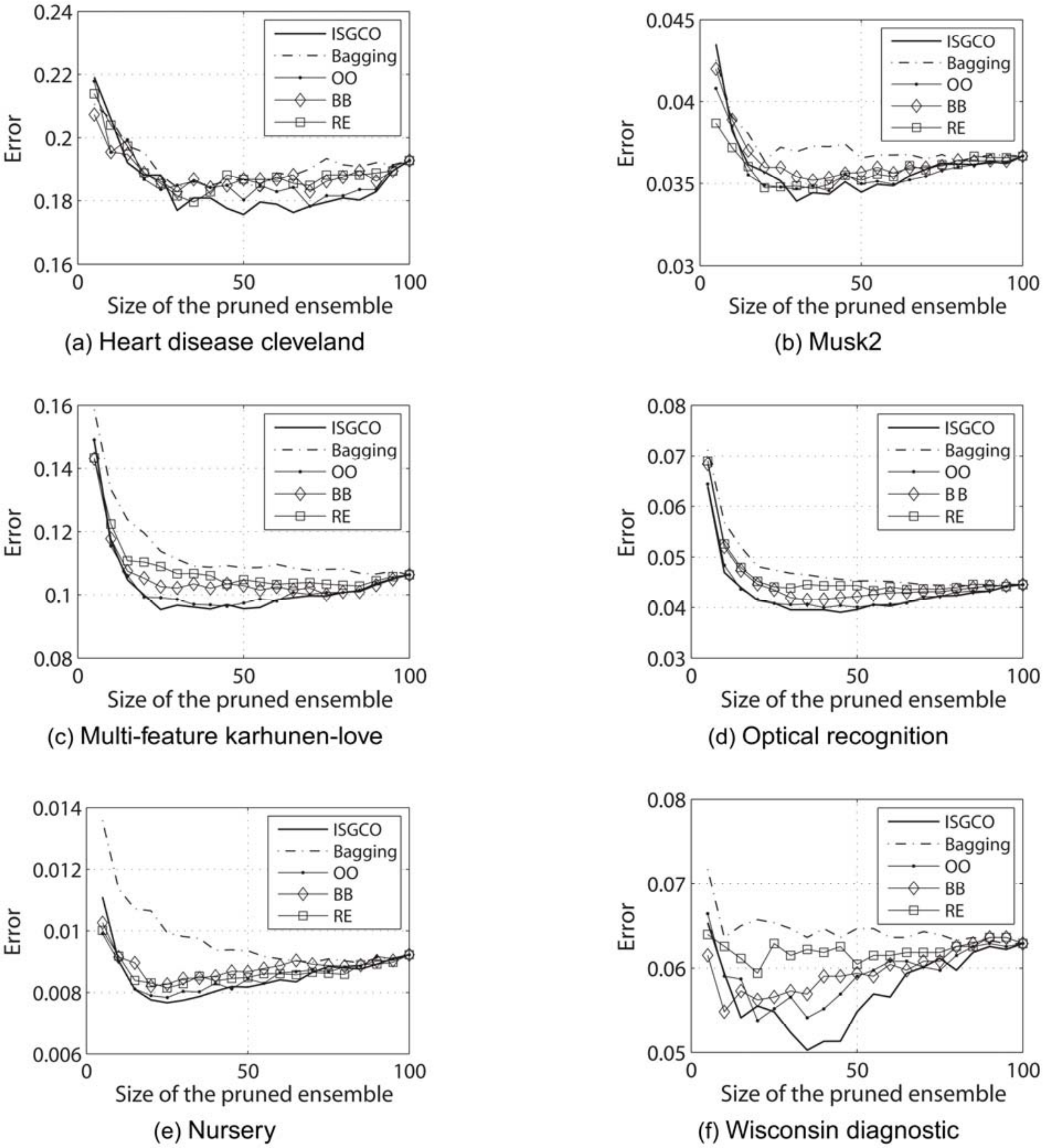
This section is devoted to investigate how the size of the pruned ensemble L influences the performance of Isgco, OO, BB, and Re. We carried out the following experiment: We varied L from 5 to 100, and plotted the error curves for six datasets: Heart disease cleveland, Musk2, Multi-feature karhunen-love, Optical recognition, Nursery, and Wisconsin diagnostic breast cancer. We also reported the error curves of BAGGING. To this end, we aggregated the individual classifiers in the same order as they were included in the initial ensemble. We generated the ensemble members using the training fold and estimated the error rates on the test fold. The reported results are averaged measurements of 10 partitions of the six datasets. The results of this experiment are illustrated by Figure 6.3.

The curves shown by Figure 6.3 indicate that, in case of unordered BAGGING, the test error decreases as the number of selected classifiers increases, then it settles at a certain rate and keeps it with little variations, whereas the error obtained by the pruning techniques drops rapidly and attains lower rates than BAGGING; after that, it increases until reaching the error rate of the entire ensemble. Particularly, as reported by Figure 6.3 (c)-(e), we notice that ISGCO and OO exhibit comparable performance. Furthermore, overall, ISGCO achieves better error rates than the other alternatives.

6.5.6 Comparison with SCG-Pruning

In order to determine which approach performs better SCG-Pruning or ISGCO, we carried out the following experiment. We generated an ensemble composed of 100 J48 trees using BAGGING. We set q_1 and q_2 to the values that maximize the accuracy rate on the training fold. We compared ISGCO with the SCG-Pruning variant SCG-MI. It is worth underscoring that we skipped step 3 of ISGCO and set the size of the final ensemble to the same value obtained by SCG-MI. The results of this experiment are given in Table 6.4. Columns 2 and 3 represent the accuracy rates scored by ISGCO and SCG-MI, respectively. Column 4 specifies the ranks for the difference in performances between these two pruning techniques. Column 5 shows the pruning ratio results obtained by SCG-Pruning.

The next step consists of testing weather the observed differences are statically significant or are merely random using Wilcoxon signed-ranks test. We have found: the sum for ranks for positive and negative differences $R^+ = 342.5$, $R^- = 287.5$, and the statistics $z = -0.45 > -1.97$ for a significance level $\alpha = 0.05$. The value of z provides a strong evidence that the observed differences are *not significant*; hence, SCG-MI and ISGCO perform similarly.



