Ph.D. in Electronic and Computer Engineering
Dept. of Electrical and Electronic Engineering
University of Cagliari

The Role of Diversity in the Design of Multiple Classifier Systems

Muhammad Atta Othman Ahmed

Supervisor: **Prof. Luca Didaci**
Co-Supervisor: **Prof. Fabio Roli**
Curriculum: **ING-INF/05**
Sistemi di Elaborazione delle Informazioni

XXX Cycle
November 2017

Ph.D. in Electronic and Computer Engineering
Dept. of Electrical and Electronic Engineering
University of Cagliari

The Role of Diversity in the Design of Multiple Classifier Systems

Muhammad Atta Othman Ahmed

Supervisor: **Prof. Luca Didaci**
Co-Supervisor: **Prof. Fabio Roli**
Curriculum: **ING-INF/05**
Sistemi di Elaborazione delle Informazioni

XXX Cycle
November 2017

Dedicated to my near and far family **Father, Mother, Brothers and Sisters**, who are to me as my soul.

Dedicated to my wife **Dr.Reham** and my daughter **Rose**, who have been supporting me all the time.

Dedicated to my Professors, Teacher’s and Mentors. To every human taught me a letter of knowledge.

Dedicated to every human believe in science as a path for peace and love, for a better world.

Acknowledgement

I express my indebtedness and gratefulness to my supervisor **Prof. Fabio Roli**[1] for his continuous encouragement and hands of aid. I needed his support, guidance and encouragement throughout the PhD research period. I am obliged to him for his moral support through all the stages during this research work. I am indebted to him for the valuable time he has spared for me during this work.

This dissertation would not have been possible without the constant encouragement I have received from my supervisor **Prof. Giorgio Fumera**[2], he has constantly encouraged me to remain focused on achieving the goal. His observations and comments helped me to establish the overall direction of the research and to move forward with investigation in depth. No words can express my gratitude to one who has been a guru in the truest sense.

I would like to express my deepest thanks to **Prof. Luca Didaci**[3] for successful guidance and he was never too busy to listen me and offer his advice whenever I need either as an academic supervisor or even a true friend. Without his invaluable comments and suggestions, this work wouldn't have been accomplished. By learning from him, I feel I have become a better academic.

Finally but not lastly I'd like to express a true gratefulness to **PRALab**[4] members, who taught me how to be a team member inside and outside academic life.

[1] http://pralab.diee.unica.it/en/FabioRoli
[2] http://pralab.diee.unica.it/en/GiorgioFumera
[3] http://pralab.diee.unica.it/en/LucaDidaci
[4] http://pralab.diee.unica.it/en

Notions, Notations and Abbreviations

Summary of notions, notations and abbreviations used in this thesis:

- $\Omega = \{w_1, w_2, \ldots, w_c\}$ - the set of class labels.
- c- The number of class labels.
- $\Re^n$ - The feature space.
- $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ or $\mathbf{x} \in \Re^n$ - an object $\mathbf{x}$ composed of n descriptive features.
- y_i - the class label of x_i.
- $\mathbf{Z} = \left\{\vec{z}_1, \vec{z}_2, \ldots, \vec{z}_N\right\}, \vec{z}_j \in \Re^n$ - the training set.
- $\mathcal{Z} = \left\{(\vec{x}_1, y_1), (\vec{x}_2, y_2), \ldots, (\vec{x}_N, y_N)\right\}, \vec{x}_j \in \Re^n$ - A labelled training set.
- $D : \Re^n \to \Omega \quad \forall \mathbf{x} \in \Re^n s.t. D(\mathbf{x}) \in \Omega$ - A classifier D.
- $\mathscr{D} = \{\mathscr{D}_1, \mathscr{D}_2, \ldots . \mathscr{D}_L\}$ - A set of classifiers.
- $D_i(\mathbf{x}) = [d_{i,1}(\mathbf{x}), d_{i,2}(\mathbf{x}), \ldots, d_{i,c}(\mathbf{x})]^T$ - A classifier Outputs.
- $\mathscr{D}_i(\mathbf{x}) = [\mu_1(\mathbf{x}), \mu_2(\mathbf{x}), \ldots, \mu_c(\mathbf{x})]^T$ - Combined classifiers c, outputs for an object $\mathbf{x}$.

Diversity measures abbreviations:

- Q - The Q statistic.
- ρ - The correlation coefficient.
- D - Disagreement measure.
- DF - The double-fault measure.
- KW - The Kohavi-Wolpert measure.
- κ - The interrater agreement measure.
- Ent - The entropy measure.
- θ - the difficulty measure.
- GD - The generalized diversity.
- CFD - The coincidence failure diversity.

Multiple classifiers systems abbreviations:

- MCS - Multiple Classifiers Systems.
- E - Ensemble of classifiers.
- MVR - Majority Voting Rule.

- F/BS - Foreword/Backward Selection Ensemble Pruning.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Thesis Abstract

Multiple Classifiers Systems (MCS) perform information fusion of classification decisions at different levels overcoming limitations of traditional approaches based on single classifiers. We address one of the main open issues about the use of *Diversity* in Multiple Classifier Systems: the effectiveness of the explicit use of diversity measures for creation of classifier ensembles. So far, diversity measures have been mostly used for ensemble pruning, namely, for selecting a subset of classifiers out of an original, larger ensemble. Here we focus on pruning techniques based on forward selection, since they allow a direct comparison with the simple estimation of accuracy of classifier ensemble. We empirically carry out this comparison for several diversity measures and benchmark data sets, using bagging as the ensemble construction technique, and majority voting as the fusion rule. Our results provide further and more direct evidence to previous observations against the effectiveness of the use of diversity measures for ensemble pruning, but also show that, combined with ensemble accuracy estimated on a validation set, diversity can have a *regularization* effect when the validation set size is small. Whereas several existing pruning methods use some combination of individual classifiers accuracy and diversity, it is still unclear whether such an evaluation function is better than the bare estimate of ensemble accuracy. We empirically investigate this issue by comparing two evaluation functions in the context of ensemble pruning: the estimate of ensemble accuracy, and its linear combination with several well-known diversity measures. This can also be viewed as using diversity as a *regularizer*, as suggested by some authors. To this aim we use a pruning method based on forward selection, since it allows a direct comparison between different evaluation functions. Experiments on thirty-seven benchmark data sets, four diversity measures and three base classifiers provide evidence that using diversity measures for ensemble pruning can be advantageous over using only ensemble accuracy, and that diversity measures can act as regularizers in this context. Focusing on ensemble creation technique well-known as Bagging, the computational power and demand of Neural Networks (NNs) approved in both researches or in applications. The weight connections of the NNs holds the real ability for the NNs model to efficient performance. We aim to analyze the weight connections of the trained ensemble of NNs, as well as investigating their statistical parametric distributions, we present a framework to estimate the best-fit statistical distribution from a list of well-known statistical parametric distribu-

tions. This work is the first attempt in the state-of-art to explore and analyze the weights of a trained ensemble of 1000 neural networks. Consequently we aim in our future work to employ the outcomes to withdraw the weight connections value from approximated best-fit distribution instead of training the ensemble of NN classifiers from scratch.

1.2 Problem Statement

Several studies have shown that ensemble learning can outperform the single classifier approach [1, 2, 3, 4]. The rationale behind this methodology is that by combining different and accurate models, we may improve the ensemble decision over each single classifier decision. The *Diversity* plays the main rule in the success of ensemble learning techniques. Despite many attempts in state-of-art [5, 6, 7, 8, 9, 10], till now there is still no agreement on how to measure, define or even manage diversity for classifier ensembles. This lack of a unifying approach differs from the parallel field of regression ensembles, where diversity is a well known problem. There is no clear definition of Good and Bad diversity measure [11, 12, 13, 14, 15]. Ensemble pruning has exponential complexity in the size of the original ensemble [16, 17, 18, 1]. Discussing and investigating the ensemble performance through the trade-off between the ensemble accuracy and diversity measures is a matter of first-order importance, as it would provide us with a clear understanding of the conditions and circumstances under which an ensemble succeeds over a single classifier approach and overall might result in building more efficient ensembles. Considering the high computational cost of ensembles, many attempts aimed to analyse those concerned rules of ensemble constructions aiming to reduce the complexity of constructing efficient ensemble [19, 20, 21, 22]. There is no clear predefined method of how to reduce the ensemble creation computational cost.

1.3 Thesis Objective

The main aims of thesis is:

- To present a comprehensive study of the diversity measures and the relationship between them.
- To compare the effectiveness of explicitly using existing diversity measures in ensemble pruning, against the direct estimation of ensemble performance.
- To investigate the possible methods to improve the diversity measures use with ensemble accuracy when combined together.
- To investigate the Diversity measures behaviour as a regularizer in the ensemble pruning.
- To explore the trained ensembles of Neural Network weight connections distributions, what is the first step to reduce their computational cost.

1.4 Thesis Outlines

- **Chapter 2:** Presents a literature review on the concepts of pattern recognition, a description of classifiers and ensembles highlighting their creation techniques and characteristics.
- **Chapter 3:** Introduce the diversity measures, relations between them and their use in ensemble creation.
- **Chapter 4:** Presents a study of neural networks ensemble, statistical analysis of their weight connections.
- **Chapter 5:** Conclusion and summary of thesis highlighting limitations and possible future considerations.

Chapter 2

Literature Review on Multiple Classifier Systems

Machine Learning (ML) is a branch of computer science that enables machines to act without being explicitly programmed and make them deliver faster, more accurate results in order to identify profitable outcomes or dangerous risks. Combining machine learning with AI and cognitive technologies can make it even more effective in processing large volumes of information [23]. **Pattern recognition** is a branch of machine learning that concentrate on the recognition process of patterns and similarities/variations in data, although it is in some cases considered to be nearly similar with machine learning [24]. Pattern recognition systems are in most cases trained/learned from labeled data, what is known as *supervised learning* strategy. When there is no available labeled data other algorithms/strategies can be used to discover unknown patterns. This kind of strategies are known as *unsupervised learning* strategy. Pattern recognition algorithms generally aim to find a reasonable answer for all possible inputs and to perform matching of the inputs, considering their statistical characteristics such as similarities and variations [25].
Using pattern recognition to solve any related problem includes some basic definitions such as:

- **A pattern object:** composed as a vector of n descriptive features or attributes, $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T, \mathbf{x} \in \Re^n$.
- The set of all **classes labels**, denoted as: $\Omega = \{w_1, w_2, \ldots, w_c\}$.
- **Features** of an object are its descriptive characteristics in a numerical form.
- **Feature Space** consists of all possible values of features generally denoted as $\Re^n$.
- **Training Se** a set of objects described via numerical features denoted as $\mathbf{Z} = \{\vec{z}_1, \vec{z}_2, \ldots, \vec{z}_N\}, \vec{z}_j \in \Re^n$.

2.1 The Classifier

The Classifier: An algorithm that implements classification, sometimes also refers to the mathematical function implemented by a classification algorithm, that maps input data into a certain categories. A classifier is any mapping D which assign a class label to an object $\mathbf{x}$, i.e.,

$$D : \Re^n \rightarrow \Omega \quad \forall \mathbf{x} \in \Re^n, D(\mathbf{x}) \in \Omega \tag{2.1}$$

The classifier outcome for an object $\mathbf{x}$ suppose to be the corresponding class label Ω for the object $\mathbf{x}$. Classifiers can be designed in different ways, so they vary in the accuracy of correctly classifying an object $\mathbf{x}$ to its belonging class. A clear definition of classifier $Accuracy$ on a set $\mathbf{Z}$ of N objects;

$$Accuracy = \frac{N_C}{N}. \tag{2.2}$$

Where N_C is the number of objects correctly classified by the classifier D.

2.2 Bayes Decision

Assuming that ω is the class label taking values $\Omega = \{\omega_1, \omega_2, \ldots, \omega_c\}$ the posterior probability for an object $\mathbf{x}$ to be classified to class ω_i using the Bayes formula:

$$P(\omega_i|\mathbf{x}) = \frac{p(\omega_i)P(\mathbf{x}|\omega_i)}{p(\mathbf{x})} = \frac{p(\omega_i)P(\mathbf{x}|\omega_i)}{\sum_{j=1}^{c} p(\omega_j)P(\mathbf{x}|w_j)}. \tag{2.3}$$

Equation 2.3 defines the propability mass function regarding the object $\mathbf{x}$ to be classified tothe class ω. The class with highest posterior probability is chosen tobethe correct class for a given object $\mathbf{x}$ what what achieves the smallest possible error.

2.3 Parametric Classifiers

Parametric classifier rely on approximating the *parameters* of the probabilistic density function $P(x|w_i)$ of the class conditional problem. from which we obtain the posterior probability shown in equation 2.3 [26].

2.4 Linear Classifiers

A great number of methods for classification can be formed in terms of a linear function that assigns a score to each possible category k by combining the features which is represented as a vector of an object with a vector of weights, using the dot product. The predicted category is the one with the highest score [27]. This type of score function is known as a linear predictor function and has the following general form:

$$\mathbf{score}(\mathbf{X}_i, k) = \boldsymbol{\beta}_k \cdot \mathbf{X}_i, \tag{2.4}$$

where X_i is the feature vector for instance i, β_k is the vector of weights corresponding to category k, and $\mathbf{score}(\mathbf{X}_i, k)$ is the score associated with assigning instance i to category k. In discrete choice theory, where instances represent people and categories represent choices,

the score is considered the utility associated with person i choosing category k. Algorithms with this basic setup are known as linear classifiers. What distinguishes them is the procedure for determining (training) the optimal weights/coefficients and the way that the score is interpreted.
Examples of such algorithms are:

- The Perceptron Algorithm [28].
- Support Vector Machines [29].
- Linear Discriminant Analysis [30, 31].

2.4.1 Linear Discriminant Classifier

The Linear Discriminant Classifier (LDC) is one of the simplest forms of classifiers. LDC is the preferable linear classifier because of it's simplicity and good performance. Given a pattern $\mathbf{x} \in \mathbb{R}^n$ aimed to be classified to a class c. Considering a vector of coefficient parameters $w_i \in \mathbb{R}^n$ with a $w_i 0$ free term, a linear classification function per class $g_i(\mathbf{x})$ with the highest tag defines the class label where:

$$g_i(\mathbf{x}) = w_{i0} + \mathbf{W}_i^T \mathbf{x}. \tag{2.5}$$

2.4.2 Nearest Mean Classifier

Nearest Mean Classifier (NMC) simply varies from the **LDC** where it classifies a given pattern $\mathbf{x}$ to its nearest mean. **NMC** discriminant functions is the negative square of the Euclidean distance to the class means where:

$$g_i(x) = -(\mu_i - x)^T(\mu_i - x). \tag{2.6}$$

$$= -\mu_i^T \mu_i + 2\mu_i^T x - x^T x. \tag{2.7}$$

Dropping the term $-x^T x$ because it does not depend on the class label, the **NMC** classifier discriminant function can be rewritten as a linear form on $\mathbf{x}$ as follows:

$$g_i(x) = -\mu_i^T \mu_i + 2\mu_i^T x = w_{i0} + w_i^T x. \tag{2.8}$$

If the covariance matrices for all classes are the identity matrices this classifier defined in equation 2.8 is identical to the LDC.

2.5 Quadratic Discriminant Classifier

The Quadratic Discriminant Classifier (QDC) is defined by a quadratic discriminant function as follows:

$$g_i(x) = w_{i0} + W_i^T x + x^T W_i x, \qquad where \quad x, W_i \in \mathbb{R}^n \tag{2.9}$$

And W_i is an $[n * m]$ matrix.

2.6 Decision Tree

Decision Tree Classifier is a simple and widely used classification technique. It applies a strait forward idea to solve the classification problem. Decision Tree Classifier poses a series of carefully crafted questions about the attributes of the test record. Every-time time it receive a feed-back, a follow-up question is asked until a conclusion about the class label of the record is reached. A Decision Tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility, it is a one way to display an algorithm [32]. A Decision Tree is a flowchart-like structure as shown in figure 2.1, in which each inner node represents an operation of testing on an attribute. E.g. whether a coin flip comes up heads or tails for example, each branch represents the output of the test, and each leaf node represents the corresponding class label, where the decision is taken after computing all attributes at hand. The paths from root to leaf represent classification rules. In decision analysis, a Decision Tree and the closely related influence diagram are used as a visual and analytical decision support tool, where the expected values or expected utility of competing alternatives are computed. A Decision Tree have three types of nodes; Decision nodes, Chance nodes and End nodes.
Decision Trees are commonly used in operations research and operations management. If,

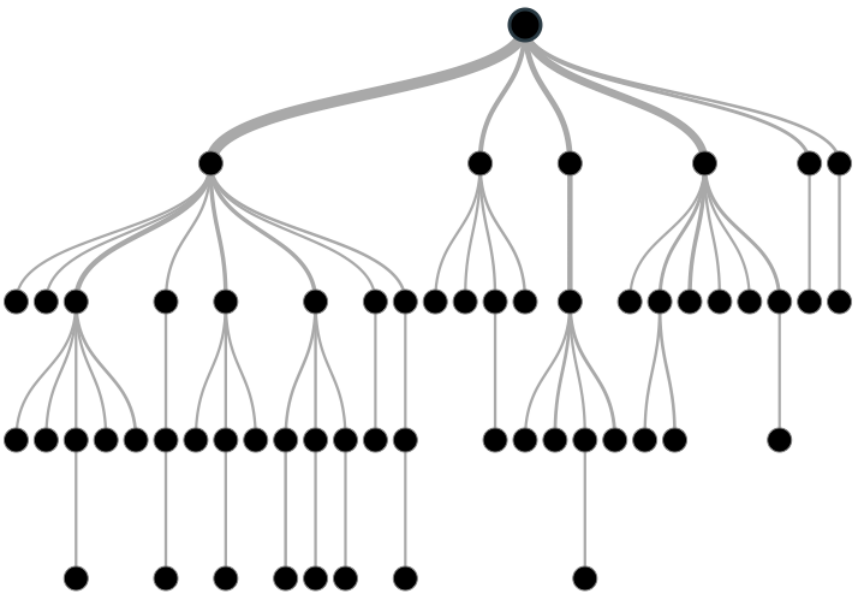


Figure 2.1: The Decision Tree classifier general diagram.

in practice, decisions have to be taken online with no recall under incomplete knowledge, a Decision Tree should be paralleled by a probability model as a best choice model or online selection model algorithm [33]. Another use of Decision Trees is as a descriptive means for calculating conditional probabilities.
Types of Decision Tree is based on the type of target variable we have. It can be of two types:

- **Categorical Variable Decision Tree:** Decision Tree which has categorical target variable then it called as categorical variable Decision Tree. Example:- In above scenario

of student problem, where the target variable was 'Student will play cricket or not' i.e. YES or NO.

- **Continuous Variable Decision Tree:** Decision Tree has continuous target variable then it is called as Continuous Variable Decision Tree.

Important Terminology related to Decision Trees [34]:

- **Root Node:** It represents entire population or sample and this further gets divided into two or more homogeneous sets.
- **Splitting:** It is a process of dividing a node into two or more sub-nodes.
- **Decision Node:** When a sub-node splits into further sub-nodes, then it is called decision node.
- **Leaf / Terminal Node:** Nodes do not split is called Leaf or Terminal node.
- **Pruning:** When we remove sub-nodes of a decision node, this process is called pruning. You can say opposite process of splitting.
- **Branch / Sub-Tree:** A sub section of entire tree is called branch or sub-tree.
- **Parent and Child Node:** A node, which is divided into sub-nodes is called parent node of sub-nodes where as sub-nodes are the child of parent node.

Advantages [35]:

- **Easy to Understand:** Decision Tree output is very easy to understand even for people from non-analytical background. It does not require any statistical knowledge to read and interpret them. Its graphical representation is very intuitive and users can easily relate their hypothesis.
- **Useful in Data Exploration:** Decision Tree is one of the fastest way to identify most significant variables and relation between two or more variables. With the help of Decision Trees, we can create new variables / features that has better power to predict target variable.
- **Less Data Cleaning Required:** It requires less data cleaning compared to some other modeling techniques. It is not influenced by outliers and missing values to a fair degree.
- **Data Type is not a Constraint:** It can handle both numerical and categorical variables.
- **Non Parametric Method:** Decision Tree is considered to be a non-parametric method. Meaning that Decision Trees have no assumptions regarding the space distribution and the classifier inner structure.

Disadvantages [36]:

- **Over Fitting:** Over fitting is one of the most practical difficulty for Decision Tree models. This problem gets solved by setting constraints on model parameters and pruning criteria.

- **Continuous Variables:** Decision Tree is not fit for continuous variables because it looses information when it categorizes variables in different classes.

2.6.1 Applications of Decision Tree:

- **Agriculture:** Application of a range of machine learning methods (mainly Decision Trees) to problems in agriculture and horticulture is described in [37].
- **Astronomy:** Use of Decision Trees for filtering noise from Hubble Space Telescope images was reported recently in [38]. Decision Trees have helped in star-galaxy classification [39], determining galaxy counts [40] and discovering quasars [41] in the Second Palomar Sky Survey. Use of neural trees for ultraviolet stellar spectral classification is described in [41].
- **Biomedical Engineering:** Use of Decision Trees for identifying features to be used in implantable devices are presented in [42].
- **Control Systems:** Automatic induction of Decision Trees was recently used for control of nonlinear dynamical systems [43].
- **Financial Analysis:** Use of Classification and Regression Tree (CART) [34] for asserting the attractiveness of buy-writes is reported in [44].
- **Manufacturing and Production:** Decision Trees have been recently used to non-destructively test welding quality [45], for semiconductor manufacturing [46], for increasing productivity [47], for material procurement method selection [48], to accelerate rotogravure printing [49], for process optimization in electrochemical machining [50], to schedule printed circuit board assembly lines [51], to uncover flaws in a Boeing manufacturing process [52] and for quality control [53]. For a recent review of the use of machine learning (Decision Trees and other techniques) in scheduling [54].
- **Medicine:** Medical research and practice have long been important areas of application for Decision Tree techniques. Recent uses of automatic induction of Decision Trees can be found in diagnosis [55], cardiology [56, 57, 58], psychiatry [59], gastroenterology [60], for detecting microcalcifications in mammography [10], to analyze Sudden Infant Death (SID) syndrome and for diagnosing thyroid disorders [61].
- **Molecular Biology:** Initiatives such as the Human Genome Project and the GenBank database offer fascinating opportunities for machine learning and other data exploration methods in molecular biology. Recent use of Decision Trees for analyzing amino acid sequences can be found in [62, 63].
- **Object Recognition:** Tree based classification has been used recently for recognizing three dimensional objects [64, 65] and for high level vision [66].
- **Pharmacology:** Use of tree based classification for drug analysis can be found in [67].
- **Physics:** Decision Trees have been used for the detection of physical particles [68].
- **Plant Diseases:** CART [34] was recently used to assess the hazard of mortality to pine trees [69].

- **Power Systems:** Power system security assessment [70] and power stability prediction [71] are two areas in power systems maintenance for which Decision Trees were used.
- **Remote Sensing:** Remote sensing has been a strong application area for pattern recognition work on Decision Trees [36, 72]. A recent use of tree-based classification in remote sensing can be found in [73].
- **Software Development:** Regression trees (and backpropagation networks) were recently used to estimate the development effort of a given software module in [74], where it is argued that machine learning methods compare favorably with traditional methods.
- **Text Processing:** A recent use of ID3 [75] for medical text classification can be found in [76].
- **Miscellaneous:** Decision Trees have also been used recently for building personal learning assistants [77] and for classifying sleep signals [78].

2.7 k-Nearest Neighbour

The k-nearest neighbours classifier (k-NN) is a non parametric kind of classification and regression algorithm [79]. It depends on a parameter k that represents the number of pattern neighbours to be considered in the decision of classification. The parameter k is usually preferred to be selected as an odd number to avoid the decision ambiguity, The output depends on whether k-NN is used for classification or regression:

- In classification kind of problems it's output is the class label of the pattern considering the pattern k-Neighbours.
- The average property value for the object of its k-Nearest Neighbours.

k-NN is a type of instance based learning, where the decision function is only estimated locally and all computation is deferred until classification is done, a common weighting scheme consists in giving each neighbour a weight of $\frac{1}{d}$, where d is the distance to the neighbour.

2.7.1 The 1-Nearest Neighbour Classifier

The prime k-NN classifier is the one presented in figure 2.2, it assigns an object x to the class of its nearest neighbour where:

$$C_n^{1nn}(x) = Y_{(1)}. \tag{2.10}$$

Given that the training set size almost infinite, the one nearest neighbour classifier ensures an error rate twice less than the **Bayes** error.

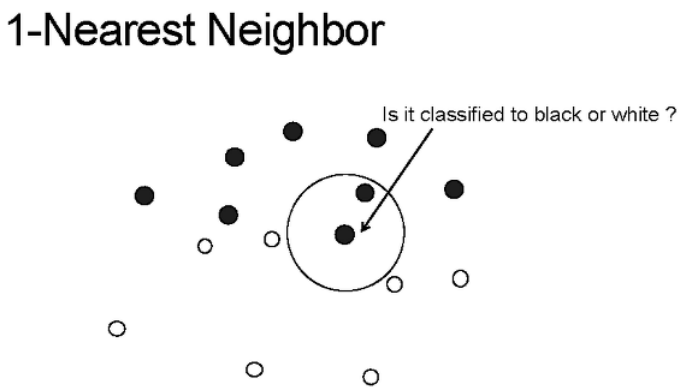


Figure 2.2: 1-Nearest Neighbour classification.

2.8 Artificial Neural Network

The idea of ANNs is based on the belief that working of human brain by making the right connections, can be imitated using silicon and wires as living **neurons** and **dendrites**. The human brain is composed of 100 billion nerve cells called neurons. They are connected to other thousand cells by **Axons**. Stimuli from external environment or inputs from sensory organs are accepted by dendrites [80]. These inputs generate electric spikes, which quickly travel through the connections of the neural network. A neuron can then send the message

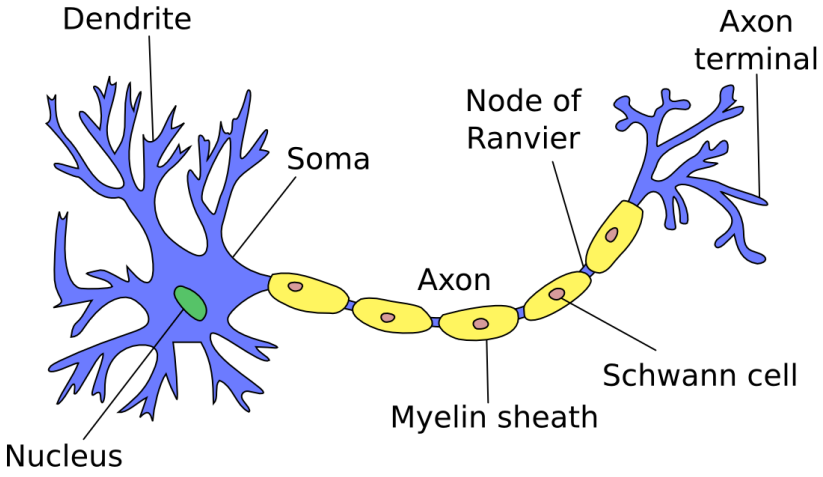


Figure 2.3: Biological neuron in a nervous system.

to other neuron to handle the issue or does not send it forward. Figure 2.3 describes the components of a biological neuron in a nervous system[1]. ANNs are composed of multiple nodes (see figure 2.4), which imitate biological neurons of human brain. The neurons are connected by links and they interact with each other. The nodes can take input data and perform simple operations on the data. The result of these operations is passed to other neurons. The output at each node is called its activation or node value [81]. Each link is associated with weight. ANNs are capable of learning, which takes place by altering weight

[1]Figure source: MAREK REI Thoughts on Machine Learning and Natural Language Processing.

