UNIVERSIDAD DE LAS PALMAS
DE GRAN CANARIA

SIANI

Diversidad en ensamblados neuronales.
Aplicación a la detección simultánea de fungicidas

(Diversity in neural ensembles.
Application in simultaneous resolution of fungicides)

Yeray Álvarez Romero

Tutor:
Antonio Carlos Domínguez Brito

Co-Tutora:
Carmen Paz Suárez Araujo

Master en Sistemas Inteligentes y Aplicaciones Numéricas

Diciembre de 2013

Una ciencia que ve la formación de la praxis,
a la cual sirve y es inherente, como algo que está más allá de ella,
y que se satisface con la separación del pensar y el actuar,
ya ha renunciado a la humanidad.
Horkheimer 1974, 270

Por democracia real entiendo la unión de lo que hay, de lo existente, y los conceptos que se usan para legitimar lo existente. La relación entre ambos consiste en que el concepto está ya presente en lo que hay, configura su orientación futura y sirve de canon crítico para la realización presente.
Cortina 1991, 223

Todo lo que pretendía reivindicar para sí la palabra democracia o lo que se aclamó como democracia real ha desaparecido de la noche a la mañana. El vencedor es la democracia liberal, ya sólo queda la democracia formal.
Sartori 1991, 459

El derecho en tanto que pretensión es ético; pero en tanto positividad, es político.
Aranguren 1991, 212

¿No tienes enemigos?,
¿Es que jamás dijiste la verdad o amaste la justicia?
Santiago Ramón y Cajal.

Libertad sin socialismo es privilegio e injusticia;
Socialismo sin libertad es esclavitud y brutalidad
Piotr Alekséyevich Kropotkin

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter will be a brief description of the main pillars that will support the understanding and conceptualization of the problem to be solved, i.e. the resolution of difficult fluorescent spectra of multi-fungicides mixtures employing diverse neural ensembles. Theoretical foundations of the computational methods, fluorescence and fungicides characteristics will be given.

1.1 Fungicides and Fluorescence

1.1.1 Fungicides

Fungi and oomycetes are the causative agents of many diseases, producing numerous losses in agricultural production worldwide. These pathogens (phyto-, from the Greek, vegetable) have co-evolved with their hosts, developing extremely efficient mechanisms to cause infections, grow, multiply and spread to all floors. Considering the ubiquitous nature and ability of fungi and oomycetes to cause epidemics in a relatively short period of time, the infection control strategies are necessary in order to ensure the productivity of current agricultural production. Chemical control measures is particularly common in the control of fungal infections where a lot of these measures are due to the use of fungicides. Fungicides are a specific type of pesticide that helps control fungal infections, preventing the growth of or removing them. Represented as the group of pesticides with greatest potential risk to humans, since about 90% of the fungicides currently or in the recent past have proven carcinogenic effects in experimental animals [Loewy, 2000]. Furthermore, in the present state of knowledge, plant protection products as fungicides are essential, and which form the basis of defending crops against pests, and make profitable production of quality food. Thus, without the use of fungicides would result in a loss of production between 20 and 30% averaged, often reached values of up to 75%. Consequently, the use of pesticides is necessary for technical, economic and social reasons, in so that it can feed an exponentially growing society, to the point that feeding problems in many developing countries would be eliminated by simply reducing crop losses where they occur, improving production techniques and conservation of crops [Márquez, 2008]. It is estimated that more than 3500 organic pesticides are used, they can all be pollutants being washed away from the fields to aquatic ecosystems, which are introduced into the food chain, killing various life forms necessary in the balance of some ecosystems. The movement and dispersal in ecosystems of a pesticide causes environmental pollution. Dispersion and fate will depend on the characteristics of the ecosystem and the pesticide, formulation type, application method, environmental and agricultural conditions. In intensive agriculture, water is one of the resources that are at high risk of pesticide contamination due to the dynamics that have these products on the environment and the many factors involved (soil characteristics, climate, agricultural management practices and forest, among others).

1.1.1.1 Benzimidazole fungicides

The class of fungicides that are to be analysed in the proposed system are Benzimidazol type, i.e., fungicides whose main component is the Benzimidazol. Fungicides under study are benzimidazole derivatives, benomyl (BM), carbendazim, (MBC), fuberidazole (FB) and thiabendazole (TBZ). These fungicides are the longest of agrochemicals family, which have an imidazole ring containing both acidic nitrogen atom as core. Are often used for prevention in the treatment of parasitic infections in agriculture and aquaculture as they are efficient at low doses, in addition to inhibiting the development of a wide variety of fungi. Benzimidazoles founds some applications such as pre or post product collection for the control of a large number of pathogens. Applied directly to the soil or sprayed over crop fields [Wu et al., 2009]. Many of these components remain in the environment after application, and some even do so for many years. This group includes analogues to thiabendazole and benzimidazole carbamates components. The changes that have been made in the brief history of fungicides have produced benzimidazoles with a lower rate of elimination, more power and a broader spectrum of activity. Carbendazim (MBC), which is the most common stable metabolite of benomyl (BM) and thiophanate -methyl (TPM), is considered more toxic fungal main precursor of benzimidazoles. Thus, the regulatory limits for these fungicides are expressed in terms of the MBC, the only such measure for food safety or environmental impact measurement of total benzimidazole found in a sample [Danaher et al., 2007]. Although there are numerous benefits from the use of benzimidazole in the field of agriculture, the effects that can cause both the environment and public health can not be neglected. Have been associated many toxic effects of chronic exposure to these components, such as teratogenicity, birth defects, polyploidy, diarrhea, anemia, pulmonary edema, or necrotic lymphadenopathy. Because of its use, both intense and extensive, the regulatory framework imposes higher concentration limits, set in $0.01-10mg*kg^{-1}$, depending on the combination of fungicides (Plant Protection-Pesticide Residues -Regulation (EC) No. 396, 2005). Particularly for many benzimidazoles, the tolerance to a residue has been recently defined as a sum of precursors or metabolites relations elements instead of individual components [Danaher et al., 2007]. The European water framework directive (Directive 2006/11/CE/4) establishes a maximum concentration (MCL) of $0.1mg*L^{-1}$ for many of the components that have the benzimidazoles in natural waters and total concentration of all pesticides of $0.5mg*L^{-1}$. It is necessary therefore to determine low concentrations of these substances, increasing the complexity of the analytical methods with high sensitivity, selectivity and resolution that must be applied to both samples of soil, sediment, water and other types of samples environmental. The selection of an appropriate treatment protocol that allows to develop a determination in a sample with multiple residues of benzimidazole components is currently a challenge due to the properties of the chemical sample. The development of highly sensitive methods for this purpose is necessary, and it is usually done before the test steps and appropriate instrumental techniques. Many efforts have been invested in the past decades to develop and validate analytical methods for quantifying components of benzimidazole and its metabolites in environmental samples at concentration levels below those established by law and maximum concentration levels [Rodríguez et al., 2010].

Fungicides in the Canary Islands Certain fungicides used in this project, such as TBZ, are extensively used in banana crops, the agricultural production base in the Canary Islands. After making cuts in the crown, the area is sprayed with thiabendazole to prevent the growth of fungi that cause rot. For this reason, bananas are impregnated with these toxins in the outer layers. In order to move this product to market, it is necessary to submit the banana up to 3 washings to eliminate these and other components. Although chemical methods pose toxicity reduction by photocatalysis, achieving effective results, it still requires the measurement of the concentrations and the indentification of components through faster, effective and inexpensive methods. Thus, the results of this project can be considered as a fast, cheap, effective and modern identification of TBZ in the islands.

1.1.2 Fluorescence

Also called fluorometry or spectrofluorometry, is a type of electromagnetic spectroscopy that analyse the fluorescence of a sample, i.e. the electromagnetic radiation to a stimulus in the same nature. When a chemical compound absorbs ultraviolet or visible electromagnetic radiation passes to an excited electronic state. Many substances in that state dissipate excess energy as heat, through collisions with neighbouring atoms or molecules, as in the absorption spectrophotometry. However, many compounds lost only part of this excess energy as heat, and residual energy emitted in the form of electromagnetic radiation of different frequency than that absorbed, and that can be used for analytical purposes, see Figure 1.1.

All fluorescent systems, such as the BFs, are generally characterized by an excitation (or absorption) spectra and an emission spectra. These kind of spectra also allow their synchronous spectra to be acquired. Carbendazim 99.7% (methyl (1 H-benzimidazol-2-yl) carbamate), benomyl 99.3% (methyl 1-(butylcarbomayl) benzimidazole-2-yl carbamate), thiabendazole 99.6% (2-(4-thiazol)benzimidazole) and fuberidazole 99.6% (2-(2furanyl)-1 h-benzimidazole) were obtained from Riedel-de Haen (Seelze, Germany). All conventional and

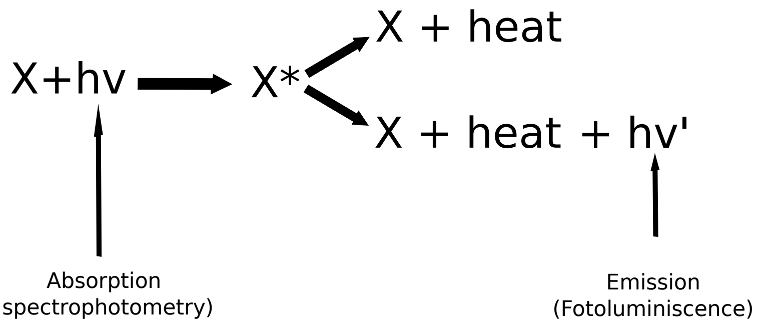


Figure 1.1: Paths of energy relaxing after an electromagnetic excitation

synchronous fluorescence spectra in the study were acquired using a Perkin-Elmer LS-50 luminescence spectrophotometer (Beaconsfeld, Buckinghamshire, UK) fitted with a xenon discharge lamp.

Fluorescent spectroscopic techniques in mixture resolution has been proven as a versatile analytical tool because of its selectivity [Suarez Araujo et al., 2010]. It main drawback lies in the spectral interferences of the fluorescence systems present in a mixture. Thus, the resolution of complex mixtures with a high degree of overlap among its compounds constitute a challenge in chemical analysis. Amongst the most commonly instrumental techniques we found layer chromatography, gas chromatography, and high performance liquid chromatography, that are used to tackle this difficult problem [Suarez Araujo et al., 2010, Vassilakis et al., 1998, Sabik and Jeannot, 1998]. However complications such as cost, time, analytical complexity, and the need for substance pre-treatment must also be considered if they are to be used. A search for alternative techniques and methods would certainly be convenient when studying this problem. A complementary approach to the instrumental and chemometric methods to obtain the resolution of multi-analyte systems could be based on neural computation which has been biologically inspired. With this purpose in mind, the use of instrumental methods are the only one which are capable of be combined with a computational solution and thus obtaining its advantages. Following, we make a short comparison of chemical and instrumental methods, offering reasons to consider preferable the instrumental techniques in conjunction with computational methods in the chemical field over the traditional chemical methods.

1.1.2.1 Comparison of chemical and instrumental methods

We include the chemical and physical methods within the scope of the analytical method, but we understand the classic chemical method with analysing chemical reactions, observation, measurement, determination of chemical changes, while contemplating the instrumental as a method that performs measurements on physical interactions, such as spectrometry. We can add instrumental methods that require very complex instruments compared to traditional pipettes, burette, etc. of the chemical instrumentation and therefore more expensive, so we have to analyse those compared methods in terms of precision, accuracy, sensitivity, speed, selectivity and cost.

- Chemical methods are generally more accurate in absolute terms than the instrumental [González Pérez and Hernández Hernández, 2002]

- Chemical methods are more accurate for high concentrations, increasing the errors of these methods as lower concentrations, conditions under which the methods are more accurate instrumental [González Pérez and Hernández Hernández, 2002]

- Instrumental methods are more sensitive than chemical [González Pérez and Hernández Hernández, 2002]. This means that are capable of analyzing much smaller measures analyte chemical method because the useful range which are capable of measuring is wider

- Selectivity measures the ability of identifying elements. For this reason, this parameter depends on the degree of interference or overlap between values of different analytes. The selectivity in instrumental methods is extremely diverse, with some very selective, especially, some optical methods such as spectrofluorimetric [González Pérez and Hernández Hernández, 2002].

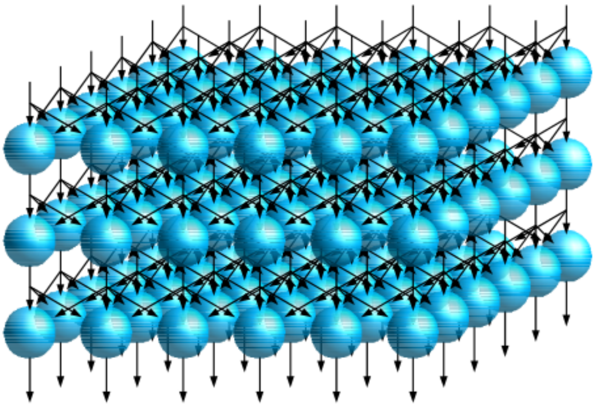


Figure 1.2: Tridimensional architecture of an Artificial Neural Network. Fuente: [García Báez, 2005]

- With regard to speed, instrumental methods are faster for serial determinations or routine. Require a long setup, but then are usually faster [González Pérez and Hernández Hernández, 2002].
- While the apparatus necessary for the chemical method is much cheaper in terms of time and personnel is much more expensive, not only for the training required to develop, but by the productivity that a chemical could offer.
- Finally instrumental methods are more susceptible of being automated by computer, due to the nature of the signals with which they work. Chemical methods suffer from this feature generally.

1.2 Neural computation

In much of the classical definitions of artificial intelligence underlies the idea of imitating human reasoning. This being so, it seems logical that since the beginning of the information age arose the curiosity to understand and try to imitate the functioning of our brain, but not just from the point of given applications which perform similar to it, but even, to imitate physiological behaviour occurred in neuronal and non-neuronal systems. Logically, emulating such behaviour poses a clear way to get those functionalities we sought. The first artificial neural network model (ANN) with universal computational capabilities was proposed by Warren McCulloch and Walter Pitts. A network of formal neurons, based on the knowledge of which is available on nerve function [McCulloch and Pitts, 1943]. These studies were then followed by others that gave completeness and improved the McCulloch-Pitts proposal. Among the most featured, were the researches and developments on learning conducted by Donald Hebb [Hebb, 1949], Lashley [Lashley, 1950], McCulloch [McCulloch, 1959] and Blum [Blum, 1962], to increase the computational power of networks of formal neurons with afferent interaction. Lately, Rosenblatt [Rosenblatt, 1962] with the Perceptron entered in the analog level of artificial neural networks. Since then, there have been many advances, that have allowed to turn the neural computation as a key element, in the resolution of many problems in a wide domain of applications, and a suitable computational paradigm for determining the style of computing in the brain, the algorithms used in it, and the implementation thereof.

The neural computation can be understood as distributed computing and parallel, adaptive and selfprogrammable conducted through modular tridimensional architectures and organized in layers consisting of a large number of processing elements with a high degree of connectivity, see Figure 1.3, and adaptive capability, under different learning models [Suárez Araujo, 1996]. Its main information processing structure are artificial neural networks, where the adjective neural means that the inspiration are biological neural networks. The ANNs studies and uses some strategies used by biological neurons for processing information. Neural computing is one of the several models of computation that have been defined throughout history, considering the ANNs as another approach to the problem of computing and is one of the five alternatives that were studied between 30 and 40, in which clearly came out winning the Von Neumann architecture.

1.2.1 Introduction to artificial neural networks

We can understand the artificial intelligence as a science of the artificial. This concept is the opposite or complementary to the natural science model, which is guided by the experimental scientific method and aspires to formalize the reality in order to obtain predictions of nature. The science of the artificial, understood as opposed to experimental science, does not begin from a set of experiences as a method to generate a general solution, but based on a functional specification aims at the synthesis of a system that satisfies the partial reality defined. Artificial intelligence works with information and knowledge, abstracts elements independent of the physical system that supports them. In the computer field, mainly two methodologies differ leading to this synthesis. Symbolic computation and connectionist computing. In the symbolic-mathematical model, we defined by logical-mathematical formalisms human behaviour with intent to imitate. In a connectionist model, we aims to provide intelligent behaviour to a system by partial representation of bio-inspired structures in the brain, which is composed of many relatively simple elements but strongly interconnected. Artificial neural networks (hereinafter ANNs), born from this desire of imitating human brain structures. The ANNs can be defined as massive networks, parallel and interconnected composed of simple elements (usually adaptive) and organizational hierarchically, trying to interact with real world objects in a manner analogous to as biological nervous system does [Kohonen, 1988]. There is not an unique definition of ANN accepted by the entire scientific community, but exists a diversity consistent with the exact meaning of these information processing systems. They have a great use that reveal its rich and varied character, from the pure mathematics to cognitive, through the engineering.

1.2.2 Framework of ANN characterization

Following the Rumelhart criteria, the general framework of an ANN is determined by eight components that are [Rumelhart et al., 1987]:

- A set of process units, containing, each of them, local memory.
- An activation state for every process unit.
- An Output function for each process unit.
- A connectivity patterns between process units, where each connection has an associated weight.
- A propagation rule for propagating patterns of activities through the network of connections. Also called network function
- An activation rule for combining the inputs arriving to a unit with the current state of the unit to produce a new level of activation. It is also called activation function.
- A learning rule, by which can change the connectivity patterns based on experience. It can be based in the development of new synaptic connections, in the loss or modification of synaptic weights.
- An environment representation. A local information environment and a global information environment.

Following this framework we characterize the ANNs in the following three levels:

1. Topology connections (covers neural structure)
2. Neurodynamics
3. Learning

1.2.2.1 Connections topology

The connections topology is an essential part of the neural structure called an artificial neural network and indicates the way in which the various elements of a network are interconnected [Hecht-Nielsen, 1990]. Formally, a network structure is described by a directed graph in which the nodes are neurons or processing units and arcs oriented communication channels between neurons. Each arc has an associated real value indicating the weight

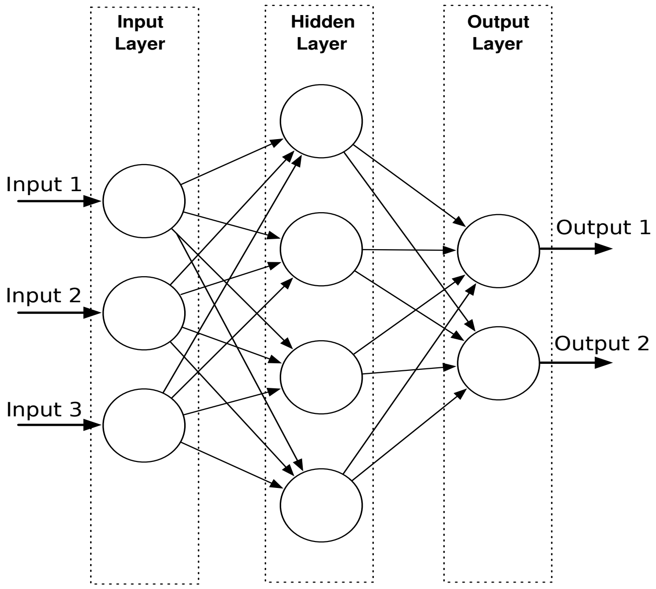


Figure 1.3: ANN with hierarchical structure. Source: own elaboration

of the connection. The communication scheme between different neurons can be of different forms: essentially distinguishes between hierarchical and non-hierarchical scheme of connections.

The most relevant topology, the hierarchical one, see figure 1.3, the units are structured in layers, which consist of sets of units with similar characteristics in terms of type of connection and type of processing they perform. The lower level corresponds to the input neurons, the upper level to the output neurons. Between them, are interspersed any number of hidden layers of cells, in principle, respecting each inner layer only receives input signals from the preceding layer and output the signals only to the next layer. Notation cells is performed in the inlet to the outlet and from top to bottom. The computer then takes place according to the bottom-up scheme. By convention, the outputs of the layer number zero are the input signals of the network.

Typically, the structural organization of the processing elements that comprise an ANN is layered. We can find plain ANNs neuronal structure, with a single layer of processing elements, also called monolayer ANNs where all constituent units and wiring are similar in processing. ANNs also exist in varying degrees of depth, this degree of depth is given by the number of layers that compose and are known as multilayer networks.

1.2.2.2 Neurodynamics

The neurodynamics of an ANN can come expressed as continuous functions over time or using discrete functions. Generically these changes can be performed synchronously or asynchronous to all or part of the network elements. Among the most common forms within asynchronous methods have the random, consisting of randomly selected unit process that will compute its output. Another common way is by following the topological order imposed by the connection, the computations are carried out synchronously layer by layer, from the next to the inputs to the output.

1.2.2.3 Learning

We can define learning as the ability of a system to absorb information from the environment without the need for a system to be programmed externally. The ANNs learning follows a model composed of two phases [Judd, 1990]: loading mode (learning) and recovery mode (execution). Charging mode is where learning takes itself, from the data received from the environment, the ANN is able to process and store the information extracted from it in their synaptic connections. On the other hand, the recovery mode from issues raised from the environment allows to get answers by using proper processing knowledge stored in ANN. The vast majority of models distinguish ANNs clearly between these two phases, however in other models this distinction is not

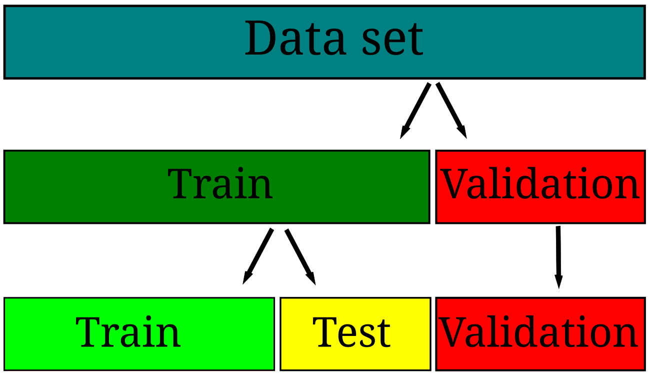


Figure 1.4: Data set distribution on supervised scheme

so clear, keeping these ANNs in a single continuous learning mode and execution. Processes learning are given because ANN is stimulated by the environment, causing changes in the network to attempt to reach a new way of efficiently responding to the task specified. Changes in the network may come given by changes in the topology of connections andor changes in the synaptic weights. A first way could be to leave that responsibility to the designer, who can set the necessary changes by proper formulation of the problem to resolve. Many other times, the network is left in charge making the learning process itself from a training data set, which is called learning from examples, in such cases, use learning algorithms that update iteratively such necessary weights andor topological changes. sUnlike neurodynamics, learning algorithms allows performing non-local processes.

The most common classification is based on how the learning process is guided, thus, three different paradigms are distinguish [Haykin, 1994]:

Supervised learning for each input pattern to the network provides the correct answer to this pattern, providing guidance for the necessary adjustments to the synaptic weights. To obtain the generalization capability on a supervised ANN it is necessary to divide the dataset into three different subgroups, see Figure 1.4. The first one is called the train or learning set and it is used during the learning phase. A second one, called test set is used as a first measure of generalization, helping to determine the stopping criteria of the learning process, in order to avoid the overfitting problem, or the threshold value. The last one, called the validation set, is used to give a final measure of generalization-accuracy of the ANN.

Reinforcement learning: from the response of the network from an input pattern, you have to return a scalar evaluation of the response, indicating correctness of the response. Many authors consider this paradigm as part of supervised learning, matching it to what would be a learning tutor.

Unsupervised learning: in this case the network does not receive any tutoring, having to organize their outputs based on redundancy and structures that can be detected on the inputs. This paradigm is often called self-organized learning.

All learning process, regardless of the type to which it belongs, according to the taxonomy made, entails a way that the synaptic weights are modified or updated. This form is what we call learning rules (laws, algorithms), which are mathematically expressed, depending on whether continuous or discrete, differential equations or difference equations based-systems. Considering the different types of rules we can talk about another classification of learning processes, which distinguish four different types [Haykin, 1994] [Jain et al., 1996]:

Hebbian: as mentioned, it is the oldest type of learning, forming the basis of many subsequent learning laws. It is within the category of learning by coincidence [Hecht-Nielsen, 1990]. This law, in the field of neural computation comes from statements made by Donald Hebb in 1949 [Hebb, 1949], based on neurobiological observations indicate that tend to reinforce synaptic connections to correlated firing neurons. They are based on many unsupervised learning.

Competitive: also have a high inspiration neurobiological since experiments have demonstrated their use in the formation of topographical maps in brain and sensory nerve cell targeting striatum cortex. It is based on a process of competition between units assigned exclusive representation, (trigger) to a group of input patterns.

Error correction: such rules, based on error correction, try to correct the error that occurs in the network, by comparing the desired output with the actual output of the network, which are mainly applied in the monitored paradigm. The way to fix the error is essentially based on gradient descent method, which consists in building a global function error that is minimized by moving in the direction of maximum slope. The biological plausibility is not as clear as in the previous cases, however practical results have been achieved very good at some examples of this type of rules as the perceptron rule [Rosenblatt, 1962], delta rule [Widrow, 1962] or backpropagation algorithm [Werbos, 1974].

Energy optimization: within them, the best known is the Boltzmann learning rule [Hinton and Sejnowski, 1986]. They are characterized by using an energy function, determined by individual states occupied by individual neurons, that is optimized. It is related to error correction in the sense that both are based on the minimization of a function, but in this case the function is not defined in terms of network error. Specifically Boltzmann rule is considered a derivative from stochastic learning information theory and thermodynamic principles.

Other features that have the ANNs are computational capacity, seen as the kind of problems that a network is able to solve, and a computational complexity, which determines how fast or slow they can prove their learning to reach a convergence in their results.

1.2.2.4 Features and Capabilities of ANNs:

The ANNs have a lot of architectural and functional properties that make them particularly suited to address highly complex problems, based on behavior, etc. in real time. In this section we present an overview of all those properties that characterize ANNs, not only by direct quoting that properties, but also inferring them by analyzing characteristics that describe the types of problems that ANNs naturally and successfully addressed. So we began by presenting a set of relevant properties of ANNs [Haykin, 1994]:

Generalization: ANNs are not limited to solving only those problems for which they are trained, they are able to carry out a process of generalization and solve problems that had never been presented before. Contextual information is treated with great ease, since the activation process of any element may be potentially influenced by any other network element.

Fault Tolerance: ANNs exhibit graceful degradation, meaning that fail over processing elements or connections overall system, does not come to a standstill, its effectiveness may be degraded somewhat but, as happens in the brain neuronal death, this need not entail catastrophic failures. This property is supported primarily by the redundancy that may exist between the processing units, as the network knowledge is distributed among the different synaptic connections. Even some types of neural architectures are able to undertake a process of rehabilitation and neuronal reconfiguration such that the responsibility for processing and or function of the damaged areas are redistributed to healthy areas, as occurs in certain brain areas.

Graded responses: the answers can be graded, thus indicating network confidence in their results. This is an important property that may, in some cases, must be taken into account in the decision making.

Uniformity in the analysis and design: the guidelines followed in any domain for analysis and design of ANNs based applications are the same, which is difficult to achieve by other computational approaches.

Apart of these properties do not forget either those already discussed in previous sections, such as massive parallelism inherent to the ANNs, which allows high-speed processing and facilitates hardware implementations, and the propensity towards learning, being the most important existing paradigm in this regard.

Considering the nature and characteristics of the problems to be solved, we find another way to view the properties and capabilities of ANNs. It is absolutely necessary to analyze the nature of the problems to be addressed to determine the viability of the neural computing paradigm for such problems. That is, if this paradigm is presented as the most suitable, or can be combined with other approaches, because what we must understand that is not all the problems could be resolved using ANNs. In general, the problems addressed by ANNs usually need any of these features, which are characteristic of artificial neural networks:

Data Availability: since ANNs are guided by examples we need to have a high amount of data to be used in the learning process, testing and validation. If there were insufficient data we must go to the creation of the so-called false data.

Problems based on behaviour, not knowledge: problems in which there is no knowledge on how to solve the function that represents it, we could say, the algorithm to address resolution. These are clear cases in which ANNs have much to say because by learning ANNs can infer and represent that knowledge and achieve the optimal solution to the problem.

Presence of inaccuracy and or incompleteness: as happens with biological systems, ANNs are qualified to address inaccuracies and incompleteness in the data, such as those due to noise, interference or overlapping data and missing data.

Need for real-time execution: upon completion of the learning process of the ANNs, performance is very fast, making them particularly suited to solve problems in real time. This happens both with software implementations or simulations, and more with hardware ones.

Non-stationary environments: ANNs are trained to work in non-stationary environments where the statistical characteristics of the data or the problem to solve vary over time, allowing residual learning processes that adapt to such changes.

In general, applications that are useful for ANNs is marked by the following tasks that are able to address [Haykin, 1994] [Jain et al., 1996]:

Pattern Classification: the set of data or patterns that is presented to the ANN are divided into a number of categories or classes. The ANN has to learn to distinguish the class of a given pattern, typically generating discriminant functions or decision boundaries from the training set. In cases where there is not a priori information about which category the patterns belong, this task is often called clustering.

Approximation of functions: from a data set, possibly noisy, indicating the input and output of an unknown function, it consists in estimate this function. This problem is also known in the statistical literature as regression. The pattern classification problem could also be considered as a particular function approximation problem, for both tasks it is used the term of functions mapping.

Associative memories: is able to store information so that it can be retrieved later. The event that triggers the recovery may be the presentation of a partial or distorted input contained in memory, in these cases is usually called content addressable memory. If the dimensions of the entries match the outputs is also called self-association, as opposed to the hetero-association, in which it does not occur. This kind of behaviour, as in the above tasks, is also called pairings mappings of input patterns with output patterns.

Prediction: is, from a temporal sequence of data occurred in the past, predict the next value in the sequence. Weather forecasts or predictions of the stock market are among their typical applications.