(a) Heart disease cleveland

(b) Musk2

(c) Multi-feature karhunen-love

(d) Optical recognition

(e) Nursery

(f) Wisconsin diagnostic

Figure 6.3: Test error curves of various ensemble approaches.

Table 6.4: Comparison of ISGCO with SCG-Pruning.

Datasets	ISGCO	SCG-MI	Rank	Pruning ratio (%)
Arrhythmia	70.22	72.04	32.00	71.40
Australian	85.57	85.19	-19.00	87.20
BCW	95.85	95.99	10.00	83.20
Car	95.42	95.65	14.50	76.00
Glass	69.63	71.40	30.00	71.20
Hayes-Roth	73.75	81.50	34.00	82.00
HDC	80.66	79.87	-22.00	71.80
HDH	80.95	79.32	-28.00	78.20
HDS	41.47	39.69	-31.00	61.40
Hepatitis	81.80	78.83	-33.00	76.60
Ionosphere	90.77	91.34	21.00	76.80
Labor	82.75	84.51	29.00	77.20
Lenses	77.50	78.33	24.00	68.00
Letter	91.50	91.52	4.00	18.00
Lymphography	78.24	77.30	-26.00	74.60
MFF	79.19	79.36	11.00	39.40
MFKL	89.56	89.17	-20.00	38.80
MFM	71.03	71.28	16.00	65.20
MFPC	93.98	93.85	-9.00	60.80
MFZ	76.07	75.86	-13.00	25.00
Musk1	81.22	82.27	27.00	71.60
Musk2	96.11	96.34	14.50	64.20
Nursery	99.09	99.10	3.00	57.20
Optical	95.45	95.56	8.00	37.00
Pen	97.61	97.54	-6.00	55.20
Soybean L	91.48	91.48	1.50	73.20
Soybean S	88.53	99.13	35.00	85.00
Spambase	92.92	92.85	-7.00	73.20
Thyroid D	99.55	99.58	5.00	86.80
Waveform1	83.63	83.27	-18.00	36.20
Waveform2	82.88	82.70	-12.00	39.40
Wine	93.93	93.93	1.50	78.00
WDBC	93.08	93.88	23.00	84.40
WPBC	76.16	75.25	-25.00	77.00
Yeast	59.34	59.04	-17.00	54.20

Combining the above results and those reported in Chapter 5, we can derive two important conclusions:

- Generally the average pruning ratio of SCG-Pruning ($\approx 65.01\%$) is *higher* than that of ISGCO. Consequently, the notion of the minimal winning coalition provides a more *powerful* and *better* criterion for determining the pruned ensemble size than ISGCO.

- The technique ISGCO is *10 times* faster and has *lower* computational complexity than SCG-Pruning.

6.6 Summary

This chapter introduced an induced subgraph game for classifier ordering. We have devised a selection criterion that measures the contributions of the ensemble members by considering the tradeoff between the individual accuracies and the diversity of the group based on Shapley value. The experimental results indicate that ISGCO provides a reliable ranking, succeeds in improving the initial ensemble performance, and outperforms some major state-of-the art pruning approaches.

The comparison between ISGCO and SCG-Pruning reveals that ISGCO achieves comparable accuracy rates to SCG-Pruning, with slightly lower pruning ratio and better computational costs.

CHAPTER 7
CONCLUSION

The primary research questions that motivated this thesis are: *can we derive a selection criterion which promotes interactive base learner?*, and *at what cost?* These concerns are of paramount importance because of two reasons: First, most scoring functions assess the utility of the base learners according to their *individual* contributions to the ensemble performance, but neglect the interactions that might exist among the different component classifiers. Consequently, these approaches underestimate the efficacy of members that have strong discriminatory power as a group but are weak as individuals, which often yields poor generalization performance. Second, many evaluation criteria are hand-designed and sometimes involve the computation of large multivariate densities. As a result, it became of urgent importance to develop selection criteria within a powerful mathematical framework like game theory.

In the present chapter, we first summarize the contributions of this thesis, providing answers to the above questions. Then, we present several areas for improvement.

7.1 Contributions and novelty of this thesis

This thesis introduced original ensemble selection criteria which have been founded upon game theory principles. The idea of modeling the pruning task in terms of games constitutes the novelty of our work in the field of ensemble learning.

In Chapter 5, we proposed a simple coalitional game-based framework that extracts sub-ensembles with *moderate diversities*. First, we considered a player as an individual learner and defined the worth of a coalition based on the notion of diversity. Then, we ordered the component learners according to their Banzhaf power indices. It is worth mentioning that the computation of this solution concept is intractable for moderate and large ensemble sizes. Nevertheless, we were able to derive a new formulation and showed that its time complexity is pseudo-polynomial. Finally, we mapped the pruned ensemble to the minimal winning coalition, a well-known concept in CGT and political science, of the proposed game.

In Chapter 6, we perused an area of improvement: instead of considering *diversity as the sole indicator of the predictive performance*, we explored *balancing accuracy with diversity* to acquire a better approximation of the ensemble generalization ability. We formulated classifier ranking as an induced subgraph game played among the ensemble members. The weights assigned to a self-loop measures the corresponding classifier accuracy, whereas the weight of an edge that links two members is defined based on a pairwise diversity function. We evaluated the utility of the base learners according to their marginal contribution in achieving a proper balance between accuracy and diversity using Shapley value. Specifically, given two ensemble members h_1 and h_2, if they contribute similar diversity, then Shapley value promotes the one which yields better predictive performance estimated on a separate set of samples; otherwise, if h_1 and h_2 are similarly accurate, then the one that increases the ensemble diversity receives a higher rank. Finally, we conducted the pruning by discarding the individual learners whose ranks are below a preset selection threshold.

7.2 Strengths and limitations

We introduced throughout this thesis new theoretical frameworks to address ensemble pruning. These frameworks were developed within the context of game theory, which allowed us to provide a strong characterization of the pruning task in terms of solution concepts. To the extent of our knowledge, this work is one of the very few attempts that addressed an ensemble learning problem from a game theory perspective. This direction of research distinguishes this thesis from others in the literature and can inspire the machine learning community to conduct more studies guided by game theory principles.