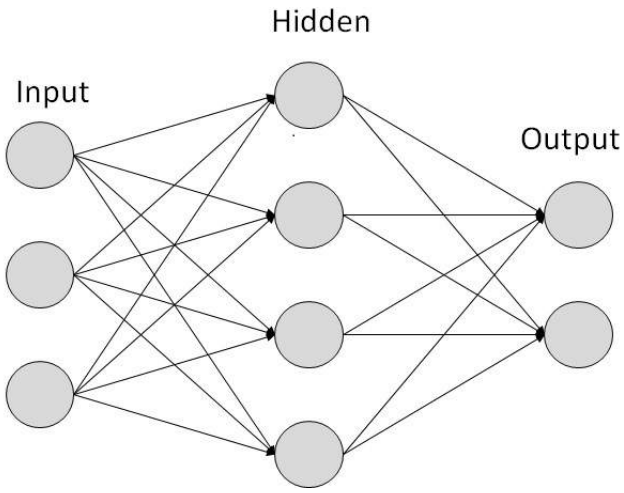


Figure 2.4: A simple ANN structure diagram.

values. Typically, neurons are organized in layers. Different layers may perform various types of transformations on their inputs. Signals travel from the first layer known as the input layer, to the last layer known as the output layer, possibly after traversing the layers multiple times. The original goal of the neural network approach was to solve problems in the same way that a human brain would. Over time, attention focused on matching specific mental abilities, leading to deviations from biology such as back-propagation, or passing information in the reverse direction and adjusting the network to reflect that information. Neural networks have been used on a variety of tasks, including computer vision, speech recognition, machine translation, social network filtering, playing board and video games, medical diagnosis and in many other domains.

2.8.1 History of ANNs

McCulloch *et. al.* [82] created a computational model for neural networks based on mathematics and algorithms called threshold logic. This model paved the way for neural network research to divide into two categories. An approach that is relevant to biological processes in the brain while the other focused on the application of neural networks to artificial intelligence. This mentioned study led to work on nerve neural networks and their link to finite automate. Widrow *et. al.* [83] developed a mathematical method for adapting the weights. Assuming that a desired response existed, a gradient search method was implemented, which was based on minimizing the error squared. This algorithm would later become known Least Mean Squares (LMS). LMS, and its variations, has been used extensively in a variety of applications, especially in the last few years. This gradient search method provided a mathematical method for finding an answer that minimized the error. The learning process was not a trial and error process. Although the computational time decreased with Selfridge's work, the LMS method decreased the amount of computational time even more, which made use of perceptrons feasible. The back propagation algorithm changes the schematic of the perceptron by using a sigmoidal function as the squashing function. Earlier versions of the perceptron used a signum function. The advantage of the sigmoidal function over the signum function is that the sigmoidal function is differentiable [84].

2.8.2 Components of an Artificial Neural Network

- **Connections and weights:** The network consists of connections, each connection transferring the output of a neuron i to the input of a neuron i. In this sense i is the predecessor of j and j is the successor of i. Each connection is assigned a weight w_{ij} [85].

- **Output function:** The propagation function computes the input p_j to the neuron j from the outputs o_i of predecessor neurons and typically has the form :

$$f_j = \sum_i o_i w_{ij}. \tag{2.11}$$

- **Activation Function:** A mathematical function defines the final output of the neurons [86]. There is various kinds of activation function (see figure 2.5):

  - The Threshold Activation function:

$$\Theta(\zeta) = \begin{cases} 1, & \zeta \geq 1 \\ 0, & otherwise. \end{cases} \tag{2.12}$$

  - The Sigmoid function:

$$\Theta(\zeta) = \frac{1}{1+\exp(-\zeta)} \tag{2.13}$$

  - The identity function

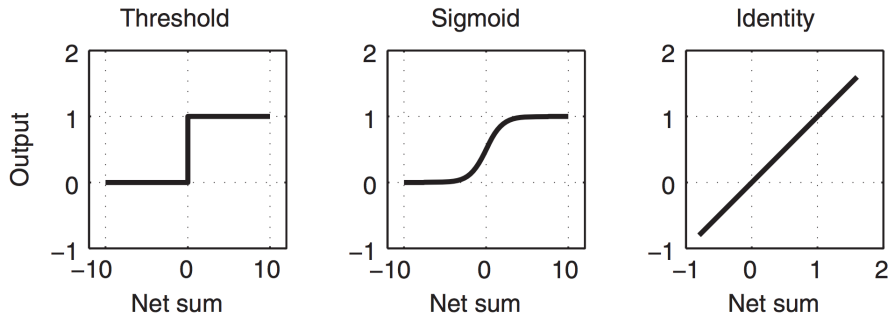


Figure 2.5: Three kinds of activation, Threshold, Sigmoid and Identity function.

- **Learning rule:** The learning rule is a rule or an algorithm which modifies the parameters of the neural network, in order for a given input to the network to produce a favoured output. This learning process typically amounts to modifying the weights and thresholds [85].

2.8.3 Applications of neural networks

They can perform tasks that are easy for a human but difficult for a machine such as:

- **Aerospace:** Autopilot air-crafts [87], aircraft fault detection [88].
- **Automotive:** Automobile guidance systems [89].
- **Military:** Weapon orientation and steering [90], target tracking [91], object discrimination, facial recognition, signal/image identification.
- **Electronics:** Code sequence prediction, IC chip layout, chip failure analysis, machine vision, voice synthesis [92].
- **Financial:** Real estate appraisal, loan advisor, mortgage screening, corporate bond rating, portfolio trading program, corporate financial analysis, currency value prediction, document readers, credit application evaluators [93].
- **Industrial:** Manufacturing process control, product design and analysis, quality inspection systems, welding quality analysis, paper quality prediction, chemical product design analysis, dynamic modeling of chemical process systems, machine maintenance analysis, project bidding, planning, and management [94].
- **Medical:** Cancer cell analysis, EEG and ECG analysis, prosthetic design, transplant time optimizer [95].
- **Speech:** Speech recognition, speech classification, text to speech conversion [96].
- **Telecommunications:** Image and data compression, automated information services, real-time spoken language translation [97].
- **Transportation:** Truck Brake system diagnosis, vehicle scheduling, routing systems[98].
- **Software:** Pattern Recognition in facial recognition, optical character recognition, etc[99].
- **Time Series Prediction:** ANNs are used to make predictions on stocks and natural calamities [100].
- **Signal Processing:** Neural networks can be trained to process an audio signal and filter it appropriately in the hearing aids [92].
- **Control:** ANNs are often used to make steering decisions of physical vehicles [101].
- **Anomaly Detection:** As ANNs are expert at recognizing patterns, they can also be trained to generate an output when something unusual occurs that misfits the pattern [102].

2.9 Classifiers Outputs Types

Given an ensemble L composed of D classifiers where: $D = \{D_1, D_2, \ldots, D_L\}$, with a set of classes $\Omega = \{\omega_1, \omega_2, \ldots, \omega_c\}$.
We can categorize the classifiers individual outputs as **four categories** [103, 86]:

- **Class labels:** Considered as the basic or universal level of classifier output representation. Each classifier D_i produce a class label $s_i \in \Omega, i = 1, 2, \ldots, L$; for any object $\mathbf{x} \in \mathbb{R}^n$.
- **Ranked class labels:** Each classifier D_i output is a subset of the class labels Ω, ranked in a plausibility order [104, 105]. This type is prefered to use with classification problems with many classes, such as biometrics , text recognition and classification.
- **Numerical support for the classes:** Each classifier D_i produces a c-dimensional vector $[d_{i,1}, d_{i,2}, \ldots, d_{i,c}]^T$.The value $d_{i,j}$ represents the support for the hypothesis that the vector $\mathbf{x}$ submitted for classification comes from class ω_j.The outputs are functions of the input $\mathbf{x}$, but to simplify the notation we will use just $d_{i,j}$ instead of $d_{i,j}(\mathbf{x})$.Without loss of generality, we can assume that the outputs contain values between 0 and 1, spanning the space $[0,1]^c$.
- **Oracle:** The output of classifier D_i for a given $\mathbf{x}$ is only known to be either correct or wrong. We deliberately disregard the information as to which class label has been assigned. The oracle output is artificial because we can only apply it to a labeled data set. For a given data set $\mathbf{Z}$, classifier D_i produces an output vector y_i such that:

$$y_{ij} = \begin{cases} 1, & If \quad \mathrm{D}_i \text{ classifies } \text{ object } \mathbf{z}_j \text{ correctly.} \\ 0, & Otherwise. \end{cases} \tag{2.14}$$

2.10 Pattern Recognition Applications

In medical science, pattern recognition is considered the basis for computer-aided diagnosis (CAD) systems. CAD describes a strategy that supports the doctor's decisions. Pattern Shape Recognition Technology (SRT) in a people counter system Other typical applications of pattern recognition techniques are automatic speech recognition, classification of text into several categories (e.g., spam/non-spam email messages), the automatic recognition of handwritten postal codes on postal envelopes, automatic recognition of images of human faces, or handwriting image extraction from medical forms [106]. The last two examples form the subtopic image analysis of pattern recognition that deals with digital images as input to pattern recognition systems [107] [108]. Optical character recognition is a classic example of the application of a pattern classifier [109]. The method of signing one's name was captured with stylus and overlay starting in 1990 [110]. The strokes, speed, relative min, relative max, acceleration and pressure is used to uniquely identify and confirm identity [111]. Banks were first offered this technology, but were content to collect from the FDIC[2] for any bank fraud and did not want to inconvenience customers.
Artificial neural networks (neural net classifiers) and deep learning have many real-world applications in image processing, a few examples:

[2]The Federal Deposit Insurance Corporation

- **Identification and authentication:** e.g., license plate recognition [112], fingerprint analysis and face detection/verification [113].
- **Medical diagnosis:** e.g., screening for cervical cancer (Papnet) [114] or breast tumours.
- **Defence:** various navigation and guidance systems, target recognition systems, shape recognition technology [115].

2.11 Combining Classifiers

Recently in the area of machine learning the concept of combining classifiers is proposed as a new direction for the improvement of the performance of individual classifiers. These classifiers could be based on a variety of classification methodologies, and could achieve different rate of correctly classified individuals. The goal of classification result integration algorithms is to generate more certain, precise and accurate system results. Dietterich (2001) provides an accessible and informal reasoning, from statistical, computational and representational viewpoints, of why ensembles can improve results[86]. Ensemble individual classifiers must be accurate and different from each other in order to efficiently contribute in the ensemble final decision [116]. Numerous methods have been suggested for the creation of ensemble of classifiers, for example:

- Using different subset of training data with a single learning method.
- Using different training parameters with a single training method (e.g. using different initial weights for each neural network in an ensemble).
- Using different learning methods.

2.11.1 Weakness of Classifier Ensembles

- **Increased Storage:** The first weakness, increased storage, is a direct consequence of the requirement that all component classifiers, instead of a single classifier, need to be stored after training. The total storage depends on the size of each component classifier itself and the size of the ensemble (number of classifiers in the ensemble) [117, 13].
- **Increased Computation:** The second weakness is increased computation: to classify an input query, all component classifiers (instead of a single classifier) must be processed, and thus it requires more execution time.
- **Decreased Comprehensibility:** The last weakness is decreased comprehensibility. With involvement of multiple classifiers in decision-making, it is more difficult for users to perceive the underlying reasoning process leading to a decision [86, 11].

2.11.2 Ensemble Size

While the number of component classifiers of an ensemble has a great impact on the accuracy of prediction, there is a limited number of studies addressing this problem. A priori determining of ensemble size and the volume and velocity of big data streams make this even

more crucial for online ensemble classifiers. Mostly statistical tests was used for determining the proper number of components [4, 118, 119]. More recently, a theoretical framework suggested that there is an ideal number of component classifiers for an ensemble which having more or less than this number of classifiers would deteriorate the accuracy. It is called "the law of diminishing returns in ensemble construction". Their theoretical framework shows that using the same number of independent component classifiers as class labels gives the highest accuracy [120].

2.11.3 Majority Voting Rule (MVR)

There are several theoretical and experimental analysis of how to combine the outputs of individual classifiers [121, 122, 123, 14, 124, 125, 6, 126, 127, 128, 129], also some reviews [130, 118]. Assuming a c-dimensional binary vectors classifiers outputs
$D_i = [d_{i,1}, d_{i,2}, \ldots, d_{i,c}]^T \in \{0,1\}^c, i = 1,2,\ldots,L,$ where $d_{i,j}$=1 if the classifier D_i classified the object **x** as it belongs to the class ω_j,and 0, otherwise. The class ω_k will be returned by *the plurality vote* if:

$$\sum_{i=1}^{L} d_{i,k} = \max_{j=1}^{c} \sum_{i=1}^{L} d_{i,j} \tag{2.15}$$

This rule is quite known as **Majority Voting Rure** (**MVR**)

Algorithm 1 Describes the Majority Voting rule as an ensemble decision combiner.

Training Phase: None.
Classification Phase: For each new object **x** Do:

1. Find the class labels $s_1, s_2, \ldots, s_L$, assigned to this object by the L base classifiers.
2. Calculate the number of votes for each class $\omega_k, k = 1,2,\ldots,c$.

$$P(k) = \sum_{i=1}^{L} I(s_i, \omega_k),$$

where $I(a,b)$=1 if $a = b$ and 0 otherwise.

3. Assign label k^*=arg $\max_{k=1}^{c} P(k)$.
Return the ensemble label of the object **x**.

2.12 Bagging Ensemble Creation Technique

Ensembles tend to yield better results when there is a significant diversity among the models [11, 131]. Many ensemble methods, therefore, seek to promote diversity among the models they combine. Although perhaps non-intuitive, [7] [132] more random algorithms (like random decision trees) can be utilized to obtain a stronger ensemble than very deliberate techniques such as entropy-reducing decision trees. [133] Using a variety of strong learning algorithms, however, has been shown to be more effective than using techniques that attempt to dumb-down the models in order to promote diversity [134]. Bagging is a method

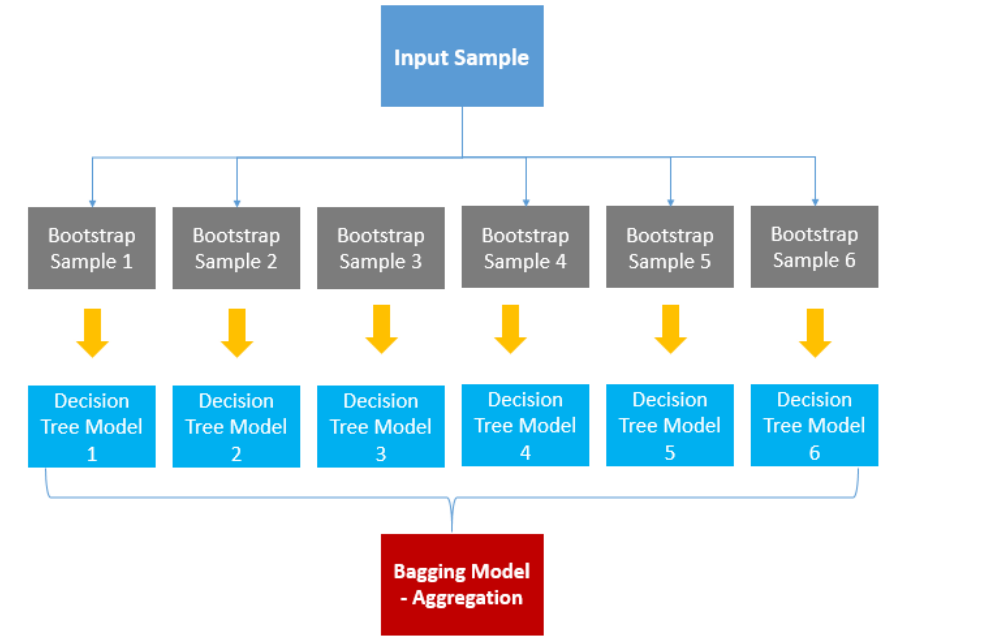


Figure 2.6: Describes the diagram of Bagging model aggregation for ensemble creation.

Algorithm 2 Describes the Bagging ensemble creation technique.

Training Phase:

1. Initialize the ensemble: $\mathscr{D} = \phi$.
2. $\mathscr{L}$, the number of classifiers to train. **For** k**=1,...,**$\mathscr{L}$ **Do:**
   a) Take a bootstrap sample S_k from original training set **Z**.
   b) Build a classifier D_k using S_k as the training set.
   c) Add the classifier to the current ensemble $D = D \bigcup D_k$.

Classification Phase

1. Run $D_1, \ldots, D_L$ on the input **x**.
2. The class with the maximum number of votes is chosen as the label for **x**.

of the first category (Breiman, 1996 [135]). Considering a training set of size t, it is possible to withdraw t random samples from the training set with replacement, these t samples can be learned, and this process can be repeated many times. Since the withdraw include replacement, usually the patterns drawn will include some duplicates and some omissions as compared to the original training set. Each iteration through the withdraw operation results in an individual classifier. After the building of several individual classifiers, taking a vote of the predictions of each classifier for the aimed to be classified instances performs the final prediction. Later, bagging is a smoothing operation which turns out to be advantageous when aiming to improve the predictive performance of regression or classification trees. In case of decision trees, the theory in [136] ensures the Breiman intuition that bagging can be described as a variance reduction technique. The figure 2.6 describes the bagging model aggregation.

The Algorithm 2 describes the steps of creating an ensemble of base classifiers via bagging aggregation and using the ensemble to classify an input pattern **x** [86]. Considering an independent classifier outputs, all classifiers scored same individual accuracy p, the majority voting rule is guaranteed to outperform individual performance [137]. Bagging aims at developing independent classifiers by taking bootstrap replicates as the training sets. The samples are pseudo-independent because they are taken from the same **Z**. However, even if they were drawn independently from the distribution of the problem, the classifiers built on these training sets might not give independent outputs.

2.13 Diversity Measures

Generally there is no completely perfect classifier that can solve all classification problems, that's why ensembles of classifiers work. In the same time if the ensemble members are identical there will be no value to combine them. Diversity measures plays an important rule in building ensembles of diverse classifiers. Considering the classifiers outputs types presented in section 2.9 equation 2.14, For a given dataset **Z**, a classifier D_i gives an output vector y_i, The diversity measures describe how diverse are the classifiers from each measure point of view[11]. The diversity measures are calculated using the contingency behaviour of two classifiers d_i and d_j across a dataset (see Table 2.1). Let **a** denotes the number of examples in the dataset correctly classified by both d_i and d_j; **b** denotes the number of examples correctly classified by d_i and misclassified by d_j; **c** denotes the number of examples misclassified by d_i , correctly classified by d_j; and **d** denotes the number of examples misclassified by both classifiers. The letter **m** denotes the total number of samples where, $m = a+b+c+d$. Let **Z**=$\{z_1,\ldots,z_n\}$, be a labelled data set, $z_j \in \Re^n$ coming from a classification problem. Let **Y**=$\{y_1,\ldots,y_n\}$; be the data assigned label by a classifier, $y_j \in \Re^n$. The diversity measures can be considered as a one of two types:

- Measures looking for diversity: the higher the value the more diverse (↑).
- Measures looking for similarity: the higher the value the less diverse (↓).

2.13.1 Pairwise Diversity Measures

The pairwise diversity measures require the consideration of a pair of classifiers and then we can average the value for a set of classifiers.

Table 2.1: The 2x2 Relationship table with probabilities

	$D_j correct(1)$	$D_j wrong(0)$
$D_i correct(1)$	a	b
$D_i wrong(0)$	c	d
Total=a+b+c+d		

1. **The Q-Statistic (Q ↓) [138]**
   Yule's Q statistic for two classifiers d_i, d_j is:

$$Q_{ij} = \frac{ad - bc}{ad + bc}. \tag{2.16}$$

2. **The Correlation Coefficient** ($\rho \downarrow$) [139]
   A well-known in mainstream statistics. The correlation between two classifiers d_i, d_j is:

$$\rho_{ij} = \frac{ad - bc}{\sqrt{(a+b)(a+c)(c+d)(b+d)}}. \tag{2.17}$$

3. **The Disagreement measure (D** $\uparrow$) [133, 140]
   The disagreement measure (D) between d_i, d_j, describes the samples classified to different classes by both classifiers it is given by:

$$D_{ij} = \frac{b+c}{m}. \tag{2.18}$$

4. **The Double Fault (DF** $\downarrow$) [141]
   The double-fault (D) of d_i, d_j, describes the samples wrongly classified by both classifiers is given by:

$$DF_{ij} = \frac{d}{m}. \tag{2.19}$$

2.13.2 Non-Pairwise Diversity Measures

The non-pairwise measures for a set of L classifiers:

1. **The Kohavi-Wolpert variance** ($KW \uparrow$) [142]
   The Kohavi-Wolpert measure of diversity for

$$KW = \frac{1}{NL^2} \sum_{j=1}^{N} l(z_j)(L - l(z_j)). \tag{2.20}$$

2. **The Interrater Agreement** ($\kappa \downarrow$) [143]
   Let $\bar{p}$ to be the average individual classification accuracy over all classifiers in the ensemble. the κ is given by:

$$\kappa = 1 - \frac{1}{2\bar{p}(1-\bar{p})} D_{ij}. \tag{2.21}$$

3. **The Entropy measure** ($Ent \uparrow$) [13]
   The diversity measure based on the concept of entropy is given by:

$$Ent = \frac{1}{N(L - \lfloor L/2 \rfloor - 1)} \sum_{j=1}^{N} min\{l(z_j), L - l(z_j)\}. \tag{2.22}$$

4. **The measure of Difficulty** ($\theta \downarrow$) [144]
   Let's define a discrete random variable X which values defined to be in $\{\frac{0}{L}, \frac{1}{L}, \ldots, 1\}$, the measure of difficulty θ is defined as:

$$\theta = Var(X). \tag{2.23}$$

5. **The Generalized Diversity** ($GD \uparrow$) [145]
   Let's define a random variable Y expressing the proportion of classifiers (out of L) that are incorrect on a randomly object drawn from $x \in \Re^n$, let p_i be the probability that i randomly chosen classifiers are incorrect for randomly chosen x, i.e., $p(Y = \frac{i}{L})$, if we denote,

$$p(1) = \sum_{i=1}^{L} \frac{i}{L} p_i, \tag{2.24}$$

   and

$$p(2) = \sum_{i=1}^{L} \frac{i}{L} \frac{(i-1)}{(L-1)} p_i. \tag{2.25}$$

   Then the generalised diversity measure, GD, is calculated as:

$$GD = 1 - \frac{p(2)}{p(1)}. \tag{2.26}$$

6. **The Coincident Failure Diversity** ($CFD \uparrow$), [145] The coincident failure diversity, CFD is a modification of GD proposed by [145]:

$$CFD = \begin{cases} 0, & p_0 = 1 \\ \frac{1}{1-p_0} \sum_{i=1}^{L} \frac{L-i}{L-1} & p_0 < 1 \end{cases} \tag{2.27}$$

   The table 2.2 summarize the pairwise and non-pairwise diversity measures names, abbreviations, description, formulas and references.

Table 2.2: Summary of diversity measures names, abbreviations, description, formulas and references.

Measure (Abbreviation)	Description	Formula	Reference
(1) Q-statistic (Q)	Pair-wise, require the true label.	$Q_{ij} = \frac{ad-bc}{ad+bc}$	Yule, 1919 [138]
(2) Correlation (ρ)	Pair-wise, require the true label.	$\rho_{ij} = \frac{ad-bc}{\sqrt{(a+b)(a+c)(c+d)(b+d)}}$	Sneath , Sokal, 1973 [139]
(3) Disagreement (Dis)	Pair-wise, do not use the true label.	$Dis_{ij} = \frac{b+c}{m}$	Ho, 1998; Skalak, 1996 [140]
(4) Double Fault (DF)	Pair-wise, require the true label.	$DF_{ij} = \frac{d}{m}$	Giacinto , Roli, 2001 [141]
(5) Entropy (E)	Non pair-wise , do not use the true label.	$Ent = \frac{1}{N(L-\lfloor L/2 \rfloor - 1)} \sum_{j=1}^{N} min\{l(z_j), L - l(z_j)\}$	Cunningham, Carney, 2000 [9]
(6) Kohavi-Wolpert (KW)	Non pair-wise , use the true label.	KW=L/1-L . (Dis_{ij})	Kohavi , Wolpert, 1996 [142]
(7) Interrater Agreement (κ)	Non pair-wise , use the true label.	$\kappa = 1 - \frac{1}{2p(1-p)} Dis_{ij}$	Dietterich, 2000b[6]; Fleiss, 1981 [143]
(8) Difficulty (Theta:θ)	Non pair-wise , use the true label.	$\theta = Var(Dis_{ij})$	Hansen , Salamon, 1990 [146]
(9) Generalized Diversity (GD)	Non pair-wise , use the true label.	GD=1-p(2)/p(1).	Partridge , Krzanowski, 1997 [145]
(10) Coincident Failure Diversity (CFD)	Non pair-wise , use the true label.	CFD=$\begin{cases} 0, & p_0 = 1 \\ \frac{1}{1-p_0} \sum_{i=1}^{L} \frac{L-i}{L-1} & p_0 < 1 \end{cases}$	Partridge, Krzanowski, 1997[145]

After about twenty years of active research in the classifier ensemble field, understanding the notion of diversity remains one of the main open problems [119, 86]. On the one hand, there is a general agreement on the qualitative definition of diversity and on its role, e.g.: "it is desired that the individual learners should be *accurate and diverse*" [119]; "Common sense suggests that the classifiers in the ensemble should be as accurate as possible and should not make coincident errors" [86] (ch. 8). On the other hand, measuring diversity and explicitly using it for ensemble construction exhibits several open issues.

A number of **diversity measures** have been proposed over the years [11, 119, 86]. Most of diversity measures have been derived intuitively, as attempts to formally characterize the pattern of individual classifiers' errors (e.g., the Double-Fault and Disagreement measures [86]). In particular, it has been clearly pointed out that diversity measures alone can not be monotonically related to ensemble accuracy, since the latter depends instead on a trade-off between diversity and individual classifiers' performance [147, 86]; quoting from [86] (ch. 8), A few other measures have been inspired by *exact* error decompositions derived in the regression field, despite the lack of a direct analogy with regression problems was pointed out in [7]: the Kohavi-Wolpert Variance [11] (and our attempt in [2]) was inspired by the bias-variance-covariance error decomposition [148], and the measure derived in [149] (which we extended in [2]) by the ambiguity decomposition [150]. The rationale of such measures is to look for exact, additive decompositions of the ensemble error into terms accounting for individual classifiers' performance, and terms hopefully interpretable as diversity; the results of [149] provided useful insights, leading to the concept of "good" and "bad" diversity. Several authors also analyzed, empirically or analytically, the connection between ensemble performance on one side, and the pattern of individual classifiers' performance and existing diversity measures on the other side (e.g., [147, 117]). Such a relationship turned out to be far from clear-cut, and no "right" diversity measure has emerged so far.