Table 1.1: Most commonly used radial basis functions. Source: [Wirth, 2001]

Name	Mathematical expression	Parameters
Gaussian	$\varphi(r) = e^{\frac{-r^2}{2\sigma^2}}$	With normalization parameter $\sigma > 0$
Multi-quadratic	$\varphi(r) = (r^2 + \sigma^2)^{1/2}$	With normalization parameter $\sigma > 0$
Generalized multi-quadratic	$\varphi(r) = (r^2 + \sigma^2)^{\beta}$	$\sigma > 0$ and $\beta > 0$
Inverse multi-quadratic	$\varphi(r) = (r^2 + \sigma^2)^{-1/2}$	With normalization parameter $\sigma > 0$
Generalized & inverse multi-quadratic	$\varphi(r) = (r^2 + \sigma^2)^{-\beta}$	$\sigma > 0$ and $\beta > 0$
Cubic	r^3	

Control: the process control is to maintain a working system within a range of parameters considered appropriate, precisely, one of the most complex systems, the human body, consists of thousands of muscle fibers that must act synchronously, or the brain, so it is clear how interesting solutions based on artificial neural networks for these tasks.

Optimization: tasks where there is a set of independent variables or parameters, an objective function or cost, which depends on such variables, and a series of restrictions on the range thereof. They should look for the values of the variables without violating the constraints, minimize or maximize the objective function. In general there are many problems in different areas of knowledge that can be addressed from this perspective, even the aforementioned control tasks are closely related to optimization tasks.

Classification and evaluation problems The networks are distinguished into two main groups according to the type of problem to solve. Thus, the network of a first type, associate a particular input configuration stimulate, input or standard output. They are referred as associators, classificatory or patterns recognizers networks. The second type of networks, handle problems that requires judgement, so called evaluative networks.

Sort a network as an evaluator or idolatress can help you decide how many neurons should have and how to organize. Evaluating network has a single neuron in the output layer that indicates the result of the evaluation performed with the input information. A network recognizer has many output neurons, at least one for each element that should be recognized.

1.2.3 Radial basis function networks

The networks of radial basis neurons (RBFN) came from the hand of Moody, Darken, Renals, Poggio and Giross. The RBFN are multilayer feed-forward networks and generally fully connected . It is composed of a single hidden layer neuron whose activation function consist of a radial basis function. The output layer are linear combinations of the activation functions of the hidden layer, in perspective, a RBFN can be understood as a linear combination of nonlinear multiple local functions.

Park and Sandberg formally demonstrated that RBFN are universal approximators [Park and Sandberg, 1991], and based on those functions defines hyperspheres or hiperelipses that divides the input space, so that each neuron constructs a nonlinear and local approximation in a region of the input space. Applications of this type of ANN are varied in either analysis series, image processing , automatic speech recognition , or to perform medical diagnoses among others. In general, all kind of classification and pattern recognition problems [Sahin, 1997].

1.2.3.1 Radial basis function

Radial basis function are those whose output depends on the distance from the entrance to a point called the center. Are symmetric about $x = 0$. Define hyperspheres or hiperelipses that divide the input space.

A function of this type is defined by at least two parameters:

- Center C: Point where the function has one outer

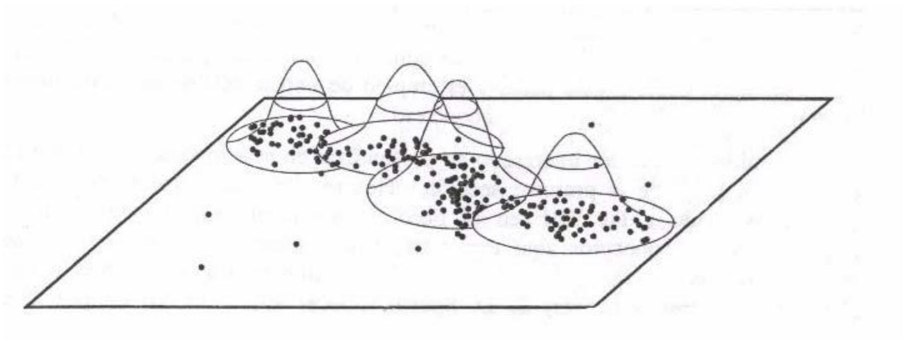


Figure 1.5: Local approximations representation

- Width σ: Magnitude of the variation of the function as it moves away from the center.

Radial basis functions have a local character functions as they have the maximum level close to the maximum of its travel when the pattern input $X(n)$ is close to the center of a neuron. As the pattern moves away from the center, the value of the function is tending to the minimum route. The outputs of the RBFN are therefore a linear combination of Gaussian functions, each of which is activated to a certain portion of the space defined by the input patterns. The most common function is the Gaussian, see equation 1.1, although there are many others in Table 1.1:

$$\varphi(r) = e^{\frac{-r^2}{2\sigma^2}} \tag{1.1}$$

1.2.3.2 Architecture

Radial basis function networks, as ANN described are composed of three main layers:

- Input layer: Feed the hidden layer transmitting the information from the environment.
- Hidden layer: Each process element i from the hidden layer has a radial basis function associated, in which manner that represents an existing class or category in the dataset, where each class is given as (C_i, σ_i). C_i represents a cluster center (weights associated with each neuron i) and σ_i represents the deviation of the width or dilation radial basis function associated with that item. The output of each hidden layer element $Z_I(n)$ is calculated as the distance that exists between the input pattern $X(n)$ to the center of the cluster C_i weighted inversely by σ_i and then applying that value of radial basis function:

  $$z_i(n) = \varphi(\sum_{j=1}^{p}(x_j(n) - c_{ji})^2)^{1/2} \tag{1.2}$$

  Where φ is a radial basis function.

- Output layer:

  Each processing element calculates the net value as a linear combination of the outputs of the processing elements in the hidden layer. The transfer function is linear activation and therefore: For a standard n, $X(n) = (x_1(n), x_2(n), ...x_p(n))$, the output of the network associated with each element k of the output layer is obtained as follows:

  $$y_k(n) = \sum_{i=1}^{m} w_{ik} z_i(n) + \mu_k \; para \; k = 1,2,...,r \tag{1.3}$$

  Where w_{ik} are the weights associated to the element k of the output layer and element i of the hidden layer, weighting each output $Z_I(n)$ from corresponding processing element of the hidden layer. The term μ_k, is a threshold term known and associated with each processing element of the output layer.

1.2.3.3 Learning

The learning process of the RBFN is produced in two steps. First, set the various parameters of the activation functions of hidden neurons and then adjusting the connection weights and thresholds of the network. These two processes can be performed either separately or simultaneously [Font Fernández et al., 2009]. The hybrid method learning consists in using the classification algorithm to calculate the input space and deviations centres, while the weights of the network connections are calculated using the back propagation algorithm.

For the calculation of the centres and deviations of radial basis functions network, first, we perform the calculation of centres giving out the input space among the centroids of the whole network. Thereby establishing a centroid dominated regions, such that one of these regions is defined by the input patterns group m nearest to the centroid which dominates the region, providing a local character to the RBFN. Observe the figure 1.5 to visualize the concept of local approximations. In it, each centroid represents a function of a neuron in the hidden layer, containing the various patterns according to the distance from the centroid of this function.

1.2.4 Backpropagation ANN

The multilayer perceptron [Werbos, 1974] or MLP ANN model is the Most Widely used in practice for solving Both classification and regression problems, having Demonstrated capacity as universal approximator its functions.

The MLP is known as a neural network model with forward propagation, which is characterized by layering of neurons whose inputs are the outputs of the preceding layer, except for the input layer, which receives information from the environment. The most commonly learning algorithm is the back propagation, based on gradient descent of an arbitrary error function [Rumelhart et al., 1986], from which we give the name of backpropagation network (BPN).

BPNs are generally used in solving complex problems. These networks learn to classify a set of pairs of inputs and output by employing a two phase cycle understood as propagation and adaptation or propagation. After the stimulus generated by the pattern units in the input layer, the generated signal is propagated to the last layer, producing an output. The output generated is then compared with the desired output and then generates an error signal that is computed for each unit of the output layer. Generated error signals are transmitted back to each of the nodes of the previous layer that receive a portion of the total error generated by the network in proportion to their contribution in it, thus correcting the weights of the network in the process called Backpropagation [Freeman and Skapura, 1991].

Each processing unit of the network learns certain characteristics of the input space, thus being able to "chop" the input space and establishing its corresponding pair in the output set. Providing a mathematical approach, the backpropagation algorithm seeks the minimum of the error function of the weight space, using the gradient descent method [Rojas, 1996]. The combination of weights which minimizes the error function is considered a solution of the learning process. Backpropagation networks, despite the similarities in the MLP architecture, differs because the use thereof downward gradient is necessary to ensure the continuity and differentiability of the error function, so that the step function of the MLP generates inconsistency with the above statement.

1.2.4.1 Architecture

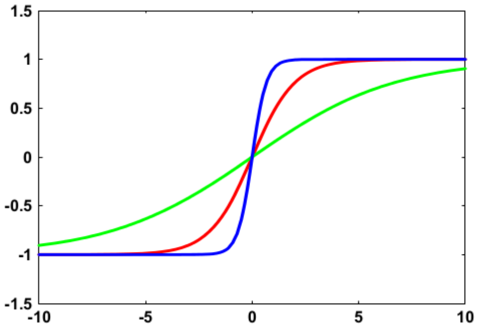
A BPN is composed of three types of layers associated with a completely hierarchical model. The number of hidden layers is undetermined, although the most common is using just one hidden layer for the vast majority of problems.

- Input Layer: The input layer is fully connected to the next hidden layer. Generally, the input layer usually employs an identity function that propagates the values associated with the data set given. Thus the output provided by the input layer is expressed as follows:

$$y_i(x) = f_{identidad}(x) \tag{1.4}$$

  where f is an identity function and i indicates the ith neuron input.

Figure 1.6: Influence of the parameter α in the hyperbolic tangent function. blue ($\alpha = 3$), red ($\alpha = 1$), green ($\alpha = 0.3$) [Villasana, 2010]



- Hidden layer: Each unit of the hidden layer, as described in neural network section, performs a weighted sum of the spread signals from the input layer to the weight associated with each connection. The result of this operation is operated by an activation function that results in a given trigger value. This value of activation will be used in spreading the signal to the next layer. The output value of each neuron in this layer can be expressed as follows:

$$y_j(x) = tanh(\sum_{i}^{n_e}(y_i * w_{ij})) \tag{1.5}$$

  Where j indicates the value of the jth neuron in the hidden layer, $tanh$ activation function is applied, n_e is the total neurons in the input layer, w_{ij} the weight associated between each neuron of the input layer and the hidden layer neuron, and where y_i is the value of output corresponding input neuron.

- Output layer: The output layer performs the same process that hidden layers, although the functions used for the activation are of threshold type, usually this is not necessary, specially in estimating the posterior probability, for which is recommend to use the same configuration in all hidden neurons.

$$y_z(x) = tanh(\sum_{j}^{n_h}(y_j * w_{jz})) \tag{1.6}$$

  Where z indicates the value of $z-th$ neuron of the output layer, $tanh$ activation function is applied, n_h is the total number of neurons of the hidden layer, w_{jz} is the weight associated between each hidden layer neuron and the output neuron, and where y_z is the output value of the corresponding hidden neuron. To determine the output value, threshold function is used with the following expression:

$$\varphi(y_z(x)) \tag{1.7}$$

Activation functions

- The hyperbolic tangent function is a continuous adaptation of the step function. Is continuous between the values $[-1,1]$ and infinitely differentiable. See Figure 1.6 to observe the influence of alpha parameter of the equation 1.8.

$$tanh(x) = \frac{e^{x\alpha} - e^{-x\alpha}}{e^{x\alpha} + e^{-x\alpha}} \tag{1.8}$$

- The sigmoid function is continuous between the values $[0,1]$ and infinitely differentiable. The trigger value is calculated as follows:

$$sigmoid(x) = \frac{1}{1+e^{-\alpha * x}} \tag{1.9}$$

  The form taken by the sigmoid function is also possible to transform via parameter α.

Output functions The threshold function is applied as output function of activation values. Thus, the output function is expressed as:

$$\varphi(x) = \begin{cases} 1 & \text{si x>=} \alpha \\ 0 & \text{si x<} \alpha \end{cases}$$

Where α represents the value determined as a discriminant. The continuous output of an MLP, under the proper assumptions, can be interpreted as a posterior probability estimation to belong to a class C [Lippmann, 1994].

In some ANNs as the MLP is used an optional unit that provides constant stimulation entry node called bias. This unit is fully connected to the corresponding layer and generates a bias node for each of the layers, except for the input. This unit has a weight associated involving both the learning process and generation of the error signal, and in the subsequent classification process. Bias node may provide in certain problems the necessary stimulation to overcome the threshold value discussed above.

1.2.4.2 Learning in MLP

Many algorithms have been proposed so far to deal with the problem of appropriate weight-update by doing some sort of parameter adaptation during learning. They can roughly be separated into two categories: global and local strategies.

Global adaptation techniques make use of the knowledge of the state of the entire network to modify global parameters, as the backpropagation learning algorithm does modifying the direction of the previous weight-step, whereas local strategies use only weight-specific information (e.g. the partial derivative) to adapt weight specific parameters. Besides the fact that local adaptation strategies are more closely related to the concept of neural learning and are better suited for parallel implementations, their superiority over global learning algorithms has been demonstrated [Schiffmann et al., 1994]. Firstly, we will describe the backpropagation algorithm, used in this work. Secondly, we will study briefly the RPROP algorithm to compare it with a local strategy.

Backpropagation algorithm The learning process begin with a random assignment of weighted connections between neurons.

- Feed Forward: Calculate the output value for each neuron on the output layer, where the output value is determined by the equation network described above in 1.6 .

- Backpropagate: Calculate the mean square error (RMSE) between the output activation of the output neuron and the value of the standard input provided to the network. This error is calculated by the expression:

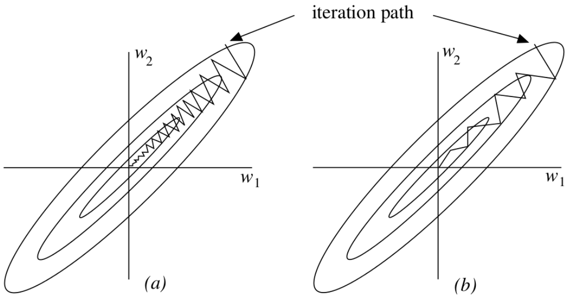
  $$\frac{1}{2}\sum_{i}^{n}||y_i - p_i||^2 \tag{1.10}$$

  where n is the number of output neurons , y_i the output value of the ith neuron and p_i the desired output value of the ith pattern presented at the input. This error is back-propagated layer by layer, calculating the contribution of each element to the total error in the direction of the gradient and updating the value of the weights to correct this contribution. The algorithm will terminate when the mean square error is small enough or depending on the number of epochs defined. An epoch consists on introduce to the net all the patterns of the learning set. Thus, it is understood that the purpose of the algorithm consists in reducing the RMSE, tracing the descent towards the global minimum of the error. The correction of the weights in each iteration , will be as follows:

  $$\Delta w(t) = w(t-1) + \lambda \nabla E|_{w(t)} + \mu * \Delta w(t-1) \tag{1.11}$$

  Where w is the value of the weights in an instant or iteration t, λ is the learning rate, ∇E is the partial derivative of the error function and μ is the momentum parameter, that is multiplied by the change in the weights in the previous moment. This parameter was introduced to the standard backpropagation algorithm of Werbos [Werbos, 1974] by Rumelhart [Rumelhart et al., 1987]. The momentum is trend that reduces oscillations always considering the changes made in the previous iteration, see Figure 1.7. It

Figure 1.7: Backpropagation with momentum (a) and without momentum (b). [Rojas, 1996]



also increases the amount of correction when the slope or first derivative is constant, thus improving the avoiding of the minimum locals.

1.3 Ensembles

The Non-free-lunch theorem [Wolpert and Macready, 1995] states that there is no learning algorithm that could achieve in every domain an optimal classifier, since each learning paradigm converge differently under distinct input spaces creating various decision regions in the features space. As each ANN makes generalization errors on different subsets of the input space, it is possible to argue that the collective decision produced by the complete set, or a screened subset, of networks, with an appropriate collective decision strategy, is less likely to be in error than the decision made by any of the individual networks [Hansen and Salamon, 1990, Yates and Partridge, 1996,Opitz and Maclin, 1999,Dietterich, 2000]. In general terms, a strong classifier could be constructed combining diverse decision regions aiming to achieve an overall higher accuracy [Kittler et al., 1998, Kuncheva and Whitaker, 2003, Brown et al., 2005, Ko and Sabourin, 2013]. It is also common to name them Ensemble of Classifiers (EoC) [Ko and Sabourin, 2013,Kittler et al., 1998]. Being able to tackle complex tasks in an efficient way has been another proof feature of the neural networks ensemble (NNE) approach. A NNE combines a set of neural networks which learn to subdivide the task and thereby solve it more efficiently and elegantly. In a sense, the NNE follows a divide-and-conquer approach by dividing the data space into smaller and easier-to learn partitions, where each ANN learns only one of the simpler partitions. The underlying complex decision boundary can then be approximated by an appropriate combination of different ANNs. NNEs are also very appropriate in applications where large volumes of data must be analyzed. It is necessary partitioning the data into smaller subsets, training different ANNs with different partitions of data, and combining their outputs using an intelligent combination rule. The situation of having too little data can also be handled using ensemble systems [Polikar, 2006]. A NNE offers several advantages over a monolithic ANN: It can perform more complex tasks than any of its components. It is more robust than a monolithic neural network. It can produce a reduction of variance and increase in confidence of the decision, and can show graceful performance degradation in situations where only a subset of neural networks in the ensemble are performing correctly [Liu and Higuchi, 2003].

Brown and Kuncheva showed in a simulated study how the smaller ensembles tend to exhibit a large variance and how it is reduced with larger ensembles [Brown and Kuncheva, 2010]. In such a way, the greater the number of classifiers we use in ensembles, the greater we reduce the risk of making a particularly poor selection. This property emerges from the diversity of decision made by the base classifiers. In this sense, the ideal situation is made up of a large number of classifiers with relative good accuracy and high diversity among them. Diversity can be sought in a explicit or implicit manner [Brown et al., 2005]. Algorithms like bagging [Breiman, 1996] and its variations (random forest [Breiman, 2001] or pasting small votes [Breiman, 1999]), the boosting algorithm [Schapire, 1990] or the random subspaces [Ho, 1998] constitute an example of an implicit method of creating diversity in ensembles. Hereafter, we are going to study all of them trying to make an updated categorization, adding the study of diversity measures and how it has been exploited explicitly as a measure in newer algorithms.

As we have pointed out, the critical success of an ensemble creation routine lies in the diversity creation because an ensemble can not perform well without some amount of diversity [Kuncheva et al., 2002, Kuncheva

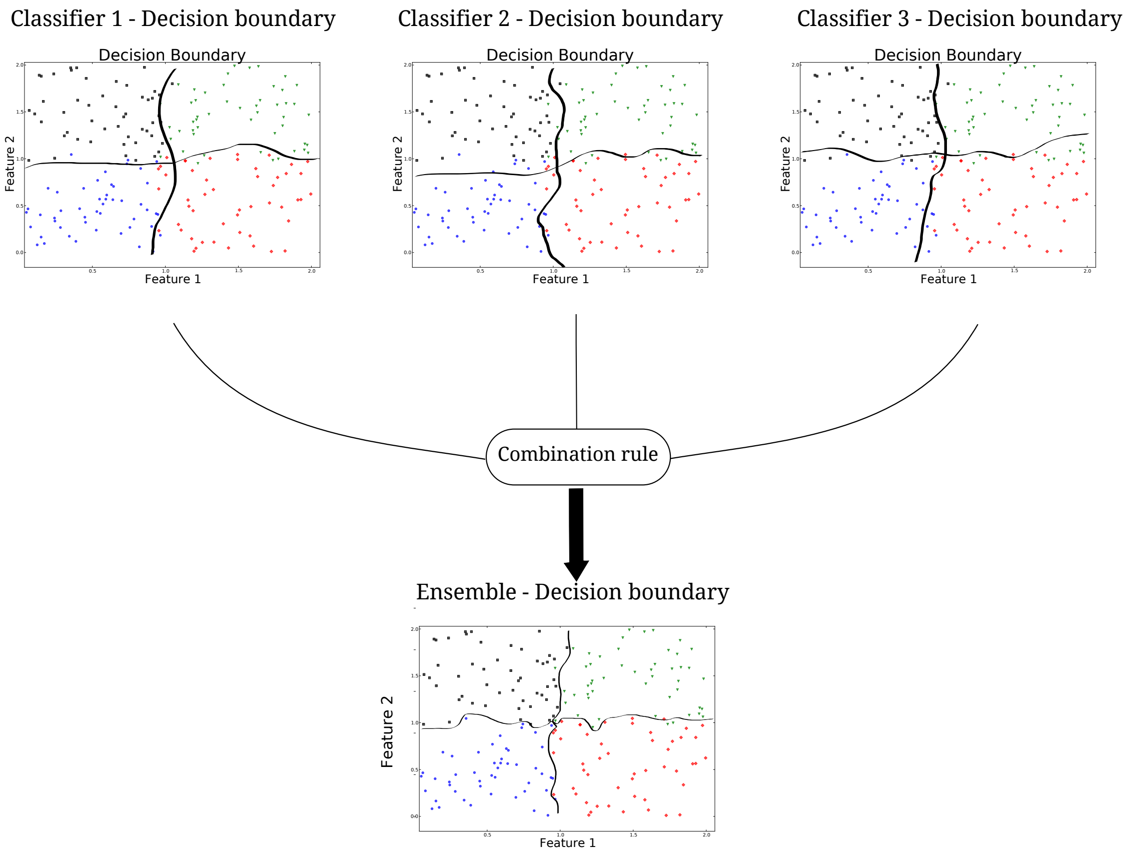


Figure 1.8: Combining classifiers that are trained on different subsets of the training data. Retrieved from [Polikar, 2006]

and Whitaker, 2003, Kittler et al., 1998, Ruta and Gabrys, 2001, Ruta and Gabrys, 2005]. It is preferable that the classifiers commit errors on different instances and therefore cancelling out those errors while combining the decisions, see figure 1.8, to achieve a more accurate recognition rate [Fumera et al., 2008, Kuncheva et al., 2002, Kuncheva and Whitaker, 2003, Kittler et al., 1998, Ruta and Gabrys, 2001, Ruta and Gabrys, 2005]. It is unusual to generate both high levels of success as a high diversity, at least in homogeneous input spaces, being relatively preferable to get weak learners and combine their decision [Schapire, 1990].

The conventional ensemble generating methods to enhance classification accuracy, referred as Multiple Classifier System (MCS) [Ko and Sabourin, 2013] create diversity from combining the decisions of multiple classifiers, but a new interesting approach referred as Simple-Classifier-based Multiple Classifier System (SMCS) aim to create diversity from combining the decision from one classifier introducing seudo data points correlated to a original reference sample and thus, generating diversity transferring the computational cost from training to the decision phase with interesting results and defining new odds [Ko and Sabourin, 2013]. We can derive a summary process from the MCS construction consisting of three steps [Ho, 1998, Kuncheva et al., 2002, Schapire, 1990, Ko and Sabourin, 2013]:

1. Prepare the data subsets taking into account features or instances diversity. In a heterogeneous input space it could be a straightforward process.
2. Train a classifier/s for each data subset available.
3. Select the best subset of classifiers that make up the ensemble with the higher accuracy.

These steps are justified by the fact that we know that diversity is necessary to enhance ensemble performance, but we also know that diversity is not sufficient to improve the ensemble accuracy (Step 3). As Hansen and Salamon stated, a necessary and sufficient condition for a majority voting ensemble of classifiers to be more accurate than any of its individual members is if the classifiers are accurate and diverse [Hansen and Salamon, 1990]. There is an important drawback of computational complexity underlying this accepted method of overproduction and select classifiers, but this is because nowadays it has not been possible to find complete solutions to the questions that Kuncheva and Whitaker have been doing for over a decade [Kuncheva and Whitaker, 2003], which we reproduce in its entirety:

1. How do we define and measure diversity?
2. How are the various measures of diversity related to each other?
3. How are the measures related to the accuracy of the group?
4. Is there a measure that is best for the purposes of developing committees that minimize error?
5. How can we use the measures in designing the classifier ensemble?

With this questions in mind the necessity of capitalize the diversity measures in the process of selecting the best subset borns, or wrapping all the process into a termed Ensemble learning that learn a target function by training a number of individual learners by combining their predictions [Opitz and Shavlik, 1996a, Liu et al., 2000, Brown, 2004], exploiting explictly the diversity measures. For example, we would see how the Negative Correlation Learning has shown a directly control over the covariance term in the bias-variance-covariance tradeoff [Brown, 2004], dealing the problem like a regression one, where the output of each classifier is defined as the posterior probability value because the zero-one loss function employed in the classification field do not apply as well. As Brown stated in [Brown et al., 2005] if an ensemble is better than a single classifier depends on if it operates on a regression or classification context. To find an effectiveness diversity measure to apply it on ensemble learning as a zero-one loss function, ergo, in a classification context, is still an active research [Brown and Kuncheva, 2010]. Hereafter, we introduce those discussed explicit and implicit methods following the taxonomy of methods for creating diversity developed by Brown, Wayatt, Harris and Yao in [Brown et al., 2005].

1.3.1 Diversity Generating Methods

During the learning phase, a function approximator follows a trajectory in hypothesis space and its desirable for creating diversity that each learner define different trajectories. In one hand, the capabilites of the implicit methods for generating diversity lies on random alterations in the hypothesis space to construct distinct trajectories, in the other hand, explicit methods define deterministically how different are those trajectories [Brown et al., 2005]. This dichotomy rises a high level taxonomy, but we must descend to a more specific level in which we can explain all the methods developed until now. In [Brown et al., 2005], the authors defined a new taxonomy from the one of four factors defined by Sharkey [Sharkey, 1999] for artificial neural networks (ANN): The initial weights, the training data used, the architecture of the networks and the training algorithm used, adding the concepts that use penalty terms in the learning algorithm as in [Brown, 2004, Liu, 1998, Rosen, 1996].

Hence, there are 3 general ways according to the hypothesis space to generate diversity in ensembles:

- **Starting point in Hypothesis Space.** Manipulate the initial weigths alter the starting point of the trajectory in the hypothesis space. It has been proved that this is the least effective mechanism to reach a good diversity [Opitz and Maclin, 1999]. Sharkey [Sharkey et al., 1995] discovered that multiple ANN trained with backpropagation starting from random points converge to the same or very similar local optima. Partridge and Yates [Partridge and Yates, 1996, Yates and Partridge, 1996] conducted a very extensive experiment to demonstrate empirically that the more important factor for creating diversity in ANN, in order, were : a) network type (RBF and MLP were proved), b) training set structure, c) number of hidden units and d) random initialization of weights. [Parmanto et al., 1995] compared the 10k-fold, the bagging and random initialization of weights, being the last one, the worst option. In the field of using an explicit use of diversity we can find the studies of Maclin and Savlik [Maclin and Shavlik, 1995] where they used competitive learning to initialize ANN to create a set of initial weights so far from the origin weight space, thereby potentially increasing the set of reachable local minima, or the fast committe learning study of Swann and Allinson [Swann and Allinson, 1998] where they generate diversity by secuencially selecting the members of an ensemble as weights snapshots during the learning of an individual.

- **Set of Accessible Hypothesis.** We understand by set of accessible hypothesis as the set of instances with a set of features that make up the dataset through which we will train, test and validate our classifier. It is expected that the alteration in some way of the training set will affect the diversity of the model, thus, different classifiers trained with variations of the training set will result in different convergences and therefore in different errors. This alteration could be made in the features, in the instances or both spaces. It is also possible to generate diversity through the alteration of the classifier architecture.

- **Manipulating Training Data** In a problem with *K* features and *N* instances we might want to train some classifiers with different subset of features but with the same instances. A method of generation diversity that follows this scheme is known as random subspaces [Ho, 1998,Zenobi and Cunningham, 2001]. Another possibility is the distribution of instances into different train subsets which train implicit diverse classifiers. The hypothesis of Schapire was that given a weak learning algorithm or a *C* class weakly learnable, we can construct a strong classifier based on a weak one producing some different distributions of the instances that each classifier learns. This idea led to the development of algorithms such as Adaboost [Freund and Schapire, 1997], Bagging [Breiman, 1996], Arcing [Breiman, 1998], and many other variations. The bagging algorithm propose to create a set of train subsets, randomly selecting *N* patterns with replacement from the original set of *N* patterns, leaving the remaining instances for testing and validation. Bagging almost always produces a better classifier than a single neural network [Opitz and Maclin, 1997]. Other option could be the combination of both views as in the random forest approach [Breiman, 2001], or the rotation forest [Rodríguez et al., 2006], only applicable to decision trees, where each node creation requires consideration of a feature that is selected by different statistical methods or randomly. The most simple method (no honest) is to combine the decision of the k-fold validation [Krogh and Vedelsby, 1995], where different classifier are trained under different distributions, generally random, of the data set. Parvin [Parvin et al., 2013] suggest an interesting idea, based on the boosting methods, that consists in distributing those instances in different subsets depending on the distance from the clusters they occupy in the space, thus, creating subsets of boundaries instances, cluster instances, etc. in a heuristic way, more than randomly.

  Some authors assert that combining the results of one classifier on different feature sets is far more effective than combining the results of different classifiers on one feature set [Duin and Tax, 2000]. Ho [Ho, 1998] concludes that is a good approach to problems where there is a large amount of redundant features. The input decimation (ID) approach [Oza and Tumer, 2001] decompose problems in L-class problems, training a set of classifiers attending to different features in specific classes. As the ID is orthogonal to approaches that search in the pattern space as bagging or boosting, Oza and Tumer [Tumer and Oza, 2003] suggest to use the ID as a complementary method in bagging and boosting. More methods of classes codification, such the error correcting output codes [Dietterich and Bakiri, 1991] or combination of class codification with features manipulation were proposed recently [Sesmero et al., 2012]. One versus all strategy (OVA) consists on a division technique that train *N* binary classifiers that distinguish one class from all the remaining classes. Although it is not a diversity generation method itself, this kind of strategy is supposed to be at least as good as multiclass classification [Rifkin and Klautau, 2004], seeking to reduce the instability of the systems that is produced by the high correlation among the classes of the problem under study.