The proposed games were defined based on several measures such as mutual information, double fault, or Cohen's kappa, and could operate with other metrics as well. We demonstrated the efficacy of our methodologies through extensive experimental investigations using a large set of benchmark datasets which cover a wide range of application domains. We also supported our analysis based on powerful statistical tests.

In the previous section, we summarized the main contributions of this thesis. However, it is of paramount importance to specify the limitations of our work. Some of these cases will be revisited in the next section as potential future lines of research.

The underlying game presented in Chapter 5 is defined in terms of two quotas q_1 and q_2. The success of this approach relies considerably on determining their values. In our experiments, we only conducted a cross-validation strategy to set these parameters. However, this approach adjusts the thresholds for each task and can therefore be less efficient computationally. Another matter that this thesis did not explore is that, the experimental analysis was conducted only on ADABOOST and BAGGING ensembles and did not perform comparisons with other ensemble techniques such as random forest.

7.3 Future work

This thesis has revealed several interesting areas for improvement. The first area is based upon the insights gained from Chapter 5. In that chapter, we evaluated the contribution o an ensemble member based on Banzhaf power index. The selection mechanism is considerably affected by the two thresholds q_1 and q_2. A natural extension of this work would be to exploit the findings reported in Section 5.3.2 so that they can be properly set for real world applications.

Another appealing work direction would be to study in depth the impact of pruning on the performance of other committee generators such as random subspace, random forest, or even simulating artificial ensemble predictions (high vs. low correlation of the base learners, etc.) . This study will provide practical guidelines for incorporating ensemble methods in real world applications.

Finally, with the increasing interests in Big data problems, it has become of paramount importance to fully make sense of the challenges and develop new ways of thinking to address them. For instance, an ensemble can be trained to learn from data that come from numerous sources and have different representations. Each ensemble member is built locally; hence, it is specialized on a portion of the feature space. The aggregation of all the component learners covers the whole representation of the problem. In this context, an ensemble can therefore improve the predictive performance.

APPENDIX A

NOTATION AND ACRONYMS

CGT	: Coalitional Game Theory
EGT	: Evolutionary Game Theory
NE	: Nash Equilibrium
ESS	: Evolutionary Stable Strategies
SCG-Pruning	: Simple Coalitional Game-based Pruning (Chapter 5)
Isgco	: Induced Subgraph Game for Classifier Ordering (Chapter 6)

$\mathcal{X}$	: feature space
$\mathcal{Y}$	: set of class labels
Γ	: a set made of m labeled samples $\Gamma = \{(x_1, y_1), ..., (x_m, y_m)\}$
x	: a feature vector
y	: a class label
$h(.)$	: hypothesis (learner)
$\mathcal{H}$	: hypothesis space
$h^\star$	: optimal hypothesis
Ω	: an ensemble composed of n classifiers $\Omega = \{h_1, ..., h_n\}$
ω	: pruned ensemble
$\mathbb{I}(c)$	: indicator function which returns 1 if the condition c is met, 0 otherwise
$\lvert.\rvert$	: size of a set
$[.]^T$	: column vector
$[.]$	: row vector
$\{...\}$	: set
$I(.;.)$	: mutual information between two random variables
$I(.;.\lvert.)$	: conditional mutual information

APPENDIX B
GUIDELINES FOR THE DESIGN OF ENSEMBLE LEARNING EXPERIMENTS

B.1 Datasets

We conducted the experimental analysis based on a large set of UCI benchmark datasets [114]. The collected datasets cover a wide range of application domains. The dimensions of these databases vary from 4 to 262, and the number of samples ranges from 16 to 20000. Some datasets contain missing values due to several factors such as: inaccurate measurements, defective equipment, and human errors. Table B.1 summarizes the properties of these datasets.

B.2 Ensemble generation

In order to create the initial committee, we invoked numerous classification models chosen from WEKA 3.6 [4], PRTOOLS 5.0.2 [116], and LIBSVM 3.18 [117]. Some classifiers do not support missing values. To overcome this problem, we replaced every missing entry with the mean and the mode for numeric and nominal features, respectively. Table B.2 gives a summary of these learning algorithms and their settings. We set the rest of the parameters to their default values.

B.3 Model evaluation and comparison

B.3.1 Performance evaluation

We resampled every dataset following Dietterich's 5×2 cross validation (cv), where stratified 2-fold cv was performed five times. Specifically, 2-fold cv divides the sample set into two equal-sized folds denoted $train$ and $test$. We trained the ensemble members on $train$, whereas the other fold $test$ was employed to measure some performance metrics (refer to Section B.3.2). Then, the roles of these two folds were swapped in order to obtain another estimate of the evaluation metric. Repeating these steps five times, we obtained at the end ten trained ensembles and evaluation measures. It is noteworthy that we report only the average of these ten measurements.

Table B.1: Properties of all datasets used in the experiments.