2.14 Previous Work on Using Diversity for Ensemble Design

Since this problem has exponential complexity in the size of the original ensemble, many heuristics have been presented. In this context, diversity measures have been used as an objective function of the pruning algorithms, to look for a trade-off between individual classifiers' performance and diversity. The usefulness of such an approach has however been doubted by many researchers, based also on empirical evidences [147], [86] (chapter 8.3). In particular, its real advantage over directly evaluating ensemble performance (estimated, e.g., from validation data) is not crystal yet. On the other hand, it is well known that famous and effective ensemble building techniques like bagging and boosting do not use any kind of diversity measure. Didaci *et. al.* [2] focused on the derivation of exact decomposition's of the ensemble error, and outlined several research directions. One of them, which we start addressing in this thesis, consists of comparing the effectiveness of explicitly using diversity measures in ensemble pruning, with the simple estimation of ensemble performance. Although many pruning methods have been proposed so far, the above comparison has been carried out by only a few authors, and with a limited scope. This work focus on pruning methods based on forward selection (FS) algorithms, which are the easiest ones on which such a comparison can be made, and carry out an empirical investigation on 23 benchmark data sets, using the popular bagging as the ensemble construction technique, and majority voting as the fusion rule. We evaluate ten well known diversity measures analyzed in [11], and one measure specifically defined for ensemble pruning [151]. In particular, we specifically evaluate the effect of the validation set size on ensemble pruning effectiveness. During twenty years of research in the classifier ensemble field, understanding the notion of *diversity* has been one of the main goals [118, 86]. A general agreement exists on the qualitative definition of diversity and on its role in classifier ensembles; basically, to obtain an effective

(accurate) ensemble, its members should be as accurate *and* diverse as possible, where "diverse" means that they should not make coincident errors [118, 86]. Individual accuracy and diversity are well-known to be contrasting goals, which means that a trade-off between them has to be achieved. On the other hand, formally defining and measuring diversity, as well as explicitly using it for ensemble construction, turned out to be not straighforward. Most measures have been derived intuitively, as attempts to formally characterize the pattern of error of individual classifiers (e.g., the Double-Fault and Disagreement measures) [86]. In particular, it has been clearly pointed out that diversity measures alone can not be monotonically related to ensemble accuracy, since the latter depends on a trade-off between diversity and individual classifiers' performance [147, 86]. A few other measures have been inspired by *exact* error decompositions derived in the regression field, despite the lack of a direct analogy to classification problems [7]. The Kohavi-Wolpert Variance [11] (and our attempt in [152] was inspired by the bias-variance-covariance error decomposition of [153]. The measure derived in [149] (which we extended in [152]) was inspired by the *ambiguity* decomposition of [154], and provided useful insights, leading to the concept of "good" and "bad" patterns of diversity. Such measures were motivated by the goal of obtaining exact, additive decomposition's of the ensemble error into terms accounting for individual classifiers' performance, and terms hopefully interpretable as diversity. Several authors also analyzed, empirically or analytically, the connection between ensemble performance on one side, and the pattern of individual classifiers' performance and existing diversity measures on the other side (e.g., [147, 12]). Such a relationship turned out to be far from clear-cut, and no "right" diversity measure has emerged so far.

Beside theoretical investigations on defining diversity and using this concept to explain ensemble performance, a considerable research effort has been spent toward the practical goal of *explicitly* using diversity measures for ensemble construction. Among existing methods, almost all follow the *overproduce and choose* approach. It consists of first generating a large ensemble (e.g., using Bagging) and then selecting the most accurate subset of classifiers. The overproduce and choose approach is also known as ensemble *pruning*, *selection* or *thinning*. It is supported by theoretical and empirical evidence showing that a (suitable) subset of the available classifiers could outperform the original ensemble [16, 155, 156].

Since ensemble pruning has exponential complexity in the size of the original ensemble, several heuristics have been proposed. In this context, diversity measures have been used in the objective function of pruning methods, to attain a trade-off between individual classifiers' performance and diversity. The effectiveness of using diversity measures to this aim has however been questioned by several authors, based also on empirical evidence [156, 147, 11], and [86] (ch. 8.3). In particular, its actual advantage over directly evaluating ensemble performance (estimated, e.g., from validation data) is not clear yet. It is also well known that popular and effective ensemble construction techniques like Bagging and Boosting do not use any explicit diversity measure. Nevertheless, despite the questionable effectiveness of heuristic pruning approaches, a theoretically grounded analysis in [5] related to ensembles of binary classifiers combined by majority voting has shown that (a suitable measure of) diversity can have a regularization effect in ensemble pruning.

Based on the above premises, the aim of this work is to compare the effectiveness of explicitly using existing diversity measures in ensemble pruning, against the direct estimation of ensemble performance. This is a follow-up of our preliminary work [1]. In particular, inspired by [5], we evaluate whether several well-known diversity measures can have a regularization effect to the (estimate of) ensemble accuracy. To this aim we consider a pruning

method based on the forward selection (FS) algorithm, since it allows a direct comparison between evaluation functions. We then compare the estimated ensemble accuracy against its linear combination with a given diversity measure, using the latter as a regularizer. We carry out experiments on 37 benchmark data sets. We use the popular Bagging as the ensemble construction technique and majority voting as the fusion rule, and evaluate a subset of the ten well-known diversity measures analyzed in [11]. Our results show that using diversity measures for ensemble pruning can be advantageous over using only ensemble accuracy, and that diversity measures can act as regularizers in this context. As pointed out in Sect. 2.14, diversity measures have been explicitly used so far for ensemble construction only in pruning methods. The only exception is [157], where a diversity measure was used in an ensemble *learning* algorithm.

2.15 Other Ensemble Creation Techniques

There are several approaches to create an ensemble of classifiers, rather than the popular creation techniques such as bagging, boosting,...,etc. **Genetic Algorithm** is widely used to involve in ensemble creation[158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172]. Genetic algorithm is used in [173] to wight each classifier contribution in the ensemble decision, also presented a framework for ensemble thinning. Feature selection for ensemble creation using genetic algorithm search approach presented by [174], a comparison between bagging and boosting is presented to show that their approach produces a more reliable ensemble and up to 80% in memory reduction. A neural network ensemble creation using genetic programming is presented in [175], they independently train a fixed number of neural networks, then the genetic programming is applied to combine the trained NNs into an ensemble. Several **mathematical frameworks** are presented to construct classifiers ensembles [176, 177, 178, 179, 180, 181, 182, 183, 184]. A proposal to improve microaneurysm detection [176] using ensemble-based framework, they propose a combination of the internal microaneurysm components, their approach assumed to ranked the first by this time.

2.16 Ensemble Pruning

Given a set of trained individual learners, rather than combining all of them, ensemble pruning tries to select a subset of individual learners to comprise the ensemble. An apparent advantage of ensemble pruning is to obtain ensembles with smaller sizes; this reduces the storage resources required for storing the ensembles and the computational resources required for calculating outputs of individual learners, and thus improves efficiency. There is another bene- fit, that is, the generalization performance of the pruned ensemble may be even better than the ensemble consisting of all the given individual learners[86]. In [185] ensemble pruning methods have been categorized as follows:

- **Ranking-based**: individual classifiers are first ranked according to some criterion, and then the top-L ones are selected as the final ensemble.
- **Clustering-based**: individual classifiers are first clustered based on the similarity of their predictions; each cluster is then pruned to remove redundant classifiers, and the remaining ones in each cluster are finally combined.

- **Optimization-based** methods search for a subset of the original ensemble that optimizes a given objective function, which can include a diversity measure. To avoid exhaustive search, three main heuristic search strategies have been proposed: hill climbing, genetic algorithms, and semi-definite programming.

In particular, several optimization-based pruning methods use the forward or backward search (FS/BS) strategy [186, 17, 187, 188, 151]. Given an initial ensemble, FS picks the best individual classifier and iteratively selects among the remaining classifiers the one that maximizes a given objective function. It stops either when a predefined ensemble size is reached, or when all the classifiers from the original ensemble have been selected; in the latter case, FS returns the best ensemble among the ones obtained at each iteration. The BS algorithm works similarly, iteratively removing from E one classifier at a time. More refined versions of FS/BS have also been proposed, which include a back-fitting step [17]. In the context of optimization-based pruning, three kinds of objective functions have been proposed so far:

- The ensemble accuracy [17, 189], combined with a diversity measure in [5].
- A given diversity measure (disregarding the performance of individual classifiers and of the ensemble) [17, 188].
- Ad hoc measures specifically devised for ensemble pruning, which combine into a single scalar the individual classifiers' performance and the *complementarity* (diversity) between their errors [186, 187, 151, 188].

A different and theoretically grounded view on the role of diversity in ensemble pruning was proposed in [5], in the context of ensembles of binary classifiers combined by majority voting: using a suitable diversity measure it was shown that promoting diversity can be seen as a regularization technique. A pruning method was also proposed based on these results, which exploits a strategy similar to FS: it starts with the most accurate classifier from the original ensemble, then iteratively sorts the remaining classifiers based on their diversity (evaluated using the proposed measure) with the current sub-ensemble, and among the most diverse ones it selects the classifier which leads to the next most accurate sub-ensemble. It is also worth mentioning two ensemble construction techniques [190, 8] which are not pruning techniques, but are related to the pruning criteria considered in this work. They consist of building individual classifiers from different subsets of the available features, analogously to the well known Random Subspace Method [191]. The difference with respect to RSM is that they use a feature selection criterion analogous to the optimization-based pruning criterion mentioned above (including FS in [8]), and evaluate the individual classifiers on the basis of a trade-off between individual classifiers' accuracy and diversity. In particular, in [8] a linear combination of these two quantities was used as the objective function, and five different measures of diversity were considered.

2.16.1 Ordering Based

Order the individual learners according to some criterion, and only the learners in the front-part will be put into the final ensemble.Ordering-based pruning originated from Margineantu and Dietterich *et. al.* [17] work on boosting pruning. Later, most efforts were devoted to pruning ensembles generated by parallel ensemble methods. Given N individual learners

$h_1,\ldots,h_N$, suppose they are combined sequentially in a random order, the generalization error of the ensemble generally decreases monotonically as the ensemble size increases, and approaches an asymptotic constant error. It has been found that, however, if an appropriate ordering is devised, the ensemble error generally reaches a minimum with intermediate ensemble size and this minimum is often lower than the asymptotic error. Hence, ensemble pruning can be realized by ordering the N individual learners and then putting the front T individual learners into the final ensemble. It is generally hard to decide the best T value, but fortunately there are usually many T values that will lead to better performance than the allmember ensemble, and at least the T value can be tuned on trainin more crucial problem is how to order the individual learners appropriately. During the past decade, many ordering strategies have been proposed. Most of them consider both the accuracy and diversity of individual learners, and a validation data set V with size $|V|$ is usually used (when there are not sufficient data, the training data set V or its sub-samples can be used as validation data). In the following we introduce some representative ordering-based pruning methods[16].

2.16.2 Clustering Based

Identify a number of representative prototype individual learners to constitute the final ensemble. Usually, a clustering process is employed to partition the individual learners into a number of groups, where individual learners in the same group behave similarly while different groups have large diversity. Then, the prototypes of clusters are put into the final ensemble. An intuitive idea to ensemble pruning is to identify some prototype individual learners that are representative yet diverse among the given individual learners, and then use only these prototypes to constitute the ensemble. This category of methods is called as clustering-based pruning because the most straightforward way to identify the prototypes is to use clustering techniques. first step, the individual learners are grouped into a number of clusters. Different clustering techniques have been exploited for this purpose. For example, Giacinto *et. al.* [192] used hierarchical agglomerative clustering and regarded the probability that the individual learners do not make coincident validation errors as the distance; Lazarevic and Obradovic [193] used $k-means$ clustering based on Euclidean distance; Bakker and Heskes [194] used deterministic annealing for clustering; etc.

2.16.3 Optimization Based

Formulate the ensemble pruning problem as an optimization problem which aims to find the subset of individual learners that maximizes or minimizes an objective related to the generalization ability of the final ensemble. Many optimization techniques have been used, e.g., heuristic optimization methods, mathematical programming. Optimization-based pruning originated from [4] which employs a genetic algorithm[195] to select individual learners for the pruned ensemble. Later, many other optimization techniques, including heuristic optimization, mathematical programming and probabilistic methods have been exploited. Bhatnagar *et. al.* [196] presented an ensemble pruning approach that takes into account both the ensemble accuracy and pair-wise diversity between pruned ensemble members, they aimed to achieve the smallest pruned ensemble size meanwhile keeping the final accuracy reasonable.

Chapter 3

Diversity Measures For Ensemble Creation

Almost all the existing methods that **explicitly use diversity for ensemble construction** follow the overproduce and choose approach (except for [197], where a diversity measure is used in an ensemble *learning* algorithm). It consists of first generating a large ensemble (e.g., using bagging) and then selecting the most accurate subset of classifiers (usually with a predefined size). This is known as ensemble *pruning*, *selection* or *thinning*. Since this problem has exponential complexity in the size of the original ensemble, several heuristics have been proposed. In this context, diversity measures have been used in the objective function of pruning methods, to look for a trade-off between individual classifiers' performance and diversity. The effectiveness of such an approach has however been questioned by several authors, based also on empirical evidences [147], [86] (chapter 8.3). In particular, its actual advantage over directly evaluating ensemble performance (estimated, e.g., from validation data) is not clear yet. On the other hand, it is well known that popular and effective ensemble construction techniques like bagging and boosting do not use any explicit diversity measure.

3.1 Using Diversity for Classifier Ensemble Pruning: An Empirical Investigation

In [2] we discussed the above issues, focusing on the derivation of exact decompositions of the ensemble error, and outlined several research directions. One of them, which we start addressing in this work, consists of comparing the effectiveness of explicitly using diversity measures in ensemble pruning, with the simple estimation of ensemble performance. Although many pruning methods have been proposed so far, the above comparison has been carried out by only a few authors, and with a limited scope. In this work we focus on pruning methods based on forwardselection (FS) algorithms, which are the easiest ones on which such a comparison can be made, and carry out an empirical investigation on 23 benchmark data sets, using the popular bagging as the ensemble construction technique, and majority voting as the fusion rule. We evaluate ten well known diversity measures analyzed in [11], and one measure specifically defined for ensemble pruning [151]. In particular, we specifically evaluate the effect of the validation set size on ensemble pruning effectiveness.

Algorithm 3 Forward Selection algorithm for ensemble pruning

Input: An ensemble E of N classifiers; a desired ensemble size $L < N$; a validation set V; an objective function m (to be computed on V)
Output: A subset of L classifiers from E.
$C \leftarrow$ the most accurate individual classifier from E .
$S \leftarrow \{C\}$.
FOR $k = 2, \ldots, L$ $C^* \leftarrow \arg\max_{C \in E \setminus S} m(S \bigcup \{C\})$.
$S \leftarrow S \bigcup \{C^*\}$.
END FOR
RETURN S

The following kinds of objective functions have been proposed:

- The ensemble performance, [17] (reduce-error pruning technique), [198, 5].
- Diversity measures, disregarding the performance of individual classifiers, [17] (Kullback-Leibler Divergence pruning), [18] and [16] (kappa-thinning).
- Measures combining into a single scalar the individual classifiers' performance and the *complementarity* between their errors [199, 187, 151] and [188] (AID thinning and Concurrency thinning).

Among the latter measures we focus on the followings:

- A measure aimed at minimizing the number of coincident errors between ensemble members, when majority voting is used, proposed in [199] to be used in the FS algorithm. It selects the classifier that correctly labels the highest number of validation samples, among the ones misclassified by the majority of classifiers in the current ensemble, i.e., the one which minimizes:

$$\begin{array}{ll} \sum_{(\mathbf{x},y)\in V} & I\left[C^*(\mathbf{x}) \neq y \wedge I\left[\sum_{C\in S} C(\mathbf{x}) \neq y > \lceil \frac{|S|}{2} \rceil\right]\right] \\ & -I\left[C^*(\mathbf{x}) = y \wedge I\left[\sum_{C\in S} C(\mathbf{x}) \neq y > \lceil \frac{|S|}{2} \rceil\right]\right] , \end{array} \tag{3.1}$$

where $I[A] = 1$ if $A =$True, and $I[A] = 0$ otherwise.

- Two measures proposed in [187] to be used in the FS algorithm, with the majority voting rule: Complementariness (the sum of validation samples which are wrongly classified by the current ensemble, but not by the candidate classifier, to be maximized: it is a variant of Eq. 3.1), and Margin Distance (to be minimized), respectively defined as:

$$\sum_{(\mathbf{x},y)\in V} I\left[C^*(\mathbf{x}) = y \wedge I\left[\sum_{C\in S} C(\mathbf{x}) \neq y > \lceil \frac{|S|}{2} \rceil\right]\right] , \tag{3.2}$$

$$\left\| \mathbf{o} - \frac{1}{|E|}\left(\mathbf{c}_{C^*} + \sum_{C\in S} \mathbf{c}_C\right)\right\|_2^2 , \tag{3.3}$$

where $\mathbf{c}_C$ is a $|V|$-dimensional vector whose i-th element is defined as $2I[C(\mathbf{x}_i) = y_i] - 1 \in \{-1, +1\}$ and the objective point $\mathbf{o}$ is defined as a constant vector with equal components $o_i = p$, $0 < p < 1$.

-Two measures proposed also in [188], related to their Accuracy In Diversity (AID) thinning and Concurrency thinning techniques, based on BS. The former removes the east accurate classifier on validation samples that are correctly classified by a fraction from L to U of the classifiers in S, where L and U are constant values set as functions of the average accuracy of individual classifiers and the number of classes (see [188] for the details). The latter removes the classifier that minimizes the following measure, aimed at penalizing the agreement on correctly classified samples (again a variant of Eq. 3.1):

$$\begin{array}{ll} \sum_{(\mathbf{x},y)\in V} I\left[C^*(\mathbf{x}) = y \wedge S(\mathbf{x}) = y\right] & +2I\left[C^*(\mathbf{x}) = y \wedge S(\mathbf{x}) \neq y\right] \\ & -2I\left[C^*(\mathbf{x}) = y \wedge S(\mathbf{x}) = y\right] \ . \end{array} \tag{3.4}$$

- The Uncertainty Weighted Accuracy (UWA), proposed in [151], as a variant of the Concurrency measure of Eq. (3.4):

$$\begin{array}{lll} \sum_{(\mathbf{x},y)\in V} & NF(\mathbf{x}) \times I\left[C^*(\mathbf{x}) = y \wedge S(\mathbf{x}) = y\right] \\ + & NT(\mathbf{x}) \times I\left[C^*(\mathbf{x}) = y \wedge S(\mathbf{x}) \neq y\right] \\ - & NF(\mathbf{x}) \times I\left[C^*(\mathbf{x}) \neq y \wedge S(\mathbf{x}) = y\right] \\ - & NT(\mathbf{x}) \times I\left[C^*(\mathbf{x}) \neq y \wedge S(\mathbf{x}) \neq y\right] \ , \end{array} \tag{3.5}$$

where $NT(\mathbf{x})$ and $NF(\mathbf{x})$ are the number of classifiers in S that classify $\mathbf{x}$ respectively correctly and wrongly.

A comparison between the effectiveness of directly using ensemble performance as the objective function, and using measures involving diversity, has been carried out by a few authors [188, 151, 5], often limited to the specific evaluation measure they were proposing, and using different and incomparable experimental setups (different data sets, base classifiers, ensemble construction methods, etc.). We also point out that only in [151, 5] the use of diversity provided a statistically significant improvement over the use of ensemble performance.

3.2 Experimental (1) Settings

Our aim is thus to carry out an extensive experimental investigation of FS-based ensemble pruning methods, focused on the comparison between the use of ensemble performance as the objective function, and the use of measures involving diversity. To this aim, we focus on the basic FS algorithm without back-fitting, and consider three kinds of objective functions:

1. Ensemble accuracy.
2. A generic diversity measure, focusing on well known ones analyzed in [11]. Although diversity alone is deemed to be not effective for ensemble pruning [147, 86], we consider it to provide more direct evidence to these findings.
3. Measures that combine individual classifiers' performance and complementarity: we consider the UWA measure of Eq. (3.5) [151].

We also consider another way to combine ensemble performance and diversity. Since diversity measures are not homogeneous to classification accuracy, to avoid combining them with individual classifiers' accuracy in a arbitrary ways (e.g., by a linear combination), we use a two-stage FS/BS: first we select $M < N$ classifiers using either ensemble accuracy or diversity; then we further select $L < M$ classifiers using the other measure. Algorithm 4 shows

Algorithm 4 Two-stage Forward Selection algorithm for ensemble pruning

Input: a classifier ensemble E of size N; a desired ensemble size $L < N$; an intermediate ensemble size M, with $L < M < N$; a validation set V; a diversity measure d

Output: a subset of L classifiers from E

step 1 (accuracy-based pruning): select from E an ensemble E' of size M using Algorithm 3, and using classification accuracy as the objective function m

step 2 (diversity-based pruning): select from E' an ensemble S of size L using Algorithm 3, and using d as the objective function m

RETURN S

the version in which ensemble accuracy is used at the first stage. In our experiments we considered both versions.

We chose 23 benchmark data sets from the UCI Machine Learning Repository Database,[1] with at least 350 samples, only numerical attributes, and without missing values (see Table 3.1). We used bagging to construct the original ensemble, majority voting as the combining rule, and two different base classifiers: multi-layer perceptron neural networks (MLP-NN) with one hidden layer containing ten units, and decision trees (DT). For MLP-NN we used the standard Matlab implementation[2], learning rate $\eta = 0.05$, and maximum number of training epochs equal to 300. For DTs we used the code of [86] (par. 2.A.2.1), with the Gini impurity criterion, χ^2 stopping criterion, and the default threshold equal to 1 for the pre-pruning stopping criterion. We set the size of the original ensemble to $N = 100$, and considered four different sizes of the pruned ensembles: $L = 5, 15, 25$ and 35.

We used only FS-based pruning. In the two-stage Algorithm 4 we set the size M of the first-stage pruned ensemble to $M = L + \lfloor (N-L)/2 \rfloor$. Since FS-based pruning starts from the best individual classifier, to better appreciate its effectiveness we chose the training set size of each data set in preliminary experiments, by maximizing the difference between the accuracy of an ensemble of 100 classifiers (constructed by bagging) and of the best individual classifier (see the right-most column of Table 3.1). The size of the validation set is one third of the training set, whilst remaining patterns form the testing set. We also considered a small validation set (one sixth of the training set) to evaluate its effect on the performance of ensemble pruning. We evaluated all the diversity measures analyzed in [11] (see Table 3.2), and the UWA measure of Eq. (3.5).

We carried out 20 runs of the experiments. At each run we selected the training, validation and testing sets by stratified random sampling (no data set was originally subdivided into a training and a testing set). We applied bagging to the training set, to construct the original ensemble of $N = 100$ classifiers. We then run Algorithm 3 separately using as the objective function the ensemble accuracy, each diversity measure, and the UWA measure. We also run the two-stage Algorithm 4 in both versions (using accuracy either at the first or at the second stage), for each diversity measure. We finally computed, separately for each data set, pruning method, base classifier, ensemble size L and validation set size, the average accuracy and its standard deviation on testing samples, over the 20 runs. Due to space limits, we make these results available only from our web site,[3] and only report the results of

[1] http://www.ics.uci.edu/~mlearn/MLRepository.html
[2] http://it.mathworks.com/help/nnet/ref/patternnet.html
[3] http://pralab.diee.unica.it/en/MCS2015Appendix1

				Tr. set size	
Dataset	Samples	Classes	Features	MLP-NN	DT
Australian	690	2	14	0.42	0.42
Balance Scale	625	3	4	0.18	0.42
Blood Transfusion	748	2	4	0.48	0.60
Breast Cancer	699	2	9	0.30	0.12
Bupa	345	2	6	0.54	0.06
Checker Board	1000	2	2	0.36	0.30
Coil 2000	9822	2	85	0.06	0.18
Cone tours	2000	3	2	0.06	0.24
Contraceptive	1473	3	9	0.36	0.60
ILPD	583	2	9	0.50	0.06
Laryngeal 2	692	2	16	0.06	0.48
Monk2	432	2	6	0.48	0.06
Page Blocks	5473	5	10	0.06	0.42
Phoneme	5404	2	5	0.36	0.30
Pima Indians	768	2	8	0.54	0.30
Pop Failures	540	2	20	0.42	0.30
Ring	7400	2	20	0.42	0.30
SaHeart	462	2	4	0.54	0.18
Sata Log Image Seg	2310	7	19	0.44	0.30
Landsat Satellite	6435	7	36	0.60	0.48
Spam Base	4601	2	57	0.42	0.30
Townorm	7400	2	20	0.12	0.30
Wine Quality	4898	7	11	0.18	0.30

Table 3.1: Characteristics of the data sets. The two rightmost columns report the size of the training set for the two base classifiers, as a fraction of the whole data set.

Diversity measure	Abbreviation
Entropy	E
Kohavi-Wolpert	KW
Coincidence Failure Diversity	CFD
Generalized Diversity	GD
Interrater Agreement	Kappa
Difficulty	Theta
Q Statistic	Q
Correlation	Rho
Disagreement	D
Double Fault	DF
Uncertainity Weighted Accuracy	UWA
Partridge and Yates' measure	PYM
Complementariness	Cs
Margin Distance	MD
CONCURRENCY	Cy

Table 3.2: Diversity measures used in the experiments.

the statistical significance test. We compared the accuracy of pruned ensembles attained by Algorithm 3 using ensemble accuracy as the objective function, and using each of the other measures (both by Algorithm 3 and Algorithm 4). To this aim we used the Wilcoxon signed-rank test, which is recommended in [200] for comparing two algorithms over multiple data sets. Our goal was to assess whether the difference was significant, and, if so, whether using ensemble accuracy as the objective function was the best or the worst option. Accordingly, we made two one-sided tests (at the $\alpha = 0.05$ level), evaluating the null hypotheses that FS-based pruning using ensemble accuracy (or a measure involving diversity) is not better than using a given measure involving diversity (or ensemble accuracy). Only if *both* null hypotheses are rejected, it can be concluded that there is no statistically significant difference between the two options.

3.3 Experimental (1) Results

For each pruned ensemble size L, base classifier, and validation set size. Tables 3.3–3.8 report the comparison between FS-based pruning (Algorithm 3) using ensemble accuracy, and FS-based pruning implemented by Algorithm 3 using either a diversity measure or UWA, and by Algorithm 4 combining ensemble accuracy and diversity.