  Other approaches tries to generate diversity through the introduction of noise into de data [Raviv and Intrator, 1996,Sharkey et al., 1997] or in the outputs of the dataset [Breiman, 2000]. The distortion methods [Sharkey et al., 2000] propose to alter the training data set in order to generate diversity introducing some kind of noise into the patterns. It is proved that whit a gaussian noise alteration the ensemble could achieve a better generalisation error [Raviv and Intrator, 1996]

- **Manipulation of Architectures** Manipulation of architectures is composed of two mainly procedures: 1) To build hybrid ensembles, with any kind of algorithm working to make up an unified decision [Gutta and Wechsler, 1996,Duin and Tax, 2000]. 2) To build ensembles which consist of member with the same architecture but varying elements into that architecture. Another possibility could be the ensemble of members with variations in both the architecture and configuration. Partridge [Partridge, 1996,Partridge and Yates, 1996] concludes that variations into the same architecture lead to poorly results in ensembles diversity, finding the opposite in the diversity generations through the use of multiple architecture, concretely he did it with MLPs and RBFs. More recently reviews point to this approach as the most growing trend in the number of publications [Wozniak et al., 2014]. Wolpert stated that each classifier has its specific competence domains where they overcome other competing algorithms. In this sense, it is not possible to design a single classifier exceeds all others for each classification task. MCS try to select always the local optimal model from the available pool of trained classifiers [Wolpert, 2002]. Some, but not many researches have been done in this scope [Partridge, 1996, Opitz and Shavlik, 1996b, Islam et al., 2003, Wang et al., 2000, Langdon et al., 2002, Woods et al., 1997].

- **Traversal of Hypothesis Space.** Given a particular data set, that is, a search space of instances and features, and given a particularly selected architecture as MLP, we can modify the method of learning that features space. Modifying how we traverse the space determines a method of diversity creation. We distinguish two kind of methods here, the penalty term based and population-based. The first one enforce a specific bias for each members of the ensemble, creating implicitly diversity. The second one discard those elements that does not create diversity into the ensemble, exploring all possible kind of modifications of the members to increase diversity.

  - **Penalty Methods** The diversity creation methods based on a penalty term are based on a modification of the error function, taking into account the diversity that each member puts into the ensemble to increase or decrease the penalty term. It could be understood as an ensemble learning algorithm where the it forces the bias of each member, some overfitting or underfitting indeed. For a learning algorithm based on the minimization of the error, i.e. the backpropagation algorithm, the penalty method introduction means an addition term as expressed in equation 1.12

$$\frac{1}{2}\sum_{i}^{n}||y_i - p_i||^2 + \lambda R \tag{1.12}$$

where λ is the weight of the penalty term R, usually guided by a diversity function, y_i the output of the classifier for the i-th instance, p_i the desired output of the i-th pattern. For a $\lambda = 0$ case, we have the standard backpropagation algorithm. The first penalty method was created by Rosen in 1996 [Rosen, 1996], consisting in a negative correlation learning posteriorly extended by Liu [Liu, 1998].

Ensemble Learning using Decorrelated Neural Networks [Rosen, 1996] The first work on penalty methods was conducted by Rosen in 1996, consisting on solving regression problems with neural network ensembles. The back propagation networks (BPN) were trained with a Fletcher-Reeves conjugate gradient algorithm because is faster than Delta-Bar-Delta and the quickprop implementations [Rosen, 1996], and also there is no need of parameter tunning. The BPNs was combined linearly in the ensemble. The neural networks ensembles decorrelated are produced by introducing a penalty term in the error function of the neural networks, in this case, a squared error function as in the equation 1.13.

$$E_j = \sum_{p=1}^{N}\left((y_p - f_j(\vec{x}_p))^2 + \sum_{i=1}^{j-1}\lambda(t)d(i,j)P(\vec{x}_p,y_p,f_i,f_j)\right) \tag{1.13}$$

where p is the $p-th$ pattern, y_p is the desired output for the correspondent p, f_j is the output of the j member of the ensemble, $\lambda(t)$ is a (possibly) time dependent scaling function, d is an indicator function for decorrelation between networks i and j, P is a correlation penalty function and N is the number of patterns. The findings point to an improvement in the results obtained by a moderate decorrelation in the network training.

Ensemble learning via negative correlation [Liu and Yao, 1999] The proposal of Liu and Yao [Liu and Yao, 1999] is practically identical to the previous one, except for slight differences in the formulation given in the equation 1.14, where we have tried to keep the same nomenclature: E is the error of the network j, N is the number of patterns, λ is now independent from time or iterations and regulates the strong of the penalty, P is the correlation penalty function, y_p is the desired output for pattern p and $f_j(p)$ is the output of the network j for the pattern p.

$$E_j = \frac{1}{N}\sum_{p=1}^{N}\frac{1}{2}(y_p - f_j(p))^2 + \frac{1}{N}\sum_{p=1}^{N}\lambda P_j(p) \tag{1.14}$$

The penalty function P_j can be seen in the equation 1.15.

$$P_j(p) = (f_j(p) - f(p)) \sum_{i \neq j} (f_i(p) - f(p)) \tag{1.15}$$

The training process is produced simultaneously in all the networks that compose the ensemble, such that the networks tries to minimize not only the difference respect to the pattern but also the diversity of the members into the ensemble.

- **Evolutionary Algorithms** Evolutionary Algorithms (EAs) were proposed by Holland [Holland, 1992] as a global optimization approach inspired by natural evolution and survival of the fittest. EAs use a solution population (chromosomes) which evolves by means of selection, crossover and mutation operators [Mitchell, 1998]. There are many works that surround the idea of combine different diversity creation methods under a EA. Opitz [Opitz, 1999] proposed a genetic ensemble feature selection that improves the wrapper proposal of [Kohavi and John, 1997] in terms of computational efficiency. The proposal of [Zhou et al., 2002,Junfei et al., 2010] consist in using diversity measures into the fitness function of the ensemble generation while produce diverse members by manipulating the feature space.

  The EAs are not a diversity creation method itself, but allow us to create ensembles with different criteria searching in the whole (features, instances, learning parameters, learning architecture, learning algorithms, etc.) space. Thus, leading to a set of individuals that are accurate and diverse in a efficient manner [Lofstrom et al., 2010, Kim and Cho, 2008, Nabavi-Kerizi et al., 2010, Chandra and Yao, 2006, Rahman et al., 2010, Liu et al., 2000, Yao et al., 1998].

- **Others diversity creation methods**

  Mixture of Experts The gating network could be seen as the paradigm in which mixture experts consists in. The gating network is an ANN on a second level in a ensemble, which environment data is composed of the outputs of a group of classifiers, thus, creating a set of weights for the outputs of each classifier member in the ensemble [Polikar, 2006]. An important point of this methods is that individual classifiers are experts in some portion of the feature space and the gating network selects the most appropriate classifier, or classifiers weighted with respect to their expertise. It can therefore be seen as a classifier selection algorithm, or mostly as a combination rule, similar to the weighted majority voting scheme that we will present in the next subsections.

1.3.2 Diversity measures

In a diversity creation method, we have one overproduction member phase and a selection of members phase. The first one consist in one or multiple methods described above, the second part of the ensemble creation could be based on a selection of the best members (by accuracy or other measures), or on a selection of the most diverse members. Diversity measures described below are examples to be used in a member selection of ensembles. Diversity in ensemble systems is an active research area due to there is no general nor effective measure that ensure a good generalization performance in every application domain. We distinguish two kind of diversity measures, depending on if we measure by pairs of classifiers or if we measure a global diversity.

It has been tested that there is no diversity measure strongly correlated with the validation set accuracy in real data sets [Johansson et al., 2007], and in the commonly used data sets [Kuncheva and Whitaker, 2003], but it is also proved that ensembles as an aggregating of diverse classifiers reach higher accuracy [Johansson et al., 2007].

Classification of classifier outputs Following the classification made in [Kuncheva and Whitaker, 2003], if we denote $D = \{D_1, ..., D_L\}$ as a set (pool, committee, mixture, team, ensemble) of classifiers, $\Omega = \{\omega_1, ..., \omega_c\}$ be a set of class labels and $x \in \mathbb{R}^n$ be a vector of n features to be labelled in Ω. There are three general possibilities for the classifier outputs:

1. A c-element vector $\mu^i = [d_{i,1}(x), ..., d_{i,c}]^T$ where a special case of this vector is a probability distribution vector over Ω estimating the posterior probabilities $P(\omega_s|x)$ where $s = 1, ..., c$.

2. Class label $D_i(x) \in \Omega, i = 1, ..., L$

3. Correct/incorrect decision (the oracle output) is the most commonly used and represent a vector where $D_i(x)$ is 1 if x is recognized correctly by D_i, and 0 otherwise.

Hereafter, we work with the oracle outputs in the diversity measures, as they were employed in this work and it is the most simply manner to work with the information.

1.3.2.1 Pairwise diversity measures

Between a pair of classifiers, we define the notation of the diversity of a given instance as in the table 1.2, where total, $N = N^{00} + N^{01} + N^{10} + N^{11}$. Based on this notation we can build some pairwise diversity measures, see 1.3. A summary interpretation of those expression can be observed in Table 1.5.

Table 1.2: A 2 × 2 table of the relationship between a pair of classifiers [Kuncheva and Whitaker, 2003], where correct means a coincidence with the desired output and wrong a misclassification error.

	D_k correct (1)	D_k wrong (0)
D_i correct (1)	N^{11}	N^{10}
D_i wrong (0)	N^{01}	N^{00}

Table 1.3: Pairwise diversity measures summary [Kuncheva and Whitaker, 2003]

Diversity measure	Expression
Q-statistic	$Q_{i,k} = \frac{N^{11}N^{00} - N^{01}N^{10}}{N^{11}N^{00} + N^{01}N^{10}}$
Correlation coefficient p	$p_{i,k} = \frac{N^{11}N^{00} - N^{01}N^{10}}{\sqrt{(N^{11}+N^{10})(N^{01}+N^{00})(N^{11}+N^{01})(N^{10}+N^{00})}}$
The disagreement measure	$Dis_{i,k} = \frac{N^{01}+N^{10}}{N^{11}+N^{00}+N^{01}N^{10}}$
The double-fault measure	$DF_{i,k} = \frac{N^{00}}{N^{11}+N^{00}+N^{01}N^{10}}$

Q statistic promotes selection of members which have a certain balance between correct and incorrect decisions, penalizing members who are not complementary. The correlation coefficient is similar to Q statistic, but look for more complete complementarity penalizing the cases in which anyone member detect the class correctly. The disagreement measure also promotes the selection of members whose classifications are more absolutely complementary in all the instances. The double-fault mainly penalizes the members pairs where no one classify correctly the instance presented. Except the correlation coefficient, all the other measures are interesting because present simple computational cost, these being a problem in pairwise measures.

In general, pairwise diversity measures lack the ability to select members that may be key in solving certain instances.

1.3.2.2 Non Pairwise diversity measures

A good review of non pairwise diversity measures have been described in [Kuncheva and Whitaker, 2003], see table 1.4. To understand the notation of the table, N is the number of instances of the dataset, L the number of classifiers, l is the number of classifier that correctly identify the presence of the class j in a given instance from the vector of classes z. The entropy expression tries to promote those combinations that maintain a balance between classifiers that classify correctly and those who do not. The Kohavi Wolpert is similar to the entropy expression but promoting combinations in a exponential manner. The interrater k also take into account the members accuracy, so that the expression promotes combinations of diverse and accurate members. It is noteworthy that the non pairwise measures select best combinations of members than the pairwise ones. The non pairwise diversity measures also select combinations with less redundant information, thus, with less members. The inconvenience of the application of non pairwise measures relies to computational costs. The interpretation of those measures can be observed in Table 1.5.

Table 1.4: Non-pairwise diversity measures summary

Diversity measure	Expression
Entropy	$E = \frac{1}{N}\sum_{j=1}^{N}\frac{1}{L-\lceil L/2\rceil} min\{l(z_j), L-l(z_j)\}$
Kohavi-Wolpert	$KW = \frac{1}{NL^2}\sum_{j=1}^{N} l(z_j)(L-l(z_j))$
Interrater agreement k	$k = 1 - \frac{\frac{1}{L}\sum_{j=1}^{N} l(z_j)(L-l(z_j))}{N(L-1)\bar{p}(1-\bar{p})}$, where $\bar{p} = \frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{L} y_{ji}$

Table 1.5: Summary of the 7 previous measures of diversity. Source: [Kuncheva and Whitaker, 2003]. The arrow specifies whether diversity is greater if the measure is lower (↓) or greater (↑). P stands for Pairwise and S stands for Symmetrical

Name		↓ / ↑	P	S	References
Q-statistic	Q	(↓)	Y	Y	[Yule, 1900]
Correlation coefficient	p	(↓)	Y	Y	[Sneath & Sokal, 1973]
Disagreement measure	Dis	(↑)	Y	Y	[Ho, 1998; Skalak, 1996]
Double-fault measure	DF	(↓)	Y	N	[Giacinto & Roli, 2001]
Kohavi-Wolpert variance	KW	(↑)	N	Y	[Kohavi & Wolpert, 1996]
Interrater agreement	k	(↓)	N	Y	[Dietterich, 2000b; Fleiss, 1981]
Entropy measure	Ent	(↑)	N	Y	[Cunningham & Carney, 2000]

1.3.3 Combination Rules

Independently from the diversity of the classifiers, their accuracy and other considerations taken into account to achieve a better ensemble system, the combination rules define the last step on the process of building such systems. We can combine the continuous or the discrete outputs of the classifiers. The decision can be given if we work in a classification or in a regression context, thus, we have to combine the continuous outputs in a regression context, but we could also combine continuous or discrete outputs in a classification context. Kuncheva and Duin [Kuncheva et al., 2001] describe a simple method of creating a decision profile matrix scheme in order to easily apply different fusion methods and functions.

1.3.3.1 Fusion of Continuous outputs

The continuous outputs allow to get a richer ensemble output for a given instance, as it could be possible to offer a posterior probability of the classes and the relevance of the outputs given. In the table 1.6 we summarize the different schemes of continuous output fusion [Polikar, 2006], where D is the number of classifiers, d is the output of the classifier, c is the index of the classes array, C is the number of classes. The weight w of each classifier is equal to the inverse of the number e elevated to the test error.

Table 1.6: Summary of continuous outputs fusion techniques and its expressions

Name of the Fusion Techniques	Acronym	Expression
Mean	Mean	$c[c]_{c=0}^{C}(x) = \frac{1}{D}\sum_{i=1}^{D} d_{ic}$
Weighted Average	WA	$c[c]_{c=0}^{C}(x) = \sum_{i=1}^{D} w_{ic}d_{ic}$
Trimmed Mean	TM	The most optimistic and pessimistic classifiers are removed from the ensemble before calculating the mean. Trimmed mean at limit 50% is equivalent to the median rule.
Minimum	Min	$c[c]_{c=0}^{C}(x) = min(\sum_{i=1}^{D} d_{ic})$
Maximum	Max	$c[c]_{c=0}^{C(x)} = max(\sum_{i=1}^{D} d_{ic})$
Median	Med	$c[c]_{c=0}^{C}(x) = median(\sum_{i=1}^{D} d_{ic})$
Product	Prod	$c[c]_{c=0}^{C}(x) = \frac{1}{D}(\prod_{i=1}^{D} d_{ic})$

1.3.3.2 Fusion of Class labels

On the other hand, in a fusion label context there are mechanism to find out the relevance of the outputs of each classifier, usually relating the relevance of each classifier to a test-set evaluation error. We summarize the fusion of class labels techniques described in [Polikar, 2006] in the table 1.7, but generalizing the expressions for a multi-label context, where *D* is the number of classifiers, *d* is the output of the classifier, *c* is the index of the classes array, *C* is the number of classes. The weight *w* of each classifier is equal to the inverse of the number *e* elevated to the test error.

Table 1.7: Summary of class label fusion techniques and it expressions

Name of the Fusion Techniques	Acronym	Mathematical expression
Unanimous voting	UV	$c[c]_{c=0}^{C} = \begin{cases} 1 \text{ if } (\sum_{i=1}^{D} d_{ic}) == D \\ 0 \text{ otherwise} \end{cases}$
Simple majority voting	SMV	$c[c]_{c=0}^{C} = \begin{cases} 1 \text{ if } (\sum_{i=1}^{D} d_{ic}) == \frac{D}{2} + 1 \\ 0 \text{ otherwise} \end{cases}$
Majority voting	MV	$max_{c=0}^{C} \sum_{i=1}^{D} d_i$
Weighted majority voting	WMV	$c[c]_{c=0}^{C} = \begin{cases} 1 \text{ if } \sum_{c=0}^{C}\sum_{i=1}^{D} w_{ic}d_{ic} > \theta \\ 0 \text{ otherwise} \end{cases}$

1.4 State of the art

Related to the problem defined by the dataset of fluorescence spectra of BFs compounds, we describe some works that deal with the same problem employing machine learning techniques also. It should be noted that there is no work related to this field that uses ensembles nor diversity methods for creating ensembles except those produced by authors of [Suárez Araujo et al., 2013].

Supervised techniques The review of the state of the art in the resolution of complex mixtures from fluorescence spectra allow us to know that there are some supervised techniques applied to this problem. In [Vasilescu et al., 2011], ANN were used to detect the presence of pollution in the black sea of Romania, obtaining an accuracy of 90%. The data source was a LIDAR and a channel relationship method was used to create some different kinds of spectra, at different wave lengths and intensity, improving the selectivity of the signals [Almhdi et al., 2007]. On the other hand, [Almhdi et al., 2007] made a comparison of the architectures ANN and support vector machines (SVM), in identifying three types of oil in water samples from fluorescence spectra. The results were not very relevant, but similar among architectures. Another works compared RBF, MLP and SMVs in identifying nanocrystals encoded microspheres in flow cytometry [Clarke, 2008]. The results were very satisfactory, obtaining an accuracy of 97.1% with the SVM. Another work based on SVM was performed in [Römer et al., 2011]. The aim of that work is to design an intelligent system that will identify different pathogens which may cause diseases in plants. Dataset comprises fluorescence spectra corresponding to samples of the pathogen in inoculated leaves. The results obtained in the SVM were 79.2% for day 2, 77.8% for the third day and 87.5% on the fourth day from the beginning of the pathogen activity. Other studies have tried to identify these components in fluorescence spectra with mathematical techniques, rather than machine learning techniques [Albani, 2008]. The authors of this work also produced some related in this field by creating ensembles [Suárez Araujo et al., 2013, García Báez et al., 2012, Álvarez Romero et al., 2013], with diversity measures based on error correlations, employing also data fusion schemes and comparing ensembles of unsupervised and supervised members. In [García Báez et al., 2012] was performed a learning process with mixture spectra also in the learning and test sets. Other studies have tried to test the suitability of the unification of the BM classes and MBC [Suárez Araujo et al., 2013]. Misclassification with separate classes were very high.

Unsupervised techniques In [Chowdary et al., 2009], we can find a PCA analysis of the data, comparing the subsequently supervised and unsupervised methods of learning, in order to diagnose different breast tumours from autofluorescence spectra at $325nm$ excitation, belonging normal, tissues and tissues from benign and malignant tumours. In [Lv and Gu, 2012] were employed a kernel PCA technique plus a canonical correlation analysis in order to extract the features of the fluorescence spectroscopy samples. In [Dorney et al., 2012], was developed a clustering with k-means algorithm and principal component analysis on a data set obtained from a fluorescence spectroscopy applied to the detection of biological cells. In [Suarez Araujo et al., 2010] a unsupervised technique based on ANN were used (HUMANN). This work employed also pure substances spectra to perform a system that identify complex mixtures of the same components.

Chapter 2

Design of Experiments

2.1 Methods

We present here the chemical experiments needed to make up the information environment of fluorescence spectra of BFs, the data set provided with that information, and the computational solution adopted to solve this application problem.

2.1.1 Chemical experiments

There is a need for a rich and representative data environment when aiming to develop automatic systems for resolution of fungicides mixtures with a high degree of overlap and which are friendly, sensitive and yet efficient and powerful, in other words, an intelligent solution. To this end, we have carried out a series of experiments which have allowed us to build up a corpus of spectrum data which are apt for the development, fine adjustment and validation of the artificial system which we propose. We have taken into account the following requirements that facilitate its study using our system based on artificial neuronal networks:

- Working with a sufficiently high number of compounds, which allows us to come to reliable conclusions about the results obtained.
- Establishing a group of physical parameters for the generation of spectra which avoid undesirable alterations. These parameters include the $\Delta\lambda$ and the wavelength interval belonging to each spectrum.
- Obtaining synchronous spectra from different $\Delta\lambda$ for the compounds used and for any mixtures of them. In this way, we will contrast the performance of the resolution of the mixture depending on the $\Delta\lambda$ of spectrum used in the process of resolution.
- Varying the concentration range of each compound to automatically determine and model the alterations that took place, in the resolution, as a result.

These requirements will be established in the selection of some groups of spectra with particular characteristics. Some of these characteristics will define the problem to be resolved. Others are designed to avoid the destabilising effect that the variability of these characteristics could have on the shape of the spectra, which would make it difficult to obtain good results in the identifications. We used six varying concentrations adapted to the luminescent characteristics of each compound, see Fig 2.1.

The data set was provided by the Environmental Chemical Analysis Group at the ULPGC. For each of the solutions, we generated the corresponding synchronous spectra for the optimum, middle and average $\Delta\lambda$ values, see Table 2.1. The optimum $\Delta\lambda$ is the value which allows for the maximum spectral intensity at a wavelength and which allows it to be distinguished from other member compounds of the same family. Usually, the optimum $\Delta\lambda$ for a compound is around the difference between its optimum λ_{em} and its optimum λ_{ex}. All spectra were repeated three times to guarantee measurements and define error margins in measurements. One hundred combinations of mixtures were generated automatically for each kind of spectra and $\lambda_{em}, \lambda_{ex}, \Delta\lambda$ values, with the only conditioning factors being that the compound distributions should be as balanced as possible with respect to the number of compounds present in each mixture, the concentrations of the same and the type of compound used, see Tables 2.2 and 2.3.

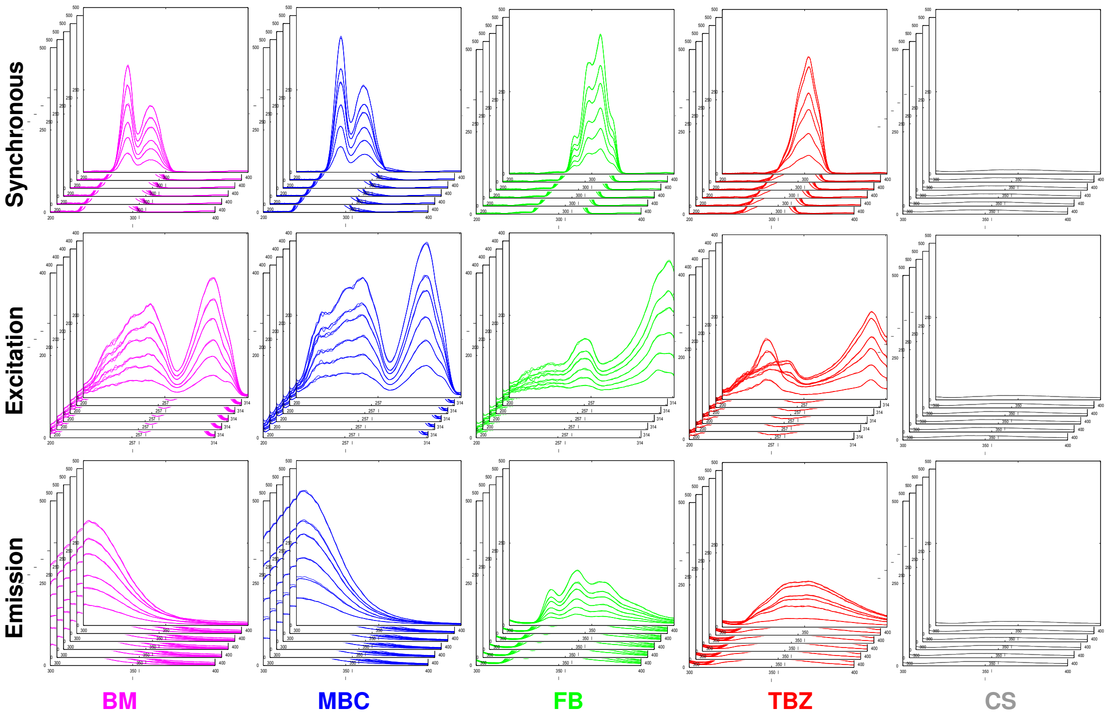


Figure 2.1: Fluorescence spectra representation of the data set. Clean sample (CS) contains the spectra of a sample without pesticides.

This experimental chemical design, make up the information environment that is going to be used in the neural computation solution.

Table 2.1: General characteristics of the data set of benzimidazole fungicides. *em* means emission and *ex* excitation.

	Characteristics	Benzimidazole family
	Compounds	4: Benomyl (BM), Carbendazim (MBC), Fuberidazol (FB), Thiabendazol (TBZ)
	Concentrations/compound	C0=absence, C1 to C6:
	BM	Interval = $250-1,500\mu g/l$, $\Delta c = 250\mu g/l$
	MBC	Interval = $250-1,500\mu g/l$, $\Delta c = 250\mu g/l$
	FB	Interval = $25-150\mu g/l$, $\Delta c = 25\mu g/l$
	TBZ	Interval = $2.5-15\mu g/l$, $\Delta c = 2.5\mu g/l$
	S1: Mean/TBZ optimum	$\Delta\lambda = 47nm$, interval = 200−400nm
4 synchronous	S2: Median	$\Delta\lambda = 53nm$, interval = 200−400nm
	S3: MBC-BM optimum	$\Delta\lambda = 59nm$, interval = 200−400nm
	S4: FB optimum	$\Delta\lambda = 29nm$, interval = 200−400nm
	S5: Mean	$\lambda_{em} = 327nm$, interval = 200−315nm
3 excitation	S6: Median	$\lambda_{em} = 325nm$, interval = 200−315nm
	S7: FB optimum	$\lambda_{em} = 341nm$, interval = 200−315nm
1 emission	S8: Mean/Median	$\lambda_{ex} = 277nm$, interval = 300−400nm

Table 2.2: Experiment design criteria, guided by concentration of compounds in the mixture

Concentrations	Number of Compounds/Mixture			Total
	2	3	4	
1	11	17	17	45
2	21	13	14	48
3	15	14	17	46
4	13	15	18	46
5	16	20	9	45
6	16	14	17	47
Total	92	93	92	277
Mixtures	46	31	23	100

Table 2.3: Experiment design criteria, guided by number of compounds in the mixture

Compounds	Number of Compounds/Mixture			Total
	2	3	4	
BM	23	23	23	69
MBC	23	24	23	70
FB	22	24	23	69
TBZ	24	22	23	69
Total	92	93	92	277
Mixtures	46	31	23	100

2.1.2 Computational experiments

The neural computation system proposed consist of a pre-processing and processing modules, both based on ANNs with a supervised learning scheme, see Figure 2.2.

2.1.2.1 Preprocessing stage

The pre-processing stage included the fluorescence spectra modelling and the attainment of the feature vector developed in [García Báez, 2005, Suárez Araujo et al., 2009]. Is usually possible to use generic methods during the feature extraction. If you possess knowledge about the particular problem we want to represent, it can be very helpful in design. Using as a base the developments and later experimental studies made by Lloyd and Evett [Lloyd and Evett, 1977] and later by Cabaniss [Cabaniss, 1991], the fluorescence spectra can be modelled by a Gaussian distribution of intensity versus reciprocal wavelength (frequency). Synchronous spectra can be modelled also by double Gaussian distributions. All spectra used in our developments have been previously normalized to the unit.

In this way any spectrum can be approximate, for a given wavelength λ, according to expression 2.1.

$$I(\lambda) \simeq = \sum_i a_i \cdot exp\left(-\frac{(\lambda^{-1} - \mu_i^{-1})^2}{2\sigma_i^{-2}}\right) \tag{2.1}$$

Where μ_i are the different values (cm) where the gaussian are centred, σ_i are it standard deviation(cm) and a_i are the amplitude of those gaussians. Changing the notation as in equation 2.2:

$$gaus_i(\lambda^{-1}) \simeq = exp\left(-\frac{(\lambda^{-1} - \mu_i^{-1})^2}{2\sigma_i^{-2}}\right) \tag{2.2}$$

We obtain the approximation of the spectrum as:

$$I(\lambda) \simeq a \cdot \mathbf{gaus}(\lambda^{-1}) \tag{2.3}$$

Where $\mathbf{gaus}(\lambda^{-1})$ is the vector formed by each of the $\mathrm{gaus}_i(\lambda^{-1})$ and a are the features. We will work with a linear approximation for the mixture model, such that the spectrum of a mixture will be represented by a linear combination of reference spectra [Lawton and Martin, 1985]. The reference spectra are the spectra of the compounds which can be identified in a mixture.

$$I(\lambda) \simeq c \cdot r(\lambda) = \sum_i c_i r_i(\lambda) \tag{2.4}$$

$$I(\lambda) \simeq c \cdot A_r(\lambda) \cdot \mathrm{gaus}(\lambda^{-1}) \tag{2.5}$$

Where $\mathbf{r}(\lambda)$ is the reference spectra and $\mathbf{c}$ is a vector with the contributions of each of the spectra of $\mathbf{r}(\lambda)$ in the mixture. $\mathbf{c}$ is then a vector which characterises a mixture and which is ideal for use as a vector of characteristics for BPN. Spectral representation via Gaussian distribution will be carried out using RBFNs [Bachiller et al., 1998] and the approximation of concentration coefficient vector ($\mathbf{c}$) through a BPN. Two RBFNs are used to obtain the spectral representation. One will determine the parameters which define the Gaussian distribution and the other will approximate the intensities of the different spectra modelled. An initial radial basis function network (RBFN1) is made up of an input layer of one neuron, a hidden layer whose number of neurons coincides with the number of Gaussians which carry out the approximation of the real spectrum (in our case 14 Gaussians) and an output layer the size of which determines the number of compounds which can be identified in any mixture to be analysed. This network will allow us to determine the parameters that define the Gaussian distributions which model the fluorescence spectra of each compound belonging to the fungicide family and mixtures that can be found in real environmental samples. A second RBFN (RBFN2) is designed to approximate the amplitudes of the spectra. Once the Gaussian approximations of the fluorescence spectra are developed, the feature vector ($\mathbf{c}$) is determined. The results obtained in this stage make up the information environment of the BPN-based systems for the BFs fluorescence identification, which are employed in the processing module. An environment made up of instances of 14 features.