Datasets	Abbreviations	Samples	Features	Missing values	Classes
Anneal	*Anneal*	898	38	Yes	6
Arrhythmia	*Arrhythmia*	452	262	Yes	13
Audiology	*Audiology*	226	69	Yes	24
Australian credit approval	*Australian*	690	14	No	2
Balance	*Balance*	526	4	No	3
Balloons adult+stretch	*Balloons1*	20	4	No	3
Balloons adult-stretch	*Balloons2*	20	4	No	3
Balloons small-yellow	*Balloons3*	20	4	No	3
Balloons small-yellow+adult-stretch	*Balloons4*	16	4	No	3
Breast cancer wisconsin	*BCW*	699	9	Yes	3
Breast cancer	*BC*	286	9	Yes	2
Car evaluation	*Car*	1728	6	No	4
Chess King-Rook vs King-Pawn	*Chess*	3196	36	No	2
Congressional voting records	*CVR*	435	16	Yes	2
Credit approval	*Credit*	690	15	Yes	2
Cylinder bands	*Cylinder*	540	39	Yes	2
Dermatology	*Dermatology*	366	34	Yes	6
Ecoli	*Ecoli*	336	8	No	8
Glass identification	*Glass*	214	10	No	6
Hayes-Roth	*Hayes-Roth*	160	5	No	4
Heart disease cleveland	*HDC*	303	13	Yes	5
Heart disease hungarian	*HDH*	294	13	Yes	5
Heart disease switzerland	*HDS*	123	13	Yes	5
Hepatitis	*Hepatitis*	155	19	Yes	2
Ionosphere	*Ionosphere*	351	34	No	2
Iris	*Iris*	150	4	No	3
Labor	*Labor*	57	16	Yes	2
Lenses	*Lenses*	24	4	No	3
Letter recognition	*Letter*	20000	16	No	26
Low resolution spectrometer	*LRS*	531	102	No	48
Lymphography	*Lymph*	148	18	No	4
Monks1	*Monks1*	556	6	No	2
Monks2	*Monks2*	601	6	No	2
Monks3	*Monks3*	554	6	No	2
Multi-feature fourier	*MFF*	2000	76	No	10
Multi-feature karhunen-love	*MFKL*	2000	64	No	10
Multi-feature morphological	*MFM*	2000	6	No	10
Multi-feature profile correlations	*MFPC*	2000	216	No	10
Multi-feature zernike	*MFZ*	2000	47	No	10
Mushroom	*Mushroom*	8124	22	Yes	2
Musk1	*Musk1*	476	166	No	2
Musk2	*Musk2*	6598	166	No	2
Nursery	*Nursery*	12960	8	No	5
Optical recognition of handwritten digits	*Optical*	5620	64	No	10
Page blocks	*Page blocks*	5473	10	No	5
Pen-based recognition of handwritten digits	*Pen*	10992	16	No	10
Pima indians diabetes	*Pima*	768	8	No	2
Post-operative patient	*POP*	90	8	Yes	3
Soybean large	*Soybean L*	683	35	Yes	19
Soybean small	*Soybean S*	47	35	No	4
Spambase	*Spambase*	4601	57	No	2
SPECT heart	*SPECT*	267	22	No	2
SPECTF heart	*SPECTF*	267	44	No	2
Teaching assistant evaluation	*TAE*	151	5	No	3
Thyroid domain	*Thyroid D*	7200	21	No	3
Thyroid gland	*Thyroid G*	215	5	No	3
Tic-Tac-Toe endgame	*Tic-Tac-Toe*	958	9	No	2
Waveform (version 1)	*Waveform1*	5000	21	No	3
Waveform (version 2)	*Waveform1*	5000	40	No	3
Wine	*Wine*	178	13	No	3
Wisconsin diagnostic breast cancer	*WDBC*	569	30	No	2
Wisconsin prognostic breast cancer	*WPBC*	198	32	Yes	2
Yeast	*Yeast*	1484	8	No	10
Zoo	*Zoo*	101	16	No	7

B.3.2 Performance comparison

We compared our approaches with numerous pruning techniques such as: Semi Definite Programming (SDP) [31], Genetic Algorithm (GASEN) [34], Orientation Ordering (OO) [37], Margin Distance Minimization (MDSQ) [77], and Kappa pruning (KAPPA)

Table B.2: List of all learning algorithms used in the experiments.

Algorithm	Platform	Description
AdaBoost	Weka	ADAptive BOOSTing.
Bagging	Weka	Bootstrap AGGregatING. The size of a bootstrap sample is set to the number of training instances.
J48	Weka	C4.5 decision tree with the confidence factor set to 0.25. $2/3$ of the training data were used for growing the tree, and $1/3$ for pruning it.
Cart	Weka	Decision tree learner using CART's minimal cost complexity pruning.
Decision Stump	Weka	This learning algorithm builds one-level decision trees.
Logistic	Weka	Multinomial logistic regression.
IBk	Weka	K-nearest neighbors classifier using linear search with the Euclidean distance, and 3 values for $k = 1, 3, 5$.
OneR	Weka	1R rule-based learning algorithm.
NaïveBayes	Weka	Standard probabilistic naïve Bayes classifier using supervised discretization.
Multilayer Perceptron	Weka	Multilayer perceptron classifier using backpropagation algorithm run for 500 epochs with $(f + 1 + k)/2$ layers, where, f designates the number of features and k is the number of classes of a dataset. The learning rate was set to 0.3, and the momentum coefficient to 0.2.
Decision Table	Weka	Simple decision table majority classifier using BestFirst or Genetic search methods with accuracy as the evaluation measure.
JRip	Weka	RIPPER (Repeated Incremental Pruning to Produce Error Reduction) algorithm for rule induction. $2/3$ of the training data were used for growing rules, and $1/3$ for pruning them.
PART	Weka	PART decision list built using J48 with the confidence factor set to 0.25. $2/3$ of the training data were used for growing rules, and $1/3$ for pruning them.
Fisherc	PRTools	Fisher's least square linear classifier.
Ldc	PRTools	Linear Bayes normal classifier. No regularization was performed.
Qdc	PRTools	Quadratic Bayes normal classifier. No regularization was performed.
Parzendc	PRTools	Parzen density based classifier. The smoothing parameters were estimated from the training data for each class.
SVM	LibSVM	Support vector machines using a radial (Gaussian) kernel with $\gamma = 1/f$ where f is the number of features, a polynomial kernel of degree 3, or a linear kernel. The cost parameter C was set to 1.0.

[36]. A summary of these pruning algorithms and their settings are given in Table B.3. It is worth noting that we invoked MIToolbox library [115] in order to compute the information theory concepts.

In our experiments, we measured the following metrics:

Accuracy performance: is the ratio between the number of correctly classified samples to the total number of samples estimated on a separate set of instances.

Table B.3: Summary of all invoked pruning techniques and their settings.