Tables 3.3 and 3.4 clearly show that using ensemble accuracy often provides a better or comparable pruned ensemble than using any diversity measure alone, or UWA. The only exception is the GD measure, using DT as the base classifier, $L = 15$, and a small validation set (see Table 3.3). Interestingly, most of the cases when using diversity attained comparable results occur for three only measures: Entropy, Generalized Diversity and Kappa.

Tables 3.5–3.8, which refer to the two-stage FS algorithm combining ensemble performance and diversity, show a different pattern, instead. When a larger validation set is used, ensemble accuracy still produces often a better or comparable pruned ensemble; however, for ensembles of DTs it never outperforms the combination of ensemble performance and diversity; moreover, it almost always performs worse with respect to the Double Fault (DF) measure. When a smaller validation set is used, instead (in this case only the results for DTs are available), combining ensemble accuracy and diversity is often better, or at least not worse, than using only ensemble accuracy (four right-most columns of Tables 3.5 and 3.7, vs the same columns of Table 3.3). Remarkably, this happens for most diversity measures.

These results seem to suggest that estimating the ensemble performance is the best option for FS-based pruning, provided that a sufficiently large validation set is available. Otherwise, a combination of ensemble performance and diversity can be advantageous, at least for some types of base classifiers. One possible explanation is that diversity measures have a *regularization* effect capable of preventing over-fitting, to some extent, as already argued in [5]. This is an interesting and non-straightforward property, which is worth investigating more throughly.

We empirically investigated the effectiveness of explicitly using diversity measures for FS-based ensemble pruning, vs the simple estimation of ensemble accuracy. On the one hand, our results provide a more direct evidence in support of previous findings that using diversity measures alone is not effective for ensemble pruning [147, 86], and in particular are in agreement with the well-established fact that diversity is not monotonically related to ensemble accuracy [86]. On the other hand, they suggest that, combined with the ensemble performance, diversity can be useful to FS-based pruning when a small validation set is available. It seems therefore that diversity has a regularization effect. This possible effect has already been argued through the derivation of generalization bounds in [197], in the context of constructing ensembles of support vector machines, as well as in [5], in the context of FS-based ensemble pruning. However, in [5] the effect of different validation set sizes was not assessed, and only one diverstity and two complmentarity measures were considered for comparison. An apparently opposite issue was raised in [147], where one of the drawback of existing diversity measures was claimed to be the lack of a regularization term: this was however referred to ensemble construction approaches based on maximizing *only* diversity, and thus it is not in contradiction of the findings of [5] and of ours. To sum up, what our results provide is not a sharp conclusion either in favour or against the effectiveness of explicitly using diversity measures for ensemble pruning. Instead, and perhaps more interestingly, they provide some hints on the conditions under which diversity can be useful, and clearly suggest as a future research direction a more thorough investigation of the effect of

Diversity Measure	Ensemble size L							
	Val. size: 1/3 Tr. size				Val. size: 1/6 Tr. size			
	5	15	25	35	5	15	25	35
E	-	-	-	-	-	-	-	-
KW	A	A	A	A	A	A	A	A
CFD	A	A	A	A	A	A	A	A
GD	-	-	-	-	-	D	-	-
Kappa	-	-	-	-	-	-	-	-
Theta	-	A	-	A	A	-	-	-
Q	-	A	-	A	A	-	-	-
Rho	A	A	A	A	-	A	A	A
D	A	A	A	A	A	A	-	-
DF	A	A	A	A	A	A	A	A
UWA	A	A	A	A	-	-	-	D
PYM	-	-	-	-	-	-	-	-
Cs	A	A	A	A	A	A	A	A
MD	-	-	-	-	-	-	-	-
Cy	-	-	-	-	-	-	-	-

Table 3.3: Comparison of FS-based pruning (Algorithm 3) using ensemble accuracy vs. using each diversity measure and UWA, PYM, Cs, MD and Cy measures for different ensemble sizes L and validation set sizes. Base classifier: DT. 'A': using accuracy is statistically significantly better than using the corresponding diversity/other measures, over the 23 data sets; 'D': using the corresponding diversity/UWA measure is better than ensemble accuracy; '-': there is no statistically significant difference between the two measures.

Diversity Measure	Ensemble size L							
	Val. size: 1/3 Tr. size				Val. size: 1/6 Tr. size			
	5	15	25	35	5	15	25	35
E	-	-	-	-	A	-	-	-
KW	A	A	A	A	-	-	-	-
CFD	A	A	A	A	-	-	-	-
GD	-	-	-	-	-	-	-	-
Kappa	-	-	-	-	-	-	-	-
Theta	A	A	A	A	-	-	-	-
Q	A	A	A	A	-	-	-	-
Rho	A	A	A	A	-	-	-	-
D	A	A	A	A	-	-	-	-
DF	A	A	A	A	-	-	-	-
UWA	-	-	-	-	-	-	-	-
PYM	-	-	-	-	-	-	-	-
Cs	A	A	A	A	A	A	A	A
MD	-	-	-	-	-	-	-	-
Cy	-	-	-	-	-	-	-	-

Table 3.4: Comparison of FS-based pruning (Algorithm 3) using ensemble accuracy vs. using each diversity measure and UWA, PYM, Cs, MD and Cy for a validation set size equal to 1/3 and 1/6 of the training set size. Base classifier: MLP-NN. See caption of Table 3.3 for the meaning of table entries.

validation set size. Our analysis can also be extended to other pruning methods categorized in [185] as optimization-based, which use genetic algorithms [4, 150] or a kind of best-first search [201], where ensemble accuracy can also be used as the objective function. Finally, this investigation can be extended to regression problems, in which the exact Ambiguity decomposition includes a diversity term which does *not* depend on ground truth, contrary to most diversity measures for classification problems, including all the ones in [11] considered in this work, and the one in [149] derived from an *exact* Ambiguity-like decomposition; this allows it to be computed also on a set of *unlabeled* samples, thus potentially reducing the

	Ensemble size L							
Diversity	Val. size: 1/3 Tr. size				Val. size: 1/6 Tr. size			
Measure	5	15	25	35	5	15	25	35
E	-	-	-	-	D	D	D	D
KW	-	-	-	-	D	D	D	D
CFD	-	-	-	D	D	D	D	D
GD	-	-	-	D	D	D	D	-
Kappa	-	-	-	-	D	D	D	-
Theta	-	-	-	-	D	D	D	-
Q	-	-	-	-	-	D	D	-
Rho	-	-	-	-	D	D	D	-
D	-	-	-	-	D	D	D	-
DF	-	D	D	D	D	D	D	D

Table 3.5: Comparison of FS-based pruning (Algorithm 3) using ensemble accuracy vs Algorithm 4 using ensemble accuracy at the first stage and each diversity measure at the second stage. Base classifier: DT. See caption of Table 3.3 for the meaning of table entries.

	Ensemble size L							
Diversity	Val. size: 1/3 Tr. size				Val. size: 1/6 Tr. size			
Measure	5	15	25	35	5	15	25	35
E	A	A	A	A	A	A	A	A
KW	A	A	A	A	A	A	A	A
CFD	-	-	-	-	A	-	D	-
GD	-	-	-	-	A	-	-	-
Kappa	A	A	-	-	A	A	A	A
Theta	A	-	-	-	A	-	-	-
Q	A	-	-	A	A	A	A	A
Rho	A	A	A	-	A	A	A	A
D	A	A	A	A	A	A	A	A
DF	D	D	D	D	-	-	-	-

Table 3.6: Comparison of FS-based pruning (Algorithm 3) using ensemble accuracy vs Algorithm 4 using ensemble accuracy at the first step and each diversity measure at the second stage, for a validation set size equal to 1/3 and 1/6 of the training set size. Base classifier: MLP-NN. See caption of Table 3.3 for the meaning of table entries.

	Ensemble size L							
Diversity	Val. size: 1/3 Tr. size				Val. size: 1/6 Tr. size			
Measure	5	15	25	35	5	15	25	35
E	-	-	-	-	D	D	D	-
KW	-	-	-	-	D	D	-	-
CFD	-	-	-	-	D	D	-	D
GD	-	-	D	D	-	D	D	D
Kappa	-	-	-	-	-	D	D	-
Theta	-	-	-	-	-	D	D	-
Q	-	-	-	-	-	D	-	-
Rho	-	-	-	-	D	D	D	-
D	-	-	-	-	D	D	-	-
DF	-	-	D	D	-	D	D	D

Table 3.7: Comparison of FS-based pruning (Algorithm 3) using ensemble accuracy vs Algorithm 4 using each diversity measure at the first stage and ensemble accuracy at the second stage. Base classifier: DT. See caption of Table 3.3 for the meaning of table entries.

Diversity Measure	Ensemble size L							
	Val. size: 1/3 Tr. size				Val. size: 1/6 Tr. size			
	5	15	25	35	5	15	25	35
E	-	A	A	A	-	-	-	-
KW	-	A	A	A	A	A	-	A
CFD	-	-	-	-	A	D	D	-
GD	-	D	-	-	-	D	D	-
Kappa	-	A	A	-	A	A	-	-
Theta	-	A	A	-	A	A	-	-
Q	A	A	A	A	A	-	A	A
Rho	-	A	A	A	A	A	-	A
D	A	A	A	A	A	-	-	A
DF	-	-	-	-	-	-	-	-

Table 3.8: Comparison of FS-based pruning (Algorithm 3) using ensemble accuracy vs Algorithm 4 using each diversity measuree at the first stage and ensemble accuracy at the second stage, for a validation set size equal to 1/3 and 1/6 of the training set size. Base classifier: MLP-NN. See caption of Table 3.3 for the meaning of table entries.

Table 3.9: The correlation value between each pair of diversity measures.

Diversity	ρ	Dis	DF	KW	κ	E	θ	GD	CFD
Q	0.9945	-0.9840	0.5578	-0.9840	-0.9840	0.9943	0.9352	-0.8210	-0.8396
ρ		-0.9710	0.5491	-0.9710	-0.9710	0.9998	0.9546	-0.8256	-0.8463
Dis			-0.5648	1.000	1.000	-0.9713	-0.8619	0.7978	0.8258
DF				-0.5648	-0.5648	0.5490	0.4922	-0.8879	-0.8951
KW					1.000	-0.9713	-0.8619	0.7978	0.8258
κ						-0.9713	-0.8619	0.7978	0.8258
E							0.9548	-0.8257	-0.8462
θ								-0.7970	-0.8002
GD									0.9927

effect of over-fitting when a small set of (labelled) validation samples is available.

3.4 Approach to Study Diversity Measures

In our previous work we used a set a set of measures as a decision functions in ensemble pruning. In order to limit the number of diversity measures to be used we could find those that are equivalent and obtain a reduced set of measures. For the sake of this purpose we studied diversity measures theoretically and experimentally. After taking note of some research papers in this state of the art, specially in the paper [11] , there is a direct pairwise comparison between each pair of diversity measures, this comparison shows directly how is the similarity between each pair of measures as shown in the figure 3.1. We also performed some experiments to hold a comparison between each pair of measures. We used a randomly generated binary classifiers with 1000 pattern in each run and we compute the diversity value between the classifiers for the ten diversity measures. After repeating this experiment with different tuned agreement and disagreement between the classifiers in both true and false classification. In each run with a different adjusted classifier votes we compute the correlation between each pair of measures and consider the mean and standard deviation as shown in the table 3.9.

For the sake of reduce the set of diversity measures, we set a threshold for the correlation value to select the least correlated measures, the figure 3.1 shows the study results made by [11], meanwhile the table 3.10 shows the selected examples from our study of the diversity measures representing the correlation value between each pair of diversity measures and a scatter plot that represents the correlation relation between them. Obviously there is agreement between our results and results presented by [11].

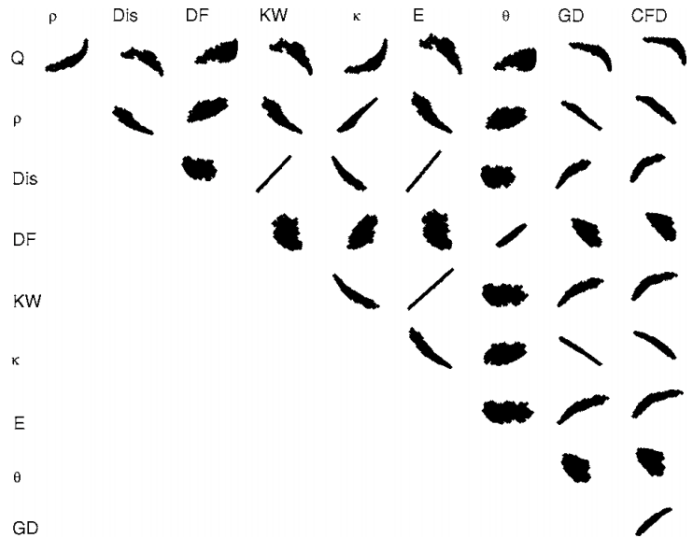
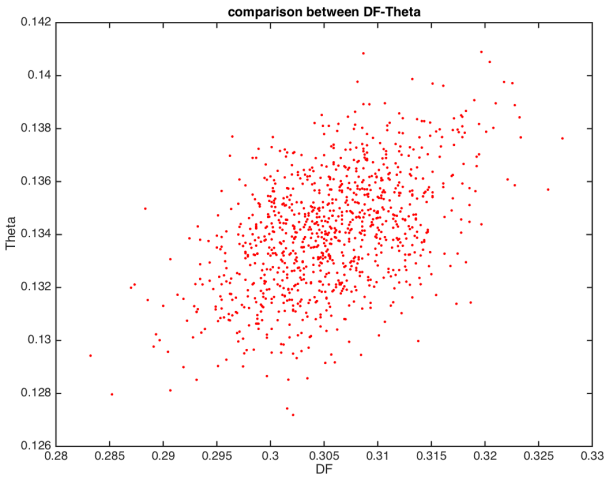
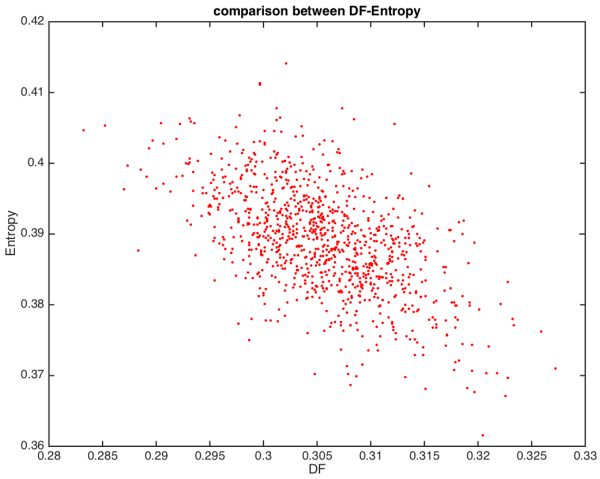
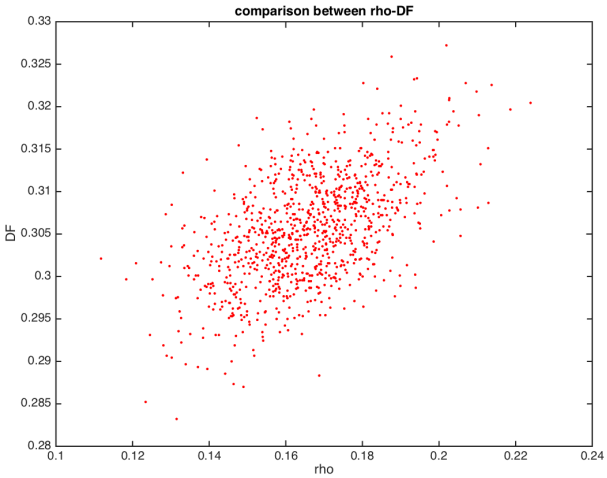
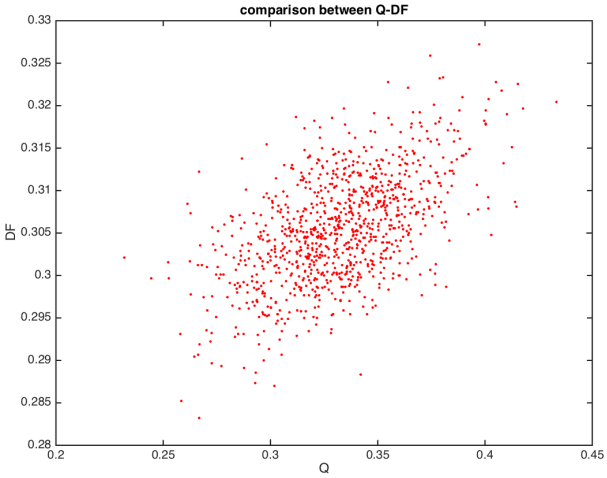
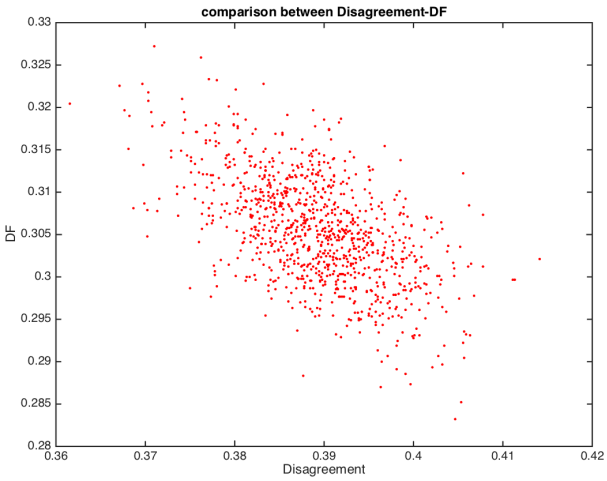
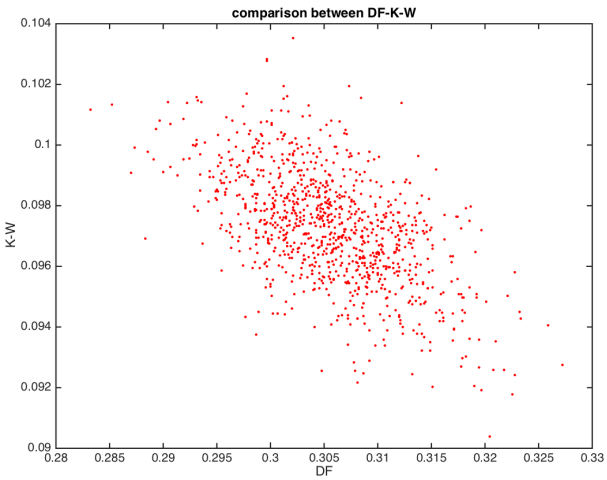
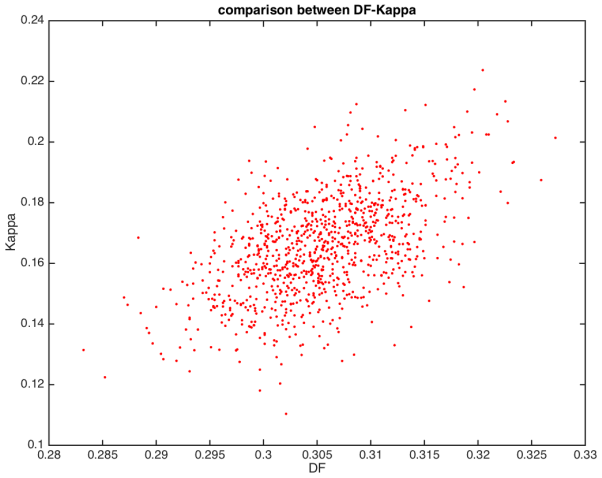
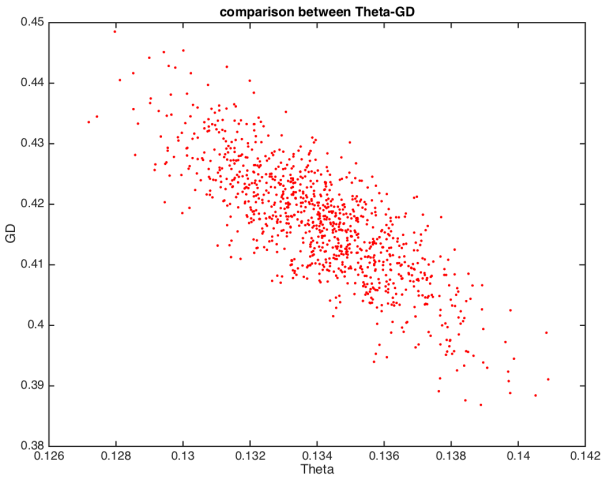
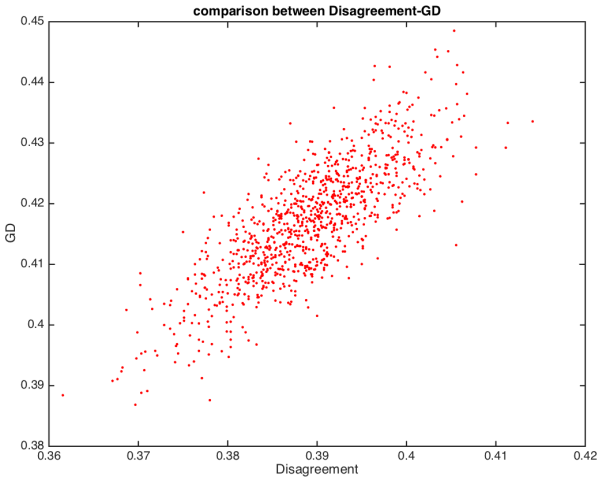


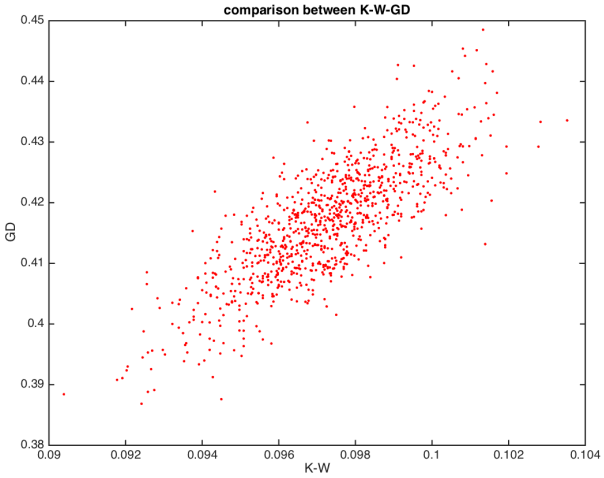
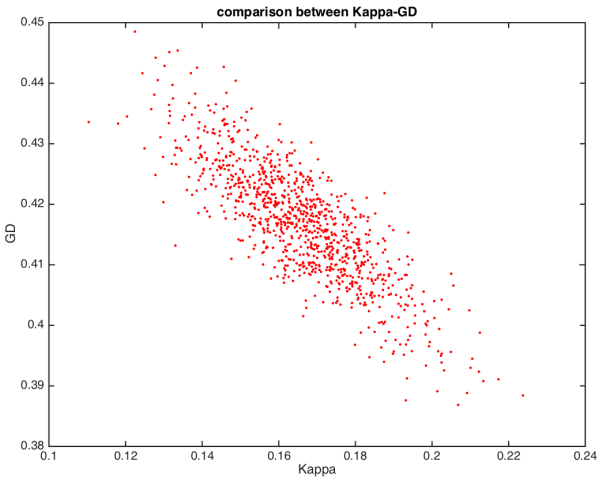
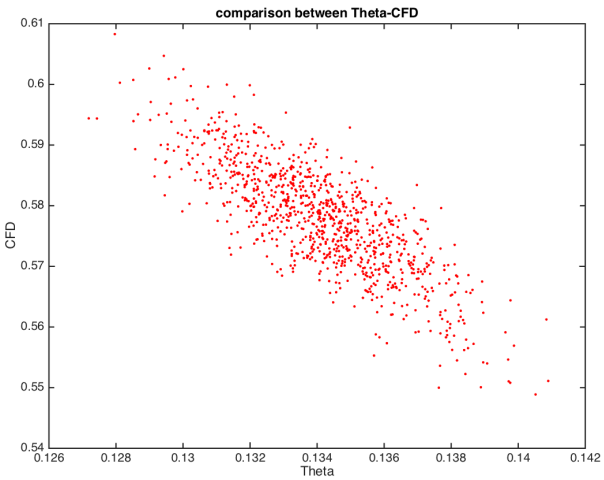
Figure 3.1: Pairwise scatterplots of 10 diversity measures.

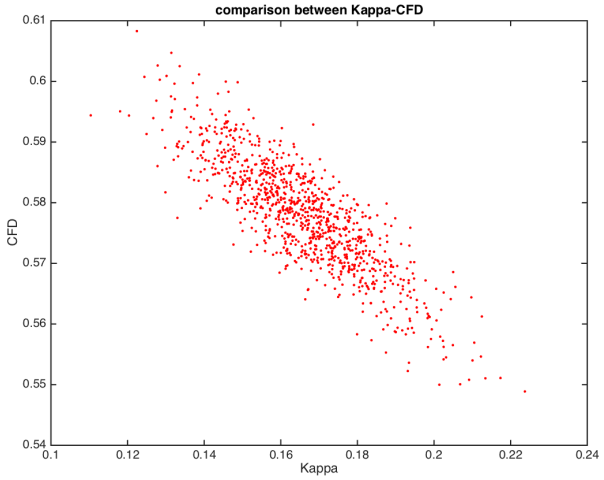
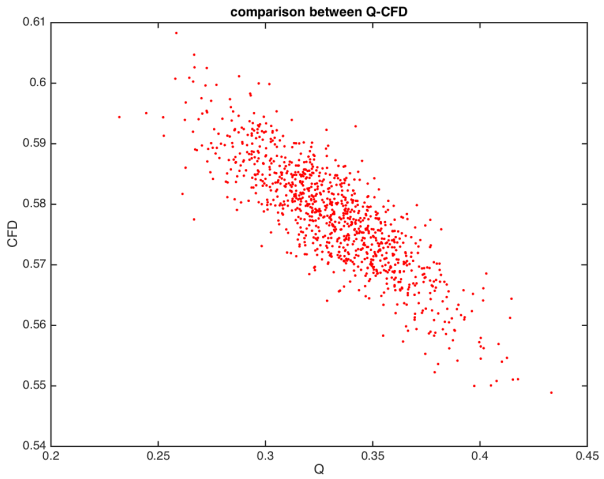
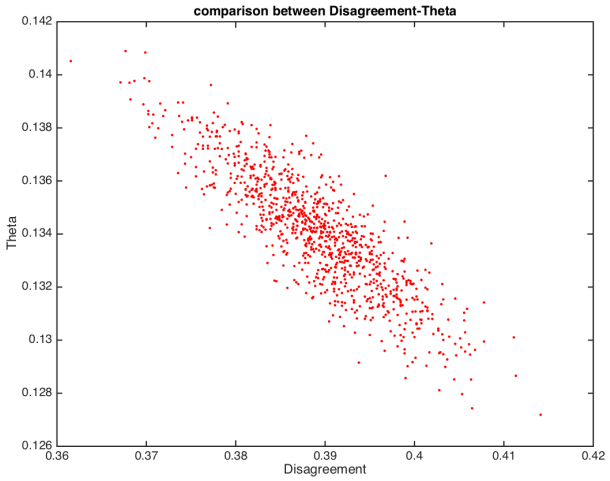
Table 3.10: The selected pairwise correlation between diversity measures.