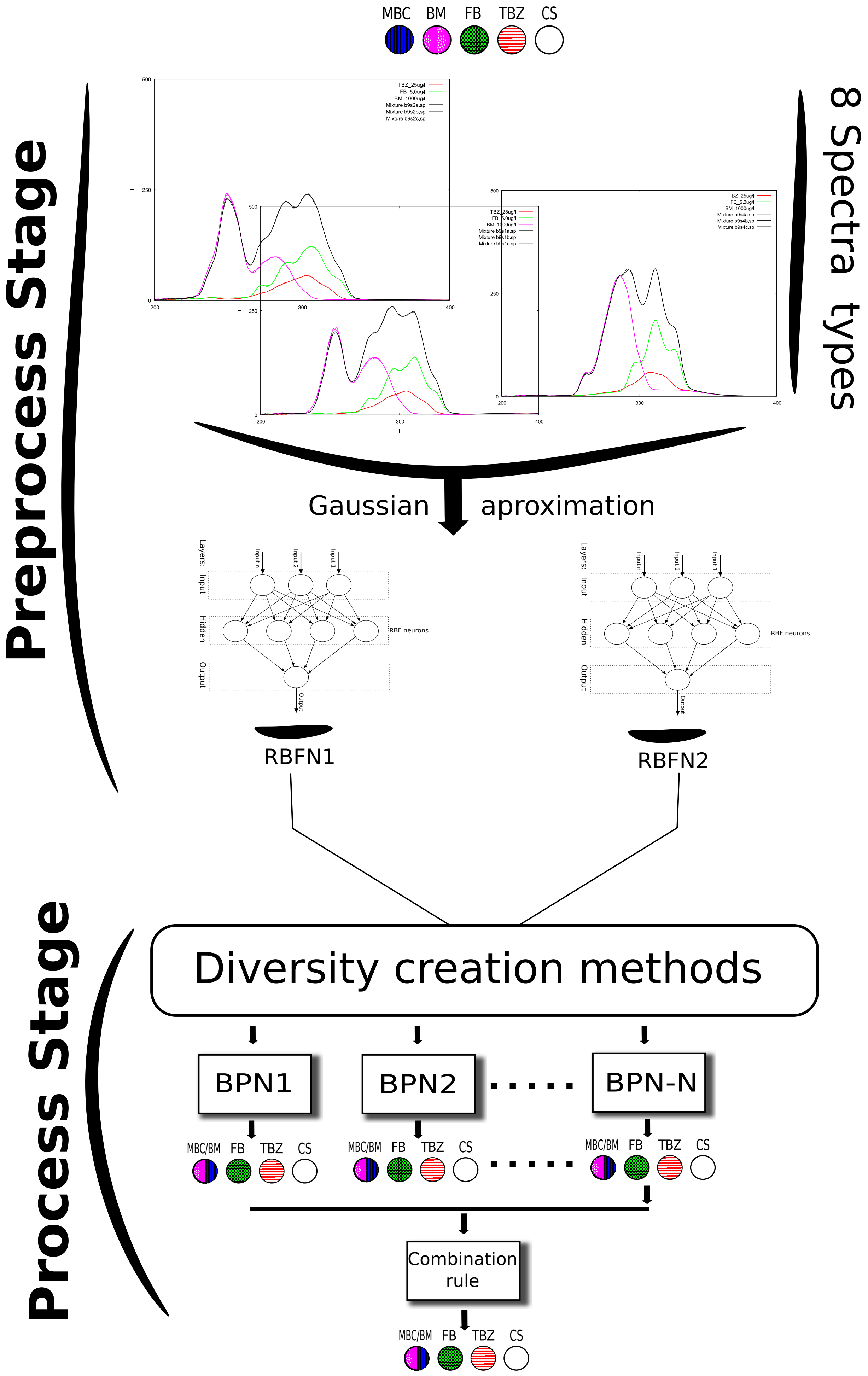


Figure 2.2: Neural system scheme for fungicides detection with diversity creation methods.

2.1.3 Processing stage

The proposal consists in diversity creation method of neural classifiers, in order to build different kind of ensembles with the data set provided. The data set in a supervised scheme as the BPN, must be divided into 3 subsets to avoid overfitting. The set which measures the final capability of the ANN to generalise those spectra from pure substances is the validation set, which is composed of all the mixtures available in all those diversity creation methods. The pure substances spectra available are instances from 24 solutions of pure substances of concentrations C1, C2, C3, C4, C5, and C6 for each substance, plus the spectra from the clean sample (CS), a total of 25 instances. Each concentration for each substance has 3 spectra repetitions, being always in the same subset. Hence, we have 75 spectra instances of pure substances for each kind of spectra. The distributions we have done contains the 13,6% of the instances in the train set, the 6,4% into the test set and the 80% in the validation set. The distribution into train and test sets, was made depending on the diversity creation method as follows:

- **Original distribution:** We have made some previous experiments with the same distribution of the patterns. That train set consisted in concentrations C2, C3, C5 and C6 of each substance plus de CS spectra. The test set was made up of spectra from the remaining 8 solutions with concentrations C1 and C4.
- **Bagging (Bgg):** A total of 17 unique instances had to be part of the training set. The others goes to the test set. The distribution was produced with replacement, in such a way that until the training set contains 17 unique instances new instances become part of this set. The process was repeated 96 times. Hence, we had 96 different distributions for each type of spectra.
- **Random Subspaces (RS):** From the original distribution we selected a number of features randomly such that we create 96 different distributions where all the spectra of each distribution contains the same selected features. The selected features has to be the same in learning, test and validation sets.
- **Negative Correlation Learning (NCL)**: We employed the original data set distribution. This method simplifies the process of manipulating the instances and create diversity in an explicit manner.
- **Weights Initialization (WI) and Hidden Nodes Variation (HNV):** We made the same unified process to compare the influence of the WI and the HNV over the original distribution.

The members of those ensemble are BPN with momentum [Rumelhart et al., 1987], a bias node, a hidden layer and an output layer with the same number of neurons as BF classes we defined in both experiments, plus the clean sample case, where there is no fungicide. The activation function employed in the BPN modules is produced by an hyperbolic tangent function.

In order to determine the optimal configuration of the BPNs, 30 ANNs with aleatory weights initialization were trained by each number of hidden nodes, from 3 to 10 elements, selecting the configuration that achieve the lowest root mean square error (RMSE) and the lowest standard deviation from the test set, in that order. The learning process stop criteria was guided by the sum of the test + learning RMSE . When an objective RMSE of 0.01 is achieved the learning process stops. The maximum number of epochs is fixed to 20.000. When it reaches the limit of epochs then the weights of the lowest total RMSE are recovered. This stopping criteria is defined for this specific problem, because there is no overfitting observed in the test set. The thresholds of the output nodes were determined by reducing the false negatives as most as possible, i.e., choosing the lowest threshold while evaluating at each point of the learning plus test set.

The input in these single BPN systems acts as the feature vector of one single type of fluorescence spectra, i.e. one BPN per each kind of spectra. Our ANN system consists in a neural ensemble approach, see Fig. 2.3, whose members are the single BPN systems. Three strategies are needed to build an ensemble system: diversity creation, selection members (nearly always) and a combination strategy. Firstly, we create diversity through the use of n BPN members, in ensembles with diverse fluorescence type in the input space. Over that given diversity, we propose different diversity creation methods, all of them based on ensemble scheme, see Fig. 2.3.

2.1.3.1 Diversity creation methods employed:

We have tried to cover the three main different types of diversity creation methods described in the introduction section: Manipulation of the starting point in hypothesis space, the set of accessible hypothesis and the traversal

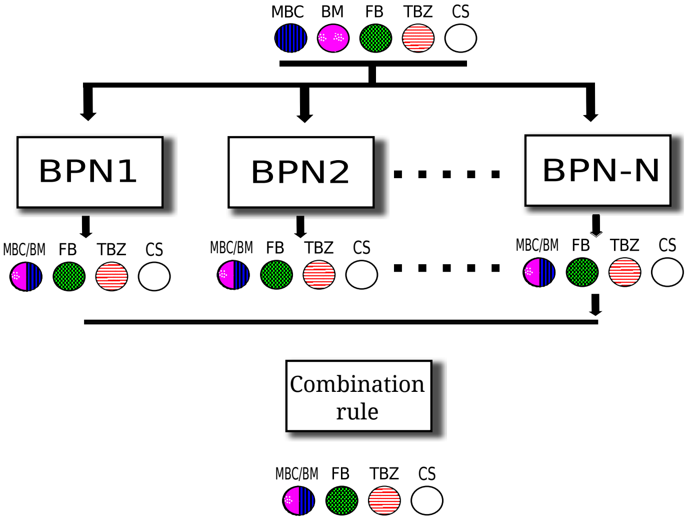


Figure 2.3: BPN-based systems for detection of BFs.: BPN ensemble system

of the hypothesis space:

- Diversity creation method by manipulating the starting point in hypothesis space
  - Initialization of weights
- Diversity creation method by manipulating the set of accessible hypothesis
  - Manipulating Training Data
    - Bagging
    - Random subspaces
  - Manipulation of architectures
    - Number of hidden neurons
- Traversal of Hypothesis Space
  - Penalty methods
    - Negative correlation learning

Experiments in WI and HNV: The process may start with the definition of the specific parameters to use in the architecture BPN. We summarize the process as follows:

- **Overproduction**
  - Generate ANNs with different hidden nodes, from 3 process elements to 10.
  - Generate 30 ANN by each number of hidden nodes with different randomly initialization weights.
- Selecting the best configuration:
  - For each amount of hidden neurons, we select for each type of spectra the one which have the lowest mean of the RMSE.
  - If the case of some different amount of hidden neurons and the same RMSE is produced, then we select the one with a lower standard deviation of the RMSE.

Once the best number of hidden nodes is defined for each spectra type, we will use that number to train each BPN in the later experiments.

Experiments in Bagging, Random Subspaces and NCL: Given the configuration from the experiments described above, for each spectra type, we train by groups of multiple of 8 (one per spectra), from 8 to 96 BPN members. This allow us to compare the results of the three main diversity creation methods (Bagging, NCL and Random Subspaces) at least by the number of members in the ensemble, because the NCL has already its own diversity measure during the learning process. In the NCL case we are going to test the influence of the penalty term λ by creating the same ensembles with $\lambda = 0.3, \lambda = 0.5, \lambda = 1.0$.

2.1.3.2 Selection members strategy:

- **Selection.** We used two methods of selecting members, always guided by the test set, for ensembles of multiple of 8, from 8 to 96 BPN members and taking into account the 8 type of spectra (conventional+synchronous) separated, i.e: by groups of 16 we select the two most diverse/accurate classifiers from each spectra:
  - The most diverse members. For each diversity measure employed. We understand by most diverse members, the member with a higher or lower (depending on the diversity measure) mean of the diversity measure in pairwise cases. In the NCL case we didn't applied diversity measure after the learning process.
  - The best members. We select the nth best elements by groups of the amount defined above.

Diversity measures employed. To select the most diverse members in order to get a higher accuracy in the ensemble, we are going to evaluate the effectiveness of some diversity measures. Thus, allowing us to compare methods with the overproduction and best members selection.

The diversity measures we are going to test are the pairwise-based described in the introduction. The non pairwise diversity measures poses a computational problem when we try to calculate diversity in groups of a determined amount of members, because we have to create all the combinations of classifiers and calculate the diversity among them.

2.1.3.3 Combination Strategy

The combination strategies used are the SMV and WMV as a collective decision strategy of class label fusion. The SMV collective decision strategy allows us to group the individual ANNs outputs that makes up the ensemble in such a way that the correct decisions are amplified, and incorrect ones are eliminated. Furthermore, SMV offers the possibility of comparing the diversity based on the same classifier to highlight the significance of the different spectral characteristics in a simple manner to understand. The WMV gives more relevance to those classifiers with higher accuracy.

We will also test the fusion of continuous outputs, which introduce more complexity in the experiments because it is not irrelevant to measure the diversity as oracle outputs. One possibility would be to measure the diversity from the discretized outputs and then fusion the continuous outputs of those selected members in the ensemble, determining the threshold only in the ensemble over the test set. From the fusion of continuous outputs we will test the mean, product, min, max and median combination rules. The decision of the classifier is based on a threshold that is calculated in the same way in which the classifiers threshold is calculated, but based on the continuous outputs of the classifiers test subset.

2.1.3.4 Software

The simultaneous resolution of complex BF mixtures is a multi-label problem, thus any input pattern will be associated as belonging to as many classes as neurons having been fired in the output layer. This particular problem characteristic is extremely important in the design and implementation of the proposed systems. A new computational tool (MULLPY) has been designed and developed to efficiently and easily design, optimize and ensemble the proposed systems. The tool is written in Python v3.3 [Oliphant, 2007], a language with high attributes in scientific computation, and proven performance in intensive computational processes despite being a scripting language [Álvarez Romero, 2012]. All of the architectures, diversity creation methods, member selection, diversity measures, ensembles, as well as the validation and visualization process have been fully developed in this language using numpy, scipy and matplotlib libraries support [Jones et al., 2013].

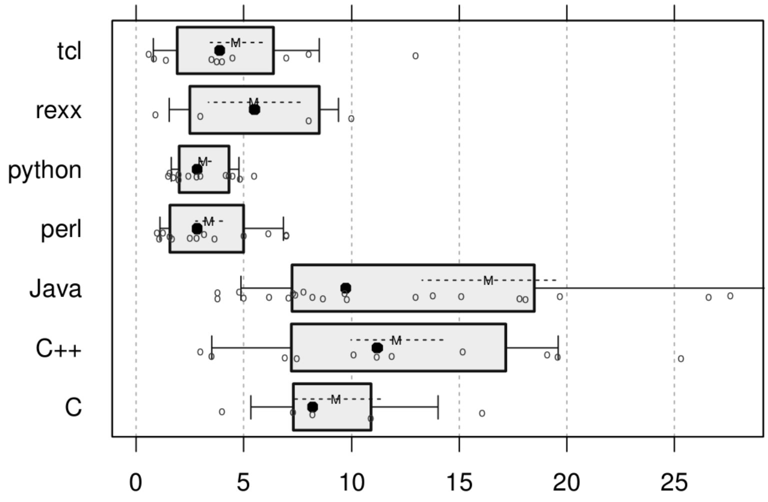


Figure 2.4: Total time in hours for programming in different languages the same programs. Source: [Prechelt, 2000]

Python. Python is a high level language, interpreted (byte code), multi-platform, multi-paradigm and fully object oriented, structured and functional. Its greatest potential lies in its wide variety of libraries, specifically scientific libraries, as shown in Figure 2.5. Its aim consist in providing a very clean syntax to generate code as readable as possible. The community that supports python looks to building code that will allow it to be reviewed and improved in the future. We will quote a few sentences of the python developer, which demonstrates the convergence that has occurred in software engineering in recent decades:

- Explicit is better than implicit
- Simple is better than complex
- Complex is better than complicated
- Sparse is better than dense
- If the implementation is hard to explain, it's a bad idea
- If the implementation is easy to explain, it may be a good idea

These ideas are reflected in the figure 2.4, making a python programmer much more productive.

Languages or programming environments very flexible and with high productivity typically offer less satisfactory computational performance, see Figure 2.6. Despite that, the python language is highly flexible, dynamic and productive, and offer similar performance to those considered intensive computing languages such Fortran, see Figure 2.7. To properly understand the table, it should be noted that Numpy is the math library of the Scipy project, whose code is optimized and proven by extensive scientific community [Chudoba et al., 2013], and that Cython is a Python library that allows introducing explicit code interpreter, to improve performance substantially in those critical fragments of intensive computational cost. The memory consumption is usually one of the weaknesses of interpreted languages. However, as shown in Figure 2.8, we see that not only python presents these shortcomings, but obtains very satisfactory results to be an interpreted language, showing an improvement factor of 2 over java in this aspect [Prechelt, 2000].

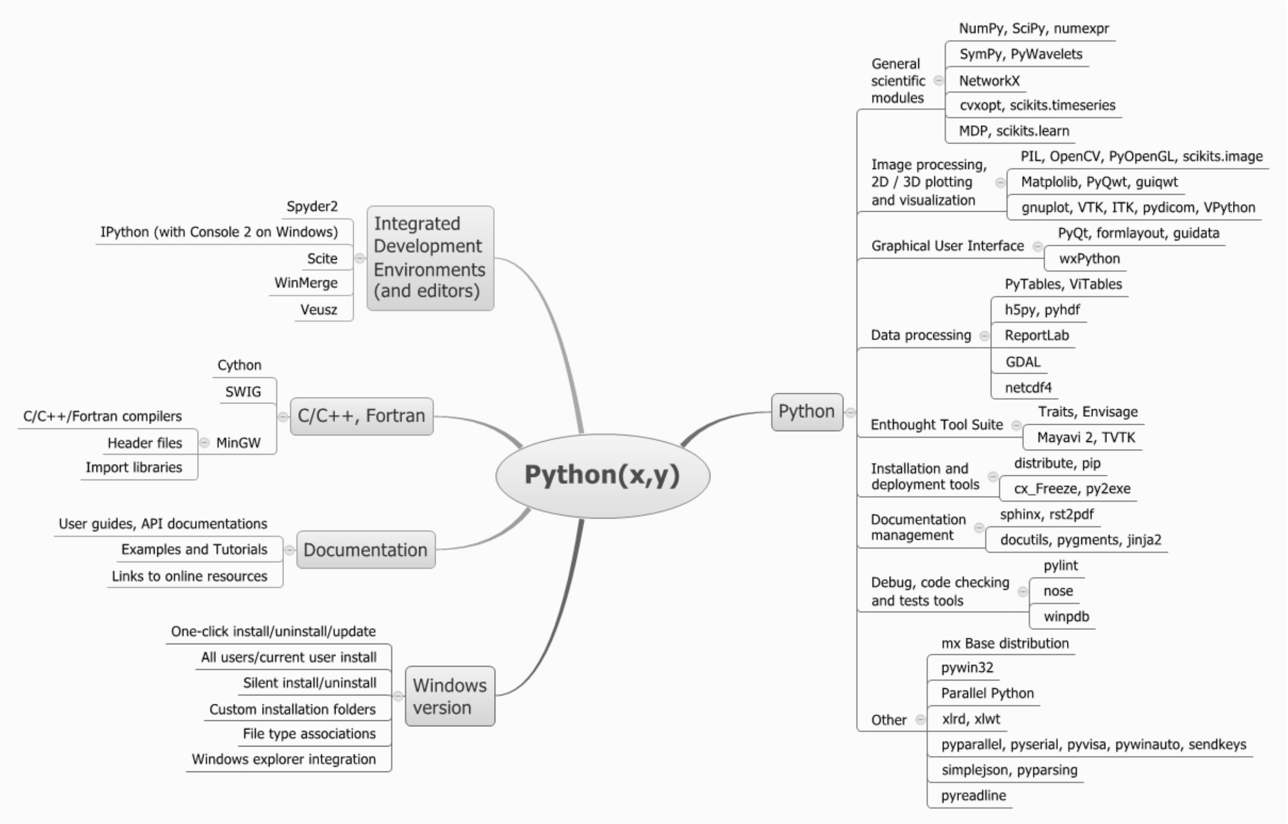


Figure 2.5: Python‘s Library tree

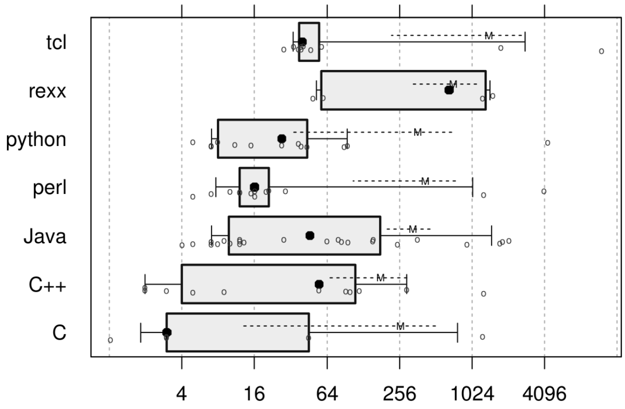
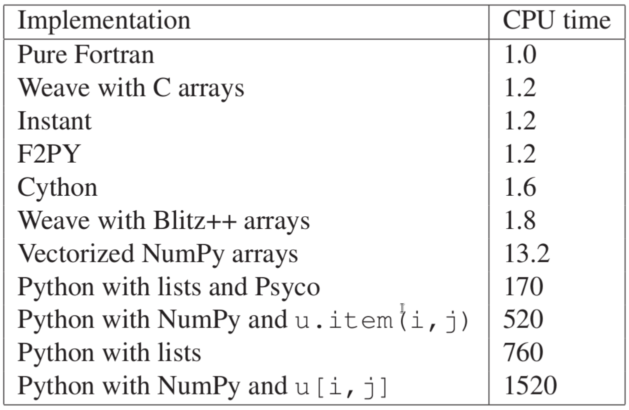


Figure 2.6: Program run time on the z1000 data set. Three programs were timed out with no output after about 21 minutes. The bad/good ratios range from 1.5 for Tcl up to 27 for C++. Note the logarithmic axis. Source: [Prechelt, 2000]



Implementation	CPU time
Pure Fortran	1.0
Weave with C arrays	1.2
Instant	1.2
F2PY	1.2
Cython	1.6
Weave with Blitz++ arrays	1.8
Vectorized NumPy arrays	13.2
Python with lists and Psyco	170
Python with NumPy and `u.item(i,j)`	520
Python with lists	760
Python with NumPy and `u[i,j]`	1520

Figure 2.7: Performance Comparison of programming languages for intensive mathematical computation. Source: [Wilbers et al.,]

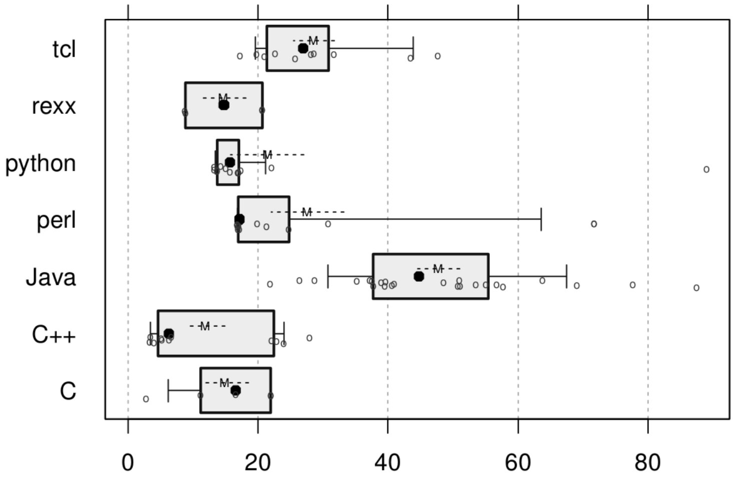


Figure 2.8: Amount of memory required by multiple programs in different languages. Source: [Prechelt, 2000]

Chapter 3

Analysis of generating diversity methods in the design of intelligent systems for BFs detection in complex mixtures

In this chapter, we present the analysis of diversity creating methods, applied to the resolution of complex mixtures of BF from fluorescence spectra with neural ensembles. We also analyse the quantity of diversity generated by each method and the results obtained by using different combination rules. We will evaluate the influence of the different diversity measures on the accuracy of the ensemble. These analyses also describes the influence of the concentrations of substances in complex mixtures, as well as the influence of the number of substances in mixtures. Also, an evaluation study of the various assemblies is presented by using the ROC curves.

3.1 Classifiers evaluation

Many times we have the need to assess whether a supervised classifier is better than another. One possible comparison that can be made is that the classifier that best percentage of correctly classified have is the best classifier. In this and other approaches, the basic process involves calculation of positives (false and true) and in the calculation of the negative (true and false). Positive are understood as all those active classifier outputs, as well as negative are inhibitory outputs of the classifier. In both cases, when we speak of true positive or negative, we are considering that the output given by the classifier corresponds to the desired output of the input pattern. False negatives (FN) or false positives (FP) occur when the expected output does not match with the output of the classifier. However, there is a more formal approach, based on the calculation of the area under the ROC curve of the classifier. The larger the area under the ROC curve, the better the classifier. As well as the ROC curve, there are many others formal approach that takes advantage of the computation of FN and FP, true negatives (TN) and true positives (TP), as the accuracy, hamming loss, error rates, f measure, recall, kappa and so on. We are going to use a more particular approach for this specific problem, because the systems used requires a mixture error function, which we defined with respect to the class detected in any one mixture. This error function is shown in the equations (3.1), where *NCND* is the number of classes present in the mixture undetected by the system, FN; *NCBD* is the number of classes detected by the system which were not present in the mixture, FP and *NCIM* is the total number of classes in the mixture.

$$E = E_{FP} + E_{FN} \; ; E_{FP} = \frac{NCBD}{NCIM} \; ; E_{FN} = \frac{NCND}{NCIM} \quad . \tag{3.1}$$

The upper limit of this function error is equal to the *NCBD* in cases where *NCIM* is equal to one. It means that the worst E that we can find in this work is equal to 4. The validation set has no instances with *NCIM* equal to zero, so it can not be infinite.

As mentioned before, the highest diagnostic accuracy of a test results in a shift up and to the left of the ROC curve. This suggests that the area under the ROC curve can be used as a convenient index of the overall accuracy of the test: maximum accuracy corresponds to the area under the curve of 1 and a minimum value of 0.5. When the ROC curve is generated by the empirical method, regardless of any ties or the area can be approximated by the trapezoidal rule, that is, as the sum of the areas of all rectangles and trapezoids (corresponding to the ties) which form under the curve [Forcada, 2003]. In Figure 3.1, graphically describes the process of analysis of the ROC curve. The dashed red line represents the value that a random classifier or classifier with hit rate of 50%, presented in a curve of this type.

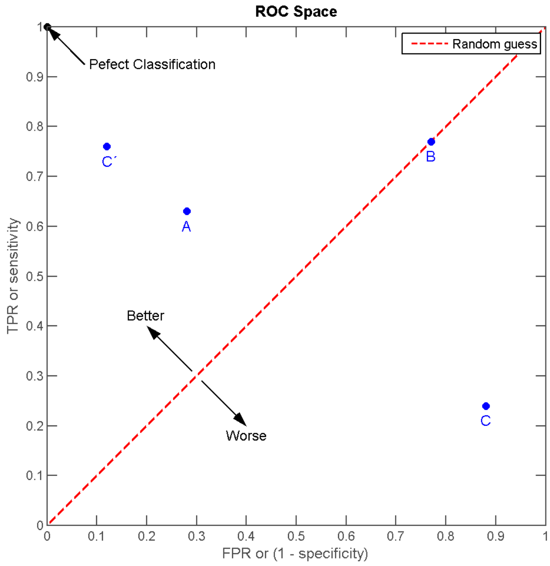


Figure 3.1: Receiver operating characteristic (ROC). Source: wikipedia

3.2 Results and discussion

We evaluate the results from the tables of Errors, the figures of Error by concentration and the figures of error by the number of classes in any mixture. The results on the tables present a set of ensembles with a different amount of members, from 8 to 96, where 8 means to have 1 member from each kind of the 8 spectra which we work with. The different groups of columns, result from the selection of the most diverse BPNs, applying different diversity pairwise measures, where each amount also means that we select the most diverse members from each kind of spectra in the test set, i.e. the ensemble of 24 members selected by diversity measure Q take the 3 most diverse members from each kind of spectra. The best members selection follows the same procedures using test error instead of diversity measures. The figures of error by concentration and classes present in the mixture of the best combination of each diversity creation method are also shown. Firstly, we present a briefly comparison among diversity creation methods and the better ensembles under the SMV scheme. After that, we compare within the same method the results of different combination rules application.

First of all, is important to remark that the results presented by this classifiers vary greatly with respect to previous works [García Báez et al., 2012, Álvarez Romero et al., 2013, Suárez Araujo et al., 2013] because of the limit imposed on the number of iterations performed due to the large amount of computational time required for all diversity creation methods. In these previous works, in the absence of overfitting the learning process continues up to almost absolute zero, which has resulted in a generalization accuracy increase. Another important different about error reduction through the combination of members in that previous works is that we made a greedy search into the all possible combinations space, finding a combination with an $E = 0.016$. The best ensemble with all types of spectra gave an error value of 0.05 which constitutes the value to overcome in this study. In order to draw conclusions about the best method of creating diversity is considered more interesting to build weaker classifiers that can bring greater diversity to the ensembles. On the other hand, without the presence of classifiers with an error different to 0.0 in test set, the selection of the members to better cover the decision space becomes a very complex task.

Table 3.1: Average mixture errors and their standard deviation between parentheses using neural ensembles with HNV and WI diversity creation methods and diversity measures with SMV. The best member E was 0.521.