Pruning technique	Description	Configurations
ALL	No pruning.	ALL combines all classifiers predictions using majority vote.
BESTN	Accuracy ordering [9].	The size of the pruned ensemble L was set to same values obtained by our approaches.
BEST	Accuracy ordering with $L = 1$ [8].	The size of the pruned ensemble L was set to 1.
KAPPA	Kappa pruning [36].	The size of the pruned ensemble was either set to 9 or to the same values found by our approaches.
FS-MID	Forward Selection [76].	We have implemented the MID criterion proposed by Meynet [2] to measure the goodness of a candidate ensemble. The size of the pruned ensemble was set 9.
BE-MID	Backward Elimination [76].	We have implemented the MID criterion proposed by Meynet [2] to measure the goodness of a candidate ensemble. The size of the pruned ensemble was set to 9.
EXH-MID	Exhaustive search [8].	We have implemented the MID criterion proposed by Meynet [2] to measure the goodness of a candidate ensemble. The size of the pruned ensemble was set to 9.
RE	Reduce Error [36]	The size of the pruned ensemble was set to same values obtained by our approaches.
CC	Complementarity Measure [77]	The size of the pruned ensemble was set to same values obtained by our approaches.
MDSQ	Margin Distance Minimization [37].	We have used a moving reference point set to either $2\sqrt{2 \times i}/n$ or $\sqrt{i}$ at iteration i, or 0.075, where n is the size of the initial committee. The size of the pruned ensemble was set to same values obtained by our approaches.
OO	Orientation Ordering [37]	The size of the pruned ensemble was set to same values obtained by our approaches.
BB	Boosting-Based [70]	The original implementation does not have relevant parameters. The size of the pruned ensemble was set to same values obtained by our approaches.
SDP	Semi Definite Programming [31]	This technique invokes the SDPA library version 7.3.6 [118] to solve the semi definite programming problem.
GASEN	Genetic Algorithm based Selective ENsemble [34].	We have evolved a population made of 100 individuals over 100 generations. The mutation and the crossover probabilities were set to 0.05 and 0.6, respectively.
SCG-MI, SCG-κ, SCG-DIS	Simple Coalitional Game -Pruning [46].	We have implemented SCG-Pruning with three pairwise diversity functions: disagreement measure (SCG-DIS), Cohen's kappa (SCG-κ), and mutual information (SCG-MI) defined by Equations 2.11, 2.15, and 2.13, respectively. The size of the pruned ensemble was determined automatically (the size of the minimal winning coalition). The parameters q_1 and q_2 were set following a cross-validation strategy.
ISGCO	Induced Subgraph Game for Classifier Ordering [47].	We set the selection threshold σ to $\sum_{i=1}^{n} \varphi_i/n$, where φ_i denotes the rank assigned to h_i, and n is the initial committee.

Given a testing set $\Gamma_{test} = \{(x_1, y_1), ..., (x_t, y_t)\}$ and a trained committee Ω, the accuracy rate is formulated as:

$$acc(\Gamma) = \frac{1}{t} \sum_{i=1}^{t} \mathbb{I}(\Omega(x_i) = y_i), \tag{B.1}$$

where $\Omega(x_i)$ denotes the estimate of instance i's class label produced by the ensemble Ω.

In order to determine which technique performs best over multiple datasets, we followed Demšar's strategy [52]. We conducted a Friedman test. This latter first ranks the techniques for each dataset separately according to the generalization accuracy in descending order. More specifically, the best performing technique gets rank 1, the second best gets rank 2,....Then, it compares the average ranks of these algorithms. Under the null hypothesis, we assume that all the techniques perform similarly; hence their ranks should be equal. The rejection of this hypothesis confirms the existence of at least one pair of algorithms with significantly different performances. If this confirmation is obtained, we proceed with either a *Nemenyi test* or a *Bonferroni-Dunn test*. Nemenyi test is performed when all methods are compared with each other, whereas Bonferroni-Dunn test is useful when we are only interested in comparing all techniques with a control algorithm.

Pruning ratio: let Ω and ω denote, respectively, the initial committee and the pruned ensemble. The pruning ratio δ is defined as:

$$\delta = \frac{|\Omega| - |\omega|}{|\Omega|} \times 100. \tag{B.2}$$

Pruning time: the experiments were conducted on a 3.6 GHZ Intel Core $i7-4790$ processor with 8 GB of system memory.

REFERENCES

1. Dietterich, T. G., “Ensemble methods in machine learning.”, Multiple Classifier Systems, (2000), p. 1–15.

2. Meynet, J. and Thiran, J.-P., “Information theoretic combination of pattern classifiers”, Pattern Recognition, vol. 43, no. 10, (2010), p. 3412–3421.

3. Duda, R. O., Hart, P. E. and Stork, D. G., Pattern classification, Wiley-Interscience, (2000).

4. Witten, I. H. and Frank, E., Data mining: Practical machine learning tools and techniques, Morgan Kaufmann Publishers, San Francisco, California, 3rd edn., (2011).

5. Alpaydın, E., Introduction to machine learning, MIT Press, Cambridge, MA, 2nd edition edn., (2010).

6. Japkowicz, N. and Shah, M., Evaluating learning algorithms, Cambridge University Press, Cambridge, (2011).

7. Hastie, T., Tibshirani, R. and Friedman, J., The elements of statistical learning: data mining, inference and prediction, Springer-Verlag, New York, USA, second edn., (2008).

8. Zhou, Z.-H., Ensemble methods: Foundations and algorithms, Taylor & Francis, Boca Raton, FL, 1st edn., (2012).

9. Rokach, L., Pattern classification using ensemble methods, World Scientific Publishing Company, Singapore, 1st edn., (2010).

10. Martínez-Muñoz, G., Hernández-Lobato, D. and Suárez, A., “An Analysis of Ensemble Pruning Techniques Based on Ordered Aggregation”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 2, (2009), p. 245–259.

11. Breiman, L., “Bagging predictors”, Machine Learning, vol. 24, no. 2, (1996), p. 123–140.

12. Freund, Y. and Schapire, R. E., “A decision-theoretic generalization of on-line learning and an application to boosting”, Journal of Computer and System Sciences, vol. 55, no. 1, (1997), p. 119–139.

13. Sun, S., “An improved random subspace method and its application to EEG signal classification”, Multiple Classifier Systems, (2007), p. 103–112.

14. Ho, T. K., “The random subspace method for constructing decision forests”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 2, (1998), p. 832–844.

15. Breiman, L., “Random forests”, Machine Learning, vol. 45, no. 1, (2001), p. 5–32.

16. Sun, S., “Local within-class accuracies for weighting individual outputs in multiple classifier systems”, Pattern Recognition Letters, vol. 31, no. 2, (2010), p. 119–124.