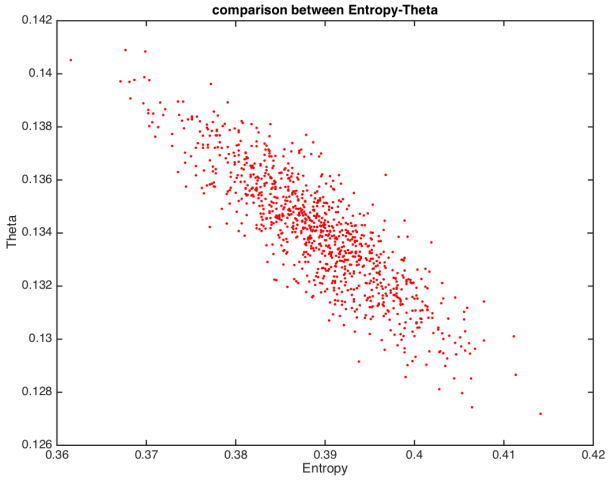
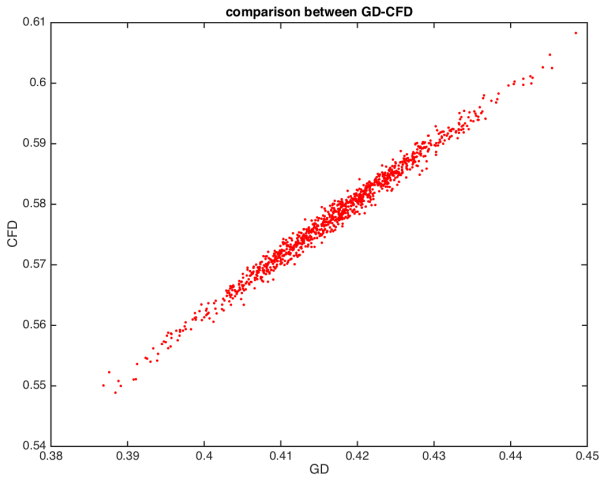
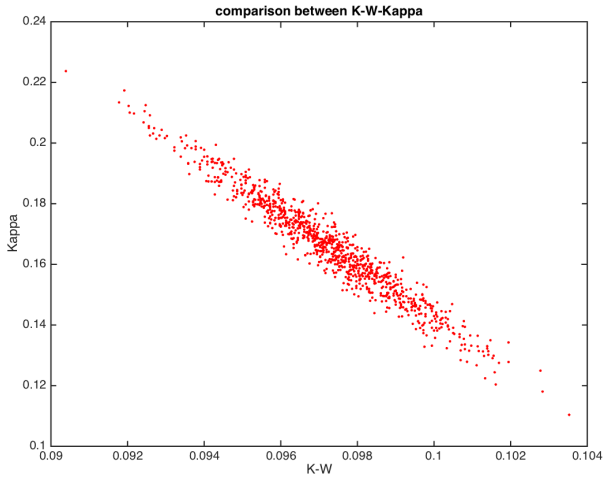
Measures	Correlation	Figure
DF-θ	0.4922	
DF-Entropy	0.5490	
ρ-DF	0.5491	

Q-DF	0.5578	
Dis-DF	0.5648	
DF-KW	0.5648	

DF-κ	0.5648	
θ-GD	0.7970	
Dis-GD	0.7978	

KW-GD	0.7978	
κ-GD	0.7978	
θ-CFD	0.8002	

κ-CFD	0.8258	
Q-CFD	0.8396	
Dis-θ	0.8619	

E-θ	0.9548	
GD-CFD	0.9927	
KW-κ	1.000	

Considering those pairs of measures that satisfies our threshold value of 0.8, One solution is ranking the measures according to their non similarity to other measures we have the following relationship: DF is the most uncorrelated measure with 7 measures, then GD with 4 measures, then theta with 3 measures, kappa, KW with 2 measures and finally Entropy,

CFD and Q with 1 measure. Due to considering the use of unlabeled data we must include Entropy and Disagreement. So Our subset of measures will be: Disagreement, Entropy, DF, GD and Theta as basic optionality, or adding Kappa, KW, CFD and Q as secondary considered measures.

Also we can select only one of the measures that use unlabeled data (Entropy or Disagreement) and after we rank the measures according to their non-similarity with this selected measures. Entropy is our first choice so we can find that the most uncorrelated measures to Entropy are DF, GD, CFD. DF is the most un correlated measure with 7 measures so we must include it. GD is in general non correlated with 4 other measures while entropy is not one of them so GD is the best second candidate. CFD is almost 0.8 correlated to Entropy and in general is correlated to most measures so according to our selection conditions we discard it. Theta in general is non correlated with three measures 2 of them are DF and GD that we include and one is CFD that we discard. So one option is Entropy, DF, GD and theta. If we choose to use Disagreement we will have the same result because Entropy and Disagreement are mostly correlated.

3.5 Approach to Combine Accuracy and Diversity

Many existing ensemble pruning methods use heuristic evaluation functions that combine the performance of individual classifiers and some measure of their diversity. It is then interesting to understand whether and under what conditions such evaluation functions are more effective (in terms of the performance of the resulting ensemble) than directly evaluating the performance of the considered ensembles (estimated, e.g., from validation data) during the pruning procedure. Quite surprisingly, so far such a comparison has been carried out by only a few authors [17, 187, 188, 151, 5], and only with a limited scope. In particular, it was often limited to the proposed evaluation measure, and using different and incomparable experimental set-up (i.e., different data sets, base classifiers, ensemble construction methods, etc.). We also point out that, among these works, only in [151, 5] the use of the proposed evaluation functions provided a statistically significant improvement over a direct estimation of ensemble performance.

To sum up, so far no clear evidence has been provided about the effectiveness of using diversity measures for ensemble pruning. A notable exception is the work of [5], where an original view of the role of diversity as a regularizer in ensemble design was proposed and theoretically investigated, in the case of binary classifiers combined by majority voting, and with a specific diversity measure. Their theoretical results showed that promoting diversity during ensemble design can actually have a regularization effect. Based on these results, a specific ensemble pruning method was then proposed in [5].

Based on the above premises, and inspired by [5], the aim of this work is to investigate whether also existing diversity measures can have a regularization effect in ensemble pruning, with respect to the (estimate of) ensemble accuracy. More precisely, we consider two evaluation functions: ensemble accuracy A alone, and its linear combination with a given diversity measure D, given by $A + \lambda D$ (with $\lambda > 0$), which is the usual form of regularization terms.

To carry out a direct comparison between such evaluation functions we consider a pruning method based on the forward selection (FS) algorithm. We first build an ensemble of N classifiers using a given ensemble construction technique, then we use FS to obtain a subset

of $L < N$ classifiers, for a given L. We consider the basic version of FS: it starts with the best (estimated) individual classifier of the original ensemble, then it iteratively selects from the remaining classifiers the one that provides the best evaluation function (either A or $A + \lambda D$) on the new candidate ensemble. The pseudo code is shown in Algorithm 3.

3.6 Experimental (2) Setting

The aim of our experiments is to compare two ensemble evaluation functions for ensemble pruning, using the basic FS pruning strategy described in Algorithm 3: the ensemble performance, evaluated as the classification accuracy A estimated from validation data, and its linear combination with a given diversity measure D evaluated on the same validation set, $A + \lambda D$, with $\lambda > 0$.

To this aim we create an initial ensemble E composed of $N = 100$ classifiers, and prune it to an ensemble of L classifiers, with $L = 5, 15, 25, 35$, using the FS algorithm. We used Bagging to obtain E, as it is a well-known ensemble creation technique, and has already been used to this aim for ensemble pruning, e.g. [16, 202]. We used majority voting as the combining rule, since it is the standard choice for Bagging [135].

In our experiments we used three different base classifiers: Multi-Layer Perceptron Neural Networks (NN), Decision Trees (DT) and K-Nearest Neighbors (K-NN). We used their standard Matlab implementation (Neural Networks and Statistics and Machine Learning Toolboxes). In particular, for NNs we used the `patternnet` function with a learning rate $\eta = 0.05$, gradient descent with momentum as the learning algorithm, and a maximum of 1000 epochs as a stop criterion. For DT we used the $Gini$ impurity criterion, the χ^2 stopping criterion, and the default threshold equal to 1 for the pre-pruning stopping criterion. For K-NN we used $K = 1$.

In the evaluation function $A + \lambda D$ we used several values of λ: 0.2, 0.5, and 0.7. We also considered the four diversity: DF, θ, Dis and GD.

We carried out our experiments on 37 benchmark data sets from the UCI Machine Learning Repository Database,[4] containing only numerical attributes and no missing values (see Table 3.11). They represent a remarkable range of classification problems: the number of patterns ranges from 160 to 10992, the number of classes from 2 to 10, and feature set size from 2 to 85. We randomly subdivided each data set, using stratified sampling, into a training set, a validation set and a test set. The size of the training set is defined as explained in Sec. 3.7. The size of the validation set was chosen as 1/3 of the training set, and the remaining instances were used as the testing set. We repeated this procedure for 20 runs, and evaluated the resulting average accuracy on testing samples.

3.7 Choice of the Training Set Size

For each data set we chose the training set size that maximizes the (estimated) difference between the highest and lowest accuracy attained by different ensembles of a given size L. The rationale is that, if all ensembles of L classifiers obtained from the initial ensemble E exhibit a similar accuracy, it becomes difficult to evaluate the difference (if any) between

[4] http://www.ics.uci.edu/~mlearn/MLRepository.html

Table 3.11: Characteristics of the data sets.

Dataset	Classes	Instances	Features
Bank Note	2	1372	4
Banana	2	5300	2
Blood Transfusion	2	748	4
Cardiotocography	3	2126	22
Pop Failures	2	540	20
SatLogLandSetSat	6	6435	36
SataLogImageSeg	7	2310	19
Spam Base	2	4601	57
Thyroid	3	7200	21
Wine Quality	7	4898	11
Australian	2	690	14
Balance Scale	3	625	4
Bands	2	365	19
Breast Cancer	2	699	9
Bupa	2	345	6
Checker Board	2	1000	2
Cleveland	5	297	13
Coil2000	2	1286	85
Contours	3	2000	2
Contraceptive	3	1473	9
Dermatology	6	358	34
Hayes Roth	3	160	4
ILPD	2	583	9
Laryngeal 2	2	692	16
Marketing	9	6876	13
Monk 2	2	432	6
Page Plocks	5	5473	10
Pen based	10	10992	16
Phoneme	2	3186	5
Pima	2	768	8
Ring	2	7400	20
Saheart	2	462	4
Segment	7	2310	19
Spectfheart	2	267	44
Vehicle	4	846	18
WDBC	2	569	30
Yeast	10	1484	8

different pruning methods (in our case, different evaluation functions used in the same priuning method). Fig. 3.2 illustrates the idea.

To this aim we carried out preliminary esperiments, considering training sets sizes ranging from 1% to 70% of the whole data set. For NNs, we also considered different numbers of hidden units, between 3 and 20. Since considering different ensemble sizes L is computa-

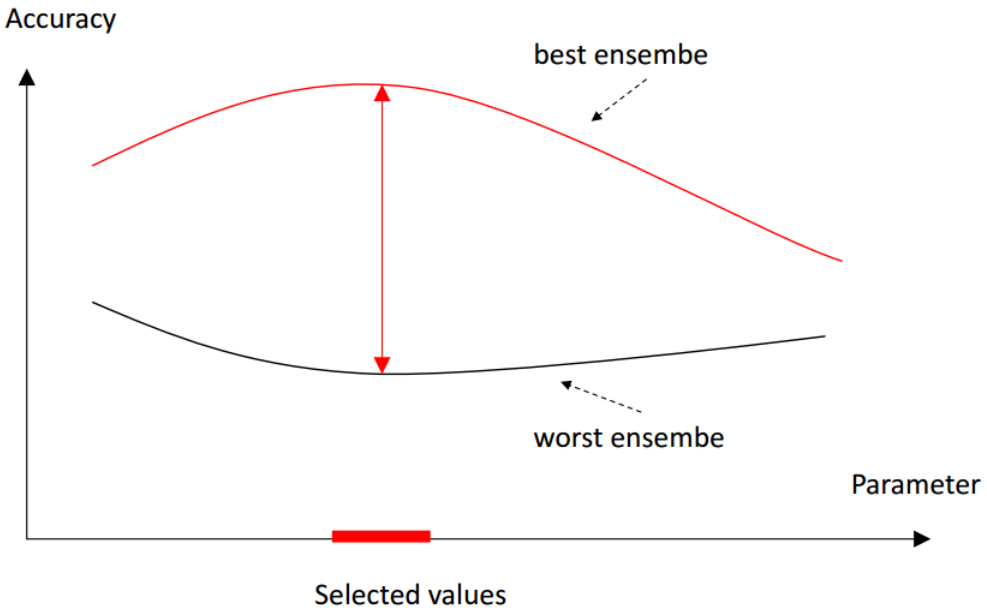


Figure 3.2: Qualitative illustration of the criterion used for choosing the training set size and the number of hidden units in NN classifiers (X axis): maximizing the accuracy gap between the best and the worst ensemble of a given size (see text for the details).

tionally costly, and obviously considering all possible subsets of size L of a given ensemble is infeasible, we only considered ensembles of size $L = \frac{N}{2} = 50$, and estimated the performance of the best and worst such ensembles with the ones of ensembles made up of the L best and by the L worst individual classifiers. The resulting training set sizes used in the rest of our experiments are shown in Table 3.12. For NNs also the number of hidden units is shown.

3.8 Statistical Evaluation of the Results

To compare the two considered ensemble pruning evaluation functions we carried out a test of statistical significance between the corresponding average testing set accuracy over the different runs of our experiments. To this aim we chose the Wilcoxon signed-rank test, as it is recommended in [203] for comparing two algorithms over multiple data sets, which is the setting considered in our experiments. This is a non-parametric statistical hypothesis test that can be used to determine whether two dependent samples were drawn from populations having the same distribution. This test is used to evaluate the statistical significance of the obtained results, i.e., whether it is possible to reject the null hypothesis that the observed values – in our case, the accuracies obtained by different ensembles – are different only by chance.

3.9 Experimental (2) Results

For each pruned ensemble size L, base classifier, diversity measure and value of λ, Table 3.13 shows the results of our experiments in terms of the statistical significance of the difference

in testing set accuracy of the FS pruning method implemented using the two considered evaluation functions. More precisely, the null hypothesis is that there is no difference between these evaluation functions. In Table 3.13 entries marked with 'A' mean that for the corresponding pruned ensemble size, base classifier, diversity measure and value of λ, using only ensemble accuracy (estimated from validation data) as the evaluation function is significantly better (according to Wilcoxon signed-rank test) than using its linear combination with the diversity measure. Entries marked with 'D' mean the opposite (the latter evaluation function is significantly better than the former). We point out that the null hypothesis has always been rejected; therefore, every entry of Table 3.13 is marked with either 'A' or 'D'.

These results provide a quite strong evidence that a linear combination of ensemble accuracy and of a diversity measure between ensemble members outperforms the use of ensemble accuracy alone as the pruning evaluation function, to a statistically significant extent. The table 3.13 clearly shows that using $A + \lambda D$ as the evaluation function in the FS algorithm provides a statistically significantly better pruned ensembles than using accuracy alone, in almost all the considered cases. The only exceptions can be observed for the largest considered ensembles ($L = 35$) of DT classifiers, when DF and θ were used as diversity measures, and the λ coefficient was 0.2 and 0.5; and for ensembles of various sizes of NN classifiers, when the other two diversity measures (Dis and GD) were used and the λ coefficient was 0.5 and 0.7. It is also worth noting that the $A + \lambda D$ evaluation function always outperfomed its counterpart A for ensembles of K-NN classifiers, and with the only exception of the largest ensembles ($L = 35$) for the DT classifier. With regard to the diversity measures, using DF, θ and GD in the $A + \lambda D$ evaluation function turned out to be worse than using A alone only for 2 out of the 108 combinations of pruned ensemble size, base classifier and value of λ (3 diversity measures, 4 ensemble sizes, 3 base classifiers and 3 values of λ); using Dis, this happened for 4 out of the 36 combinations.

As far as our experiments are concerned, we can conclude that well-known, "generic" ensemble diversity measures (i.e., not specifically devised for ensemble pruning) seem to be useful when used together with ensemble accuracy as the pruning evaluation function. In particular, such diversity measures seem to act as regularizers of the estimated ensemble accuracy, which is in agreement with the more specific results of [5].

3.10 Combining Accuracy and Diversity Using Unlabeled data

The above experiments results provided an evidence that diversity have a regular behaviour when combined with ensemble accuracy as an objective function of the FS ensemble pruning method. At least for diversity measures those use the instance true label as DF, Theta and GD, thus it outperforms the ensemble accuracy when used alone as an objective function in same algorithm. The diversity measures that don't use the pattern true label such as Disagreement and Entropy, when combined with accuracy as an objective function shows a regular behavior to outperforms ensemble accuracy for value of $\lambda = 0.2$. This behaviour of the diversity measures that don't use the instance true label motivated us to extend our experiments in order to investigate the rule of those diversity measures in the case of small λ. In this section of experiments, we investigate the case of having a few labelled instances that can be exploit by the objective functions that require the instance true label such as the

ensemble accuracy and the three diversity measures DF, Theta and GD, meanwhile growing up the amount of unlabeled data that can be exploit by Disagreement and Entropy diversity measures. The aim of this experiments is to investigate the behaviour of diversity measures that don't use the instance true label while growing the amount of instances that this two diversity measures can exploit.

3.11 Experimental (3) Settings

Real datasets: In order to carry out this section of experiments, we picked a 10 large size real datasets from those already reported in table 3.11. The chosen datasets have a large enough number of instances which after division intro training/validation/testing. We focus on the size of the validation set hence it aims to satisfy the growing up rate of the unlabeled instances portion of the validation set. The number of classes in the selected datasets varies from 2 up to 9 class.
Synthetic datasets: We use 10 synthetic datasets were generated as a 2 class problems with 10k instances. Each class is a Gaussian mixture in $[0,1]^{12}$, with equal probability and a covariance matrix that is proportional to the identity matrix $\sum = \sigma^2 I$.

For both synthetic and large size datasets. The training set size is determined using the initial setting algorithm. The full validation set size is fixed to satisfy the growing of unlabeled portion size requirement. The rest of the dataset was used as a test set. In this experiments we investigate the ensemble pruning FS algorithm using an objective function m()= Ensemble Accuracy+$\lambda\cdot$ Diversity, $\lambda = 0.2$. This objective function is applied on a fixed size labelled validation set plus a growing amount of unlabeled data. The diversity measures that use labels (DF, Theta and GD) are applied only to the small labelled validation set. The Diversity measures that don't use labels (Disagreement and Entropy) can exploit all the patterns. The fixed labelled portion of the validation set has a 10 patterns per class. The unlabeled portion of the validation set grows as 10, 25, 40, 75 and 100 patterns per class.

3.12 Initial Settings Estimation

Applying same initial settings estimation algorithm presented in section 3.7 to the datasets, we obtain results used in this section of experiments.To evaluate this section of experiments, we proceed with the selected parameters using the initial settings algorithm. The first stage of this experiments we use a validation set with 10 labeled instances that can be exploit by ensemble accuracy alone or the proposed linear combination of ensemble accuracy and diversity measure with a combination coefficient $\lambda = 0.2$. In the second stage we grow up the unlabeled portion of the validation set in order to investigate the advantage that the diversity measures that can exploit the unlabeled portion of the validation set.

3.13 Experimental (3) Results and Conclusion

We use an objective function described as a linear combination of ensemble accuracy plus weighted diversity in the FS ensemble pruning technique. The selected final pruned ensemble members using our objective function aimed to be accurate and diverse. The results analysis presented in this paper shows the fact how is our results statistically significantly

better than using the ensemble accuracy. It is believed that diverse ensembles outperform non-diverse ones this is why our results fit to the state of the art. It combines the benchmark principle of the ensemble accuracy objective function ability to select accurate final pruned ensemble members, with the role of diversity measures to produce diverse final pruned ensemble. The DF, Theta and GD measures when combined with ensemble accuracy outperform the ensemble accuracy alone. The diversity measures that don't use the pattern true label show a better performance when weighted with accuracy with a small value. We used only three values of lambda to weightily combine diversity with diversity, even the results obtained are hopeful it is possible to explore different values of lambda. The experiments using different size divisions of unlabeled validation set shows good results with measures use the true label but not with the measures don't use the label. Meanwhile the diversity measures that don't use the label still the only solution when we have no label. It is recommended to explore different unlabeled validation set size even different values of the diversity accuracy combination parameter values.

Table 3.12: For each data set, the number of hidden units for the NN base classifiers (second column) and the training set size for the three base classifiers (NNs, DTs and k-NNs) is shown.

Dataset	**hidden units**	**NN**	**DT**	K**-NN**
Bank Note	12	0.1	0.6	0.6
Banana	3	0.7	0.7	0.7
Blood Transfusion	3	0.5	0.4	0.1
Cardiotocography	7	0.1	0.6	0.2
Pop Failures	3	0.5	0.6	0.6
SatLogLandSetSat	12	0.6	0.5	0.1
SataLogImageSeg	20	0.6	0.5	0.1
Spam Base	3	0.4	0.4	0.4
Thyroid	3	0.1	0.3	0.3
Wine Quality	7	0.4	0.6	0.5
Australian	12	0.4	0.5	0.5
Balance Scale	12	0.5	0.6	0.2
Bands	3	0.1	0.6	0.3
Breast Cancer	20	0.4	0.6	0.2
Bupa	12	0.4	0.5	0.6
Checker Board	12	0.6	0.6	0.1
Cleveland	7	0.6	0.5	0.6
Coil2000	3	0.1	0.6	0.6
Contours	20	0.5	0.6	0.3
Contraceptive	3	0.6	0.6	0.6
Dermatology	7	0.4	0.3	0.3
Hayes Roth	12	0.6	0.4	0.6
ILPD	3	0.1	0.5	0.1
aryngeal 2	3	0.2	0.5	0.1
Marketing	7	0.6	0.6	0.6
Monk 2	12	0.6	0.5	0.2
Page Plocks	7	0.4	0.5	0.6
Pen based	8	0.7	0.3	0.7
Phoneme	7	0.1	0.6	0.6
Pima	12	0.6	0.6	0.1
Ring	20	0.5	0.5	0.6
Saheart	12	0.4	0.4	0.6
Segment	20	0.5	0.6	0.3
Spectfheart	20	0.2	0.4	0.6
Vehicle	12	0.6	0.5	0.6
WDBC	3	0.6	0.3	0.1
Yeast	7	0.3	0.6	0.1

Table 3.13: Outcome of the statistical significance test for the comparison between the use of the evaluation functions A and $A+\lambda D$ (see text) for ensemble pruning, for several ensemble sizes L, values of λ, base classifiers and diversity measures. 'A' means that the evaluation function A is statistically significantly better than $A+\lambda D$, 'D' means the opposite (see text for the details).

		L=5			L=15			L=25			L=35		
Base classifier	**Diversity**	λ 0.2	0.5	0.7	λ 0.2	0.5	0.7	λ 0.2	0.5	0.7	λ 0.2	0.5	0.7
DT	DF	D	D	D	D	D	D	D	D	D	**A**	**A**	D
	Theta	D	D	D	D	D	D	D	D	D	**A**	**A**	D
	DIS	D	D	D	D	D	D	D	D	D	D	D	D
	GD	D	D	D	D	D	D	D	D	D	D	D	D
KNN	DF	D	D	D	D	D	D	D	D	D	D	D	D
	Theta	D	D	D	D	D	D	D	D	D	D	D	D
	DIS	D	D	D	D	D	D	D	D	D	D	D	D
	GD	D	D	D	D	D	D	D	D	D	D	D	D
NN	DF	D	D	D	D	D	D	D	D	D	D	D	D
	Theta	D	D	D	D	D	D	D	D	D	D	D	D
	DIS	D	**A**	**A**	D	D	**A**	D	D	**A**	D	D	D
	GD	D	**A**	**A**	D	D	D	D	D	D	D	D	D

Table 3.14: Comparison of FS-based pruning presented in section 3.1 shows a comparison of FS-based pruning using ensemble accuracy vs. using each diversity measure combined with the ensemble accuracy via a parameter λ, for different ensemble sizes L. Base classifier: DT using 10 per class labelled patterns and same patterns used as unlabeled. See caption of table 3.13 for the meaning of table entries.

Diversity	Real Datasets				Synthetic Datasets			
	L=5	L=15	L=25	L=35	L=5	L=15	L=25	L=35
DF	D	D	D	D	-	A	A	A
Theta	D	D	D	D	-	A	A	A
Dis*	-	-	-	-	-	-	-	-
Entropy*	-	-	-	-	-	-	-	-
GD	D	A	D	A	D	D	D	A

Table 3.15: Comparison of FS-based pruning presented in section 3.1 shows a comparison of FS-based pruning using ensemble accuracy vs. using each diversity measure combined with the ensemble accuracy via a parameter λ, for different ensemble sizes L. Base classifier: K-Nearest Neighbor,k=1; using 10 per class labelled patterns and same patterns used as unlabeled. See caption of table 3.13 for the meaning of table entries.

Diversity	Real Datasets				Synthetic Datasets			
	L=5	L=15	L=25	L=35	L=5	L=15	L=25	L=35
DF	A	A	D	D	A	D	A	D
Theta	A	A	D	D	A	D	A	D
Dis*	-	-	-	-	-	-	-	-
Entropy*	-	-	-	-	-	-	-	-
GD	A	A	D	D	A	D	D	A

Table 3.16: Comparison of FS-based pruning presented in section 3.1 shows a comparison of FS-based pruning using ensemble accuracy vs. using each diversity measure combined with the ensemble accuracy via a parameter λ, for different ensemble sizes L. Base classifier: Neural Networks; using 10 per class labelled patterns and same patterns used as unlabeled. See caption of table 3.13 for the meaning of table entries.