		DF			Dis			p			Q			Best		
	Members	E	E_{FN}	E_{FP}	E	E_{FN}	E_{FP}	E	E_{FN}	E_{FP}	E	E_{FN}	E_{FP}	E	E_{FN}	E_{FP}
HNV	8	0.647(0.203)	0.647(0.203)	0.000(0.000)	**0.596(0.202)**	0.596(0.202)	0.000(0.000)	0.647(0.203)	0.647(0.203)	0.000(0.000)	0.647(0.203)	0.647(0.203)	0.000(0.000)	0.626(0.202)	0.626(0.202)	0.000(0.000)
	16	0.609(0.202)	0.609(0.202)	0.000(0.000)	0.663(0.203)	0.663(0.203)	0.000(0.000)	0.663(0.203)	0.663(0.203)	0.000(0.000)	0.663(0.203)	0.663(0.203)	0.000(0.000)	0.672(0.206)	0.672(0.206)	0.000(0.000)
	24	0.669(0.204)	0.669(0.204)	0.000(0.000)	0.616(0.202)	0.616(0.202)	0.000(0.000)	0.669(0.204)	0.669(0.204)	0.000(0.000)	0.669(0.204)	0.669(0.204)	0.000(0.000)	0.684(0.206)	0.684(0.206)	0.000(0.000)
	32	0.664(0.204)	0.664(0.204)	0.000(0.000)	0.620(0.202)	0.620(0.202)	0.000(0.000)	0.664(0.204)	0.664(0.204)	0.000(0.000)	0.664(0.204)	0.664(0.204)	0.000(0.000)	0.673(0.205)	0.673(0.205)	0.000(0.000)
	40	0.668(0.204)	0.668(0.204)	0.000(0.000)	0.615(0.202)	0.615(0.202)	0.000(0.000)	0.668(0.204)	0.668(0.204)	0.000(0.000)	0.668(0.204)	0.668(0.204)	0.000(0.000)	0.671(0.205)	0.671(0.205)	0.000(0.000)
	48	0.672(0.205)	0.672(0.205)	0.000(0.000)	0.617(0.202)	0.617(0.202)	0.000(0.000)	0.672(0.205)	0.672(0.205)	0.000(0.000)	0.672(0.205)	0.672(0.205)	0.000(0.000)	0.672(0.205)	0.672(0.205)	0.000(0.000)
	56	0.670(0.205)	0.670(0.205)	0.000(0.000)	0.633(0.203)	0.633(0.203)	0.000(0.000)	0.670(0.205)	0.670(0.205)	0.000(0.000)	0.670(0.205)	0.670(0.205)	0.000(0.000)	0.672(0.205)	0.672(0.205)	0.000(0.000)
	64	0.669(0.204)	0.669(0.204)	0.000(0.000)	0.642(0.204)	0.642(0.204)	0.000(0.000)	0.669(0.204)	0.669(0.204)	0.000(0.000)	0.669(0.204)	0.669(0.204)	0.000(0.000)	0.671(0.205)	0.671(0.205)	0.000(0.000)
	72	0.669(0.204)	0.669(0.204)	0.000(0.000)	0.636(0.204)	0.636(0.204)	0.000(0.000)	0.669(0.204)	0.669(0.204)	0.000(0.000)	0.669(0.204)	0.669(0.204)	0.000(0.000)	0.674(0.205)	0.674(0.205)	0.000(0.000)
	80	0.671(0.205)	0.671(0.205)	0.000(0.000)	0.636(0.204)	0.636(0.204)	0.000(0.000)	0.671(0.205)	0.671(0.205)	0.000(0.000)	0.671(0.205)	0.671(0.205)	0.000(0.000)	0.671(0.205)	0.671(0.205)	0.000(0.000)
	88	0.669(0.204)	0.669(0.204)	0.000(0.000)	0.636(0.204)	0.636(0.204)	0.000(0.000)	0.669(0.204)	0.669(0.204)	0.000(0.000)	0.669(0.204)	0.669(0.204)	0.000(0.000)	0.668(0.205)	0.668(0.205)	0.000(0.000)
	96	0.666(0.204)	0.666(0.204)	0.000(0.000)	0.636(0.204)	0.636(0.204)	0.000(0.000)	0.666(0.204)	0.666(0.204)	0.000(0.000)	0.666(0.204)	0.666(0.204)	0.000(0.000)	0.668(0.205)	0.668(0.205)	0.000(0.000)
	All	0.643(0.204) 0.643(0.204) 0.000(0.000)														
WI	8	0.589(0.198)	0.589(0.198)	0.000(0.000)	0.517(0.192)	0.517(0.192)	0.000(0.000)	0.589(0.198)	0.589(0.198)	0.000(0.000)	0.589(0.198)	0.589(0.198)	0.000(0.000)	0.592(0.199)	0.592(0.199)	0.000(0.000)
	16	0.647(0.205)	0.647(0.205)	0.000(0.000)	**0.514(0.192)**	0.514(0.192)	0.000(0.000)	0.647(0.205)	0.647(0.205)	0.000(0.000)	0.647(0.205)	0.647(0.205)	0.000(0.000)	0.637(0.203)	0.637(0.203)	0.000(0.000)
	24	0.661(0.205)	0.661(0.205)	0.000(0.000)	0.515(0.192)	0.515(0.192)	0.000(0.000)	0.661(0.205)	0.661(0.205)	0.000(0.000)	0.661(0.205)	0.661(0.205)	0.000(0.000)	0.638(0.203)	0.638(0.203)	0.000(0.000)
	32	0.663(0.206)	0.663(0.206)	0.000(0.000)	0.520(0.192)	0.520(0.192)	0.000(0.000)	0.663(0.206)	0.663(0.206)	0.000(0.000)	0.663(0.206)	0.663(0.206)	0.000(0.000)	0.639(0.203)	0.639(0.203)	0.000(0.000)
	40	0.661(0.205)	0.661(0.205)	0.000(0.000)	0.528(0.194)	0.528(0.194)	0.000(0.000)	0.661(0.205)	0.661(0.205)	0.000(0.000)	0.652(0.205)	0.652(0.205)	0.000(0.000)	0.639(0.203)	0.639(0.203)	0.000(0.000)
	48	0.652(0.205)	0.652(0.205)	0.000(0.000)	0.538(0.194)	0.538(0.194)	0.000(0.000)	0.652(0.205)	0.652(0.205)	0.000(0.000)	0.665(0.206)	0.665(0.206)	0.000(0.000)	0.646(0.204)	0.646(0.204)	0.000(0.000)
	56	0.665(0.206)	0.665(0.206)	0.000(0.000)	0.548(0.196)	0.548(0.196)	0.000(0.000)	0.665(0.206)	0.665(0.206)	0.000(0.000)	0.653(0.204)	0.653(0.204)	0.000(0.000)	0.644(0.204)	0.644(0.204)	0.000(0.000)
	64	0.653(0.204)	0.653(0.204)	0.000(0.000)	0.554(0.196)	0.554(0.196)	0.000(0.000)	0.653(0.204)	0.653(0.204)	0.000(0.000)	0.661(0.205)	0.661(0.205)	0.000(0.000)	0.644(0.204)	0.644(0.204)	0.000(0.000)
	72	0.675(0.206)	0.675(0.206)	0.000(0.000)	0.558(0.196)	0.558(0.196)	0.000(0.000)	0.675(0.206)	0.675(0.206)	0.000(0.000)	0.675(0.206)	0.675(0.206)	0.000(0.000)	0.643(0.204)	0.643(0.204)	0.000(0.000)
	80	0.672(0.206)	0.672(0.206)	0.000(0.000)	0.560(0.196)	0.560(0.196)	0.000(0.000)	0.672(0.206)	0.672(0.206)	0.000(0.000)	0.672(0.206)	0.672(0.206)	0.000(0.000)	0.643(0.204)	0.643(0.204)	0.000(0.000)
	88	0.674(0.206)	0.674(0.206)	0.000(0.000)	0.574(0.197)	0.574(0.197)	0.000(0.000)	0.674(0.206)	0.674(0.206)	0.000(0.000)	0.674(0.206)	0.674(0.206)	0.000(0.000)	0.643(0.204)	0.643(0.204)	0.000(0.000)
	96	0.667(0.206)	0.667(0.206)	0.000(0.000)	0.585(0.198)	0.585(0.198)	0.000(0.000)	0.667(0.206)	0.667(0.206)	0.000(0.000)	0.667(0.206)	0.667(0.206)	0.000(0.000)	0.640(0.203)	0.640(0.203)	0.000(0.000)
	All	0.643(0.204) 0.643(0.204) 0.000(0.000)														

The table 3.1 shows the results of HNV and WI methods. The best members in WI was made selecting the first *n* elements of the selected amount of hidden neurons, from the 30 BPNs generated with that configuration, for each *m* amount of members in the ensemble.

The best member column in HNV was made selecting the best 2 generations from the 30 BPNs of each amount of hidden neurons of each spectra, in order to be possible to build ensembles of 96 members. The HNV method offer poorest results than the WI method. These results seem pretty obvious, because it is not a method of actual generation of diversity, but the use of all classifiers trained during the selection process for members.

From the point of view of the diversity measures we found that *Dis* achieved the best results in HNV method as well as in the WI method. As we pointed in the definition of the diversity measures, the absence of products in the formula prevents any anomalies in the results due to some products with zero member. In fact, the other 3 measures employed, DF, p and Q obtained the same best results. The selection of the best members to make up the ensemble achieved best results in some cases than those 3 diversity measures in the HNV, but not in the case of WI, at least in the case of 8 members. Generally, the diversity measures allow better ensembles than selection by the best members. The number of members in the ensemble does not provide a better error of the same. This results are obviously worst than the obtained in previous works, as we mentioned before, where we achieved very optimized members, BPN_3 with an error of 0.029 until the worst member, the BPN_8, with an error of 0.283 [Álvarez Romero et al., 2013]. This represents a deterioration in the generalization ability of the members that will be difficult to overcome through their combination. The causes of this decline have several sources, some already mentioned above. A much smaller number of epochs, introducing bias nodes in the hidden layer, and the stopping criterion that sums the training error and the test to measure the objective mistake.

From this point in advance, the number of hidden neurons, in order, for the different types of spectra are defined as 4,4,4,4,10,3,9 and 4 as consequence of the evaluation by RMSE.

Table 3.2: Average mixture errors of identification values and their standard deviation between parentheses using neural ensembles with Bagging and Random subspaces diversity creation methods and diversity measures with SMV.

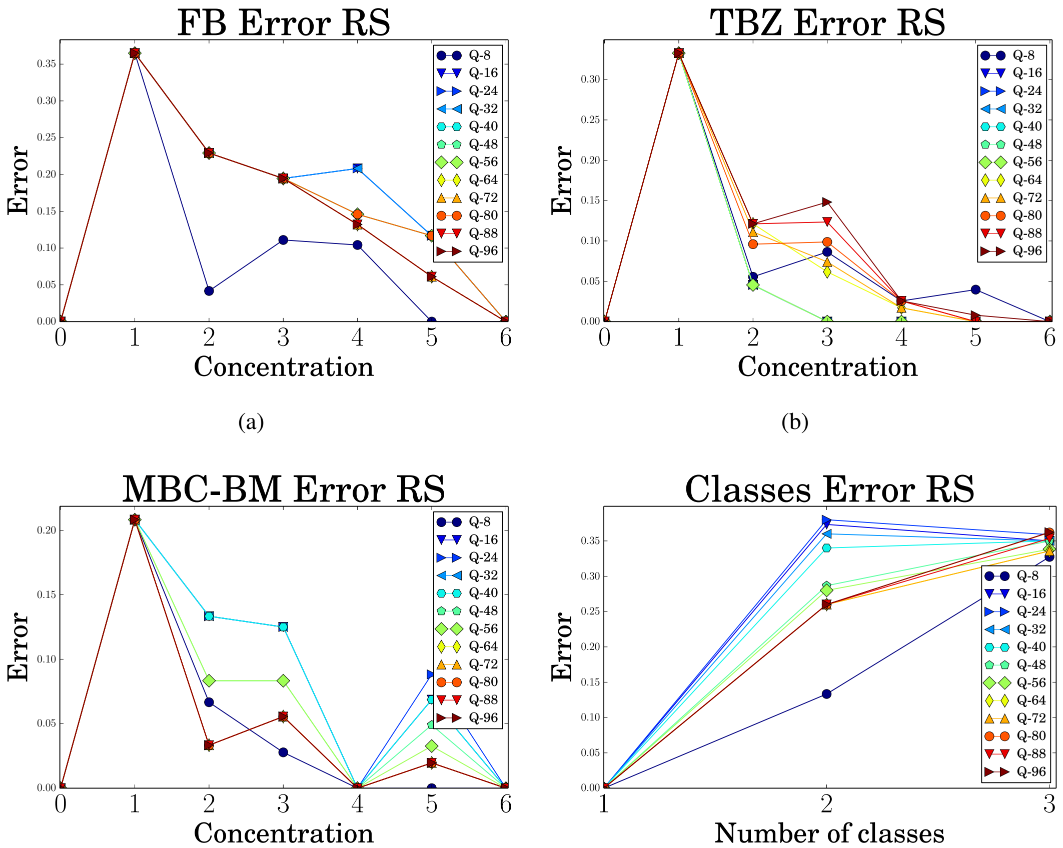
		DF			Dis			p			Q			Best		
	Members	E	E_{FN}	E_{FP}	E	E_{FN}	E_{FP}	E	E_{FN}	E_{FP}	E	E_{FN}	E_{FP}	E	E_{FN}	E_{FP}
Bagging	8	0.297(0.158)	0.267(0.149)	0.030(0.061)	0.308(0.159)	0.278(0.151)	0.030(0.061)	0.303(0.159)	0.273(0.150)	0.030(0.061)	0.296(0.157)	0.266(0.149)	0.030(0.061)	0.279(0.153)	0.249(0.144)	0.030(0.061)
	16	0.270(0.151)	0.240(0.141)	0.030(0.061)	**0.261(0.150)**	0.231(0.140)	0.030(0.061)	0.267(0.150)	0.237(0.141)	0.030(0.061)	0.298(0.158)	0.268(0.149)	0.030(0.061)	**0.260(0.149)**	0.230(0.139)	0.030(0.061)
	24	0.273(0.152)	0.243(0.143)	0.030(0.061)	0.268(0.152)	0.238(0.142)	0.030(0.061)	0.288(0.155)	0.258(0.146)	0.030(0.061)	0.280(0.153)	0.250(0.144)	0.030(0.061)	0.280(0.153)	0.250(0.144)	0.030(0.061)
	32	0.286(0.154)	0.256(0.145)	0.030(0.061)	0.292(0.156)	0.262(0.147)	0.030(0.061)	0.263(0.149)	0.233(0.139)	0.030(0.061)	0.266(0.149)	0.236(0.140)	0.030(0.061)	0.286(0.154)	0.256(0.145)	0.030(0.061)
	40	0.273(0.153)	0.243(0.143)	0.030(0.061)	0.307(0.160)	0.277(0.151)	0.030(0.061)	0.264(0.149)	0.234(0.139)	0.030(0.061)	0.262(0.148)	0.232(0.139)	0.030(0.061)	0.287(0.155)	0.257(0.146)	0.030(0.061)
	48	**0.257(0.149)**	0.227(0.139)	0.030(0.061)	0.292(0.156)	0.262(0.147)	0.030(0.061)	**0.261(0.148)**	0.231(0.138)	0.030(0.061)	**0.261(0.148)**	0.231(0.138)	0.030(0.061)	0.283(0.154)	0.253(0.145)	0.030(0.061)
	56	0.263(0.150)	0.233(0.141)	0.030(0.061)	0.300(0.158)	0.270(0.149)	0.030(0.061)	0.268(0.150)	0.238(0.141)	0.030(0.061)	0.259(0.148)	0.229(0.138)	0.030(0.061)	0.280(0.153)	0.250(0.144)	0.030(0.061)
	64	0.280(0.154)	0.250(0.144)	0.030(0.061)	0.303(0.158)	0.273(0.150)	0.030(0.061)	0.256(0.147)	0.226(0.137)	0.030(0.061)	0.258(0.147)	0.228(0.137)	0.030(0.061)	0.279(0.153)	0.249(0.144)	0.030(0.061)
	72	0.286(0.155)	0.256(0.146)	0.030(0.061)	0.307(0.159)	0.277(0.151)	0.030(0.061)	0.258(0.147)	0.228(0.137)	0.030(0.061)	0.254(0.146)	0.224(0.136)	0.030(0.061)	0.279(0.153)	0.249(0.144)	0.030(0.061)
	80	0.281(0.154)	0.251(0.145)	0.030(0.061)	0.304(0.159)	0.274(0.150)	0.030(0.061)	0.261(0.148)	0.231(0.138)	0.030(0.061)	0.254(0.146)	0.224(0.136)	0.030(0.061)	0.279(0.153)	0.249(0.144)	0.030(0.061)
	88	0.278(0.153)	0.248(0.144)	0.030(0.061)	0.310(0.160)	0.280(0.152)	0.030(0.061)	0.261(0.148)	0.231(0.138)	0.030(0.061)	0.261(0.148)	0.231(0.138)	0.030(0.061)	0.278(0.153)	0.248(0.143)	0.030(0.061)
	96	0.279(0.153)	0.249(0.144)	0.030(0.061)	0.303(0.158)	0.273(0.150)	0.030(0.061)	0.264(0.148)	0.234(0.139)	0.030(0.061)	0.265(0.149)	0.235(0.139)	0.030(0.061)	0.276(0.152)	0.246(0.143)	0.030(0.061)
	All	0.309(0.159) 0.279(0.150) 0.030(0.061)														
RS	8	0.398(0.175)	0.398(0.175)	0.000(0.000)	0.403(0.174)	0.403(0.174)	0.000(0.000)	0.278(0.180)	0.178(0.124)	0.100(0.139)	**0.162(0.116)**	0.162(0.116)	0.000(0.000)	0.425(0.180)	0.425(0.180)	0.000(0.000)
	16	0.408(0.177)	0.408(0.177)	0.000(0.000)	0.279(0.150)	0.279(0.150)	0.000(0.000)	0.329(0.190)	0.226(0.140)	0.103(0.140)	0.232(0.142)	0.232(0.142)	0.000(0.000)	0.440(0.182)	0.440(0.182)	0.000(0.000)
	24	0.394(0.174)	0.394(0.174)	0.000(0.000)	0.266(0.148)	0.266(0.148)	0.000(0.000)	0.306(0.184)	0.233(0.142)	0.073(0.126)	0.237(0.143)	0.236(0.142)	0.002(0.014)	0.407(0.176)	0.407(0.176)	0.000(0.000)
	32	0.389(0.173)	0.389(0.173)	0.000(0.000)	0.266(0.147)	0.266(0.147)	0.000(0.000)	0.267(0.167)	0.230(0.141)	0.037(0.095)	0.229(0.140)	0.229(0.140)	0.000(0.000)	0.357(0.166)	0.357(0.166)	0.000(0.000)
	40	**0.355(0.166)**	0.355(0.166)	0.000(0.000)	0.279(0.149)	0.279(0.149)	0.000(0.000)	0.224(0.139)	0.224(0.139)	0.000(0.000)	0.224(0.139)	0.224(0.139)	0.000(0.000)	0.351(0.166)	0.351(0.166)	0.000(0.000)
	48	0.372(0.169)	0.372(0.169)	0.000(0.000)	0.239(0.140)	0.239(0.140)	0.000(0.000)	**0.198(0.131)**	0.198(0.131)	0.000(0.000)	0.211(0.134)	0.211(0.134)	0.000(0.000)	**0.348(0.165)**	0.348(0.165)	0.000(0.000)
	56	0.376(0.170)	0.376(0.170)	0.000(0.000)	0.236(0.140)	0.236(0.140)	0.000(0.000)	0.203(0.132)	0.203(0.132)	0.000(0.000)	0.204(0.132)	0.204(0.132)	0.000(0.000)	0.356(0.166)	0.356(0.166)	0.000(0.000)
	64	0.405(0.175)	0.405(0.175)	0.000(0.000)	**0.218(0.136)**	0.218(0.136)	0.000(0.000)	0.210(0.134)	0.210(0.134)	0.000(0.000)	0.203(0.132)	0.203(0.132)	0.000(0.000)	0.368(0.168)	0.368(0.168)	0.000(0.000)
	72	0.404(0.175)	0.404(0.175)	0.000(0.000)	0.238(0.140)	0.238(0.140)	0.000(0.000)	0.210(0.134)	0.210(0.134)	0.000(0.000)	0.203(0.132)	0.203(0.132)	0.000(0.000)	0.369(0.168)	0.369(0.168)	0.000(0.000)
	80	0.400(0.174)	0.400(0.174)	0.000(0.000)	0.262(0.145)	0.262(0.145)	0.000(0.000)	0.211(0.134)	0.211(0.134)	0.000(0.000)	0.212(0.134)	0.212(0.134)	0.000(0.000)	0.365(0.168)	0.365(0.168)	0.000(0.000)
	88	0.394(0.173)	0.394(0.173)	0.000(0.000)	0.255(0.143)	0.255(0.143)	0.000(0.000)	0.219(0.136)	0.219(0.136)	0.000(0.000)	0.210(0.134)	0.210(0.134)	0.000(0.000)	0.365(0.168)	0.365(0.168)	0.000(0.000)
	96	0.381(0.170)	0.381(0.170)	0.000(0.000)	0.250(0.143)	0.250(0.143)	0.000(0.000)	0.212(0.134)	0.212(0.134)	0.000(0.000)	0.213(0.134)	0.213(0.134)	0.000(0.000)	0.360(0.168)	0.360(0.168)	0.000(0.000)
	All	0.268(0.150) 0.238(0.141) 0.030(0.061)														

In table 3.2, with the bagging method we does not found a correlation between the number of classifiers and a decreasing of the error. The disagreement diversity measure application results in a worst accuracy than other measures, even compared to the best member selection. Other consideration about the bagging method is the increase of the false positive error.

The RS method achieved better results than the bagging, HNV and WI ones. The best results was obtained by using the Q statistic to select members, also having more homogeneous behaviour and showing once again that no diversity measure is able to get the best combination for any problem. Taking into account that the 8 best members by test set of the RS method has an average error of 0.425, the values obtained by the combinations selected by diversity measures highlight the importance of such measures. Moreover, RS obtained the best results in combining all the classifiers generated during the experiments. The RS method has better results in some diversity measures compared to Bgg method, but the best members ensemble and those ensembles guided by the DF measure are worst than the reciprocal in Bgg. This highlight the importance of selecting diverse members over the best members, specially in the RS case. The Bgg method results does not have high variations over diversity measures and amount of members in the ensemble. This method also get better results than HNV and WI.

In both cases, Bgg and RS, the combinations over 40 members in the ensemble are more stable, despite the best case founded in RS has 8 members. In the figure 3.2, we can see how the problems in identification comes from the low concentration errors, specially in mixtures of 3 components. The same occurs in the best combination founded in the bagging method, see Figure 3.3. The Bgg had no FN problem in identifying the MBC-BM class, but had higher FP error.

Figure 3.2: Influence of the concentration of the analytes in the mixture over the average of the errors for the RS method: (a) FB, (b) TBZ and (c) MBC-BM. (d) Influence of the number of classes from the mixture in the average of the errors of the mixtures.



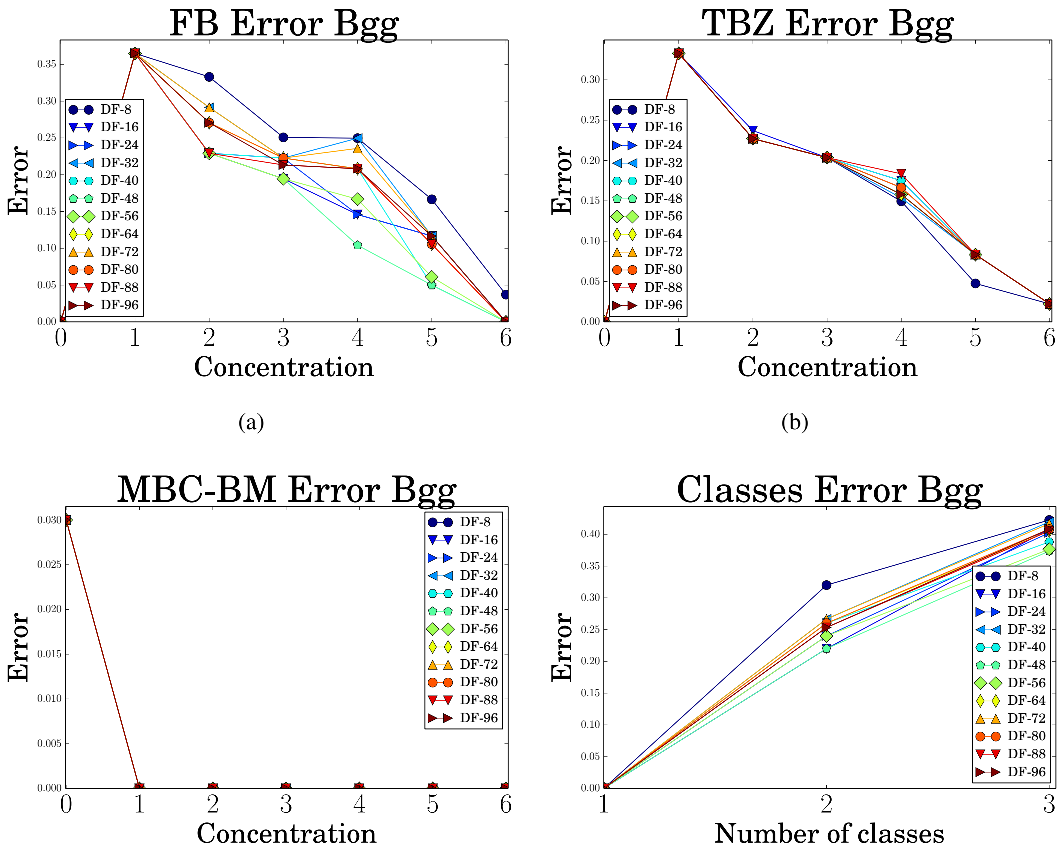
(a)

(b)

(c)

(d)

Figure 3.3: Influence of the concentration of the analytes in the mixture over the average of the errors for the Bgg method: (a) FB, (b) TBZ and (c) MBC-BM. (d) Influence of the number of classes from the mixture in the average of the errors of the mixtures.



(a)

(b)

(c)

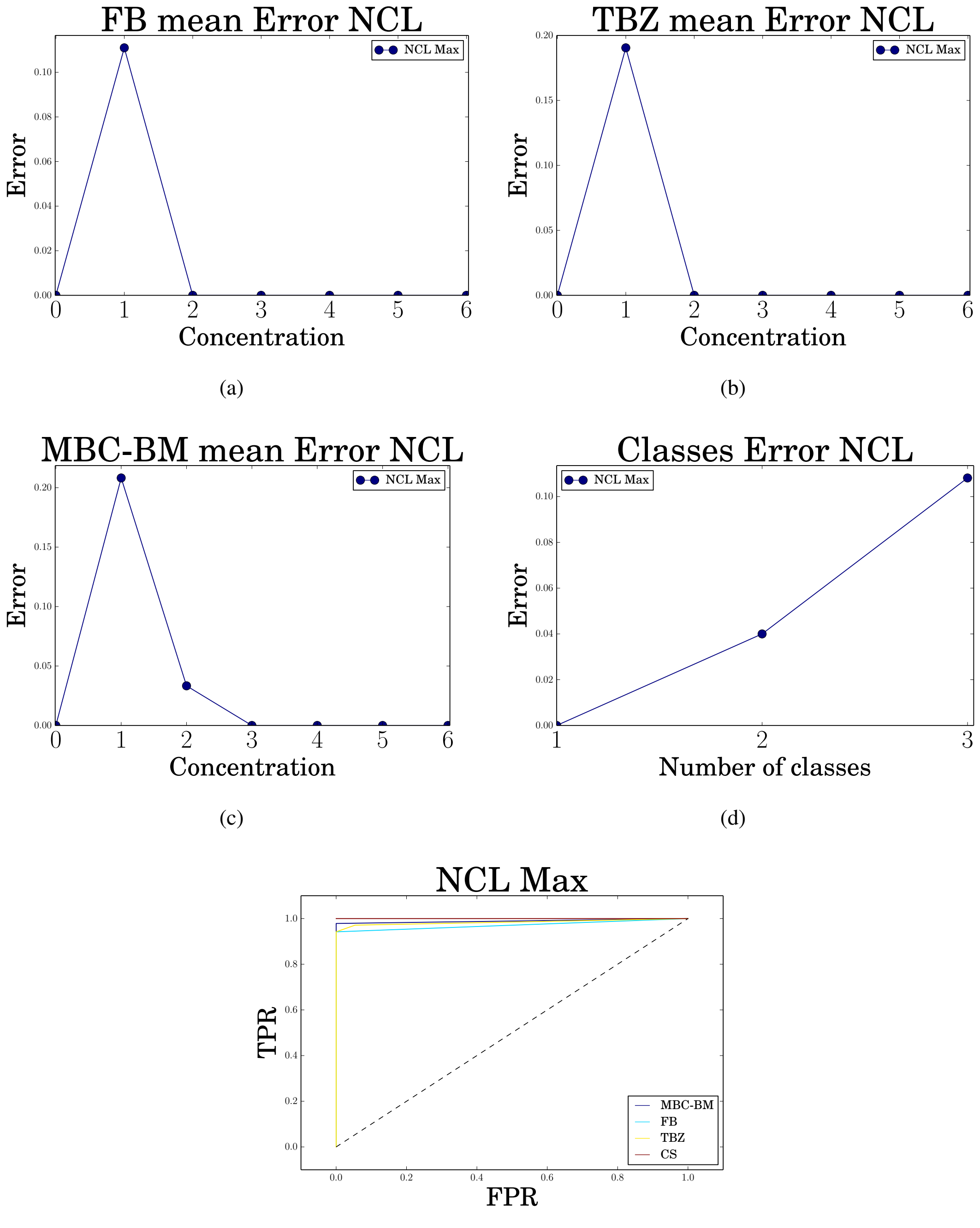
(d)

In table 3.3, we have compared the diversity creation method NCL, with different combination rules, different values of λ and different amount of members in the ensemble. As we mentioned before, the original NCL method has a correlation diversity measure and a mean combination rule. Here, we altered that consideration just while taking decisions, not during the learning process. Related to the number of members, we can observe very little variations in the results. There is not improvements in the results with higher number of members and viceversa. In the case of the value of λ there is a correlation, near always, between the increase of the value and the decrease of the ensemble error in mixture identification of BFs. As in other methods, we observe that the Max combination rule produces better results than other rules like mean. It seems again to be related with this data set, due to the difficulties in the detection of low concentration of BFs, select the max value of the ensemble members provide more probabilities to get over the threshold, avoiding many FN errors. Despite that, the mean rule tends to be more stable than others. The best result obtained in this experiment was with the Max rule and $\lambda = 1.0$, with an error of 0.051. In some cases a reduction of the EFN was reduced to 0.038, but having also a bigger EFP. The class fusion combination rules (SMV and WMV) produced acceptable results, but with the increase of the number of member in the ensemble, there is also an increase of the total error. The WMV corrects this influence by weighting the best members. In fact, the NCL with WMV and 40 members, had one of the better results achieves in this study with an error of 0.059, but the most important thing, a $EFN = 0.044$, which is one of the lowest together with the EFP. In the figure 3.4 (a), (b) and (c), we can see that low concentrations of all the classes related to fungicide are producing the errors, in mixtures of 2 and 3 classes 3.4(d). The system have a good overall efficiency in terms of area under the curve 3.4(e).