17. Han, M. and Liu, B., “Ensemble of extreme learning machine for remote sensing image classification”, Neurocomputing, vol. 149, (2015), p. 65–70.

18. Mashhoori, A., “Block-wise two-directional 2DPCA with ensemble learning for face recognition”, Neurocomputing, vol. 108, (2013), p. 111–117.

19. Kavitha, B., Karthikeyan, S. and Maybell, P. S., “An ensemble design of intrusion detection system for handling uncertainty using Neutrosophic Logic Classifier”, Knowledge-Based Systems, vol. 28, (2012), p. 88–96.

20. Rokach, L., Romano, R. and Maimon, O., “Negation recognition in medical narrative reports”, Information Retrieval, vol. 11, no. 6, (2008), p. 499–538.

21. Rokach, L., “Ensemble-based classifiers”, Artificial Intelligence Review, vol. 33, no. 1, (2010), p. 1–39.

22. Zhou, Z.-H. and Li, N., “Multi-information ensemble diversity”, Multiple Classifier Systems, (2010), p. 134–144.

23. Didaci, L., Fumera, G. and Roli, F., “Diversity in classifier ensembles: Fertile concept or dead end?”, Multiple Classifier Systems, (2013), p. 37–48.

24. Bi, Y., “The impact of diversity on the accuracy of evidential classifier ensembles”, International Journal of Approximate Reasoning, vol. 53, no. 4, (2012), p. 584–607.

25. Brown, G. and Kuncheva, L. I., “Good and Bad diversity in majority vote ensembles”, Multiple Classifier Systems, (2010), p. 124–133.

26. Kuncheva, L. I. and Whitaker, C. J., “Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy”, Machine Learning, vol. 51, no. 2, (2003), p. 181–207.

27. García-Pedrajas, N., García-Osorio, C. and Fyfe, C., “Nonlinear boosting projections for ensemble construction”, Journal of Machine Learning Research, vol. 8, (2007), p. 1–33.

28. Ulaş, A., Semerci, M., Yıldız, O. T. and Alpaydın, E., “Incremental construction of classifier and discriminant ensembles”, Information Sciences, vol. 179, no. 9, (2009), p. 1298–1318.

29. Rokach, L., “Collective-agreement-based pruning of ensembles”, Computational Statistics and Data Analysis, vol. 53, no. 4, (2009), p. 1015–1026.

30. Zanda, M., “A probabilistic perspective on ensemble diversity”, Ph.D. thesis, Manchester University, (2010).

31. Zhang, Y., Burer, S. and Street, N., “Ensemble pruning via semi-definite programming”, Journal of Machine Learning Research, vol. 7, (2006), p. 1315–1338.

32. García-Pedrajas, N. and Ortiz-Boyer, D., “Boosting random subspace method”, Neural Networks, vol. 21, no. 9, (2008), p. 1344–1362.

33. Rokach, L., “Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated bibliography”, Computational Statistics and Data Analysis, vol. 53, (2009), p. 4046–4072.

34. Zhou, Z.-H., Wu, J.-X., Jiang, Y. and Chen, S.-F., “Genetic algorithm based selective neural network ensemble”, International Joint Conference on Artificial Intelligence, (2001), p. 797–802.

35. Partalas, I., Tsoumakas, G. and Vlahavas, I., "Pruning an ensemble of classifiers via reinforcement learning", Neurocomputing, vol. 72, no. 7-9, (2008), p. 1900–1909.

36. Margineantu, D. D. and Dietterich, T. G., "Pruning adaptive boosting", International Conference on Machine Learning, (1997), p. 211–218.

37. Martínez-Muñoz, G. and Suárez, A., "Pruning in ordered bagging ensembles", International Conference in Machine Learning, (2006), p. 609–616.

38. Lu, Z., Wu, X., Zhu, X. and Bongard, J., "Ensemble pruning via individual contribution ordering", International Conference on Knowledge Discovery and Data Mining, (2010), p. 871–880.

39. Osborne, M. J. and Rubinstein, A., A Course in Game Theory, MIT Press, Cambridge, (1994).

40. Gillies, D. B., "Solutions to general non-zero-sum games", Contributions to the Theory of Games, vol. 4, (1959), p. 47–85.

41. Shapley, L. S., "A value for n-person games", Annals of Mathematical Studies, vol. 2, (1953), p. 307–317.

42. Banzhaf, J. F., "Weighted voting doesn't work: A mathematical analysis", Rutgers Law Review, vol. 19, no. 2, (1965), p. 317–343.

43. Cohen, S., Ruppin, E. and Dror, G., "Feature selection based on the shapley value", International Joint Conference on Artificial Intelligence, (2005), p. 665–670.

44. Ykhlef, H., Bouchaffra, D. and Ykhlef, F., "Coalitional game-based adaboost", IEEE International Conference on Systems, Man and Cybernetics, (2014), p. 194–199.

45. Ykhlef, H. and Bouchaffra, D., "Induced subgraph game for ensemble selection", IEEE International Conference on Tools with Artificial Intelligence, (2015), p. 636–643.

46. Ykhlef, H. and Bouchaffra, D., "An efficient ensemble pruning approach based on simple coalitional games", Information Fusion, vol. 34, (2017), p. 28–42.

47. Ykhlef, H. and Bouchaffra, D., “An induced subgraph game for ensemble selection”, International Journal on Artificial Intelligence Tools, vol. 26, no. 1, (2017), p. 1–20.

48. Bishop, C., Pattern recognition and machine learning, Springer-Verlag, New York, USA, (2006).

49. Kuncheva, L. I., Combining pattern classifiers: Methods and algorithms, J. Wiley, Hoboken, New Jersey, (2004).

50. Flach, P., Machine learning: The art and science of algorithms that make sense of data, Cambridge University Press, (2012).

51. Dietterich, T. G., “Approximate statistical tests for comparing supervised classification learning algorithms”, Neural Computation, vol. 10, no. 7, (1998), p. 1895–1923.

52. Demšar, J., “Statistical comparisons of classifiers over multiple data sets”, Journal of Machine Learning Research, vol. 7, (2006), p. 1–30.