Diversity	Real Datasets				Synthetic Datasets			
	L=5	L=15	L=25	L=35	L=5	L=15	L=25	L=35
DF	D	D	A	A	A	A	A	A
Theta	D	D	A	A	A	A	A	A
Dis*	-	-	-	-	-	-	-	-
Entropy*	-	-	-	-	-	-	-	-
GD	D	D	A	D	D	D	D	A

Chapter 4

Trained Neural Networks Ensemble Weight Analysis

Bellid *et. al.* [204], due to the lack of data and mathematical approaches to describe the inside of NN have to resort the assumption that the weight connection of a trained neural network to be like a Normal distribution. They presented an extensive empirical study of weight distribution in a back-propagation NN and test formally if the weight of trained NN has indeed a normal distribution. Even they considered a very small invalid probability of rejection of 0.005, the majority of weight distributions investigated were described as NOT Normal. In case of using a simple NN model with no hidden layers, the neural network weight distribution passes the normality test with more than 90% in the case of Gene Promotion dataset. Barbour *et. al.* [205], presented a review of theoretical and experimental techniques for analysing the distribution of synaptic weights. Comparing different approaches to analyze the distribution of synaptic weights, Barbour clearly described the obtained distributions from different approaches, as all have a similar shape. They summarise that theoretical analysis through optimality principles show various features of the weight distributions. One of the amazing approaches they highlighted is "Obtaining weight distributions from optimality principles". Considering a Perceptron with 1 binary neuron, we aim the Perceptron to learn N random input-output associations (input patterns) by modifying the weight connections. Considering that the Perceptron has a large number of weight connections $\overrightarrow{W}$, Gardner *et. al.* [206] considered $\overrightarrow{W}$-dimensional space representing all possible configurations of $\overrightarrow{W}$. Only a weight vector subspace of $\overrightarrow{W}$ will satisfy all input-output association. As the number N increases, the subspace of $\overrightarrow{W}$ that satisfy all N input-output associations decreases. They recommend estimating the distribution of weights of NN model below or at maximum capacity. Brunel *et. al.* [207] presented an interesting study to compare the Perceptron and purkinje cell from optimal capacity and the weight connections distribution. Summary of their comparison is that below maximum capacity, the non-negative weight distribution has 2 components, about 50% at least are zero weights meanwhile Gaussian distribution found to be fit to the positive connections.

4.1 Problem Overview

Multiple Classifier Systems (MCSs) is a simple system which provides a promising outcome for most of the machine learning problems; hence it train many different models on same data and consider the average of their predictions [15]. The creation of MCS is computationally expensive, however, because it requires the training of multiple learners. Snapshot Ensembles 2017 [19] recently presented to create ensemble of NN with no additional training cost. They exploit local minima of the error function; Producing N different NN (connection weights) by running the learning algorithm only once, instead of running it for N different times starting from different initial weights. Delphine *et. al.* [208] proposed a mathematical framework presenting a new alternative formalism for Training A special class of NN called Spiking NN. They train 1 SNN on a given input/signal I and stop. After training they consider the obtained weight distribution; corresponding to the used training signal/input W_I; for New J_i; $i = 1 : n$; Inputs/signals/patterns they formalized The convergence of the found weight W_I to the desired W_{J_i} without rerunning the training algorithm. Santucci *et. al.* [21] proposed a new approach for defining randomization techniques, inspired by the fact that existing ones can be seen as implicitly inducing a probability distribution on the parameters of a base classifier. Accordingly, that new randomization techniques can be obtained by directly defining a suitable parameter distribution for a given classifier, as a function of the training set at hand. An ensemble can therefore be built by directly sampling the parameter values of its members from such a distribution, without actually manipulating the available training data nor running the learning algorithm. In this way, an ensemble can be obtained even without having access to the training set but having access only to a pre-trained classifier. The constructed ensemble is built using a simulation of bagging. The proposed simulated ensemble achieved a classification performance very close to bagging when NMC used as a base classifier. For the base classifier LDC there is only a partial agreement between the proposed randomization technique and the original bagging. In the case of QDC, the difference in performance was not significant due to the p-value of the statistical test, which is used to compare the proposed randomization approach and the original bagging. Authors clearly highlighted that in the case of non-parametric classifiers such as neural networks, the number of parameters (weight connections) can be very high and at the same time they cannot be related to statistics of the data. MAO *et. al.* [1] presented the effectiveness and use of various diversity measures to construct ensembles of different base classifiers. Most of the contributions aim to improve, accelerate and make easier the concepts related to machine learning strategies. In this work we produce a detailed analysis to the weight connections of a trained ensemble of 1000 NN created and trained using bagging. The main aim of this work, is to explore the distribution of weight connections aiming to investigate highlight these unexplored informations about the nature of trained NN weight connections. This is the first step towards reaching a learning free ensemble of NN's; via estimating the correct distribution of trained weight connections what leads to not train the ensemble anymore but directly creating it via withdrawing the correct weight values from those found from the distributions. These theories can be applied to both the feed-forward and recurrent networks. Experimental testing of the link between optimal learning and the distribution of synaptic weights Analysis of synaptic weight distributions can test learning theories and offer access to difficult-to-obtain information, such as the storage capacity of a neuron. However, these analyses would obviously be strengthened by a direct demonstration that the distribution shape was indeed linked to (optimal) learning. The most promising approach would be to

compare distributions when different quantities of information have been stored. One obvious idea is to compare distributions from immature and mature animals, although this might be confounded by concurrent developmental processes. Another possibility would be to compare distributions from animals raised in feature-poor and -enriched environments; presumably, the latter would have learned more. Finally, chronic pharmacological or genetic interventions might allow manipulation of specific model parameters (e.g. activity or noise levels) and testing of their expected effects on distribution shape.

4.2 Weight Connections Distribution Approximation

The aim is to find the best fit well-known statistical distributions mentioned in Section 4.2 to the weight connection values in a trained ensemble of NN's created using Bagging. The NN's initial state has a remarkable influence on the classifier performance. A reasonable choice is to randomly initialize the NN weights. Analyzing the NN's weight distributions is our goal. The proposed aim to approximate the best-fit well-known parametric distribution to weight connections values of a trained NN ensemble created using bagging. The Algorithm 5 describes the approach by mike; it attempts to fit the weight values to a list of continuous and discrete distributions, the following list shows **the considered distribution's names, notations and parameters**:

- **Beta** (β): α, β
- **Birnbaum-Saunders** (BS): γ, μ, β, ϕ .
- **Exponential** (Exp): λ
- **Extreme value** (Ev): a, b
- **Gamma** (Γ): α, β
- **Generalized Extreme value** (GEv) : s, ξ
- **Generalized Pareto** (Pareto) : α
- **Gaussian** (Gauss): μ, σ^2
- **Logistic** (Log): μ, s .
- **Log-logistic** (Llog): α, β
- **Lognormal**(logNorm): μ, σ
- **Nakagami** (Nakg): m, Ω
- **Normal**(N): μ, σ
- **Rayleigh** (Rlh) : σ
- **Rician** (Rc): ν, σ
- **T location-scale** (TLS): ν, μ, σ

- **Weibull** (Wb) : λ, k

The test returns **a list of valid distributions** sorted by:

- **NLogL**: Negative of the log likelihood.
- **BIC**: Bayesian information criterion (default).
- **AIC**: Akaike information criterion.
- **AICc**: AIC with a correction for finite sample sizes.

The $Likelihood\ \mathscr{L}(\theta|x)$ of a parameter value, θ (or vector of parameter values), given outcomes x, is equal to the probability (density) assumed for those observed outcomes given those parameter values. Let X be a random variable with a discrete probability distribution p depending on a parameter θ. Then the function:

$$\mathscr{L}(\theta|x) = p_\theta(x) = P_\theta(X = x), \tag{4.1}$$

considered as a function of θ, is called the $likelihood\ function$ (of θ, given the outcome x of the random variable X). The Negative of Log Likelihood :

$$NLogL = -log\mathscr{L}(\theta|x) = -log\,p_\theta(x) = -logP_\theta(X = x). \tag{4.2}$$

Let X be a random variable following an absolutely continuous probability distribution with density function f depending on a parameter Î¸. Then the function:

$$\mathscr{L}(\theta|x) = f_\theta(x), \tag{4.3}$$

considered as a function of θ, is called the $likelihood\ function$ (of Î¸, given the outcome x of X). The Negative of Log Likelihood (NLogL):

$$NLogL = -log\mathscr{L}(\theta|x) = -log\,f_\theta(x). \tag{4.4}$$

Suppose that we have a statistical model $\mathscr{M}$ of some data x. Let k be the number of estimated parameters in the model. Let $\hat{\mathscr{L}}$ be the maximized value of the likelihood function for the model; i.e.$\hat{\mathscr{L}} = P(x|\hat{\theta}, \mathscr{M})$ are the parameter values that maximize the likelihood function. The Akaike information criterion is defined in equation 4.5:

$$AIC = 2k - 2\ln(\hat{\mathscr{L}}). \tag{4.5}$$

The Akaike information criterion with a correction for finite sample sizes (AICc) is defined in equation 4.6:

$$AICc = -2\mathscr{L}(\theta|x) + 2k(k+1)/(n-k-1). \tag{4.6}$$

These values of AIC and AICc can be used to compare various models for the same data set to determine the best-fitting model. The model having the smallest value, as discussed in Akaike (1974) [209], is usually the preferred model. The BIC [210] was developed by Gideon E. Schwarz, who gave a Bayesian argument for adopting it. It is closely related to the Akaike information criterion (AIC). In fact, Akaike was so impressed with Schwarz's Bayesian formalism that he developed his own Bayesian formalism, now often referred to as the ABIC

Algorithm 5 Best-fit valid parametric distribution approximation.

Input: A data X.
Output: Best-Fit parametric probability distribution to data X.

For Every Distribution in Section 4.2; **do:**
Compute the distribution parameters (μ, σ ,etc).
Compute the **NLogL, AIC, AICc,BIC**.
End for
Sort Distributions ascending according to each of **NLogL, AIC, AICc,BIC**.
Return Best-fit Distribution with **NLogL, AIC, AICc,BIC** value.

for "a Bayesian Information Criterion" or more casually "Akaike's Bayesian Information Criterion". The corrected Akaike's Information Criterion (AICc) and the Bayesian Information Criterion (BIC) are information-based criteria that assess model fit. Both are based on Negative Log Likelihood. The Bayesian Information Criterion (BIC) is defined in equation 4.7:

$$BIC = -2\mathscr{L}(\theta|x) + k\ln(n). \tag{4.7}$$

When comparing the BIC values for two models, the model with the smaller BIC value is considered better. Burnham *et. al.* [211], clarify that that AIC can be derived from the BIC approximation to the Bayes factor. Due to the model selection literature, it is wrong to consider that AIC and BIC selection are directly comparable as if they had the same objective target model, but they are not.

4.3 Experimental Setup

In this this section, we describe the used approach to approximate the best-fit distribution to data X, as in Algorithm 5. Highlighting a selected example of our experiments performed on the well-known two class problems named **Breast Cancer** dataset, using only the first two features due to the simplicity of the used neural network structure in order to restrict our investigation on a fixed small number of weight connections. Starting with a simple structure multi-layer feedforward NN with 2 neurons in the input layer (only 2 features per instance used) 3 neurons in the hidden layer and 1 neuron in the output layer; considering the weight connections of 9 neurons per each NN classifier. Using bagging ensemble creation method, we create an ensemble E of size $E = 1000$ NN classifier. The weight connection vectors of each neuron of the trained 1000 NN classifiers; $\overrightarrow{W}_i = \{w_{iC_1}, w_{iC_2}, \ldots, w_{iC_N}\}$ where N=1000 is the number of ensemble members; where i is the number of neurons per NN classifier $i = [1,2,\ldots,9]$.

4.4 Results and Discussion

The used Breast Cancer dataset is divided into training set of size 0.4 a validation set and test set of size 0.3. We trained a 1k NN classifier on 1k bootstrap replica of the training set. A validation check is one of the stop criteria for training the NN classifier. The scored accuracy on the test set of the trained single classifier and the ensemble is reasonable. We analyze the

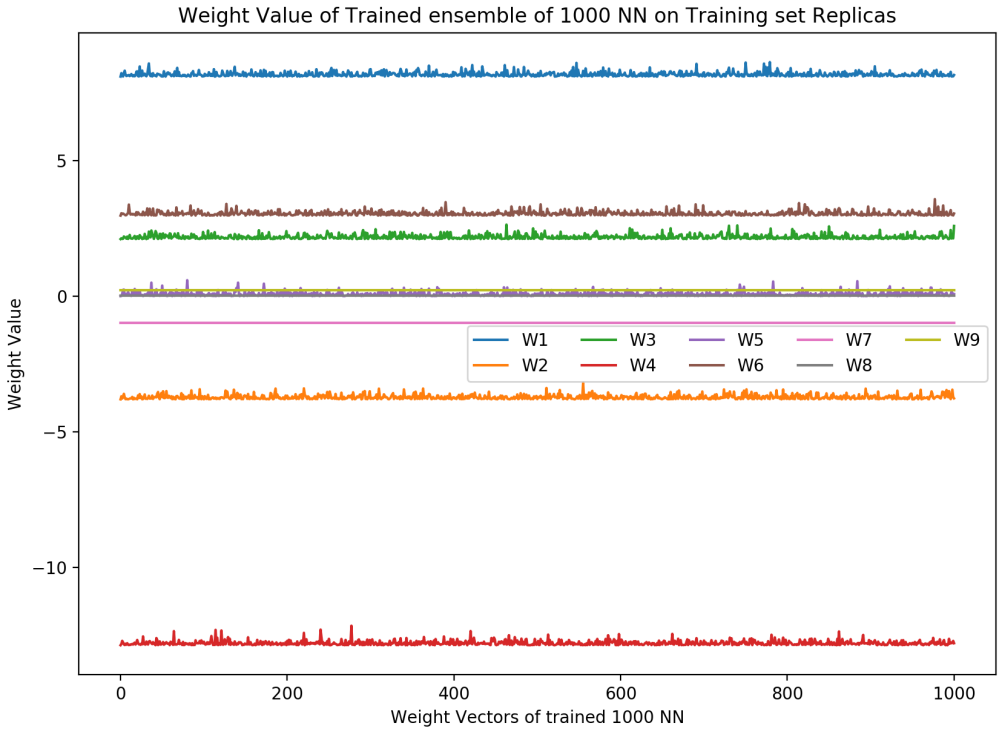


Figure 4.1: Weight vectors of trained 1000 NN classifier using Bootstrap sampling; an example of results analysis on Breast Cancer dataset.

single neuron weight values over the trained 1000 classifier created using bagging. Applying Algorithm 5 using NLogL equations 4.2, 4.4 as sort index to select the best-fit distribution, on the weight connections of a trained ensemble using the dataset breast cancer; the weight best-fit distribution results are shown in Tables 4.1,4.2. The Algorithm 5 decision is that all weight connections approximately fit the **t Location Scale** distribution. Generalizing this approach to artificial and real datasets, the summary of the datasets characteristics and the approximately best fit parametric distribution to the weight connections values are reported in Table4.3. The results conclusion is that the **t Location Scale** is approximately the best fit distribution to most of the considered weight connections of each single neuron.

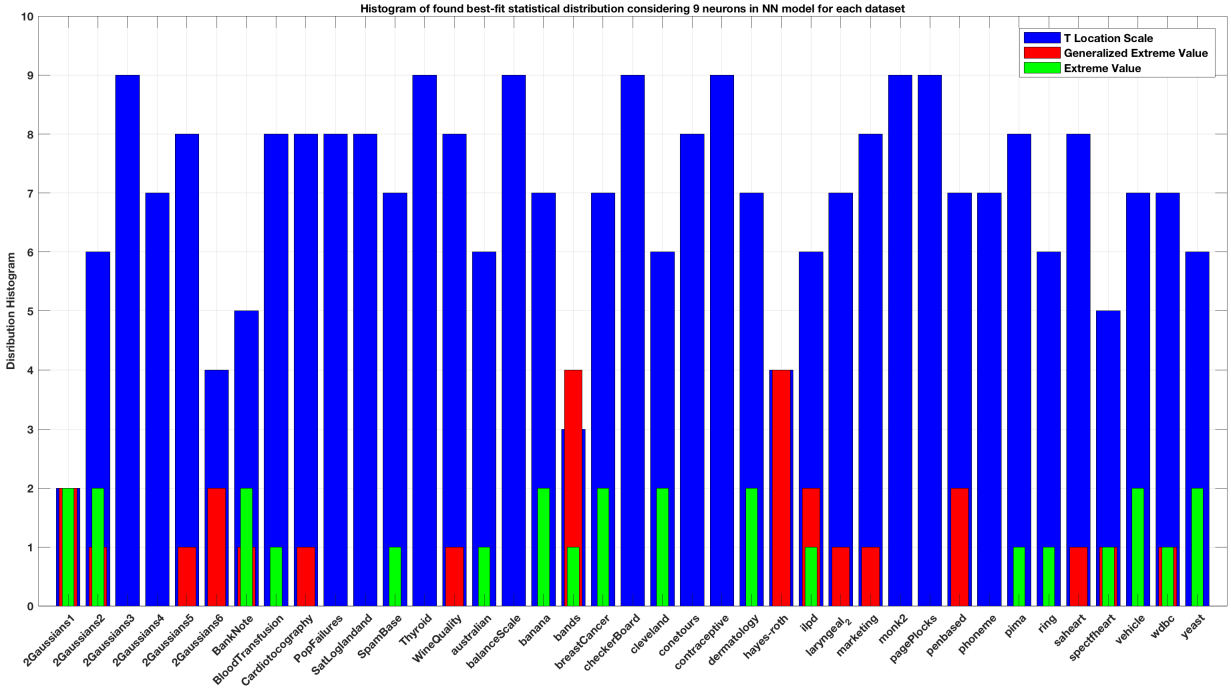


Figure 4.2: Histogram of best-fit parametric distributions to the weight connections of trained ensemble of NN perdataset

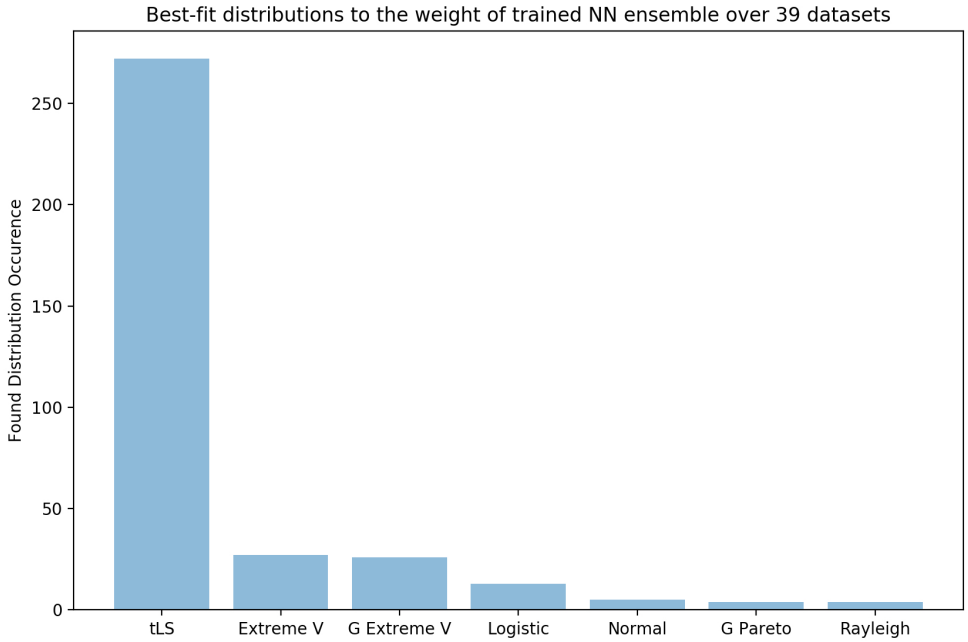


Figure 4.3: Summary of the histogram of best-fit parametric distributions over 39 datasets; 9 neurons per dataset.

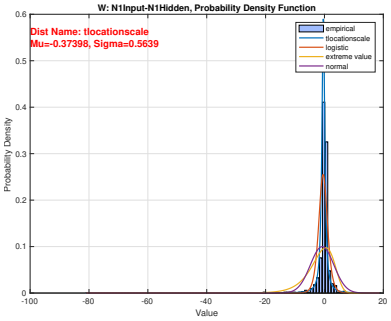
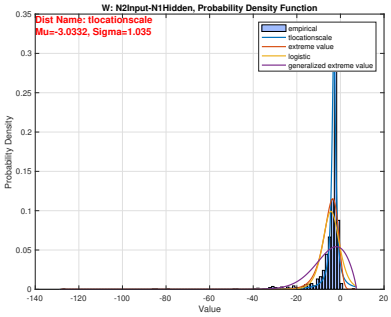
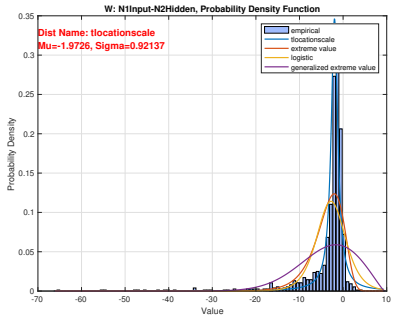
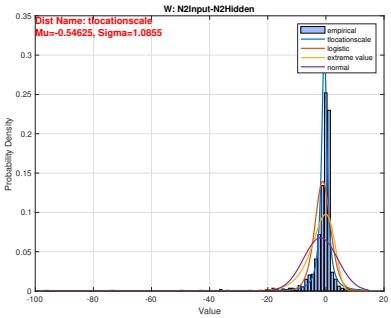
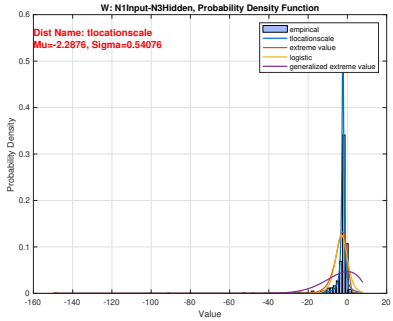
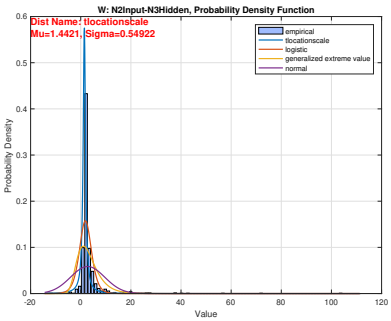
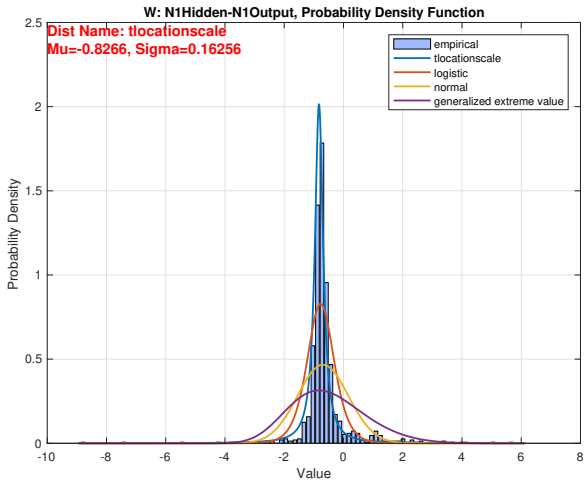
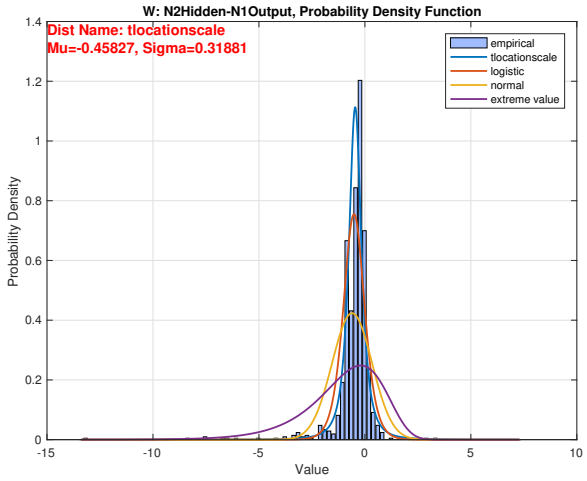
	N1 Input	N2 Input
N1 Hidden		
N2 Hidden		
N3 Hidden		

Table 4.1: Weight Connection from Input:Hidden Layer, Best-Fit distribution estimation, Breast cancer Dataset.

N1 Hidden



N2 Hidden



N3 Hidden

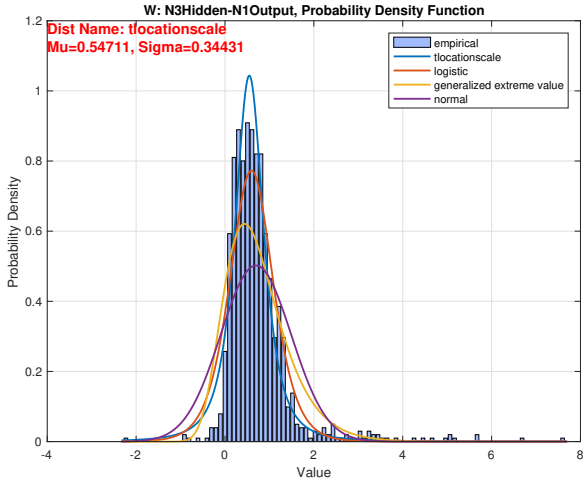


Table 4.2: Weight Connection from Hidden:Output Layer, Best-Fit distribution estimation, Breast cancer Dataset.

Table 4.3: The wight connections of a trained ensemble of 1k NN classifier created using bagging; Estimating the best-fit distribution for each single weight connection and for the full weight connections matrix between layers. The header "C" denotes the number of classes in the dataset; the header "F" denotes the number of features per instance; the header "$F^{'}$" denotes that regarding the single neuron weight analysis the table reports only the weight connections between the first 2 input neurons and the rest of the network.