Table 3.3: Average mixture errors of identification values and their standard deviation between parentheses using neural ensembles with NCL scheme for different values of λ with some combination rules.

		$k = 0.3$			$k = 0.5$			$k = 1.0$		
	Members	E	E_{FN}	E_{FP}	E	E_{FN}	E_{FP}	E	E_{FN}	E_{FP}
NCL SMV	8	0.165(0.119)	0.165(0.119)	0.000(0.000)	0.159(0.118)	0.159(0.118)	0.000(0.000)	**0.089(0.088)**	0.089(0.088)	0.000(0.000)
	16	0.175(0.123)	0.175(0.123)	0.000(0.000)	0.159(0.118)	0.159(0.118)	0.000(0.000)	0.154(0.116)	0.154(0.116)	0.000(0.000)
	24	0.171(0.121)	0.171(0.121)	0.000(0.000)	0.163(0.118)	0.163(0.118)	0.000(0.000)	0.105(0.093)	0.105(0.093)	0.000(0.000)
	32	0.187(0.128)	0.187(0.128)	0.000(0.000)	0.155(0.115)	0.155(0.115)	0.000(0.000)	0.117(0.100)	0.117(0.100)	0.000(0.000)
	40	0.165(0.119)	0.165(0.119)	0.000(0.000)	0.148(0.113)	0.148(0.113)	0.000(0.000)	0.128(0.105)	0.128(0.105)	0.000(0.000)
	48	0.166(0.119)	0.166(0.119)	0.000(0.000)	0.154(0.115)	0.154(0.115)	0.000(0.000)	0.139(0.111)	0.139(0.111)	0.000(0.000)
	56	0.325(0.159)	0.325(0.159)	0.000(0.000)	0.321(0.157)	0.321(0.157)	0.000(0.000)	0.293(0.153)	0.293(0.153)	0.000(0.000)
	64	0.323(0.159)	0.323(0.159)	0.000(0.000)	0.301(0.153)	0.301(0.153)	0.000(0.000)	0.300(0.154)	0.300(0.154)	0.000(0.000)
	72	0.326(0.159)	0.326(0.159)	0.000(0.000)	0.316(0.157)	0.316(0.157)	0.000(0.000)	0.302(0.154)	0.302(0.154)	0.000(0.000)
	80	0.313(0.156)	0.313(0.156)	0.000(0.000)	0.299(0.153)	0.299(0.153)	0.000(0.000)	0.294(0.153)	0.294(0.153)	0.000(0.000)
	88	0.318(0.158)	0.318(0.158)	0.000(0.000)	0.303(0.154)	0.303(0.154)	0.000(0.000)	0.293(0.152)	0.293(0.152)	0.000(0.000)
	96	0.319(0.158)	0.319(0.158)	0.000(0.000)	0.294(0.152)	0.294(0.152)	0.000(0.000)	0.296(0.153)	0.296(0.153)	0.000(0.000)
NCL WMV	8	0.141(0.114)	0.108(0.097)	0.033(0.064)	0.091(0.089)	0.091(0.089)	0.000(0.000)	0.073(0.080)	0.073(0.080)	0.000(0.000)
	16	0.218(0.145)	0.108(0.099)	0.110(0.113)	0.093(0.090)	0.093(0.090)	0.000(0.000)	0.100(0.098)	0.053(0.065)	0.047(0.075)
	24	0.176(0.129)	0.106(0.096)	0.070(0.092)	0.177(0.134)	0.062(0.073)	0.115(0.116)	0.086(0.087)	0.071(0.077)	0.015(0.043)
	32	0.204(0.141)	0.084(0.085)	0.120(0.119)	0.208(0.142)	0.097(0.091)	0.112(0.115)	0.124(0.115)	0.032(0.051)	0.092(0.105)
	40	0.114(0.109)	0.046(0.063)	0.068(0.091)	0.177(0.135)	0.062(0.074)	0.115(0.116)	**0.059(0.073)**	**0.044(0.060)**	**0.015(0.043)**
	48	0.265(0.154)	0.096(0.091)	0.169(0.132)	0.222(0.145)	0.092(0.089)	0.130(0.121)	0.173(0.133)	0.034(0.053)	0.139(0.125)
	56	0.172(0.122)	0.172(0.122)	0.000(0.000)	0.197(0.128)	0.197(0.128)	0.000(0.000)	0.190(0.127)	0.190(0.127)	0.000(0.000)
	64	0.196(0.130)	0.196(0.130)	0.000(0.000)	0.208(0.132)	0.208(0.132)	0.000(0.000)	0.201(0.129)	0.201(0.129)	0.000(0.000)
	72	0.177(0.123)	0.177(0.123)	0.000(0.000)	0.202(0.131)	0.202(0.131)	0.000(0.000)	0.201(0.131)	0.192(0.127)	0.008(0.032)
	80	0.154(0.113)	0.154(0.113)	0.000(0.000)	0.177(0.122)	0.177(0.122)	0.000(0.000)	0.185(0.125)	0.185(0.125)	0.000(0.000)
	88	0.145(0.112)	0.145(0.112)	0.000(0.000)	0.157(0.116)	0.157(0.116)	0.000(0.000)	0.179(0.123)	0.177(0.122)	0.002(0.014)
	96	0.191(0.127)	0.191(0.127)	0.000(0.000)	0.190(0.127)	0.190(0.127)	0.000(0.000)	0.187(0.126)	0.172(0.120)	0.015(0.043)
NCL mean	8	0.136(0.107)	0.136(0.107)	0.000(0.000)	0.129(0.103)	0.129(0.103)	0.000(0.000)	0.103(0.094)	0.103(0.094)	0.000(0.000)
	16	0.155(0.116)	0.155(0.116)	0.000(0.000)	0.120(0.100)	0.120(0.100)	0.000(0.000)	0.098(0.092)	0.098(0.092)	0.000(0.000)
	24	0.148(0.113)	0.148(0.113)	0.000(0.000)	0.119(0.099)	0.119(0.099)	0.000(0.000)	0.097(0.091)	0.097(0.091)	0.000(0.000)
	32	0.142(0.110)	0.142(0.110)	0.000(0.000)	0.121(0.100)	0.121(0.100)	0.000(0.000)	**0.074(0.080)**	0.074(0.080)	0.000(0.000)
	40	0.125(0.102)	0.125(0.102)	0.000(0.000)	0.095(0.090)	0.095(0.090)	0.000(0.000)	0.085(0.086)	0.085(0.086)	0.000(0.000)
	48	0.139(0.109)	0.134(0.106)	0.005(0.025)	0.111(0.096)	0.111(0.096)	0.000(0.000)	0.082(0.085)	0.082(0.085)	0.000(0.000)
	56	0.132(0.106)	0.127(0.103)	0.005(0.025)	0.101(0.093)	0.101(0.093)	0.000(0.000)	0.085(0.086)	0.085(0.086)	0.000(0.000)
	64	0.156(0.117)	0.136(0.107)	0.020(0.050)	0.126(0.103)	0.126(0.103)	0.000(0.000)	0.087(0.087)	0.087(0.087)	0.000(0.000)
	72	0.135(0.106)	0.135(0.106)	0.000(0.000)	0.106(0.095)	0.106(0.095)	0.000(0.000)	0.090(0.088)	0.090(0.088)	0.000(0.000)
	80	0.119(0.099)	0.119(0.099)	0.000(0.000)	0.100(0.092)	0.100(0.092)	0.000(0.000)	0.088(0.088)	0.088(0.088)	0.000(0.000)
	88	0.124(0.102)	0.119(0.099)	0.005(0.025)	0.113(0.097)	0.113(0.097)	0.000(0.000)	0.085(0.086)	0.085(0.086)	0.000(0.000)
	96	0.124(0.102)	0.124(0.102)	0.000(0.000)	0.113(0.097)	0.113(0.097)	0.000(0.000)	0.085(0.086)	0.085(0.086)	0.000(0.000)
NCL Max	8	0.112(0.098)	0.107(0.095)	0.005(0.025)	0.078(0.081)	0.078(0.081)	0.000(0.000)	0.076(0.081)	0.076(0.081)	0.000(0.000)
	16	0.173(0.128)	0.108(0.097)	0.065(0.089)	0.076(0.081)	0.076(0.081)	0.000(0.000)	0.107(0.103)	0.060(0.072)	0.047(0.075)
	24	0.122(0.107)	0.090(0.088)	0.032(0.062)	0.078(0.086)	0.058(0.071)	0.020(0.050)	0.064(0.074)	0.064(0.074)	0.000(0.000)
	32	0.180(0.134)	0.078(0.083)	0.102(0.110)	0.158(0.124)	0.088(0.088)	0.070(0.092)	0.135(0.120)	0.042(0.062)	0.093(0.105)
	40	0.077(0.087)	0.050(0.067)	0.027(0.057)	0.161(0.129)	0.059(0.073)	0.102(0.110)	**0.051(0.067)**	0.051(0.067)	0.000(0.000)
	48	0.215(0.143)	0.083(0.086)	0.132(0.120)	0.204(0.140)	0.089(0.088)	0.115(0.115)	0.158(0.129)	**0.038(0.058)**	0.120(0.118)
	56	0.167(0.130)	0.043(0.063)	0.124(0.117)	0.164(0.130)	0.052(0.068)	0.112(0.114)	0.058(0.073)	0.053(0.068)	0.005(0.025)
	64	0.191(0.139)	0.066(0.077)	0.125(0.119)	0.142(0.121)	0.061(0.074)	0.082(0.099)	0.183(0.135)	0.048(0.064)	0.134(0.122)
	72	0.177(0.133)	0.050(0.067)	0.127(0.118)	0.158(0.127)	0.063(0.075)	0.095(0.106)	0.063(0.078)	0.042(0.060)	0.022(0.052)
	80	0.103(0.106)	0.037(0.058)	0.067(0.090)	0.150(0.126)	0.047(0.065)	0.103(0.111)	0.128(0.116)	0.043(0.060)	0.085(0.101)
	88	0.173(0.135)	0.047(0.065)	0.127(0.122)	0.132(0.119)	0.040(0.060)	0.092(0.105)	0.156(0.127)	0.047(0.063)	0.108(0.113)
	96	0.138(0.119)	0.059(0.073)	0.078(0.097)	0.157(0.128)	0.052(0.068)	0.105(0.112)	0.159(0.128)	0.049(0.066)	0.111(0.113)
NCL Min	8	0.237(0.144)	0.237(0.144)	0.000(0.000)	0.223(0.140)	0.223(0.140)	0.000(0.000)	0.197(0.131)	0.197(0.131)	0.000(0.000)
	16	0.260(0.150)	0.260(0.150)	0.000(0.000)	0.239(0.143)	0.239(0.143)	0.000(0.000)	0.214(0.137)	0.214(0.137)	0.000(0.000)
	24	0.276(0.152)	0.276(0.152)	0.000(0.000)	0.259(0.150)	0.259(0.150)	0.000(0.000)	**0.166(0.120)**	0.166(0.120)	0.000(0.000)
	32	0.271(0.152)	0.271(0.152)	0.000(0.000)	0.255(0.148)	0.255(0.148)	0.000(0.000)	0.206(0.135)	0.206(0.135)	0.000(0.000)
	40	0.251(0.145)	0.251(0.145)	0.000(0.000)	0.268(0.151)	0.268(0.151)	0.000(0.000)	0.205(0.133)	0.205(0.133)	0.000(0.000)
	48	0.292(0.155)	0.292(0.155)	0.000(0.000)	0.273(0.154)	0.273(0.154)	0.000(0.000)	0.223(0.139)	0.223(0.139)	0.000(0.000)
	56	0.289(0.156)	0.289(0.156)	0.000(0.000)	0.266(0.151)	0.266(0.151)	0.000(0.000)	0.199(0.133)	0.199(0.133)	0.000(0.000)
	64	0.276(0.153)	0.276(0.153)	0.000(0.000)	0.308(0.160)	0.308(0.160)	0.000(0.000)	0.199(0.133)	0.199(0.133)	0.000(0.000)
	72	0.272(0.153)	0.272(0.153)	0.000(0.000)	0.281(0.154)	0.281(0.154)	0.000(0.000)	0.222(0.140)	0.222(0.140)	0.000(0.000)
	80	0.327(0.164)	0.327(0.164)	0.000(0.000)	0.268(0.152)	0.268(0.152)	0.000(0.000)	0.212(0.136)	0.212(0.136)	0.000(0.000)
	88	0.320(0.162)	0.320(0.162)	0.000(0.000)	0.287(0.157)	0.287(0.157)	0.000(0.000)	0.236(0.141)	0.236(0.141)	0.000(0.000)
	96	0.259(0.151)	0.259(0.151)	0.000(0.000)	0.248(0.147)	0.248(0.147)	0.000(0.000)	0.285(0.155)	0.285(0.155)	0.000(0.000)
NCL Product	8	0.607(0.198)	0.607(0.198)	0.000(0.000)	0.607(0.198)	0.607(0.198)	0.000(0.000)	**0.590(0.198)**	0.590(0.198)	0.000(0.000)
	16	0.744(0.210)	0.744(0.210)	0.000(0.000)	0.737(0.209)	0.737(0.209)	0.000(0.000)	0.705(0.207)	0.705(0.207)	0.000(0.000)
	24	0.770(0.211)	0.770(0.211)	0.000(0.000)	0.768(0.211)	0.768(0.211)	0.000(0.000)	0.723(0.208)	0.723(0.208)	0.000(0.000)
	32	0.796(0.212)	0.796(0.212)	0.000(0.000)	0.793(0.212)	0.793(0.212)	0.000(0.000)	0.788(0.212)	0.788(0.212)	0.000(0.000)
	40	0.813(0.213)	0.813(0.213)	0.000(0.000)	0.799(0.212)	0.799(0.212)	0.000(0.000)	0.793(0.212)	0.793(0.212)	0.000(0.000)
	48	0.823(0.213)	0.823(0.213)	0.000(0.000)	0.823(0.213)	0.823(0.213)	0.000(0.000)	0.810(0.213)	0.810(0.213)	0.000(0.000)
	56	0.817(0.213)	0.817(0.213)	0.000(0.000)	0.828(0.213)	0.828(0.213)	0.000(0.000)	0.834(0.214)	0.834(0.214)	0.000(0.000)
	64	0.837(0.214)	0.837(0.214)	0.000(0.000)	0.837(0.214)	0.837(0.214)	0.000(0.000)	0.839(0.214)	0.839(0.214)	0.000(0.000)
	72	0.842(0.214)	0.842(0.214)	0.000(0.000)	0.847(0.214)	0.847(0.214)	0.000(0.000)	0.843(0.214)	0.843(0.214)	0.000(0.000)
	80	0.866(0.214)	0.866(0.214)	0.000(0.000)	0.853(0.214)	0.853(0.214)	0.000(0.000)	0.850(0.214)	0.850(0.214)	0.000(0.000)
	88	0.866(0.214)	0.866(0.214)	0.000(0.000)	0.876(0.214)	0.876(0.214)	0.000(0.000)	0.858(0.214)	0.858(0.214)	0.000(0.000)
	96	0.866(0.214)	0.866(0.214)	0.000(0.000)	0.867(0.214)	0.867(0.214)	0.000(0.000)	0.866(0.214)	0.866(0.214)	0.000(0.000)
NCL Median	8	0.159(0.117)	0.159(0.117)	0.000(0.000)	0.145(0.111)	0.145(0.111)	0.000(0.000)	0.101(0.093)	0.101(0.093)	0.000(0.000)
	16	0.173(0.124)	0.173(0.124)	0.000(0.000)	0.155(0.116)	0.155(0.116)	0.000(0.000)	0.128(0.103)	0.128(0.103)	0.000(0.000)
	24	0.170(0.123)	0.170(0.123)	0.000(0.000)	0.158(0.117)	0.158(0.117)	0.000(0.000)	**0.083(0.086)**	0.083(0.086)	0.000(0.000)
	32	0.163(0.119)	0.163(0.119)	0.000(0.000)	0.134(0.106)	0.134(0.106)	0.000(0.000)	0.087(0.087)	0.087(0.087)	0.000(0.000)
	40	0.151(0.114)	0.151(0.114)	0.000(0.000)	0.120(0.100)	0.120(0.100)	0.000(0.000)	0.101(0.093)	0.101(0.093)	0.000(0.000)
	48	0.132(0.106)	0.132(0.106)	0.000(0.000)	0.131(0.105)	0.131(0.105)	0.000(0.000)	0.104(0.094)	0.104(0.094)	0.000(0.000)
	56	0.172(0.123)	0.172(0.123)	0.000(0.000)	0.130(0.105)	0.130(0.105)	0.000(0.000)	0.099(0.092)	0.099(0.092)	0.000(0.000)
	64	0.157(0.117)	0.157(0.117)	0.000(0.000)	0.136(0.107)	0.136(0.107)	0.000(0.000)	0.096(0.091)	0.096(0.091)	0.000(0.000)
	72	0.162(0.119)	0.162(0.119)	0.000(0.000)	0.131(0.105)	0.131(0.105)	0.000(0.000)	0.095(0.090)	0.095(0.090)	0.000(0.000)
	80	0.097(0.091)	0.097(0.091)	0.000(0.000)	0.121(0.101)	0.121(0.101)	0.000(0.000)	0.149(0.113)	0.149(0.113)	0.000(0.000)
	88	0.157(0.117)	0.157(0.117)	0.000(0.000)	0.134(0.106)	0.134(0.106)	0.000(0.000)	0.091(0.089)	0.091(0.089)	0.000(0.000)
	96	0.131(0.105)	0.131(0.105)	0.000(0.000)	0.119(0.101)	0.119(0.101)	0.000(0.000)	0.093(0.090)	0.093(0.090)	0.000(0.000)

Figure 3.4: Influence of the concentration of the analytes in the mixture over the average of the errors for the NCL method with 40 members, Max combination rule and $\lambda = 1.0$: (a) FB, (b) TBZ and (c) MBC-BM. (d) Influence of the number of classes from the mixture in the average of the errors of the mixtures. (e) ROC curve.



(a)

(b)

(c)

(d)

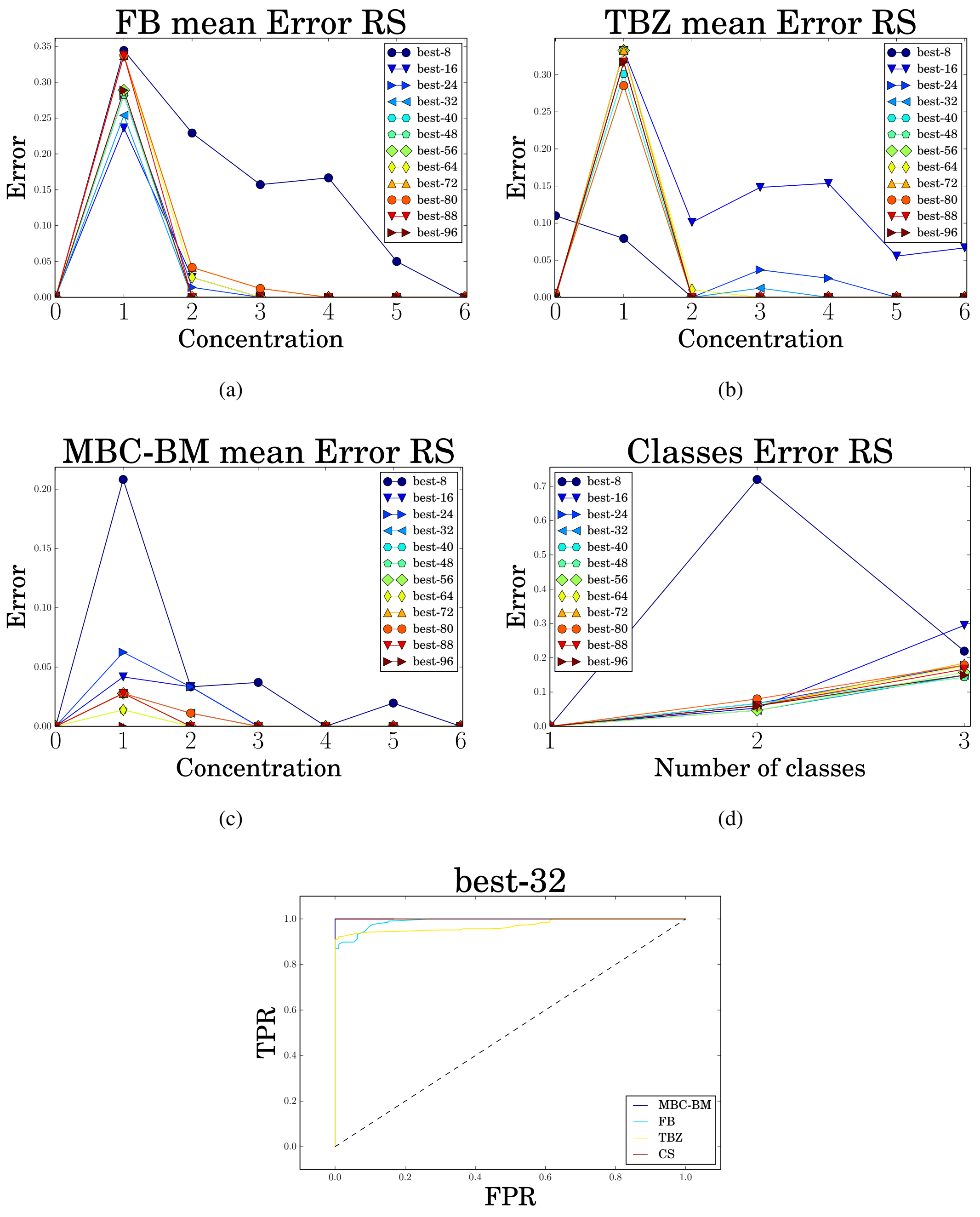
(e)

In the table 3.4 we compare different combinations rules in the random subspace method. Is is interesting to see how the mean rule with the best members achieved the best results, specifically less than the half of the SMV scheme. The median method obtain better results than SMV combining the outputs of the most diverse members with the Q diversity. The product rule produced the worst results, because of the presence of members with a value near to zero, cancelling the ensemble output. It could be an interesting rule when the reduction of FP is an objective. The min rule produces similar consequences to the product rule, increasing the FN error. The WMV scheme couldn't improve the SMV results even with the best members. As in all the other cases, we can observe in the figures 3.5 (a, b, c and d) that the main component of the errors were produced in low concentrations of the fungicides classes, specially in mixtures with more than 2 classes. In the figure 3.5(e) we showed the ROC curve of the best ensemble founded in the table 3.4, where it is possible to see that the problems are mainly produced by the TBZ and FB classes. The RS achieved good results but far from the stability founded in the NCL method results. Although the best members have provided a better result in the RS with the mean rule, we first generated diversity among members that get good use of their combination.

Table 3.4: Average mixture errors of identification values and their standard deviation between parentheses using neural ensembles with Random subspaces diversity creation methods and diversity measures with some combination rules.