53. García, S. and Herrera, F., “An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons”, Journal of Machine Learning Research, vol. 9, (2009), p. 2677–2694.

54. García, S., Fernández, A., Julián Luengo and Herrera, F., “Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power”, Information Sciences, vol. 180, (2010), p. 2044–2064.

55. Polikar, R., “Ensemble based systems in decision making”, IEEE Circuits and Systems Magazine, vol. 6, no. 3, (2006), p. 21–45.

56. Parikh, D. and Polikar, R., “An ensemble-based incremental learning approach to data fusion”, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 37, no. 2, (2007), p. 437–450.

57. Kotsiantis, S. B., Kanellopoulos, D. and Pintelas, P. E., “Local boosting of decision stumps for regression and classification problems”, Journal of Computers, vol. 1, no. 6, (2006), p. 30–37.

58. Li, X., Wang, L. and Sung, E., “AdaBoost with SVM-based component classifiers”, Engineering Applications of Artificial Intelligence, vol. 21, no. 5, (2008), p. 785–795.

59. Friedman, J., Hastie, T. and Tibshirani, R., “Additive logistic regression: A statistical view of boosting”, Annals of Statistics, vol. 28, no. 2, (2000), p. 337–407.

60. Andy Tsao, C.-H. and Chang, Y.-c. I., “A stochastic approximation view of boosting”, Computational Statistics and Data Analysis, vol. 52, no. 1, (2007), p. 325–344.

61. Islam, M. M., Yao, X. and Murase, K., “A constructive algorithm for training cooperative neural network ensembles.”, IEEE Transactions on Neural Networks, vol. 14, no. 4, (2003), p. 820–834.

62. Drucker, H., “Effect of pruning and early stopping on performance of a boosting ensemble”, Computational Statistics and Data Analysis, vol. 38, (2002), p. 393–406.

63. Sivalingam, D. M., Pandian, N. and Ben-Arie, J., “Minimal classification method with error-correcting codes for multiclass recognition”, International Journal of Pattern Recognition and Artificial Intelligence, vol. 19, no. 5, (2005), p. 663–680.

64. Langdon, W. B., Barrett, S. J. and Buxton, B. F., “Combining decision trees and neural networks for drug discovery”, Genetic Programming, Proceedings of the European Conference, (2002), p. 60–70.

65. Brown, G., “An information theoretic perspective on multiple classifier systems”, Multiple Classifier Systems, (2009), p. 344–353.

66. Gwet, K. L., Handbook of inter-rater reliability: The definitive guide to measuring the extent of agreement among raters, Advanced Analytics, LLC, Gaithersburg, Montgomery, (2014).

67. Cover, T. M. and Thomas, J. A., Elements of information theory, J. Wiley, Hoboken, New Jersey, 2nd edn., (2006).

68. Xu, L., Li, B. and Chen, E., “Ensemble pruning via constrained eigen-optimization”, IEEE International Conference on Data Mining, (2012), p. 715–724.

69. Opitz, D. and Maclin, R., “Popular ensemble methods: An empirical study”, Journal of Artificial Research, vol. 11, (1999), p. 169–198.

70. Martínez-Muñoz, G. and Suárez, A., “Using boosting to prune bagging ensembles”, Pattern Recognition Letters, vol. 28, no. 1, (2007), p. 156–165.

71. Caruana, R., Niculescu-Mizil, A., Crew, G. and Ksikes, A., “Ensemble selection from libraries of models”, International Conference on Machine Learning, (2004), p. 18.

72. Rokach, L., “Decomposition methodology for classification tasks -A meta decomposer framework”, Pattern Analysis and Applications, vol. 9, (2006), p. 257–271.

73. Arbela, R. and Rokach, L., “Classifier evaluation under limited resources”, Pattern Recognition Letters, vol. 27, no. 14, (2006), p. 1619–1631.

74. Quinlan, J. R., C4.5: programs for machine learning, Morgan Kaufmann Publishers, San Francisco, California, (1993).

75. Windeatt, T. and Ardeshir, G., “An Empirical comparison of pruning methods for ensemble classifiers”, Advances in Intelligent Data Analysis, (2001).

76. Tsoumakas, G., Partalas, I. and Vlahavas, I., “An ensemble pruning primer”, Applications of Supervised and Unsupervised Ensemble Methods, Springer, Berlin, Heidelberg, chap. 1, 1st edn., (2009), p. 1–13.

77. Martínez-Muñoz, G. and Suárez, A., “Aggregation ordering in bagging”, International Conference on Artificial Intelligence and Applications, (2004), p. 258–263.

78. Partalas, I., Tsoumakas, G. and Vlahavas, I., “Focused ensemble selection: A diversity based method for greedy ensemble selection”, European Conference on Artificial Intelligence, (2008), p. 117–121.

79. Zhou, Z.-H. and Tang, W., “Selective ensemble of decision trees”, Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, (2003).

80. Lazarevic, A. and Obradovic, Z., “Effective pruning of neural network classifier ensembles”, International Joint Conference on Neural Networks, (2001), p. 796–801.

81. Giacinto, G., Roli, F. and Fumera, G., “Design of effective multiple classifier systems by clustering of classifiers”, International Conference on Pattern Recognition, (2000), p. 160–163.

82. Bakke, B. and Heskes, T., “Clustering ensembles of neural network models”, Neural Networks, vol. 16, no. 2, (2003), p. 261–269.

83. Qiang, F., Shang-xu, H. and Sheng-ying, Z., “Clustering-based selective neural network ensemble”, Journal of Zhejiang University-SCIENCE A, vol. 6, no. 5, (2005), p. 387–392.

84. Tsoumakas, G., Angelis, L. and Vlahavas, I., “Selective fusion of heterogeneous classifiers”, Intelligent Data Analysis, vol. 9, no. 6, (2005), p. 511–525.

85. Leyton-Brown, K. and Shoham, Y., Essentials of game theory, Morgan & Claypool Publishers, San Rafael, California, (2008).

86. Fudenberg, D. and Tirole, J., Game theory, MIT Press, Cambridge, MA, (1991).

87. Peleg, B., “Coalitional TU games and solutions”, Introduction to the Theory of Cooperative Games, Springer, Berlin, chap. 2, (2007), p. 9–26.