N	Name	C	F	F'	N1In:N1H	N2In:N1H	N1In:N2H	N2In:N2H	N1In:N3H	N2In:N3H	N1H:N1O	N1H:N2O	N1H:N3O
1	2Gaussians1	2	2	2	Extreme value	G Extreme value	G Extreme value	rayleigh	Extreme value	Logistic	tLS	tLS	Logistic
2	2Gaussians2	2	2	2	tLS	tLS	tLS	tLS	G Extreme value	tLS	Extreme value	G Extreme value	tLS
3	2Gaussians3	2	2	2	tLS	tLS	tLS	tLS	tLS	tLS	tLS	tLS	tLS
4	2Gaussians4	2	2	2	tLS	tLS	Normal	Normal	tLS	tLS	tLS	tLS	tLS
5	2Gaussians5	2	2	2	Extreme value	tLS	tLS	tLS	tLS	tLS	tLS	tLS	tLS
6	2Gaussians6	2	2	2	Extreme value	tLS	G pareto	Logistic	Extreme value	tLS	tLS	tLS	Logistic
7	BankNote	2	4	2	tLS	tLS	G pareto	G Extreme value	tLS	tLS	Extreme value	tLS	G Extreme value
8	BloodTransfusion	2	4	2	tLS	tLS	tLS	tLS	G Extreme value	tLS	tLS	tLS	tLS
9	Cardiotocography	3	22	2	tLS	tLS	tLS	tLS	tLS	tLS	Extreme value	tLS	tLS
10	PopFailures	2	20	2	tLS	Logistic	tLS	tLS	tLS	tLS	tLS	tLS	tLS
11	SatLoglandand	6	36	2	tLS	tLS	tLS	Logistic	tLS	tLS	tLS	tLS	tLS
12	SpamBase	2	57	2	tLS	tLS	tLS	tLS	tLS	Logistic	G Extreme value	tLS	tLS
13	Thyroid	3	21	2	tLS	tLS	tLS	tLS	tLS	tLS	tLS	tLS	tLS
14	WineQuality	7	11	2	tLS	tLS	tLS	tLS	Extreme value	tLS	tLS	tLS	tLS
15	australian	2	14	2	tLS	Logistic	tLS	tLS	tLS	tLS	G Extreme value	Logistic	tLS
16	balanceScale	3	4	2	tLS	tLS	tLS	tLS	tLS	tLS	tLS	tLS	tLS
17	banana	2	2	2	tLS	tLS	tLS	G Extreme value	tLS	tLS	tLS	tLS	G Extreme value
18	bands	2	19	2	Extreme value	tLS	Extreme value	tLS	Extreme value	tLS	G Extreme value	Extreme value	rayleigh
19	breastCancer	2	9	2	tLS	tLS	tLS	tLS	tLS	tLS	G Extreme value	G Extreme value	tLS
20	checkerBoard	2	2	2	tLS	tLS	tLS	tLS	tLS	tLS	tLS	tLS	tLS
21	cleveland	5	13	2	tLS	tLS	tLS	tLS	tLS	tLS	G Extreme value	G Extreme value	Logistic
22	conetours	3	2	2	tLS	tLS	tLS	Logistic	tLS	tLS	tLS	tLS	tLS
23	contraceptive	3	9	2	tLS	tLS	tLS	tLS	tLS	tLS	tLS	tLS	tLS
24	dermatology	6	34	2	tLS	tLS	tLS	tLS	tLS	tLS	G Extreme value	G Extreme value	tLS
25	hayes-roth	3	4	2	tLS	tLS	Extreme value	tLS	Extreme value	tLS	Extreme value	Extreme value	rayleigh
26	ilpd	2	9	2	tLS	tLS	Extreme value	tLS	tLS	tLS	Extreme value	G Extreme value	tLS
27	laryngeal_2	2	16	2	tLS	tLS	tLS	tLS	tLS	Extreme value	tLS	tLS	Logistic
28	marketing	9	13	2	tLS	tLS	tLS	tLS	tLS	tLS	tLS	tLS	Extreme value
29	monk2	2	6	2	tLS	tLS	tLS	tLS	tLS	tLS	tLS	tLS	tLS
30	pagePlocks	5	10	2	tLS	tLS	tLS	tLS	tLS	tLS	tLS	tLS	tLS
31	penbased	10	16	2	Extreme value	Extreme value	tLS	tLS	tLS	tLS	tLS	tLS	tLS
32	phoneme	2	5	2	tLS	tLS	tLS	tLS	tLS	Logistic	tLS	G pareto	tLS
33	pima	2	8	2	tLS	tLS	tLS	tLS	tLS	tLS	tLS	G Extreme value	tLS
34	ring	2	20	2	tLS	tLS	G Extreme value	rayleigh	tLS	tLS	tLS	tLS	G pareto
35	saheart	2	4	2	tLS	tLS	tLS	tLS	tLS	tLS	Extreme value	tLS	tLS
36	spectfheart	2	44	2	tLS	Normal	tLS	tLS	Extreme value	tLS	tLS	G Extreme value	Normal
37	vehicle	4	18	2	tLS	tLS	tLS	tLS	tLS	tLS	G Extreme value	G Extreme value	tLS
38	wdbc	2	30	2	Extreme value	tLS	tLS	tLS	tLS	tLS	tLS	tLS	G Extreme value
39	yeast	10	8	2	tLS	tLS	tLS	tLS	tLS	tLS	G Extreme value	G Extreme value	rayleigh

Chapter 5

Summary and Conclusion

5.1 Summary

- **Chapter 1:** Presents Thesis abstract, problem statement thesis outlines.
- **Chapter 2:** Presents a brief introduction to the research field of Pattern Recognition, describing various types of classifiers. Starting with simple classifiers such as Linear Discriminant Classifiers (LDC), Nearest Mean Classifiers (NMC) and Quadratic Discriminant Classifiers (QDC). Decision Tree, K-Nearest Neighbours and Artificial Neural Networks (ANN) are mainly presented, highlighting their applications limits and characteristics. Introduce the concepts of Multiple Classifier Systems (MCS), their creation, size, limits and weakness. Majority Voting rule as an ensemble decision rule is presented. Diversity measures are presented, discussed in details. Ending with different ensemble pruning strategies.
- **Chapter 3:** Presents the main contributions to study diversity measures highlighting the relation between them.we focus on pruning techniques based on forward/backward selection, since they allow a direct comparison with the simple estimation of accuracy of classifier ensemble. Presenting a comparison for several diversity measures and benchmark data sets, using bagging as the ensemble construction technique, and majority voting as the fusion rule. Obtained results provide further and more direct evidence to previous observations against the effectiveness of the use of diversity measures for ensemble pruning, but also show that, combined with ensemble accuracy estimated on a validation set, diversity can have a regularization effect when the validation set size is small. Then presenting an empirical investigation of a linear combination of ensemble accuracy with diversity measures. This can also be viewed as using diversity as a regularizer, as suggested by some authors. The summary of experiments on thirty-seven benchmark data sets, four diversity measures and three base classifiers provide evidence that using diversity measures for ensemble pruning can be advantageous over using only ensemble accuracy, and that diversity measures can act as regularizers in this context. The results provided an evidence that diversity have a regular behaviour when combined with ensemble accuracy as an objective function of the FS ensemble pruning method. At least for diversity measures those use the instance true label as DF, Theta and GD, thus it outperforms the ensemble accuracy when used alone

as an objective function in same algorithm. The diversity measures that don't use the pattern true label such as Disagreement and Entropy, when combined with accuracy as an objective function shows a regular behavior to outperforms ensemble accuracy for value of $\lambda = 0.2$. This behaviour of the diversity measures that don't use the instance true label motivated us to extend our experiments in order to investigate the rule of those diversity measures in the case of small λ. We investigated the case of having a few labelled instances that can be exploit by the objective functions that require the instance true label such as the ensemble accuracy and the three diversity measures DF, Theta and GD, meanwhile growing up the amount of unlabeled data that can be exploit by Disagreement and Entropy diversity measures. The aim of this experiments is to investigate the behaviour of diversity measures that don't use the instance true label while growing the amount of instances that this two diversity measures can exploit. Whereas the usefulness of diversity measures for ensemble construction has been questioned by some authors, their specific role as regularizers has been recently pointed out in [5] based on theoretical results as well as on empirical evidence in the context of ensemble pruning, although in a specific setting (binary classifiers, and an ad hoc diversity measure). As a follow-up of our preliminary work [1], we investigated the effectiveness of well-known, generic diversity measures in ensemble pruning. In particular, we considered their use in the ensemble evaluation function of pruning methods based on the forward search strategy, by linearly combining them with ensemble accuracy (estimated from validation data). This can be viewed as using diversity measures as regularizers, in the spirit of [5]. As far as our experiments are concerned, our empirical results provided evidence that also generic ensemble diversity measures can be useful when used together with ensemble accuracy as the pruning evaluation function. This is in agreement with the results we obtained in [1], related to ad hoc evaluation functions proposed by other authors for ensemble pruning, that combine individual classifiers' (not ensemble) accuracy and diversity (more precisely, complementarity between their errors). Our results also show that also generic diversity measures can have a regularization effect on the estimated ensemble accuracy, in the context of ensemble pruning. This provides some evidence that the results of [5], related to a specific diversity measure, could be extended to generic diversity measures.

- **Chapter 4:** Presents a detailed approach to analyse the weight connections of a trained ensemble of neural networks. Presenting a framework to estimate the best-fit statistical distribution to the weight connections of the trained ensemble of neural networks. The analysis results votes for the T-location scale statistical distribution to be the best fit to the weights of the trained NN ensemble.Starting from an existing ensemble creation randomization technique known as Bagging, the aim is to analyze the distribution of the weight connections of the trained ensemble of Neural Networks. We present a framework to estimate the best-fit statistical distribution from a list of well-known statistical parametric distributions. This work is the first attempt in the state-of-art to explore and analyze the weights of a trained ensemble of 1000 neural networks. The analysis results votes for the T-location scale statistical distribution to be the best fit to the weights of the trained NN ensemble.We investigated the weight distribution of a trained ensemble of NN's created using Bagging. The used approach to estimate the best-fit parametric distribution to the weight value of a single neuron of each NN classifier in the ensemble uses maximum likelihood to compute the parameters of the

attempt to fit distribution. Considering the weight distribution of each neuron in the NN classifier for each dataset, this is our future scope to use the estimated best-fit distribution to not train the classifier on new bootstrap replicas of the training set, but automatically assign the weight value of the neuron from already estimated as a best-fit distribution. Considering the optimality principle to investigate the weight distribution highlighted by some references, we aim to re-investigate the weight distribution of a trained ensemble of neural networks created using bagging when we raise the number of instances in the training set, until the classifier becomes about to reach its maximum capacity.

5.2 Limitations of Thesis and Possible Future Considerations

In this thesis we focus only on Bagging ensemble creation techniques, using only three base classifiers and foreward selection as ensemble pruning technique. Ensemble accuracy and a set of ten diversity measures are deeply investigated to be used separately or combined via a combination coefficient of only three values, as an objective function for the foreward selection ensemble pruning algorithm. We held our main experiments on a 37 datasets only. It is possible to extend this investigation using other diversity measures or even combining them in different ways. Also the work on estimating the distribution of trained neural networks ensembles can be extended to other base classifiers, it is promising that a clear definition of the weight connections distribution of trained ensembles of neural network classifiers can save the training cost and directly withdraw the weight connections values from those distributions.

List of Publications Related to the Thesis

Published papers

1. Muhammad AO Ahmed, Luca Didaci, Giorgio Fumera, and Fabio Roli. **An Empirical Investigation on the Use of Diversity for Creation of Classifier Ensembles.** In 12th International Workshop, MCS 2015, Gunzburg, Germany, June 29-July 1, 2015, Proceedings, volume 9132, pages 206-219. Springer International Publishing, 2015 [1]

2. Muhammad AO Ahmed, Luca Didaci, Giorgio Fumera, and Fabio Roli. **Using Diversity for Classifier Ensemble Pruning: An Empirical Investigation**; Theoretical and Applied Informatics Journal 2018. [212]

3. Muhammad AO Ahmed. **Trained Neural Networks Ensembles Weight Connections Analysis.**,The International Conference on Advanced Machine Learn-ing Technologies and Applications (AMLTA2018), pages 242-251, Cham, 2018. Springer International Publishing. [213]

4. Bahram Lavi, Muhammad AO Ahmed. I**nteractive Fuzzy Cellular Automata for Fast Person Re-Identification.,**The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2018), pages 147–157, Cham, 2018. Springer International Publishing.[214]

5. Muhammad AO Ahmed *et. al.* . **Multi-filter Score-Level Fusion for Fingerprint Verification.**,The International Conference on Advanced Machine Learn-ing Technologies and Applications (AMLTA2018), pages 624–633, Cham, 2018. Springer International Publishing. [215]

Bibliography

[1] Muhammad AO Ahmed, Luca Didaci, Giorgio Fumera, and Fabio Roli. An empirical investigation on the use of diversity for creation of classifier ensembles. In *12th International Workshop, MCS 2015, Günzburg, Germany, June 29 - July 1, 2015, Proceedings*, volume 9132, pages 206–219. Springer International Publishing, 2015. [cited at p. 2, 24, 56, 66, 69]

[2] Luca Didaci, Giorgio Fumera, and Fabio Roli. Diversity in classifier ensembles: Fertile concept or dead end? In *International Workshop on Multiple Classifier Systems*, pages 37–48. Springer, 2013. [cited at p. 2, 23, 29]

[3] Xiaoqin Zeng and Daniel S Yeung. Sensitivity analysis of multilayer perceptron to input and weight perturbations. *IEEE Transactions on Neural Networks*, 12(6):1358–1366, 2001. [cited at p. 2]

[4] Zhi-Hua Zhou, Jianxin Wu, and Wei Tang. Ensembling neural networks: many could be better than all. *Artificial intelligence*, 137(1-2):239–263, 2002. [cited at p. 2, 18, 27, 35]

[5] Nan Li, Yang Yu, and Zhi-Hua Zhou. Diversity regularized ensemble pruning. *Machine Learning and Knowledge Discovery in Databases*, pages 330–345, 2012. [cited at p. 2, 24, 26, 30, 31, 34, 45, 49, 66]

[6] Robert PW Duin and David MJ Tax. Experiments with classifier combining rules. In *International Workshop on Multiple Classifier Systems*, pages 16–29. Springer, 2000. [cited at p. 2, 18, 22]

[7] Gavin Brown, Jeremy Wyatt, Rachel Harris, and Xin Yao. Diversity creation methods: a survey and categorisation. *Information Fusion*, 6(1):5–20, 2005. [cited at p. 2, 18, 23, 24]

[8] Alexey Tsymbal, Mykola Pechenizkiy, and Pádraig Cunningham. Diversity in search strategies for ensemble feature selection. *Information fusion*, 6(1):83–98, 2005. [cited at p. 2, 26]

[9] Padraig Cunningham and John Carney. Diversity versus quality in classification ensembles based on feature selection. In *European Conference on Machine Learning*, pages 109–116. Springer, 2000. [cited at p. 2, 22]

[10] Kevin S Woods, Christopher C Doss, Kevin W Bowyer, Jeffrey L Solka, Carey E Priebe, and W PHILIP KEGELMEYER JR. Comparative evaluation of pattern recognition tech-

niques for detection of microcalcifications in mammography. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(06):1417–1436, 1993. [cited at p. 2, 10]

[11] Ludmila I. Kuncheva and Christopher J. Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51(2):181–207, May 2003. [cited at p. 2, 17, 18, 20, 23, 24, 25, 29, 31, 32, 35, 37, 38]

[12] Ludmila I Kuncheva. Change detection in streaming multivariate data using likelihood detectors. *IEEE Transactions on Knowledge and Data Engineering*, 25(5):1175–1180, 2013. [cited at p. 2, 24]

[13] Ludmila I Kuncheva and Chris J Whitaker. Ten measures of diversity in classifier ensembles: limits for two classifiers. In *Intelligent Sensor Processing (Ref. No. 2001/050), A DERA/IEE Workshop on*, pages 10–1. IET, 2001. [cited at p. 2, 17, 21]

[14] Ludmila I Kuncheva, Christopher J Whitaker, Catherine A Shipp, and Robert PW Duin. Limits on the majority vote accuracy in classifier fusion. *Pattern Analysis & Applications*, 6(1):22–31, 2003. [cited at p. 2, 18]

[15] Ludmila I Kuncheva. *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons, 2014. [cited at p. 2, 56]

[16] Gonzalo Martínez-Muñoz and Alberto Suárez. Pruning in ordered bagging ensembles. In *Proceedings of the 23rd international conference on Machine learning*, pages 609–616. ACM, 2006. [cited at p. 2, 24, 27, 30, 46]

[17] Dragos D Margineantu and Thomas G Dietterich. Pruning adaptive boosting. In *ICML*, volume 97, pages 211–218, 1997. [cited at p. 2, 26, 30, 45]

[18] Andreas Prodromidis and Salvatore J Stolfo. Pruning meta-classifiers in a distributed data mining system. In *Proceedings of the First National Conference on New Information Technologies*, volume 151, 1998. [cited at p. 2, 30]

[19] Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E Hopcroft, and Kilian Q Weinberger. Snapshot ensembles: Train 1, get m for free. *arXiv preprint arXiv:1704.00109*, 2017. [cited at p. 2, 56]

[20] G. B. Huang, H. Zhou, X. Ding, and R. Zhang. Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(2):513–529, April 2012. [cited at p. 2]

[21] Enrica Santucci, Luca Didaci, Giorgio Fumera, and Fabio Roli. A parameter randomization approach for constructing classifier ensembles. *Pattern Recognition*, 69:1 – 13, 2017. [cited at p. 2, 56]

[22] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems*, pages 1135–1143, 2015. [cited at p. 2]

[23] CE Rasmussen and CKI Williams. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, USA, 1 2006. [cited at p. 5]

[24] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006. [cited at p. 5]

[25] Moira Stewart. Towards a global definition of patient centred care: the patient should be the judge of patient centred care. *BMJ: British Medical Journal*, 322(7284):444, 2001. [cited at p. 5]

[26] Ludmila Kuncheva. *Fuzzy classifier design*, volume 49. Springer Science & Business Media, 2000. [cited at p. 6]

[27] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *Journal of machine learning research*, 9(Aug):1871–1874, 2008. [cited at p. 6]

[28] Yoav Freund and Robert E Schapire. Large margin classification using the perceptron algorithm. *Machine learning*, 37(3):277–296, 1999. [cited at p. 7]

[29] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995. [cited at p. 7]

[30] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of human genetics*, 7(2):179–188, 1936. [cited at p. 7]

[31] Geoffrey McLachlan. *Discriminant analysis and statistical pattern recognition*, volume 544. John Wiley & Sons, 2004. [cited at p. 7]

[32] Yan-yan Song and LU Ying. Decision tree methods: applications for classification and prediction. *Shanghai archives of psychiatry*, 27(2):130, 2015. [cited at p. 8]

[33] Detlof Von Winterfeldt and Ward Edwards. Decision analysis and behavioral research. 1993. [cited at p. 8]

[34] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984. [cited at p. 9, 10]

[35] Matthew Hansen, R Dubayah, and R DeFries. Classification trees: an alternative to traditional land cover classifiers. *International journal of remote sensing*, 17(5):1075–1081, 1996. [cited at p. 9]

[36] Philip H Swain and Hans Hauska. The decision tree classifier: Design and potential. *IEEE Transactions on Geoscience Electronics*, 15(3):142–147, 1977. [cited at p. 9, 11]

[37] Robert J McQueen, Stephen R Garner, Craig G Nevill-Manning, and Ian H Witten. Applying machine learning to agricultural data. *Computers and electronics in agriculture*, 12(4):275–293, 1995. [cited at p. 10]

[38] Steven Salzberg, Rupali Chandar, Holland Ford, Sreerama K Murthy, and Richard White. Decision trees for automated identification of cosmic-ray hits in hubble space telescope images. *Publications of the Astronomical Society of the Pacific*, 107(709):279, 1995. [cited at p. 10]

[39] Nicholas Weir, Usama M Fayyad, and S Djorgovski. Automated star/galaxy classification for digitized poss-ii. *The Astronomical Journal*, 109:2401, 1995. [cited at p. 10]

[40] Nicholas Weir, S Djorgovski, and Usama M Fayyad. Initial galaxy counts from digitized poss-ii. *The Astronomical Journal*, 110:1, 1995. [cited at p. 10]

[41] RK Gulati, Ranjan Gupta, P Gothoskar, and S Khobragade. Ultraviolet stellar spectral classification using a multilevel tree neural network. *Vistas in Astronomy*, 38:293–298, 1994. [cited at p. 10]

[42] WJ Gibb, DM Auslander, and JC Griffin. Selection of myocardial electrogram features for use by implantable devices. *IEEE Transactions on Biomedical Engineering*, 40(8):727–735, 1993. [cited at p. 10]

[43] KJ Hunt. Classification by induction: application to modelling and control of non-linear dynamical systems. *Intelligent Systems Engineering*, 2(4):231–245, 1993. [cited at p. 10]

[44] Joseph J Mezrich. When is a tree a hedge? *Financial Analysts Journal*, 50(6):75–81, 1994. [cited at p. 10]

[45] Aytul Ercil, Yagmur Denizhan, Ahmet Okcular, and G Zora. Classification trees prove useful in nondestructive testing of spot weld quality. *Welding Journal (Miami);(United States)*, 72(9), 1993. [cited at p. 10]

[46] Keki B Irani, Jie Cheng, Usama M Fayyad, and Zhaogang Qian. Applying machine learning to semiconductor manufacturing. *iEEE Expert*, 8(1):41–47, 1993. [cited at p. 10]

[47] Davis M Kennedy. Decision tree bears fruit. *PRODUCTS FINISHING-CINCINNATI-*, 57:66–66, 1993. [cited at p. 10]

[48] SANCHOY K DAS and Sanjay Bhambri. A decision tree approach for selecting between demand based, reorder, and jit/kanban methods for material procurement. *Production planning & Control*, 5(4):342–348, 1994. [cited at p. 10]

[49] Bob Evans and Doug Fisher. Overcoming process delays with decision tree induction. *IEEE expert*, 9(1):60–66, 1994. [cited at p. 10]

[50] A Famili. Use of decision-tree induction for process optimization and knowledge refinement of an industrial process. *AI EDAM*, 8(1):63–75, 1994. [cited at p. 10]

[51] Shailendra C Palvia and Steven R Gordon. Tables, trees and formulas in decision analysis. *Communications of the ACM*, 35(10):104–113, 1992. [cited at p. 10]

[52] Patricia Riddle, Richard Segal, and Oren Etzioni. Representation design and brute-force induction in a boeing manufacturing domain. *Applied Artificial Intelligence an International Journal*, 8(1):125–147, 1994. [cited at p. 10]

[53] Yuan Guo and Kevin J Dooley. Distinguishing between mean, variance and autocorrelation changes in statistical quality control. *THE INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH*, 33(2):497–510, 1995. [cited at p. 10]

[54] Haldun Aytug, Siddhartha Bhattacharyya, Gary J Koehler, and Jane L Snowdon. A review of machine learning in scheduling. *IEEE Transactions on Engineering Management*, 41(2):165–171, 1994. [cited at p. 10]

[55] Igor Kononenko. Inductive and bayesian learning in medical diagnosis. *Applied Artificial Intelligence an International Journal*, 7(4):317–337, 1993. [cited at p. 10]

[56] William J Long, John L Griffith, Harry P Selker, and Ralph B D'agostino. A comparison of logistic regression to decision-tree induction in a medical domain. *Computers and Biomedical Research*, 26(1):74–97, 1993. [cited at p. 10]

[57] Judith A Falconer, Bruce J Naughton, Dorothy D Dunlop, Elliot J Roth, Dale C Strasser, and James M Sinacore. Predicting stroke inpatient rehabilitation outcome using a classification tree approach. *Archives of Physical Medicine and Rehabilitation*, 75(6):619–625, 1994. [cited at p. 10]

[58] Peter Kokol, Marjan Mernik, Jernej Završnik, Kurt Kancler, and Ivan Malčić. Decision trees based on automatic learning and their use in cardiology. *Journal of Medical Systems*, 18(4):201–206, 1994. [cited at p. 10]

[59] Dean P McKenzie, Patrick D McGorry, Chris S Wallace, Lee Hun Low, David L Copolov, and Bruce S Singh. Constructing a minimal diagnostic decision tree. *Methods of information in medicine*, 32(2):161–166, 1993. [cited at p. 10]

[60] Gert Judmaier, Peter Meyersbach, Gunter Weiss, Helmut Wachter, and Gilbert Reibnegger. The role of neopterin in assessing disease activity in crohn's disease: classification and regression trees. *American Journal of Gastroenterology*, 88(5), 1993. [cited at p. 10]

[61] PAD Wilks and MJ English. Accurate segmentation of respiration waveforms from infants enabling identification and classification of irregular breathing patterns. *Medical engineering & physics*, 16(1):19–23, 1994. [cited at p. 10]

[62] Shinichi Shimozono, Ayumi Shinohara, Takeshi Shinohara, Satoru Miyano, Satoru Kuhara, and Setsuo Arikawa. Knowledge acquisition from amino acid sequences by machine learning system bonsai. *Transactions of Information Processing Society of Japan*, 35(10):2009–2018, 1994. [cited at p. 10]

[63] Steven Salzberg. Locating protein coding regions in human dna using a decision tree algorithm. *Journal of Computational Biology*, 2(3):473–485, 1995. [cited at p. 10]

[64] Lilly Spirkovska. Three-dimensional object recognition using similar triangles and decision trees. *Pattern Recognition*, 26(5):727–732, 1993. [cited at p. 10]

[65] Michael E Bullock, David L Wang, Scott R Fairchild, and Tim J Patterson. Automated training of 3d morphology algorithm for object recognition. In *SPIE's International Symposium on Optical Engineering and Photonics in Aerospace Sensing*, pages 238–251. International Society for Optics and Photonics, 1994. [cited at p. 10]

[66] Yves Kodratoff and Stephane Moscatelli. Machine learning for object recognition and scene analysis. *International Journal of Pattern Recognition and Artificial Intelligence*, 8(01):259–304, 1994. [cited at p. 10]

[67] Koudou Toussaint Dago, Rémy Luthringer, Régis Lengellé, Gérard Rinaudo, and Jean-Paul Macher. Statistical decision tree: A tool for studying pharmaco-eeg effects of cns-active drugs. *Neuropsychobiology*, 29(2):91–96, 1994. [cited at p. 10]

[68] David Bowser-Chao and Debra L Dzialo. Comparison of the use of binary decision trees and neural networks in top-quark detection. *Physical Review D*, 47(5):1900, 1993. [cited at p. 10]

[69] FA Baker, David L Verbyla, CS Hodges Jr, and EW Ross. Classification and regression tree analysis for assessing hazard of pine mortality caused by heterobasidion annosum. *Plant Disease*, 1993. [cited at p. 10]

[70] ND Hatziargyriou, GC Contaxis, and NC Sideris. A decision tree method for on-line steady state security assessment. *IEEE Transactions on power systems*, 9(2):1052–1061, 1994. [cited at p. 11]

[71] Steven Rovnyak, Stein Kretsinger, James Thorp, and Donald Brown. Decision trees for real-time transient stability prediction. *IEEE Transactions on Power Systems*, 9(3):1417–1426, 1994. [cited at p. 11]

[72] Byungyong Kim and David Landgrebe. Hierarchical decision tree classifiers in high-dimensional and large class data. *IEEE Trans. on Geoscience and Remote Sensing*, 29(4):518, 1990. [cited at p. 11]

[73] Ron Rymon and Nicholas M Short. Automatic cataloguing and characterization of earth science data using set enumeration trees. *Telematics and Informatics*, 11(4):309–318, 1994. [cited at p. 11]

[74] Krishnamoorthy Srinivasan and Douglas Fisher. Machine learning approaches to estimating software development effort. *IEEE Transactions on Software Engineering*, 21(2):126–137, 1995. [cited at p. 11]

[75] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986. [cited at p. 11]

[76] Wendy Lehnert, Stephen Soderland, David Aronow, Fangfang Feng, and Avinoam Shmueli. Inductive text classification for medical applications. *Journal of Experimental & Theoretical Artificial Intelligence*, 7(1):49–80, 1995. [cited at p. 11]

[77] Tom M Mitchell, Rich Caruana, Dayne Freitag, John McDermott, David Zabowski, et al. Experience with a learning personal assistant. *Communications of the ACM*, 37(7):80–91, 1994. [cited at p. 11]

[78] Miroslav Kubat, Gert Pfurtscheller, and Doris Flotzinger. Ai-based approach to automatic sleep classification. *Biological Cybernetics*, 70(5):443–448, 1994. [cited at p. 11]

[79] Naomi S Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992. [cited at p. 11]

[80] Amy FT Arnsten. The biology of being frazzled. *Science*, 280(5370):1711–1712, 1998. [cited at p. 12]

[81] Jacek M Zurada. *Introduction to artificial neural systems*, volume 8. West St. Paul, 1992. [cited at p. 12]

[82] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943. [cited at p. 13]