		DF			Dis			p			Q			Best		
	Members	E	E_{FN}	E_{FP}	E	E_{FN}	E_{FP}	E	E_{FN}	E_{FP}	E	E_{FN}	E_{FP}	E	E_{FN}	E_{FP}
RS SMV	8	0.398(0.175)	0.398(0.175)	0.000(0.000)	0.403(0.174)	0.403(0.174)	0.000(0.000)	0.278(0.180)	0.178(0.124)	0.100(0.139)	**0.162(0.116)**	0.162(0.116)	0.000(0.000)	0.425(0.180)	0.425(0.180)	0.000(0.000)
	16	0.408(0.177)	0.408(0.177)	0.000(0.000)	0.279(0.150)	0.279(0.150)	0.000(0.000)	0.329(0.190)	0.226(0.140)	0.103(0.140)	0.232(0.142)	0.232(0.142)	0.000(0.000)	0.440(0.182)	0.440(0.182)	0.000(0.000)
	24	0.394(0.174)	0.394(0.174)	0.000(0.000)	0.266(0.148)	0.266(0.148)	0.000(0.000)	0.306(0.184)	0.233(0.142)	0.073(0.126)	0.237(0.143)	0.236(0.142)	0.002(0.014)	0.407(0.176)	0.407(0.176)	0.000(0.000)
	32	0.389(0.173)	0.389(0.173)	0.000(0.000)	0.266(0.147)	0.266(0.147)	0.000(0.000)	0.267(0.167)	0.230(0.141)	0.037(0.095)	0.229(0.140)	0.229(0.140)	0.000(0.000)	0.357(0.166)	0.357(0.166)	0.000(0.000)
	40	0.355(0.166)	0.355(0.166)	0.000(0.000)	0.279(0.149)	0.279(0.149)	0.000(0.000)	0.224(0.139)	0.224(0.139)	0.000(0.000)	0.224(0.139)	0.224(0.139)	0.000(0.000)	0.351(0.166)	0.351(0.166)	0.000(0.000)
	48	0.372(0.169)	0.372(0.169)	0.000(0.000)	0.239(0.140)	0.239(0.140)	0.000(0.000)	0.198(0.131)	0.198(0.131)	0.000(0.000)	0.211(0.134)	0.211(0.134)	0.000(0.000)	0.348(0.165)	0.348(0.165)	0.000(0.000)
	56	0.376(0.170)	0.376(0.170)	0.000(0.000)	0.236(0.140)	0.236(0.140)	0.000(0.000)	0.203(0.132)	0.203(0.132)	0.000(0.000)	0.204(0.132)	0.204(0.132)	0.000(0.000)	0.356(0.166)	0.356(0.166)	0.000(0.000)
	64	0.405(0.175)	0.405(0.175)	0.000(0.000)	0.218(0.136)	0.218(0.136)	0.000(0.000)	0.210(0.134)	0.210(0.134)	0.000(0.000)	0.203(0.132)	0.203(0.132)	0.000(0.000)	0.368(0.168)	0.368(0.168)	0.000(0.000)
	72	0.404(0.175)	0.404(0.175)	0.000(0.000)	0.238(0.140)	0.238(0.140)	0.000(0.000)	0.210(0.134)	0.210(0.134)	0.000(0.000)	0.203(0.132)	0.203(0.132)	0.000(0.000)	0.369(0.168)	0.369(0.168)	0.000(0.000)
	80	0.400(0.174)	0.400(0.174)	0.000(0.000)	0.262(0.145)	0.262(0.145)	0.000(0.000)	0.211(0.134)	0.211(0.134)	0.000(0.000)	0.212(0.134)	0.212(0.134)	0.000(0.000)	0.365(0.168)	0.365(0.168)	0.000(0.000)
	88	0.394(0.173)	0.394(0.173)	0.000(0.000)	0.255(0.143)	0.255(0.143)	0.000(0.000)	0.219(0.136)	0.219(0.136)	0.000(0.000)	0.210(0.134)	0.210(0.134)	0.000(0.000)	0.365(0.168)	0.365(0.168)	0.000(0.000)
	96	0.381(0.170)	0.381(0.170)	0.000(0.000)	0.250(0.143)	0.250(0.143)	0.000(0.000)	0.212(0.134)	0.212(0.134)	0.000(0.000)	0.213(0.134)	0.213(0.134)	0.000(0.000)	0.360(0.168)	0.360(0.168)	0.000(0.000)
RS WMV	8	**0.346(0.179)**	0.050(0.067)	0.296(0.171)	0.377(0.180)	0.107(0.095)	0.271(0.165)	0.726(0.263)	0.010(0.033)	0.716(0.263)	0.651(0.262)	0.010(0.033)	0.641(0.262)	0.383(0.185)	0.103(0.099)	0.281(0.167)
	16	0.349(0.184)	0.010(0.029)	0.339(0.183)	0.699(0.256)	0.035(0.055)	0.664(0.256)	0.789(0.272)	0.010(0.033)	0.779(0.272)	0.791(0.272)	0.010(0.033)	0.781(0.272)	0.388(0.195)	0.008(0.025)	0.381(0.194)
	24	0.593(0.221)	0.000(0.000)	0.593(0.221)	0.865(0.280)	0.000(0.000)	0.865(0.280)	0.837(0.273)	0.010(0.033)	0.827(0.273)	0.837(0.273)	0.000(0.000)	0.837(0.273)	0.387(0.196)	0.000(0.000)	0.387(0.196)
	32	0.608(0.223)	0.000(0.000)	0.608(0.223)	0.865(0.280)	0.000(0.000)	0.865(0.280)	0.837(0.273)	0.000(0.000)	0.837(0.273)	0.838(0.273)	0.000(0.000)	0.838(0.273)	0.598(0.223)	0.000(0.000)	0.598(0.223)
	40	0.608(0.223)	0.000(0.000)	0.608(0.223)	0.865(0.280)	0.000(0.000)	0.865(0.280)	0.845(0.275)	0.000(0.000)	0.845(0.275)	0.838(0.273)	0.000(0.000)	0.838(0.273)	0.620(0.227)	0.000(0.000)	0.620(0.227)
	48	0.610(0.223)	0.000(0.000)	0.610(0.223)	0.865(0.280)	0.000(0.000)	0.865(0.280)	0.865(0.280)	0.000(0.000)	0.865(0.280)	0.865(0.280)	0.000(0.000)	0.865(0.280)	0.620(0.227)	0.000(0.000)	0.620(0.227)
	56	0.640(0.226)	0.000(0.000)	0.640(0.226)	0.865(0.280)	0.000(0.000)	0.865(0.280)	0.865(0.280)	0.000(0.000)	0.865(0.280)	0.865(0.280)	0.000(0.000)	0.865(0.280)	0.620(0.227)	0.000(0.000)	0.620(0.227)
	64	0.643(0.227)	0.000(0.000)	0.643(0.227)	0.865(0.280)	0.000(0.000)	0.865(0.280)	0.865(0.280)	0.000(0.000)	0.865(0.280)	0.865(0.280)	0.000(0.000)	0.865(0.280)	0.633(0.232)	0.000(0.000)	0.633(0.232)
	72	0.661(0.230)	0.000(0.000)	0.661(0.230)	0.865(0.280)	0.000(0.000)	0.865(0.280)	0.865(0.280)	0.000(0.000)	0.865(0.280)	0.865(0.280)	0.000(0.000)	0.865(0.280)	0.640(0.233)	0.000(0.000)	0.640(0.233)
	80	0.687(0.239)	0.000(0.000)	0.687(0.239)	0.865(0.280)	0.000(0.000)	0.865(0.280)	0.865(0.280)	0.000(0.000)	0.865(0.280)	0.865(0.280)	0.000(0.000)	0.865(0.280)	0.660(0.240)	0.000(0.000)	0.660(0.240)
	88	0.692(0.240)	0.000(0.000)	0.692(0.240)	0.865(0.280)	0.000(0.000)	0.865(0.280)	0.865(0.280)	0.000(0.000)	0.865(0.280)	0.865(0.280)	0.000(0.000)	0.865(0.280)	0.661(0.241)	0.000(0.000)	0.661(0.241)
	96	0.699(0.242)	0.000(0.000)	0.699(0.242)	0.865(0.280)	0.000(0.000)	0.865(0.280)	0.865(0.280)	0.000(0.000)	0.865(0.280)	0.865(0.280)	0.000(0.000)	0.865(0.280)	0.710(0.245)	0.000(0.000)	0.710(0.245)
RS mean	8	0.243(0.154)	0.127(0.109)	0.117(0.117)	0.390(0.174)	0.259(0.144)	0.131(0.117)	0.362(0.179)	0.126(0.105)	0.235(0.157)	0.344(0.168)	0.199(0.126)	0.146(0.126)	0.263(0.158)	0.153(0.119)	0.110(0.114)
	16	0.157(0.121)	0.099(0.091)	0.058(0.084)	0.257(0.154)	0.145(0.112)	0.112(0.115)	0.598(0.217)	0.109(0.103)	0.489(0.208)	0.655(0.220)	0.167(0.125)	0.488(0.208)	0.134(0.105)	0.132(0.104)	0.002(0.014)
	24	0.090(0.088)	0.088(0.087)	0.002(0.014)	0.266(0.157)	0.111(0.098)	0.155(0.132)	0.708(0.221)	0.210(0.138)	0.498(0.207)	0.661(0.219)	0.154(0.120)	0.507(0.208)	0.085(0.085)	0.085(0.085)	0.000(0.000)
	32	0.129(0.108)	0.121(0.103)	0.008(0.032)	0.285(0.158)	0.114(0.099)	0.172(0.133)	0.607(0.215)	0.124(0.108)	0.483(0.205)	0.581(0.213)	0.092(0.093)	0.489(0.206)	**0.068(0.077)**	0.068(0.077)	0.000(0.000)
	40	0.150(0.118)	0.123(0.105)	0.027(0.057)	0.299(0.164)	0.121(0.104)	0.177(0.136)	0.577(0.212)	0.103(0.099)	0.475(0.204)	0.553(0.210)	0.095(0.095)	0.458(0.201)	0.073(0.080)	0.068(0.077)	0.005(0.025)
	48	0.153(0.118)	0.141(0.113)	0.012(0.038)	0.247(0.152)	0.118(0.103)	0.128(0.120)	0.415(0.186)	0.106(0.097)	0.309(0.172)	0.417(0.187)	0.102(0.096)	0.315(0.173)	0.070(0.078)	0.066(0.076)	0.003(0.020)
	56	0.171(0.124)	0.156(0.117)	0.015(0.043)	0.241(0.150)	0.105(0.097)	0.136(0.122)	0.461(0.196)	0.098(0.094)	0.364(0.185)	0.461(0.196)	0.098(0.095)	0.363(0.184)	0.072(0.078)	0.072(0.078)	0.000(0.000)
	64	0.191(0.132)	0.156(0.117)	0.035(0.066)	0.226(0.147)	0.098(0.094)	0.128(0.120)	0.420(0.192)	0.084(0.088)	0.336(0.181)	0.431(0.193)	0.088(0.091)	0.343(0.181)	0.082(0.084)	0.082(0.084)	0.000(0.000)
	72	0.170(0.124)	0.146(0.113)	0.023(0.054)	0.208(0.142)	0.094(0.092)	0.113(0.113)	0.392(0.188)	0.078(0.086)	0.314(0.177)	0.415(0.193)	0.079(0.087)	0.336(0.182)	0.085(0.085)	0.085(0.085)	0.000(0.000)
	80	0.153(0.118)	0.133(0.109)	0.020(0.050)	0.219(0.146)	0.084(0.089)	0.135(0.122)	0.407(0.190)	0.076(0.085)	0.331(0.179)	0.429(0.194)	0.067(0.080)	0.361(0.185)	0.088(0.088)	0.083(0.084)	0.005(0.025)
	88	0.141(0.113)	0.124(0.105)	0.017(0.045)	0.205(0.142)	0.071(0.081)	0.134(0.122)	0.397(0.190)	0.068(0.081)	0.329(0.180)	0.421(0.194)	0.075(0.085)	0.346(0.183)	0.078(0.082)	0.078(0.082)	0.000(0.000)
	96	0.130(0.108)	0.118(0.102)	0.012(0.038)	0.262(0.151)	0.084(0.086)	0.178(0.131)	0.414(0.190)	0.077(0.085)	0.337(0.179)	0.397(0.191)	0.063(0.078)	0.334(0.182)	0.072(0.079)	0.068(0.077)	0.003(0.020)
RS Max	8	0.259(0.160)	0.126(0.111)	0.133(0.125)	0.518(0.198)	0.211(0.135)	0.307(0.170)	0.531(0.209)	0.110(0.106)	0.421(0.195)	0.525(0.201)	0.189(0.130)	0.334(0.177)	0.278(0.163)	0.155(0.121)	0.123(0.120)
	16	0.212(0.144)	0.089(0.088)	0.123(0.120)	0.700(0.213)	0.388(0.177)	0.312(0.172)	0.921(0.230)	0.357(0.178)	0.565(0.216)	0.693(0.228)	0.110(0.106)	0.583(0.221)	0.182(0.133)	0.090(0.088)	0.092(0.105)
	24	0.357(0.174)	0.091(0.090)	0.266(0.159)	0.746(0.223)	0.329(0.171)	0.417(0.195)	0.697(0.222)	0.160(0.122)	0.536(0.212)	0.680(0.227)	0.074(0.089)	0.606(0.222)	**0.176(0.131)**	0.085(0.085)	0.092(0.105)
	32	0.489(0.195)	0.272(0.153)	0.217(0.148)	0.721(0.222)	0.299(0.164)	0.421(0.195)	0.679(0.222)	0.118(0.106)	0.560(0.215)	0.680(0.228)	0.050(0.074)	0.630(0.224)	0.346(0.172)	0.080(0.083)	0.266(0.159)
	40	0.482(0.194)	0.266(0.152)	0.215(0.148)	0.701(0.212)	0.391(0.176)	0.310(0.171)	0.680(0.222)	0.118(0.106)	0.561(0.216)	0.669(0.225)	0.084(0.093)	0.585(0.219)	0.356(0.175)	0.080(0.083)	0.276(0.162)
	48	0.459(0.192)	0.246(0.148)	0.214(0.147)	0.709(0.219)	0.322(0.167)	0.386(0.188)	0.665(0.225)	0.084(0.093)	0.581(0.219)	0.664(0.225)	0.083(0.093)	0.581(0.219)	0.342(0.174)	0.061(0.074)	0.281(0.164)
	56	0.532(0.208)	0.196(0.137)	0.336(0.181)	0.696(0.218)	0.310(0.164)	0.386(0.188)	0.664(0.225)	0.083(0.093)	0.581(0.219)	0.664(0.225)	0.083(0.093)	0.581(0.219)	0.342(0.174)	0.061(0.074)	0.281(0.164)
	64	0.571(0.209)	0.198(0.137)	0.373(0.185)	0.695(0.218)	0.309(0.164)	0.386(0.188)	0.658(0.225)	0.077(0.089)	0.581(0.219)	0.663(0.225)	0.081(0.092)	0.581(0.219)	0.358(0.177)	0.061(0.074)	0.296(0.168)
	72	0.573(0.210)	0.195(0.136)	0.378(0.186)	0.679(0.216)	0.292(0.159)	0.386(0.188)	0.655(0.225)	0.073(0.088)	0.581(0.219)	0.659(0.225)	0.078(0.090)	0.581(0.219)	0.375(0.181)	0.061(0.074)	0.314(0.172)
	80	0.573(0.210)	0.195(0.136)	0.378(0.186)	0.752(0.222)	0.290(0.159)	0.462(0.202)	0.649(0.225)	0.068(0.085)	0.581(0.219)	0.653(0.225)	0.072(0.087)	0.581(0.219)	0.389(0.184)	0.062(0.075)	0.327(0.176)
	88	0.563(0.208)	0.185(0.132)	0.378(0.186)	0.780(0.224)	0.279(0.157)	0.501(0.207)	0.649(0.225)	0.068(0.085)	0.581(0.219)	0.653(0.225)	0.072(0.087)	0.581(0.219)	0.394(0.185)	0.070(0.081)	0.325(0.175)
	96	0.579(0.210)	0.206(0.140)	0.373(0.185)	0.673(0.222)	0.141(0.120)	0.532(0.211)	0.651(0.226)	0.057(0.080)	0.595(0.221)	0.667(0.226)	0.106(0.106)	0.562(0.218)	0.413(0.192)	0.037(0.063)	0.376(0.187)
RS Min	8	0.659(0.206)	0.659(0.206)	0.000(0.000)	**0.548(0.194)**	0.548(0.194)	0.000(0.000)	0.809(0.213)	0.809(0.213)	0.000(0.000)	0.784(0.213)	0.784(0.213)	0.000(0.000)	0.627(0.204)	0.627(0.204)	0.000(0.000)
	16	0.673(0.206)	0.673(0.206)	0.000(0.000)	0.677(0.207)	0.677(0.207)	0.000(0.000)	0.813(0.214)	0.813(0.214)	0.000(0.000)	0.843(0.215)	0.843(0.215)	0.000(0.000)	0.678(0.207)	0.678(0.207)	0.000(0.000)
	24	0.791(0.213)	0.791(0.213)	0.000(0.000)	0.794(0.212)	0.794(0.212)	0.000(0.000)	0.855(0.215)	0.855(0.215)	0.000(0.000)	0.855(0.215)	0.855(0.215)	0.000(0.000)	0.787(0.213)	0.787(0.213)	0.000(0.000)
	32	0.805(0.213)	0.805(0.213)	0.000(0.000)	0.834(0.214)	0.834(0.214)	0.000(0.000)	0.887(0.214)	0.887(0.214)	0.000(0.000)	0.887(0.214)	0.887(0.214)	0.000(0.000)	0.796(0.213)	0.796(0.213)	0.000(0.000)
	40	0.810(0.214)	0.810(0.214)	0.000(0.000)	0.840(0.214)	0.840(0.214)	0.000(0.000)	0.893(0.215)	0.893(0.215)	0.000(0.000)	0.893(0.215)	0.893(0.215)	0.000(0.000)	0.809(0.213)	0.809(0.213)	0.000(0.000)
	48	0.815(0.214)	0.815(0.214)	0.000(0.000)	0.843(0.214)	0.843(0.214)	0.000(0.000)	0.893(0.215)	0.893(0.215)	0.000(0.000)	0.893(0.215)	0.893(0.215)	0.000(0.000)	0.809(0.213)	0.809(0.213)	0.000(0.000)
	56	0.834(0.215)	0.834(0.215)	0.000(0.000)	0.845(0.214)	0.845(0.214)	0.000(0.000)	0.893(0.215)	0.893(0.215)	0.000(0.000)	0.893(0.215)	0.893(0.215)	0.000(0.000)	0.809(0.213)	0.809(0.213)	0.000(0.000)
	64	0.839(0.215)	0.839(0.215)	0.000(0.000)	0.872(0.214)	0.872(0.214)	0.000(0.000)	0.893(0.215)	0.893(0.215)	0.000(0.000)	0.893(0.215)	0.893(0.215)	0.000(0.000)	0.809(0.213)	0.809(0.213)	0.000(0.000)
	72	0.849(0.215)	0.849(0.215)	0.000(0.000)	0.872(0.214)	0.872(0.214)	0.000(0.000)	0.893(0.215)	0.893(0.215)	0.000(0.000)	0.893(0.215)	0.893(0.215)	0.000(0.000)	0.812(0.213)	0.812(0.213)	0.000(0.000)
	80	0.849(0.215)	0.849(0.215)	0.000(0.000)	0.877(0.214)	0.877(0.214)	0.000(0.000)	0.893(0.215)	0.893(0.215)	0.000(0.000)	0.893(0.215)	0.893(0.215)	0.000(0.000)	0.812(0.213)	0.812(0.213)	0.000(0.000)
	88	0.849(0.215)	0.849(0.215)	0.000(0.000)	0.887(0.214)	0.887(0.214)	0.000(0.000)	0.893(0.215)	0.893(0.215)	0.000(0.000)	0.893(0.215)	0.893(0.215)	0.000(0.000)	0.815(0.213)	0.815(0.213)	0.000(0.000)
	96	0.852(0.215)	0.852(0.215)	0.000(0.000)	0.896(0.215)	0.896(0.215)	0.000(0.000)	0.897(0.215)	0.897(0.215)	0.000(0.000)	0.894(0.215)	0.894(0.215)	0.000(0.000)	0.818(0.214)	0.818(0.214)	0.000(0.000)
RS Product	8	0.839(0.214)	0.839(0.214)	0.000(0.000)	**0.835(0.214)**	0.835(0.214)	0.000(0.000)	0.889(0.215)	0.889(0.215)	0.000(0.000)	0.877(0.214)	0.877(0.214)	0.000(0.000)	0.848(0.214)	0.848(0.214)	0.000(0.000)
	16	0.882(0.215)	0.882(0.215)	0.000(0.000)	0.893(0.215)	0.893(0.215)	0.000(0.000)	0.899(0.215)	0.899(0.215)	0.000(0.000)	0.907(0.215)	0.907(0.215)	0.000(0.000)	0.885(0.215)	0.885(0.215)	0.000(0.000)
	24	0.913(0.215)	0.913(0.215)	0.000(0.000)	0.926(0.215)	0.926(0.215)	0.000(0.000)	0.912(0.215)	0.912(0.215)	0.000(0.000)	0.912(0.215)	0.912(0.215)	0.000(0.000)	0.913(0.215)	0.913(0.215)	0.000(0.000)
	32	0.919(0.215)	0.919(0.215)	0.000(0.000)	0.926(0.215)	0.926(0.215)	0.000(0.000)	0.931(0.216)	0.931(0.216)	0.000(0.000)	0.931(0.216)	0.931(0.216)	0.000(0.000)	0.916(0.215)	0.916(0.215)	0.000(0.000)
	40	0.926(0.215)	0.926(0.215)	0.000(0.000)	0.926(0.215)	0.926(0.215)	0.000(0.000)	0.931(0.216)	0.931(0.216)	0.000(0.000)	0.931(0.216)	0.931(0.216)	0.000(0.000)	0.921(0.215)	0.921(0.215)	0.000(0.000)
	48	0.931(0.216)	0.931(0.216)	0.000(0.000)	0.926(0.215)	0.926(0.215)	0.000(0.000)	0.931(0.216)	0.931(0.216)	0.000(0.000)	0.931(0.216)	0.931(0.216)	0.000(0.000)	0.926(0.215)	0.926(0.215)	0.000(0.000)
	56	0.931(0.216)	0.931(0.216)	0.000(0.000)	0.929(0.215)	0.929(0.215)	0.000(0.000)	0.931(0.216)	0.931(0.216)	0.000(0.000)	0.931(0.216)	0.931(0.216)	0.000(0.000)	0.928(0.215)	0.928(0.215)	0.000(0.000)
	64	0.931(0.216)	0.931(0.216)	0.000(0.000)	0.931(0.216)	0.931(0.216)	0.000(0.000)	0.931(0.216)	0.931(0.216)	0.000(0.000)	0.931(0.216)	0.931(0.216)	0.000(0.000)	0.931(0.216)	0.931(0.216)	0.000(0.000)
	72	0.931(0.216)	0.931(0.216)	0.000(0.000)	0.931(0.216)	0.931(0.216)	0.000(0.000)	0.931(0.216)	0.931(0.216)	0.000(0.000)	0.931(0.216)	0.931(0.216)	0.000(0.000)	0.931(0.216)	0.931(0.216)	0.000(0.000)
	80	0.931(0.216)	0.931(0.216)	0.000(0.000)	0.931(0.216)	0.931(0.216)	0.000(0.000)	0.931(0.216)	0.931(0.216)	0.000(0.000)	0.931(0.216)	0.931(0.216)	0.000(0.000)	0.931(0.216)	0.931(0.216)	0.000(0.000)
	88	0.931(0.216)	0.931(0.216)	0.000(0.000)	0.931(0.216)	0.931(0.216)	0.000(0.000)	0.931(0.216)	0.931(0.216)	0.000(0.000)	0.931(0.216)	0.931(0.216)	0.000(0.000)	0.931(0.216)	0.931(0.216)	0.000(0.000)
	96	0.931(0.216)	0.931(0.216)	0.000(0.000)	0.931(0.216)	0.931(0.216)	0.000(0.000)	0.931(0.216)	0.931(0.216)	0.000(0.000)	0.931(0.216)	0.931(0.216)	0.000(0.000)	0.931(0.216)	0.931(0.216)	0.000(0.000)
RS Median	8	0.386(0.173)	0.386(0.173)	0.000(0.000)	0.359(0.165)	0.359(0.165)	0.000(0.000)	0.132(0.106)	0.132(0.106)	0.000(0.000)	0.184(0.126)	0.184(0.126)	0.000(0.000)	0.411(0.176)	0.411(0.176)	0.000(0.000)
	16	0.375(0.172)	0.375(0.172)	0.000(0.000)	0.192(0.129)	0.192(0.129)	0.000(0.000)	0.202(0.135)	0.202(0.135)	0.000(0.000)	0.254(0.148)	0.254(0.148)	0.000(0.000)	0.399(0.176)	0.399(0.176)	0.000(0.000)
	24	0.382(0.173)	0.382(0.173)	0.000(0.000)	0.188(0.128)	0.188(0.128)	0.000(0.000)	0.203(0.134)	0.203(0.134)	0.000(0.000)	0.171(0.124)	0.171(0.124)	0.000(0.000)	0.386(0.173)	0.386(0.173)	0.000(0.000)
	32	0.315(0.160)	0.315(0.160)	0.000(0.000)	0.164(0.115)	0.164(0.115)	0.000(0.000)	0.160(0.119)	0.160(0.119)	0.000(0.000)	0.156(0.118)	0.156(0.118)	0.000(0.000)	0.351(0.167)	0.351(0.167)	0.000(0.000)
	40	0.275(0.151)	0.275(0.151)	0.000(0.000)	0.189(0.123)	0.189(0.123)	0.000(0.000)	0.145(0.114)	0.145(0.114)	0.000(0.000)	0.154(0.117)	0.154(0.117)	0.000(0.000)	0.311(0.159)	0.311(0.159)	0.000(0.000)
	48	0.285(0.153)	0.285(0.153)	0.000(0.000)	0.140(0.109)	0.140(0.109)	0.000(0.000)	0.130(0.106)	0.130(0.106)	0.000(0.000)	0.145(0.114)	0.145(0.114)	0.000(0.000)	0.300(0.156)	0.300(0.156)	0.000(0.000)
	56	0.284(0.153)	0.284(0.153)	0.000(0.000)	0.138(0.109)	0.138(0.109)	0.000(0.000)	0.142(0.110)	0.142(0.110)	0.000(0.000)	**0.129(0.106)**	0.129(0.106)	0.000(0.000)	0.316(0.160)	0.316(0.160)	0.000(0.000)
	64	0.297(0.156)	0.297(0.156)	0.000(0.000)	0.137(0.108)	0.137(0.108)	0.000(0.000)	0.147(0.112)	0.147(0.112)	0.000(0.000)	0.131(0.106)	0.131(0.106)	0.000(0.000)	0.320(0.161)	0.320(0.161)	0.000(0.000)
	72	0.298(0.155)	0.298(0.155)	0.000(0.000)	0.151(0.113)	0.151(0.113)	0.000(0.000)	0.143(0.111)	0.143(0.111)	0.000(0.000)	0.129(0.106)	0.129(0.106)	0.000(0.000)	0.328(0.163)	0.328(0.163)	0.000(0.000)
	80	0.291(0.154)	0.291(0.154)	0.000(0.000)	0.178(0.123)	0.178(0.123)	0.000(0.000)	0.144(0.111)	0.144(0.111)	0.000(0.000)	0.135(0.108)	0.135(0.108)	0.000(0.000)	0.310(0.159)	0.310(0.159)	0.000(0.000)
	88	0.301(0.156)	0.301(0.156)	0.000(0.000)	0.179(0.123)	0.179(0.123)	0.000(0.000)	0.151(0.113)	0.151(0.113)	0.000(0.000)	0.145(0.112)	0.145(0.112)	0.000(0.000)	0.321(0.161)	0.321(0.161)	0.000(0.000)
	96	0.285(0.152)	0.285(0.152)	0.000(0.000)	0.176(0.122)	0.176(0.122)	0.000(0.000)	0.151(0.113)	0.151(0.113)	0.000(0.000)	0.148(0.113)	0.148(0.113)	0.000(0.000)	0.302(0.156)	0.302(0.156)	0.000(0.000)

Figure 3.5: Influence of the concentration of the analytes in the mixture over the average of the errors for the RS method with 32 members and mean combination rule: (a) FB, (b) TBZ and (c) MBC-BM. (d) Influence of the number of classes from the mixture in the average of the errors of the mixtures. (e) ROC curve.



(a)

(b)

(c)

(d)

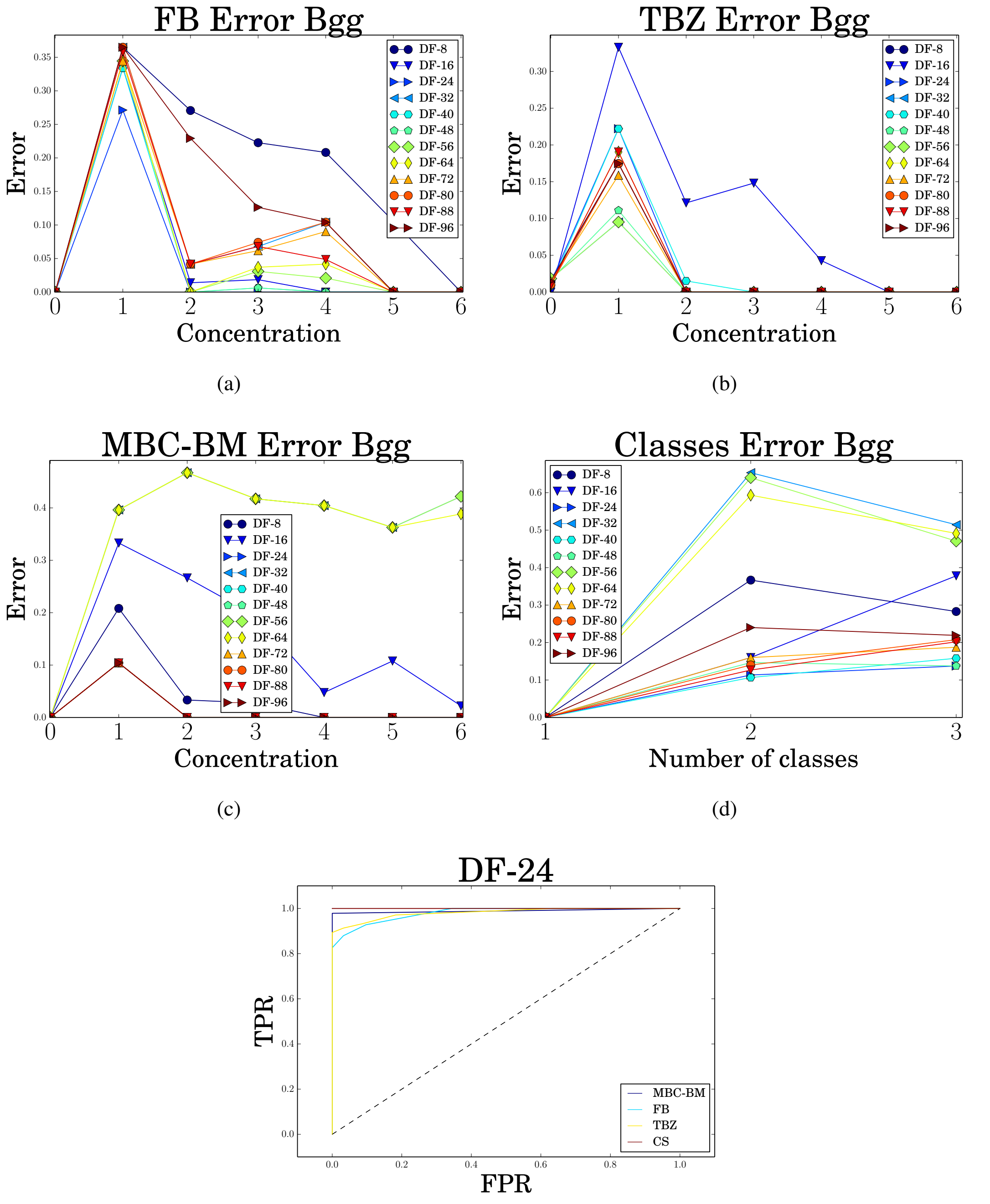
(e)

In the table 3.5 we present a comparison for the Bgg method of different combination rules for ensembles whose members was selected with different diversity measures. We also showed the influence of the number of members. In this method, the DF measure obtained the best results for three different combination rules, the SMV, the Mean and the Median. The ensemble with 24 members selected by the DF diversity measure and combined by a Median rule achieved the best results with an error of $E = 0.081$. This method allowed us to find combinations of members that are able to reduce the EFN to 0.0. The combination rules that able that possibility are the WMV and the Max rules, specially the WMV, because it reduces also the EFP. The figures 3.6 show the influence of the concentration in the error, the influence of the number of classes in the mixture and ROC curve of the best ensemble generated in the table 3.5. Again, the error is mainly produced by low concentrations of the different classes, specially FB and TBZ, with an increase of the error in mixtures of three classes.

Table 3.5: Average mixture errors of identification values and their standard deviation between parentheses using neural ensembles with Bagging diversity creation method and diversity measures with some combination rules.