88. Chalkiadakis, G., Elkind, E. and Wooldridge, M., Computational aspects of cooperative game theory, Morgan & Claypool Publishers, San Rafael, California, (2011).

89. Ieong, S. and Shoham, Y., “Marginal contribution nets: A compact representation scheme for coalitional games”, ACM Conference on Electronic Commerce, (2005), p. 193–202.

90. Kalai, E. and Zemel, E., “Totally balanced games and games of flow”, Mathematics of Operations Research, vol. 7, no. 3, (1982), p. 476–478.

91. Deng, X. and Papadimitriou, C. H., “On the complexity of cooperative solution concepts”, Mathematics of Operations Research, vol. 19, no. 2, (1994), p. 257–266.

92. Conitzer, V. and Sandholm, T., “Complexity of constructing solutions in the core based on synergies among coalitions”, Artificial Intelligence, vol. 170, no. 6-7, (2006), p. 607–619.

93. Brams, S. J. and Affuso, P. J., “Power and size: A new paradox”, Theory and Decision, vol. 7, no. 1, (1976), p. 29–56.

94. Algaba, E., Bilbao, J. M., García, J. R. F. and López, J. J., “Computing power indices in weighted multiple majority games”, Mathematical Social Sciences, vol. 46, no. 1, (2003), p. 63–80.

95. Bolus, S., “Power indices of simple games and vector-weighted majority games by means of binary decision diagrams”, European Journal of Operational Research, vol. 210, no. 2, (2011), p. 258–272.

96. Aadithya, K. V., Michalak, T. P. and Jennings, N. R., “Representation of coalitional games with algebraic decision diagrams”, International Conference on Autonomous Agents and Multiagent Systems, (2011), p. 1121–1122.

97. Sakurai, Y., Ueda, S., Iwasaki, A., Minato, S.-I. and Yokoo, M., “Compact representation scheme of coalitional games based on multi-terminal zero-suppressed binary decision diagrams”, Agents in Principle, Agents in Practice, (2011), p. 4–18.

98. Uno, T., “Efficient computation of power indices for weighted majority games”, Tech. rep., National Institute of Informatics, Tokyo, (2003).

99. Matsui, T. and Matsui, Y., “A survey of algorithms for calculating power indices of weighted majority games”, Journal of the Operations Research Society of Japan, vol. 43, no. 1, (2000), p. 71–86.

100. Weibull, J. W., Evolutionary game theory, MIT Press, Cambridge, MA, (1997).

101. Hofbauer, J., Evolutionary games and population dynamics, Cambridge University Press, Cambridge, MA, (1998).

102. Hao, Y., “Computation and analysis of evolutionary game dynamics”, Ph.D. thesis, Iowa State University, (2013).

103. Saad, W., Han, Z., Debbah, M., Hjorungnes, A. and Basar, T., “Coalitional game theory for communication networks”, IEEE Signal Processing Magazine, vol. 26, no. 5, (2009), p. 77–97.

104. Narayanam, R. and Narahari, Y., “Determining the top-k nodes in social networks using the Shapley value”, International Joint Conference on Autonomous Agents and Multi-Agent Systems, (2008), p. 1509–1512.

105. Contreras, J., Klusch, M. and Yen, J., “Multi-agent coalition formation in power transmission planning: A bilateral Shapley value approach”, International Conference on Artificial Intelligence in Planning Systems, (1998), p. 19–26.

106. Saad, W., Han, Z. and Poor, H. V., “Coalitional game theory for cooperative micro-grid distribution networks”, IEEE International Conference on Communications Workshops, (2011), p. 1–5.

107. Dimitriadis, S. I., Laskaris, N. A., Tsirka, V., Vourkas, M., Micheloyannis, S. and Fotopoulos, S., “Tracking brain dynamics via time-dependent network analysis”, Journal of Neuroscience Methods, vol. 193, no. 1, (2010), p. 145–155.

108. Bulò, S. R. and Pelillo, M., “A game-theoretic approach to hypergraph clustering”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 6, (2013), p. 1312–1327.

109. Sun, X., Liu, Y., Li, J., Zhu, J., Chen, H. and Liu, X., “Feature evaluation and selection with cooperative game theory”, Pattern Recognition, vol. 45, no. 8, (2012), p. 2992–3002.

110. Fragnelli, V. and Moretti, S., “A game theoretical approach to the classification problem in gene expression data analysis”, Computers & Mathematics with Applications, vol. 55, no. 5, (2008), p. 950–959.

111. Garg, V. K., Narahari, Y. and Narasimha Murty, M., “Novel Biobjective Clustering (BiGC) Based on Cooperative Game Theory”, IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 5, (2013), p. 1070–1082.

112. Peng, H., Long, F. and Ding, C., “Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 8, (2005), p. 1226–1238.

113. Riker, W. H., “The theory of political coalitions”, Midwest Journal of Political Science, vol. 7, no. 3, (1962), p. 295–297.

114. Bache, K. and Lichman, M., “UCI Machine Learning Repository”, (2017). URL http://archive.ics.uci.edu/ml

115. Brown, G., Pocock, A., Zhao, M.-J. and Luján, M., “Conditional likelihood maximisation: A unifying framework for information theoretic feature selection”, Journal of Machine Learning Research, vol. 13, (2012), p. 27–66.

116. Duin, R. P., Juszczak, P., Paclik, P., Pekalska, E., de Ridder, D., Tax, D. M. and Verzakov, S., “PRTools 4.1: A matlab toolbox for pattern recognition”, Tech. rep., Delft University of Technology, Delft, (2007).

117. Chang, C.-C. and Lin, C.-J., “LIBSVM : a library for support vector machines”, ACM Transactions on Intelligent Systems and Technology, vol. 2, no. 3, (2011).

118. Yamashita, M., Fujisawa, K., Fukuda, M., Kobayashi, K., Nakta, K. and Nakata, M., “Latest developments in the SDPA Family for solving large-scale SDPs”, Handbook on Semidefinite, Cone and Polynomial Optimization: Theory, Algorithms, Software and Applications, Springer, New York, USA, chap. 24, (2011), p. 687–714.