[83] Bernard Widrow and Marcian E Hoff. Adaptive switching circuits. Technical report, STANFORD UNIV CA STANFORD ELECTRONICS LABS, 1960. [cited at p. 13]

[84] Stephen Cole Kleene. Representation of events in nerve nets and finite automata. Technical report, RAND PROJECT AIR FORCE SANTA MONICA CA, 1951. [cited at p. 13]

[85] Andreas Zell, Niels Mache, Ralf Huebner, Günter Mamier, Michael Vogt, Michael Schmalzl, and Kai-Uwe Herrmann. Snns (stuttgart neural network simulator). In *Neural Network Simulation Environments*, pages 165–186. Springer, 1994. [cited at p. 14]

[86] L.I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. John Wiley & Sons, 2014. [cited at p. 14, 16, 17, 20, 22, 23, 24, 25, 29, 31, 32, 34]

[87] SN Balakrishnan and Victor Biega. Adaptive-critic-based neural networks for aircraft optimal control. *Journal of Guidance, Control, and Dynamics*, 19(4):893–898, 1996. [cited at p. 15]

[88] Dipankar Dasgupta, Kalmanje KrishnaKumar, D Wong, and Misty Berry. Negative selection algorithm for aircraft fault detection. *Artificial immune systems*, pages 1–13, 2004. [cited at p. 15]

[89] Dean A Pomerleau. *Neural network perception for mobile robot guidance*, volume 239. Springer Science & Business Media, 2012. [cited at p. 15]

[90] Wim Smit. Curtailing and steering military r&d'. In *Proceedings of the 41st Pugwash Conference (forthcoming) and B Elzen, B Enserink and WA Smit,'Weapons innovation: networks and guiding principles', Science and Public Policy*, volume 17, 1993. [cited at p. 15]

[91] Christian Goerick, Detlev Noll, and Martin Werner. Artificial neural networks in real-time car detection and tracking applications. *Pattern Recognition Letters*, 17(4):335–343, 1996. [cited at p. 15]

[92] A Cochocki and Rolf Unbehauen. *Neural networks for optimization and signal processing*. John Wiley & Sons, Inc., 1993. [cited at p. 15]

[93] Robert R Trippi and Efraim Turban. *Neural networks in finance and investing: Using artificial intelligence to improve real world performance*. McGraw-Hill, Inc., 1992. [cited at p. 15]

[94] Magali RG Meireles, Paulo EM Almeida, and Marcelo Godoy Simões. A comprehensive review for industrial applicability of artificial neural networks. *IEEE transactions on industrial electronics*, 50(3):585–601, 2003. [cited at p. 15]

[95] William G Baxt. Application of artificial neural networks to clinical medicine. *The lancet*, 346(8983):1135–1138, 1995. [cited at p. 15]

[96] Richard P Lippmann. Review of neural networks for speech recognition. *Neural computation*, 1(1):1–38, 1989. [cited at p. 15]

[97] Manfred M Fischer and Sucharita Gopal. Artificial neural networks: a new approach to modeling interregional telecommunication flows. *Journal of regional Science*, 34(4):503–527, 1994. [cited at p. 15]

[98] Leah L Rogers and Farid U Dowla. Optimization of groundwater remediation using artificial neural networks with parallel solute transport modeling. *Water Resources Research*, 30(2):457–481, 1994. [cited at p. 15]

[99] Xin Yao. Evolving artificial neural networks. *Proceedings of the IEEE*, 87(9):1423–1447, 1999. [cited at p. 15]

[100] Iebeling Kaastra and Milton Boyd. Designing a neural network for forecasting financial and economic time series. *Neurocomputing*, 10(3):215–236, 1996. [cited at p. 15]

[101] W Thomas Miller, Paul J Werbos, and Richard S Sutton. *Neural networks for control*. MIT press, 1995. [cited at p. 15]

[102] James Cannady. Artificial neural networks for misuse detection. In *National information systems security conference*, pages 368–81, 1998. [cited at p. 15]

[103] Lei Xu, Adam Krzyzak, and Ching Y Suen. Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE transactions on systems, man, and cybernetics*, 22(3):418–435, 1992. [cited at p. 16]

[104] Jack D Tubbs and William O Alltop. Measures of confidence associated with combining classification results. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(3):690–692, 1991. [cited at p. 16]

[105] Tin Kam Ho, Jonathan J. Hull, and Sargur N. Srihari. Decision combination in multiple classifier systems. *IEEE transactions on pattern analysis and machine intelligence*, 16(1):66–75, 1994. [cited at p. 16]

[106] Robert Milewski and Venu Govindaraju. Binarization and cleanup of handwritten text from carbon copy medical form images. *Pattern recognition*, 41(4):1308–1315, 2008. [cited at p. 16]

[107] Richard O Duda, Peter E Hart, and David G Stork. Pattern classification second edition john wiley & sons. *New York*, 58, 2001. [cited at p. 16]

[108] Roberto Brunelli. *Template matching techniques in computer vision: theory and practice*. John Wiley & Sons, 2009. [cited at p. 16]

[109] and others. Optical character recognition, April 8 1969. US Patent 3,437,824. [cited at p. 16]

[110] Melba C Caldwell, David K Caldwell, and Peter L Tyack. Review of the signature-whistle hypothesis for the atlantic bottlenose dolphin. *The bottlenose dolphin*, pages 199–234, 1990. [cited at p. 16]

[111] Friedrich Justen, Kai U Ziegler, and Heinz E Gallus. Experimental investigation of unsteady flow phenomena in a centrifugal compressor vaned diffuser of variable geometry. *TRANSACTIONS-AMERICAN SOCIETY OF MECHANICAL ENGINEERS JOURNAL OF TURBOMACHINERY*, 121:763–771, 1999. [cited at p. 16]

[112] ANPR. [cited at p. 17]

[113] Tom M Mitchell. Machine learning. 1997. *Burr Ridge, IL: McGraw Hill*, 45(37):870–877, 1997. [cited at p. 17]

[114] Olvi L Mangasarian, R Setiono, and WH Wolberg. Pattern recognition via linear programming: Theory and application to medical diagnosis. *Large-scale numerical optimization*, pages 22–31, 1990. [cited at p. 17]

[115] Philip D Stahl and R Alan B Ezekowitz. The mannose receptor is a pattern recognition receptor involved in host defense. *Current opinion in immunology*, 10(1):50–55, 1998. [cited at p. 17]

[116] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999. [cited at p. 17]

[117] Ludmila I Kuncheva. A bound on kappa-error diagrams for analysis of classifier ensembles. *IEEE Transactions on knowledge and data engineering*, 25(3):494–501, 2013. [cited at p. 17, 23]

[118] Zhi-Hua Zhou. *Ensemble methods: foundations and algorithms*. CRC press, 2012. [cited at p. 18, 23, 24]

[119] Zhi-Hua Zhou. *Ensemble methods: foundations and algorithms*. CRC press, 2012. [cited at p. 18, 22, 23]

[120] Hamed R. Bonab and Fazli Can. A theoretical framework on the ideal number of classifiers for online ensembles in data streams. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, 2016. [cited at p. 18]

[121] G. Fumera and F. Roli. A theoretical and experimental analysis of linear combiners for multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):942–956, June 2005. [cited at p. 18]

[122] Josef Kittler, Mohamad Hatef, Robert PW Duin, and Jiri Matas. On combining classifiers. *IEEE transactions on pattern analysis and machine intelligence*, 20(3):226–239, 1998. [cited at p. 18]

[123] Ludmila I Kuncheva. A theoretical study on six classifier fusion strategies. *IEEE Transactions on pattern analysis and machine intelligence*, 24(2):281–286, 2002. [cited at p. 18]

[124] Xiaofan Lin, Sherif Yacoub, John Burns, and Steven Simske. Performance analysis of pattern classifier combination by plurality voting. *Pattern Recognition Letters*, 24(12):1959–1969, 2003. [cited at p. 18]

[125] Amanda JC Sharkey. Multi-net systems. In *Combining artificial neural nets*, pages 1–30. Springer, 1999. [cited at p. 18]

[126] Kaushik Ghosh, Yew Seng Ng, and Rajagopalan Srinivasan. Evaluation of decision fusion strategies for effective collaboration among heterogeneous fault diagnostic methods. *Computers & chemical engineering*, 35(2):342–355, 2011. [cited at p. 18]

[127] Ludmila I Kuncheva. " fuzzy" versus" nonfuzzy" in combining classifiers designed by boosting. *IEEE Transactions on fuzzy systems*, 11(6):729–741, 2003. [cited at p. 18]

[128] Chun-Xia Zhang and Robert PW Duin. An experimental study of one-and two-level classifier fusion for different sample sizes. *Pattern Recognition Letters*, 32(14):1756–1767, 2011. [cited at p. 18]

[129] Li Zhang and Wei-Da Zhou. Sparse ensembles using weighted combination methods based on linear programming. *Pattern Recognition*, 44(1):97–106, 2011. [cited at p. 18]

[130] Sergey Tulyakov, Stefan Jaeger, Venu Govindaraju, and David Doermann. Review of classifier combination methods. *Machine Learning in Document Analysis and Recognition*, pages 361–386, 2008. [cited at p. 18]

[131] Peter Sollich and Anders Krogh. Learning with ensembles: How overfitting can be useful. In *Advances in neural information processing systems*, pages 190–196, 1996. [cited at p. 18]

[132] Juan José Garcıa Adeva, U Beresi, and R Calvo. Accuracy and diversity in ensembles of text categorisers. *CLEI Electronic Journal*, 9(1):1–12, 2005. [cited at p. 18]

[133] Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, 20(8):832–844, 1998. [cited at p. 18, 21]

[134] Mike Gashler, Christophe Giraud-Carrier, and Tony Martinez. Decision tree ensemble: Small heterogeneous is better than large homogeneous. In *Machine Learning and Applications, 2008. ICMLA'08. Seventh International Conference on*, pages 900–905. IEEE, 2008. [cited at p. 18]

[135] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, Aug 1996. [cited at p. 19, 46]

[136] Peter Büchlmann and Bin Yu. Analyzing bagging. *Annals of Statistics*, pages 927–961, 2002. [cited at p. 19]

[137] Louisa Lam and SY Suen. Application of majority voting to pattern recognition: an analysis of its behavior and performance. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 27(5):553–568, 1997. [cited at p. 20]

[138] George Udny Yule. *An introduction to the theory of statistics*. C. Griffin, limited, 1919. [cited at p. 20, 22]

[139] Peter HA Sneath, Robert R Sokal, et al. *Numerical taxonomy. The principles and practice of numerical classification*. 1973. [cited at p. 21, 22]

[140] David B Skalak et al. The sources of increased accuracy for two proposed boosting algorithms. In *Proc. American Association for Artificial Intelligence, AAAI-96, Integrating Multiple Learned Models Workshop*, volume 1129, page 1133, 1996. [cited at p. 21, 22]

[141] Giorgio Giacinto and Fabio Roli. Design of effective neural network ensembles for image classification purposes. *Image and Vision Computing*, 19(9):699–707, 2001. [cited at p. 21, 22]

[142] Ron Kohavi, David H Wolpert, et al. Bias plus variance decomposition for zero-one loss functions. In *ICML*, volume 96, pages 275–83, 1996. [cited at p. 21, 22]

[143] Udo Kamps. *Generalized order statistics*. Wiley Online Library, 1981. [cited at p. 21, 22]

[144] Lars Kai Hansen and Peter Salamon. Neural network ensembles. *IEEE transactions on pattern analysis and machine intelligence*, 12(10):993–1001, 1990. [cited at p. 21]

[145] Derek Partridge and Wojtek Krzanowski. Software diversity: practical statistics for its measurement and exploitation. *Information and software technology*, 39(10):707–717, 1997. [cited at p. 22]

[146] Stephen José Hanson and David J Burr. What connectionist models learn: Learning and representation in connectionist networks. *Behavioral and Brain Sciences*, 13(03):471–489, 1990. [cited at p. 22]

[147] E Ke Tang, Ponnuthurai N Suganthan, and Xin Yao. An analysis of diversity measures. *Machine Learning*, 65(1):247–271, 2006. [cited at p. 23, 24, 29, 31, 34]

[148] Naonori Ueda and Ryohei Nakano. Generalization error of ensemble estimators. In *Neural Networks, 1996., IEEE International Conference on*, volume 1, pages 90–95. IEEE, 1996. [cited at p. 23]

[149] Gavin Brown and Ludmila I Kuncheva. âgoodâ and âbadâ diversity in majority vote ensembles. In *International Workshop on Multiple Classifier Systems*, pages 124–133. Springer, 2010. [cited at p. 23, 24, 35]

[150] Anders Krogh and Jesper Vedelsby. Neural network ensembles, cross validation, and active learning. In *Advances in neural information processing systems*, pages 231–238, 1995. [cited at p. 23, 35]

[151] Ioannis Partalas, Grigorios Tsoumakas, and Ioannis Vlahavas. An ensemble uncertainty aware measure for directed hill climbing ensemble pruning. *Machine Learning*, 81(3):257–282, 2010. [cited at p. 23, 26, 29, 30, 31, 45]

[152] Zhi-Hua Zhou, Fabio Roli, Josef Kittler, et al. Multiple classifier systems. In *Proc. 2013 11th Int. Workshop Mult. Classifier Syst.(MCS)*. Springer, 2013. [cited at p. 24]

[153] Jeff W Labadie, James L Hedrick, and Mitsuru Ueda. Poly (aryl ether) synthesis. ACS Publications, 1996. [cited at p. 24]

[154] ES Mangalova, OV SHESTERNEVA, and MV SAVELYEVA. Methods of ensemble diversity creation in regression task. *YOUTH. SOCIETY. MODERN SCIENCE, TECHNOLOGY AND INNOVATION*, (13):196–198, 2014. [cited at p. 24]

[155] Xin Chen, Li Zhou, Zheng-Zhong Shao, Ping Zhou, David P Knight, and Fritz Vollrath. Conformation transition of silk protein membranes monitored by time-resolved ft-ir spectroscopy-conformation transition behavior of regenerated silk fibroin membranes in alcohol solution at high concentration. *ACTA CHIMICA SINICA-CHINESE EDITION-*, 61(4):625–629, 2003. [cited at p. 24]

[156] Zhi-Hua Zhou, Jianxin Wu, and Wei Tang. Ensembling neural networks: Many could be better than all. *Artificial Intelligence*, 137(1):239 – 263, 2002. [cited at p. 24]

[157] Hyuna Yang, Jeremy R Wang, John P Didion, Ryan J Buus, Timothy A Bell, Catherine E Welsh, François Bonhomme, Alex Hon-Tsen Yu, Michael W Nachman, Jaroslav Pialek, et al. Subspecific origin and haplotype diversity in the laboratory mouse. *Nature genetics*, 43(7):648–655, 2011. [cited at p. 25]

[158] Luiz S Oliveira, Robert Sabourin, Flávio Bortolozzi, and Ching Y Suen. Feature selection for ensembles: A hierarchical multi-objective genetic algorithm approach. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition-Volume 2*, page 676. IEEE Computer Society, 2003. [cited at p. 25]

[159] Luiz S Oliveira, Marisa Morita, Robert Sabourin, and Flávio Bortolozzi. Multi-objective genetic algorithms to create ensemble of classifiers. *Lecture Notes in Computer Science*, 3410:592–606, 2005. [cited at p. 25]

[160] Enzhe Yu and Sungzoon Cho. Ensemble based on ga wrapper feature selection. *Computers & Industrial Engineering*, 51(1):111–116, 2006. [cited at p. 25]

[161] David W Opitz and Jude W Shavlik. Generating accurate and diverse members of a neural-network ensemble. In *Advances in neural information processing systems*, pages 535–541, 1996. [cited at p. 25]

[162] Simon Günter and Horst Bunke. Optimization of weights in a multiple classifier handwrittenword recognition system using a genetic algorithm., 2009. [cited at p. 25]

[163] Ran Li, Jianjiang Lu, Yafei Zhang, and Tianzhong Zhao. Dynamic adaboost learning with feature selection based on parallel genetic algorithm for image annotation. *Knowledge-Based Systems*, 23(3):195–201, 2010. [cited at p. 25]

[164] Eulanda M Dos Santos, Robert Sabourin, and Patrick Maupin. Overfitting cautious selection of classifier ensembles with genetic algorithms. *Information Fusion*, 10(2):150–162, 2009. [cited at p. 25]

[165] Alexey Tsymbal and Padraig Cunningham. Sequential genetic search for ensemble feature selection. Technical report, Trinity College Dublin, Department of Computer Science, 2005. [cited at p. 25]

[166] D Opitz and J Shavlik. A genetic algorithm approach for creating neural network ensembles. *Combining artificial neural nets*, pages 79–99, 1999. [cited at p. 25]

[167] Laura EA Santana, Ligia Silva, Anne MP Canuto, Fernando Pintro, and Karliane O Vale. A comparative analysis of genetic algorithm and ant colony optimization to select attributes for an heterogeneous ensemble of classifiers. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–8. IEEE, 2010. [cited at p. 25]

[168] Hamid Parvin, Hosein Alizadeh, Behrouz Minaei-Bidgoli, and Morteza Analoui. A scalable method for improving the performance of classifiers in multiclass applications by pairwise classifiers and ga. In *Networked Computing and Advanced Information Management, 2008. NCM'08. Fourth International Conference on*, volume 2, pages 137–142. IEEE, 2008. [cited at p. 25]

[169] Symone Soares, Carlos Henggeler Antunes, and Rui Araújo. Comparison of a genetic algorithm and simulated annealing for automatic neural network ensemble development. *Neurocomputing*, 121:498–511, 2013. [cited at p. 25]

[170] Zili Zhang and Pengyi Yang. An ensemble of classifiers with genetic algorithm-based feature selection. *The IEEE intelligent informatics bulletin*, 9(1):18–24, 2008. [cited at p. 25]

[171] Guillaume Tremblay, Robert Sabourin, and Patrick Maupin. Optimizing nearest neighbour in random subspaces using a multi-objective genetic algorithm. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 1, pages 208–211. IEEE, 2004. [cited at p. 25]

[172] YongSeog Kim, W Nick Street, and Filippo Menczer. Optimal ensemble construction via meta-evolutionary ensembles. *Expert Systems with Applications*, 30(4):705–714, 2006. [cited at p. 25]

[173] Jared Sylvester and Nitesh V Chawla. Evolutionary ensemble creation and thinning. In *Neural Networks, 2006. IJCNN'06. International Joint Conference on*, pages 5148–5155. IEEE, 2006. [cited at p. 25]

[174] César Guerra-Salcedo and Darrell Whitley. Genetic approach to feature selection for ensemble creation. In *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 1*, pages 236–243. Morgan Kaufmann Publishers Inc., 1999. [cited at p. 25]

[175] Ulf Johansson, Tuve Lofstrom, Rikard Konig, and Lars Niklasson. Building neural network ensembles using genetic programming. In *Neural Networks, 2006. IJCNN'06. International Joint Conference on*, pages 1260–1265. IEEE, 2006. [cited at p. 25]

[176] Balint Antal and Andras Hajdu. An ensemble-based system for microaneurysm detection and diabetic retinopathy grading. *IEEE transactions on biomedical engineering*, 59(6):1720–1726, 2012. [cited at p. 25]

[177] Hamid Parvin, Miresmaeil MirnabiBaboli, and Hamid Alinejad-Rokny. Proposing a classifier ensemble framework based on classifier selection and decision tree. *Engineering Applications of Artificial Intelligence*, 37:34–42, 2015. [cited at p. 25]

[178] Prodip Hore, Lawrence O Hall, and Dmitry B Goldgof. A scalable framework for cluster ensembles. *Pattern recognition*, 42(5):676–688, 2009. [cited at p. 25]

[179] Mohammad Ali Bagheri, Qigang Gao, and Sergio Escalera. A framework towards the unification of ensemble classification methods. In *Machine Learning and Applications (ICMLA), 2013 12th International Conference on*, volume 2, pages 351–355. IEEE, 2013. [cited at p. 25]

[180] Lior Rokach. Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1):1–39, 2010. [cited at p. 25]

[181] Juan José Rodriguez, Ludmila I Kuncheva, and Carlos J Alonso. Rotation forest: A new classifier ensemble method. *IEEE transactions on pattern analysis and machine intelligence*, 28(10):1619–1630, 2006. [cited at p. 25]

[182] Lean Yu, Kin Keung Lai, Shouyang Wang, and Wei Huang. A bias-variance-complexity trade-off framework for complex system modeling. In *International Conference on Computational Science and Its Applications*, pages 518–527. Springer, 2006. [cited at p. 25]

[183] Arjun Chandra and Xin Yao. Evolutionary framework for the construction of diverse hybrid ensembles. In *ESANN*, pages 253–258, 2005. [cited at p. 25]

[184] John B Carlin, John C Galati, Patrick Royston, et al. A new framework for managing and analyzing multiply imputed data in stata. *Stata Journal*, 8(1):49–67, 2008. [cited at p. 25]

[185] Grigorios Tsoumakas, Ioannis Partalas, and Ioannis Vlahavas. An ensemble pruning primer. *Applications of supervised and unsupervised ensemble methods*, pages 1–13, 2009. [cited at p. 25, 35]

[186] A Terrece Pearman, Wan-Yin Chou, Kimberly D Bergman, Malini R Pulumati, and Nicola C Partridge. Parathyroid hormone induces c-fos promoter activity in osteoblastic cells through phosphorylated camp response element (cre)-binding protein binding to the major cre. *Journal of Biological Chemistry*, 271(41):25715–25721, 1996. [cited at p. 26]

[187] Gonzalo Martınez-Munoz and Alberto Suárez. Aggregation ordering in bagging. In *Proc. of the IASTED International Conference on Artificial Intelligence and Applications*, pages 258–263. Citeseer, 2004. [cited at p. 26, 30, 45]

[188] Robert E Banfield, Lawrence O Hall, Kevin W Bowyer, and W Philip Kegelmeyer. Ensemble diversity measures and their application to thinning. *Information Fusion*, 6(1):49–62, 2005. [cited at p. 26, 30, 31, 45]

[189] Sugato Basu. Semi-supervised clustering with limited background knowledge. In *AAAI*, pages 979–980, 2004. [cited at p. 26]

[190] Gabriele Zenobi and Padraig Cunningham. Using diversity in preparing ensembles of classifiers based on different feature subsets to minimize generalization error. *Machine Learning: ECML 2001*, pages 576–587, 2001. [cited at p. 26]

[191] Piotr Mirowski, Sining Chen, Tin Kam Ho, and Chun-Nam Yu. Demand forecasting in smart grids. *Bell Labs technical journal*, 18(4):135–158, 2014. [cited at p. 26]

[192] Giorgio Giacinto, Fabio Roli, and Giorgio Fumera. Design of effective multiple classifier systems by clustering of classifiers. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 2, pages 160–163. IEEE, 2000. [cited at p. 27]

[193] Aleksandar Lazarevic and Zoran Obradovic. Effective pruning of neural network classifier ensembles. In *Neural Networks, 2001. Proceedings. IJCNN'01. International Joint Conference on*, volume 2, pages 796–801. IEEE, 2001. [cited at p. 27]

[194] Bart Bakker and Tom Heskes. Clustering ensembles of neural network models. *Neural networks*, 16(2):261–269, 2003. [cited at p. 27]

[195] David E Goldberg. Genetic algorithms in search, optimization, and machine learning, 1989. *Reading: Addison-Wesley*, 1989. [cited at p. 27]

[196] Vasudha Bhatnagar, Manju Bhardwaj, Shivam Sharma, and Sufyan Haroon. Accuracy–diversity based pruning of classifier ensembles. *Progress in Artificial Intelligence*, 2(2-3):97–111, 2014. [cited at p. 27]

[197] Yang Yu, Yu-Feng Li, and Zhi-Hua Zhou. Diversity regularized machine. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011. [cited at p. 29, 34]

[198] Rich Caruana, Alexandru Niculescu-Mizil, Geoff Crew, and Alex Ksikes. Ensemble selection from libraries of models. In *Proceedings of the twenty-first international conference on Machine learning*, page 18. ACM, 2004. [cited at p. 30]

[199] Derek Partridge and William B Yates. Engineering multiversion neural-net systems. *Neural Computation*, 8(4):869–893, 1996. [cited at p. 30]

[200] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30, 2006. [cited at p. 33]

[201] Lior Rokach. Collective-agreement-based pruning of ensembles. *Computational Statistics & Data Analysis*, 53(4):1015–1026, 2009. [cited at p. 35]

[202] Peter R Ebeling. Transplantation osteoporosis. *ASBMR Primer on the Metabolic Bone Diseases and Disorders of Mineral Metabolism*, page 279, 2009. [cited at p. 46]

[203] D Mihailovic. Optical experimental evidence for a universal length scale for the dynamic charge inhomogeneity of cuprate superconductors. *Physical review letters*, 94(20):207001, 2005. [cited at p. 48]

[204] I. Bellido and E. Fiesler. *Do Backpropagation Trained Neural Networks have Normal Weight Distributions?*, pages 772–775. Springer London, London, 1993. [cited at p. 55]

[205] Boris Barbour, Nicolas Brunel, Vincent Hakim, and Jean-Pierre Nadal. What can we learn from synaptic weight distributions? *Trends in Neurosciences*, 30(12):622 – 629, 2007. [cited at p. 55]

[206] Elizabeth Gardner. The space of interactions in neural network models. *Journal of physics A: Mathematical and general*, 21(1):257, 1988. [cited at p. 55]

[207] Nicolas Brunel, Vincent Hakim, Philippe Isope, Jean-Pierre Nadal, and Boris Barbour. Optimal information storage and the distribution of synaptic weights. *Neuron*, 43(5):745 – 757, 2004. [cited at p. 55]

[208] Benoî t Perthame, Delphine Salort, and Gilles Wainrib. Distributed synaptic weights in a lif neural network and learning rules. *Physica D: Nonlinear Phenomena*, 2017. [cited at p. 56]

[209] Hirotugu Akaike. A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6):716–723, 1974. [cited at p. 58]

[210] Gideon Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978. [cited at p. 58]

[211] Kenneth P Burnham and David R Anderson. Multimodel inference: understanding aic and bic in model selection. *Sociological methods & research*, 33(2):261–304, 2004. [cited at p. 59]

[212] Muhammad AO Ahmed, Luca Didaci, Giorgio Fumera, and Fabio Roli. Using diversity for classifier ensemble pruning:an empirical investigation. *Theoretical and Applied Informatics*, 29(1):1–16, 2018. [cited at p. 69]

[213] Muhammad Atta Othman Ahmed. Trained neural networks ensembles weight connections analysis. In *The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2018)*, pages 242–251, Cham, 2018. Springer International Publishing. [cited at p. 69]

[214] Bahram Lavi and Muhammad Atta Othman Ahmed. Interactive fuzzy cellular automata for fast person re-identification. In *The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2018)*, pages 147–157, Cham, 2018. Springer International Publishing. [cited at p. 69]

[215] Muhammad Atta Othman Ahmed, Omar Reyad, Yasser AbdelSatar, and Nahla F. Omran. Multi-filter score-level fusion for fingerprint verification. In *The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2018)*, pages 624–633, Cham, 2018. Springer International Publishing. [cited at p. 69]