		DF			Dis			p			Q			Best		
	Members	E	E_{FN}	E_{FP}	E	E_{FN}	E_{FP}	E	E_{FN}	E_{FP}	E	E_{FN}	E_{FP}	E	E_{FN}	E_{FP}
Bgg SMV	8	0.297(0.158)	0.267(0.149)	0.030(0.061)	0.308(0.159)	0.278(0.151)	0.030(0.061)	0.303(0.159)	0.273(0.150)	0.030(0.061)	0.296(0.157)	0.266(0.149)	0.030(0.061)	0.279(0.153)	0.249(0.144)	0.030(0.061)
	16	0.270(0.151)	0.240(0.141)	0.030(0.061)	0.261(0.150)	0.231(0.140)	0.030(0.061)	0.267(0.150)	0.237(0.141)	0.030(0.061)	0.298(0.158)	0.268(0.149)	0.030(0.061)	0.260(0.149)	0.230(0.139)	0.030(0.061)
	24	0.273(0.152)	0.243(0.143)	0.030(0.061)	0.268(0.152)	0.238(0.142)	0.030(0.061)	0.288(0.155)	0.258(0.146)	0.030(0.061)	0.280(0.153)	0.250(0.144)	0.030(0.061)	0.280(0.153)	0.250(0.144)	0.030(0.061)
	32	0.286(0.154)	0.256(0.145)	0.030(0.061)	0.292(0.156)	0.262(0.147)	0.030(0.061)	0.263(0.149)	0.233(0.139)	0.030(0.061)	0.266(0.149)	0.236(0.140)	0.030(0.061)	0.286(0.154)	0.256(0.145)	0.030(0.061)
	40	0.273(0.153)	0.243(0.143)	0.030(0.061)	0.307(0.160)	0.277(0.151)	0.030(0.061)	0.264(0.149)	0.234(0.139)	0.030(0.061)	0.262(0.148)	0.232(0.139)	0.030(0.061)	0.287(0.155)	0.257(0.146)	0.030(0.061)
	48	**0.257(0.149)**	0.227(0.139)	0.030(0.061)	0.292(0.156)	0.262(0.147)	0.030(0.061)	0.261(0.148)	0.231(0.138)	0.030(0.061)	0.261(0.148)	0.231(0.138)	0.030(0.061)	0.283(0.154)	0.253(0.145)	0.030(0.061)
	56	0.263(0.150)	0.233(0.141)	0.030(0.061)	0.300(0.158)	0.270(0.149)	0.030(0.061)	0.268(0.150)	0.238(0.141)	0.030(0.061)	0.259(0.148)	0.229(0.138)	0.030(0.061)	0.280(0.153)	0.250(0.144)	0.030(0.061)
	64	0.280(0.154)	0.250(0.144)	0.030(0.061)	0.303(0.158)	0.273(0.150)	0.030(0.061)	0.256(0.147)	0.226(0.137)	0.030(0.061)	0.258(0.147)	0.228(0.137)	0.030(0.061)	0.279(0.153)	0.249(0.144)	0.030(0.061)
	72	0.286(0.155)	0.256(0.146)	0.030(0.061)	0.307(0.159)	0.277(0.151)	0.030(0.061)	0.258(0.147)	0.228(0.137)	0.030(0.061)	0.254(0.146)	0.224(0.136)	0.030(0.061)	0.279(0.153)	0.249(0.144)	0.030(0.061)
	80	0.281(0.154)	0.251(0.145)	0.030(0.061)	0.304(0.159)	0.274(0.150)	0.030(0.061)	0.261(0.148)	0.231(0.138)	0.030(0.061)	0.254(0.146)	0.224(0.136)	0.030(0.061)	0.279(0.153)	0.249(0.144)	0.030(0.061)
	88	0.278(0.153)	0.248(0.144)	0.030(0.061)	0.310(0.160)	0.280(0.152)	0.030(0.061)	0.261(0.148)	0.231(0.138)	0.030(0.061)	0.261(0.148)	0.231(0.138)	0.030(0.061)	0.278(0.153)	0.248(0.143)	0.030(0.061)
	96	0.279(0.153)	0.249(0.144)	0.030(0.061)	0.303(0.158)	0.273(0.150)	0.030(0.061)	0.264(0.148)	0.234(0.139)	0.030(0.061)	0.265(0.149)	0.235(0.139)	0.030(0.061)	0.276(0.152)	0.246(0.143)	0.030(0.061)
Bgg WMV	8	0.153(0.125)	0.093(0.093)	0.060(0.088)	0.155(0.126)	0.058(0.072)	0.097(0.107)	**0.112(0.108)**	0.057(0.070)	0.055(0.084)	0.166(0.136)	0.058(0.071)	0.108(0.120)	0.185(0.144)	0.064(0.076)	0.122(0.126)
	16	0.132(0.129)	0.010(0.029)	0.122(0.126)	0.161(0.131)	0.035(0.056)	0.126(0.121)	0.278(0.188)	0.018(0.038)	0.261(0.186)	0.278(0.188)	0.018(0.038)	0.261(0.186)	0.187(0.145)	0.064(0.076)	0.123(0.127)
	24	**0.132(0.131)**	**0.000(0.000)**	**0.132(0.131)**	0.181(0.139)	0.034(0.056)	0.147(0.130)	0.283(0.189)	0.018(0.038)	0.266(0.187)	0.290(0.193)	0.018(0.038)	0.272(0.191)	0.137(0.133)	0.000(0.000)	0.137(0.133)
	32	0.207(0.158)	0.000(0.000)	0.207(0.158)	0.181(0.139)	0.034(0.056)	0.147(0.130)	0.441(0.220)	0.012(0.032)	0.429(0.220)	0.433(0.215)	0.013(0.033)	0.420(0.214)	0.137(0.133)	0.000(0.000)	0.137(0.133)
	40	0.207(0.158)	0.000(0.000)	0.207(0.158)	0.186(0.141)	0.034(0.056)	0.152(0.132)	0.580(0.248)	0.012(0.032)	0.568(0.247)	0.580(0.248)	0.012(0.032)	0.568(0.247)	0.140(0.134)	0.000(0.000)	0.140(0.134)
	48	0.213(0.160)	0.000(0.000)	0.213(0.160)	0.222(0.156)	0.032(0.054)	0.191(0.149)	0.599(0.249)	0.012(0.032)	0.586(0.249)	0.599(0.249)	0.012(0.032)	0.586(0.249)	0.140(0.134)	0.000(0.000)	0.140(0.134)
	56	0.242(0.168)	0.000(0.000)	0.242(0.168)	0.219(0.156)	0.026(0.050)	0.192(0.150)	0.686(0.256)	0.012(0.032)	0.674(0.256)	0.652(0.254)	0.010(0.029)	0.642(0.254)	0.140(0.134)	0.000(0.000)	0.140(0.134)
	64	0.250(0.170)	0.000(0.000)	0.250(0.170)	0.198(0.152)	0.001(0.010)	0.197(0.152)	0.707(0.258)	0.010(0.029)	0.697(0.258)	0.675(0.255)	0.010(0.029)	0.665(0.256)	0.140(0.134)	0.000(0.000)	0.140(0.134)
	72	0.250(0.170)	0.000(0.000)	0.250(0.170)	0.198(0.152)	0.001(0.010)	0.197(0.152)	0.707(0.258)	0.010(0.029)	0.697(0.258)	0.707(0.258)	0.010(0.029)	0.697(0.258)	0.232(0.166)	0.000(0.000)	0.232(0.166)
	80	0.275(0.177)	0.000(0.000)	0.275(0.177)	0.247(0.167)	0.000(0.000)	0.247(0.167)	0.715(0.259)	0.008(0.025)	0.707(0.259)	0.733(0.260)	0.001(0.010)	0.732(0.260)	0.240(0.169)	0.000(0.000)	0.240(0.169)
	88	0.275(0.177)	0.000(0.000)	0.275(0.177)	0.266(0.172)	0.000(0.000)	0.266(0.172)	0.758(0.262)	0.000(0.000)	0.758(0.262)	0.758(0.262)	0.000(0.000)	0.758(0.262)	0.242(0.169)	0.000(0.000)	0.242(0.169)
	96	0.277(0.177)	0.000(0.000)	0.277(0.177)	0.267(0.172)	0.000(0.000)	0.267(0.172)	0.762(0.262)	0.000(0.000)	0.762(0.262)	0.762(0.262)	0.000(0.000)	0.762(0.262)	0.311(0.186)	0.000(0.000)	0.311(0.186)
Bgg Mean	8	**0.174(0.128)**	0.131(0.108)	0.043(0.073)	0.498(0.190)	0.498(0.190)	0.000(0.000)	0.526(0.200)	0.507(0.197)	0.018(0.048)	0.580(0.204)	0.568(0.202)	0.012(0.038)	0.284(0.161)	0.209(0.137)	0.075(0.095)
	16	0.528(0.191)	0.528(0.191)	0.000(0.000)	0.422(0.177)	0.422(0.177)	0.000(0.000)	0.782(0.212)	0.782(0.212)	0.000(0.000)	0.780(0.212)	0.780(0.212)	0.000(0.000)	0.269(0.153)	0.239(0.144)	0.030(0.061)
	24	0.386(0.172)	0.356(0.165)	0.030(0.061)	0.485(0.188)	0.485(0.188)	0.000(0.000)	0.815(0.213)	0.815(0.213)	0.000(0.000)	0.811(0.214)	0.811(0.214)	0.000(0.000)	0.264(0.153)	0.230(0.142)	0.033(0.064)
	32	0.495(0.190)	0.495(0.190)	0.000(0.000)	0.460(0.185)	0.460(0.185)	0.000(0.000)	0.784(0.212)	0.784(0.212)	0.000(0.000)	0.786(0.212)	0.786(0.212)	0.000(0.000)	0.273(0.154)	0.243(0.145)	0.030(0.061)
	40	0.509(0.193)	0.509(0.193)	0.000(0.000)	0.385(0.172)	0.385(0.172)	0.000(0.000)	0.746(0.211)	0.746(0.211)	0.000(0.000)	0.743(0.210)	0.743(0.210)	0.000(0.000)	0.308(0.160)	0.278(0.152)	0.030(0.061)
	48	0.524(0.193)	0.524(0.193)	0.000(0.000)	0.441(0.181)	0.441(0.181)	0.000(0.000)	0.504(0.190)	0.504(0.190)	0.000(0.000)	0.504(0.190)	0.504(0.190)	0.000(0.000)	0.271(0.154)	0.241(0.145)	0.030(0.061)
	56	0.510(0.190)	0.510(0.190)	0.000(0.000)	0.471(0.185)	0.471(0.185)	0.000(0.000)	0.489(0.188)	0.489(0.188)	0.000(0.000)	0.489(0.188)	0.489(0.188)	0.000(0.000)	0.263(0.152)	0.233(0.143)	0.030(0.061)
	64	0.496(0.189)	0.496(0.189)	0.000(0.000)	0.527(0.193)	0.527(0.193)	0.000(0.000)	0.501(0.190)	0.501(0.190)	0.000(0.000)	0.497(0.189)	0.497(0.189)	0.000(0.000)	0.286(0.156)	0.256(0.147)	0.030(0.061)
	72	0.309(0.160)	0.279(0.152)	0.030(0.061)	0.395(0.184)	0.319(0.165)	0.077(0.098)	0.502(0.190)	0.502(0.190)	0.000(0.000)	0.486(0.187)	0.486(0.187)	0.000(0.000)	0.285(0.156)	0.255(0.147)	0.030(0.061)
	80	0.319(0.162)	0.289(0.154)	0.030(0.061)	0.360(0.175)	0.322(0.165)	0.038(0.069)	0.464(0.184)	0.464(0.184)	0.000(0.000)	0.476(0.186)	0.476(0.186)	0.000(0.000)	0.290(0.157)	0.260(0.148)	0.030(0.061)
	88	0.336(0.165)	0.306(0.157)	0.030(0.061)	0.361(0.173)	0.346(0.169)	0.015(0.043)	0.473(0.185)	0.473(0.185)	0.000(0.000)	0.465(0.183)	0.465(0.183)	0.000(0.000)	0.306(0.160)	0.276(0.151)	0.030(0.061)
	96	0.324(0.163)	0.294(0.154)	0.030(0.061)	0.377(0.174)	0.377(0.174)	0.000(0.000)	0.465(0.183)	0.465(0.183)	0.000(0.000)	0.465(0.183)	0.465(0.183)	0.000(0.000)	0.310(0.161)	0.280(0.152)	0.030(0.061)
Bgg Max	8	0.361(0.186)	0.121(0.109)	0.240(0.162)	0.655(0.207)	0.641(0.206)	0.013(0.041)	0.326(0.170)	0.280(0.158)	0.047(0.075)	**0.318(0.167)**	0.284(0.158)	0.033(0.064)	0.678(0.242)	0.368(0.180)	0.310(0.201)
	16	0.366(0.200)	0.151(0.112)	0.215(0.177)	0.398(0.214)	0.101(0.101)	0.297(0.199)	0.574(0.202)	0.544(0.198)	0.030(0.061)	0.574(0.202)	0.544(0.198)	0.030(0.061)	0.677(0.232)	0.492(0.192)	0.185(0.169)
	24	0.400(0.235)	0.000(0.000)	0.400(0.235)	0.604(0.236)	0.307(0.167)	0.297(0.199)	0.574(0.202)	0.544(0.198)	0.030(0.061)	0.574(0.202)	0.544(0.198)	0.030(0.061)	0.366(0.200)	0.151(0.112)	0.215(0.177)
	32	0.400(0.235)	0.000(0.000)	0.400(0.235)	0.665(0.215)	0.553(0.201)	0.112(0.117)	0.400(0.235)	0.000(0.000)	0.400(0.235)	0.400(0.235)	0.000(0.000)	0.400(0.235)	0.366(0.200)	0.151(0.112)	0.215(0.177)
	40	0.400(0.235)	0.000(0.000)	0.400(0.235)	0.681(0.217)	0.553(0.201)	0.128(0.124)	0.400(0.235)	0.000(0.000)	0.400(0.235)	0.400(0.235)	0.000(0.000)	0.400(0.235)	0.400(0.235)	0.000(0.000)	0.400(0.235)
	48	0.400(0.235)	0.000(0.000)	0.400(0.235)	0.666(0.215)	0.538(0.199)	0.128(0.124)	0.400(0.235)	0.000(0.000)	0.400(0.235)	0.400(0.235)	0.000(0.000)	0.400(0.235)	0.400(0.235)	0.000(0.000)	0.400(0.235)
	56	0.400(0.235)	0.000(0.000)	0.400(0.235)	0.670(0.217)	0.538(0.199)	0.132(0.127)	0.400(0.235)	0.000(0.000)	0.400(0.235)	0.400(0.235)	0.000(0.000)	0.400(0.235)	0.400(0.235)	0.000(0.000)	0.400(0.235)
	64	0.400(0.235)	0.000(0.000)	0.400(0.235)	0.663(0.215)	0.538(0.199)	0.125(0.123)	0.400(0.235)	0.000(0.000)	0.400(0.235)	0.400(0.235)	0.000(0.000)	0.400(0.235)	0.400(0.235)	0.000(0.000)	0.400(0.235)
	72	0.400(0.235)	0.000(0.000)	0.400(0.235)	0.663(0.215)	0.538(0.199)	0.125(0.123)	0.400(0.235)	0.000(0.000)	0.400(0.235)	0.400(0.235)	0.000(0.000)	0.400(0.235)	0.400(0.235)	0.000(0.000)	0.400(0.235)
	80	0.400(0.235)	0.000(0.000)	0.400(0.235)	0.654(0.231)	0.469(0.189)	0.185(0.169)	0.400(0.235)	0.000(0.000)	0.400(0.235)	0.400(0.235)	0.000(0.000)	0.400(0.235)	0.400(0.235)	0.000(0.000)	0.400(0.235)
	88	0.400(0.235)	0.000(0.000)	0.400(0.235)	0.654(0.231)	0.469(0.189)	0.185(0.169)	0.400(0.235)	0.000(0.000)	0.400(0.235)	0.400(0.235)	0.000(0.000)	0.400(0.235)	0.400(0.235)	0.000(0.000)	0.400(0.235)
	96	0.400(0.235)	0.000(0.000)	0.400(0.235)	0.654(0.231)	0.469(0.189)	0.185(0.169)	0.400(0.235)	0.000(0.000)	0.400(0.235)	0.400(0.235)	0.000(0.000)	0.400(0.235)	0.400(0.235)	0.000(0.000)	0.400(0.235)
Bgg Min	8	0.418(0.184)	0.394(0.179)	0.023(0.054)	**0.298(0.159)**	0.293(0.157)	0.005(0.025)	0.518(0.194)	0.513(0.193)	0.005(0.025)	0.484(0.191)	0.479(0.190)	0.005(0.025)	0.336(0.166)	0.325(0.163)	0.012(0.038)
	16	0.440(0.181)	0.440(0.181)	0.000(0.000)	0.519(0.189)	0.519(0.189)	0.000(0.000)	0.583(0.197)	0.583(0.197)	0.000(0.000)	0.583(0.197)	0.583(0.197)	0.000(0.000)	0.465(0.185)	0.465(0.185)	0.000(0.000)
	24	0.529(0.191)	0.529(0.191)	0.000(0.000)	0.485(0.185)	0.485(0.185)	0.000(0.000)	0.634(0.203)	0.634(0.203)	0.000(0.000)	0.632(0.202)	0.632(0.202)	0.000(0.000)	0.465(0.185)	0.465(0.185)	0.000(0.000)
	32	0.582(0.197)	0.582(0.197)	0.000(0.000)	0.485(0.185)	0.485(0.185)	0.000(0.000)	0.674(0.206)	0.674(0.206)	0.000(0.000)	0.634(0.203)	0.634(0.203)	0.000(0.000)	0.573(0.197)	0.573(0.197)	0.000(0.000)
	40	0.601(0.199)	0.601(0.199)	0.000(0.000)	0.485(0.185)	0.485(0.185)	0.000(0.000)	0.674(0.206)	0.674(0.206)	0.000(0.000)	0.672(0.205)	0.672(0.205)	0.000(0.000)	0.576(0.197)	0.576(0.197)	0.000(0.000)
	48	0.608(0.199)	0.608(0.199)	0.000(0.000)	0.529(0.191)	0.529(0.191)	0.000(0.000)	0.686(0.206)	0.686(0.206)	0.000(0.000)	0.686(0.206)	0.686(0.206)	0.000(0.000)	0.594(0.199)	0.594(0.199)	0.000(0.000)
	56	0.616(0.200)	0.616(0.200)	0.000(0.000)	0.529(0.191)	0.529(0.191)	0.000(0.000)	0.710(0.208)	0.710(0.208)	0.000(0.000)	0.706(0.208)	0.706(0.208)	0.000(0.000)	0.594(0.199)	0.594(0.199)	0.000(0.000)
	64	0.616(0.200)	0.616(0.200)	0.000(0.000)	0.529(0.191)	0.529(0.191)	0.000(0.000)	0.710(0.208)	0.710(0.208)	0.000(0.000)	0.706(0.208)	0.706(0.208)	0.000(0.000)	0.594(0.199)	0.594(0.199)	0.000(0.000)
	72	0.616(0.200)	0.616(0.200)	0.000(0.000)	0.574(0.196)	0.574(0.196)	0.000(0.000)	0.710(0.208)	0.710(0.208)	0.000(0.000)	0.710(0.208)	0.710(0.208)	0.000(0.000)	0.599(0.199)	0.599(0.199)	0.000(0.000)
	80	0.616(0.200)	0.616(0.200)	0.000(0.000)	0.640(0.203)	0.640(0.203)	0.000(0.000)	0.710(0.208)	0.710(0.208)	0.000(0.000)	0.710(0.208)	0.710(0.208)	0.000(0.000)	0.606(0.199)	0.606(0.199)	0.000(0.000)
	88	0.616(0.200)	0.616(0.200)	0.000(0.000)	0.640(0.203)	0.640(0.203)	0.000(0.000)	0.710(0.208)	0.710(0.208)	0.000(0.000)	0.710(0.208)	0.710(0.208)	0.000(0.000)	0.606(0.199)	0.606(0.199)	0.000(0.000)
	96	0.616(0.200)	0.616(0.200)	0.000(0.000)	0.640(0.203)	0.640(0.203)	0.000(0.000)	0.710(0.208)	0.710(0.208)	0.000(0.000)	0.710(0.208)	0.710(0.208)	0.000(0.000)	0.611(0.199)	0.611(0.199)	0.000(0.000)
Bgg Product	8	0.895(0.215)	0.895(0.215)	0.000(0.000)	**0.875(0.214)**	0.875(0.214)	0.000(0.000)	0.881(0.214)	0.881(0.214)	0.000(0.000)	0.881(0.214)	0.881(0.214)	0.000(0.000)	0.892(0.215)	0.892(0.215)	0.000(0.000)
	16	0.926(0.215)	0.926(0.215)	0.000(0.000)	0.918(0.216)	0.918(0.216)	0.000(0.000)	0.921(0.215)	0.921(0.215)	0.000(0.000)	0.921(0.215)	0.921(0.215)	0.000(0.000)	0.926(0.215)	0.926(0.215)	0.000(0.000)
	24	0.926(0.215)	0.926(0.215)	0.000(0.000)	0.931(0.216)	0.931(0.216)	0.000(0.000)	0.931(0.216)	0.931(0.216)	0.000(0.000)	0.931(0.216)	0.931(0.216)	0.000(0.000)	0.926(0.215)	0.926(0.215)	0.000(0.000)
	32	0.931(0.216)	0.931(0.216)	0.000(0.000)	0.931(0.216)	0.931(0.216)	0.000(0.000)	0.987(0.233)	0.987(0.233)	0.000(0.000)	0.987(0.233)	0.987(0.233)	0.000(0.000)	0.931(0.216)	0.931(0.216)	0.000(0.000)
	40	0.931(0.216)	0.931(0.216)	0.000(0.000)	0.987(0.233)	0.987(0.233)	0.000(0.000)	0.987(0.233)	0.987(0.233)	0.000(0.000)	0.987(0.233)	0.987(0.233)	0.000(0.000)	0.931(0.216)	0.931(0.216)	0.000(0.000)
	48	0.931(0.216)	0.931(0.216)	0.000(0.000)	0.987(0.233)	0.987(0.233)	0.000(0.000)	0.987(0.233)	0.987(0.233)	0.000(0.000)	0.987(0.233)	0.987(0.233)	0.000(0.000)	0.940(0.219)	0.940(0.219)	0.000(0.000)
	56	0.949(0.222)	0.949(0.222)	0.000(0.000)	0.987(0.233)	0.987(0.233)	0.000(0.000)	0.987(0.233)	0.987(0.233)	0.000(0.000)	0.987(0.233)	0.987(0.233)	0.000(0.000)	0.987(0.233)	0.987(0.233)	0.000(0.000)
	64	0.987(0.233)	0.987(0.233)	0.000(0.000)	0.987(0.233)	0.987(0.233)	0.000(0.000)	0.987(0.233)	0.987(0.233)	0.000(0.000)	0.987(0.233)	0.987(0.233)	0.000(0.000)	0.987(0.233)	0.987(0.233)	0.000(0.000)
	72	0.987(0.233)	0.987(0.233)	0.000(0.000)	0.987(0.233)	0.987(0.233)	0.000(0.000)	0.987(0.233)	0.987(0.233)	0.000(0.000)	0.987(0.233)	0.987(0.233)	0.000(0.000)	0.987(0.233)	0.987(0.233)	0.000(0.000)
	80	0.987(0.233)	0.987(0.233)	0.000(0.000)	0.987(0.233)	0.987(0.233)	0.000(0.000)	0.987(0.233)	0.987(0.233)	0.000(0.000)	0.987(0.233)	0.987(0.233)	0.000(0.000)	0.987(0.233)	0.987(0.233)	0.000(0.000)
	88	0.987(0.233)	0.987(0.233)	0.000(0.000)	0.987(0.233)	0.987(0.233)	0.000(0.000)	0.987(0.233)	0.987(0.233)	0.000(0.000)	0.987(0.233)	0.987(0.233)	0.000(0.000)	0.987(0.233)	0.987(0.233)	0.000(0.000)
	96	0.987(0.233)	0.987(0.233)	0.000(0.000)	0.987(0.233)	0.987(0.233)	0.000(0.000)	0.987(0.233)	0.987(0.233)	0.000(0.000)	0.987(0.233)	0.987(0.233)	0.000(0.000)	0.987(0.233)	0.987(0.233)	0.000(0.000)
Bgg Median	8	0.200(0.134)	0.184(0.128)	0.017(0.045)	0.277(0.151)	0.277(0.151)	0.000(0.000)	0.467(0.188)	0.462(0.188)	0.005(0.025)	0.822(0.215)	0.822(0.215)	0.000(0.000)	0.178(0.132)	0.115(0.103)	0.063(0.088)
	16	0.218(0.135)	0.218(0.135)	0.000(0.000)	0.252(0.144)	0.252(0.144)	0.000(0.000)	0.791(0.215)	0.791(0.215)	0.000(0.000)	0.791(0.215)	0.791(0.215)	0.000(0.000)	0.150(0.118)	0.135(0.111)	0.015(0.043)
	24	**0.081(0.087)**	0.067(0.077)	0.013(0.041)	0.254(0.143)	0.254(0.143)	0.000(0.000)	0.404(0.179)	0.399(0.178)	0.005(0.025)	0.402(0.179)	0.397(0.178)	0.005(0.025)	0.357(0.169)	0.357(0.169)	0.000(0.000)
	32	0.475(0.193)	0.456(0.190)	0.018(0.048)	0.267(0.148)	0.267(0.148)	0.000(0.000)	0.377(0.173)	0.377(0.173)	0.000(0.000)	0.386(0.175)	0.386(0.175)	0.000(0.000)	0.406(0.180)	0.406(0.180)	0.000(0.000)
	40	0.088(0.090)	0.080(0.084)	0.008(0.032)	0.255(0.146)	0.255(0.146)	0.000(0.000)	0.742(0.213)	0.742(0.213)	0.000(0.000)	0.739(0.213)	0.739(0.213)	0.000(0.000)	0.393(0.178)	0.393(0.178)	0.000(0.000)
	48	0.089(0.092)	0.072(0.081)	0.017(0.045)	0.276(0.150)	0.276(0.150)	0.000(0.000)	0.750(0.213)	0.750(0.213)	0.000(0.000)	0.750(0.213)	0.750(0.213)	0.000(0.000)	0.405(0.179)	0.405(0.179)	0.000(0.000)
	56	0.455(0.192)	0.437(0.188)	0.018(0.048)	0.284(0.152)	0.284(0.152)	0.000(0.000)	0.756(0.213)	0.756(0.213)	0.000(0.000)	0.747(0.213)	0.747(0.213)	0.000(0.000)	0.162(0.123)	0.146(0.116)	0.017(0.045)
	64	0.451(0.190)	0.441(0.188)	0.010(0.035)	0.320(0.159)	0.320(0.159)	0.000(0.000)	0.738(0.212)	0.738(0.212)	0.000(0.000)	0.741(0.212)	0.741(0.212)	0.000(0.000)	0.410(0.181)	0.410(0.181)	0.000(0.000)
	72	0.111(0.101)	0.096(0.092)	0.015(0.043)	0.255(0.152)	0.214(0.138)	0.042(0.071)	0.738(0.212)	0.738(0.212)	0.000(0.000)	0.735(0.212)	0.735(0.212)	0.000(0.000)	0.399(0.179)	0.399(0.179)	0.000(0.000)
	80	0.113(0.101)	0.103(0.095)	0.010(0.035)	0.420(0.186)	0.378(0.177)	0.042(0.071)	0.725(0.211)	0.725(0.211)	0.000(0.000)	0.378(0.173)	0.378(0.173)	0.000(0.000)	0.400(0.179)	0.400(0.179)	0.000(0.000)
	88	0.108(0.098)	0.098(0.092)	0.010(0.035)	0.444(0.191)	0.399(0.182)	0.045(0.077)	0.665(0.206)	0.665(0.206)	0.000(0.000)	0.315(0.157)	0.315(0.157)	0.000(0.000)	0.400(0.179)	0.400(0.179)	0.000(0.000)
	96	0.143(0.115)	0.128(0.107)	0.015(0.043)	0.471(0.193)	0.453(0.189)	0.018(0.048)	0.573(0.199)	0.573(0.199)	0.000(0.000)	0.236(0.138)	0.236(0.138)	0.000(0.000)	0.400(0.179)	0.400(0.179)	0.000(0.000)

Figure 3.6: Influence of the concentration of the analytes in the mixture over the average of the errors for the Bgg method with 32 members and mean combination rule: (a) FB, (b) TBZ and (c) MBC-BM. (d) Influence of the number of classes from the mixture in the average of the errors of the mixtures. (e) ROC curve.



(a)

(b)

(c)

(d)

(e)

Diversity among different diversity generation methods and the ensembles average mixture error. To conclude the degree of influence of diversity in the error of the ensembles, we present here a summary error table, see table 3.6, without EFN and EFP, including the different values of the diversity measures employed. The combination rule selected was the SMV because is better of those rules that fusion class labels. The WI and the HNV methods are not interesting to show in this table due to the absence of diversity. In the absence of a balance between diversity and accuracy of the members of the ensemble is difficult to infer very strict conclusions, except that the process of building better ensembles can not be guided solely by the criterion of diversity. No measure of diversity offer a correlation between their values implying that a higher accuracy of the ensemble. The only coincidence occurred with the DF measure in the Bgg and RS, practically occurs in both directions, the worst accuracy with the worst diversity and viceversa.

Table 3.6: Average mixture errors and their standard deviation between parentheses using neural ensembles with all the diversity creation methods and diversity measures with SMV. Marked the best errors and best diversity values in black. Underlined values show the opposite.

		DF		Dis		p		Q	
	Members	E	DF	E	Dis	E	p	E	Q
Bgg	8	0.297(0.158)	0.525(0.047)	0.308(0.159)	0.010(0.006)	0.303(0.159)	**0.517(0.183)**	0.296(0.157)	**0.765(0.188)**
	16	0.270(0.151)	0.484(0.081)	**0.261(0.150)**	0.011(0.008)	0.267(0.150)	0.455(0.222)	0.298(0.158)	0.762(0.229)
	24	0.273(0.152)	0.454(0.090)	0.268(0.152)	0.010(0.007)	0.288(0.155)	0.451(0.225)	0.280(0.153)	0.698(0.317)
	32	0.286(0.154)	0.456(0.091)	0.292(0.156)	0.010(0.008)	0.263(0.149)	0.423(0.232)	0.266(0.149)	0.687(0.323)
	40	0.273(0.153)	0.445(0.090)	0.307(0.160)	0.010(0.008)	0.264(0.149)	0.393(0.231)	0.262(0.148)	0.641(0.356)
	48	**0.257(0.149)**	**0.439(0.086)**	0.292(0.156)	0.010(0.008)	0.261(0.148)	0.392(0.232)	0.261(0.148)	0.643(0.346)
	56	0.263(0.150)	0.442(0.088)	0.300(0.158)	0.010(0.008)	0.268(0.150)	0.403(0.226)	0.259(0.148)	0.672(0.330)
	64	0.280(0.154)	0.451(0.087)	0.303(0.158)	0.011(0.009)	**0.256(0.147)**	0.396(0.229)	0.258(0.147)	0.662(0.332)
	72	0.286(0.155)	0.458(0.087)	0.307(0.159)	0.011(0.009)	0.258(0.147)	0.399(0.223)	**0.254(0.146)**	0.663(0.329)
	80	0.281(0.154)	0.453(0.087)	0.304(0.159)	0.011(0.010)	0.261(0.148)	0.384(0.230)	0.254(0.146)	0.647(0.343)
	88	0.278(0.153)	0.454(0.088)	0.310(0.160)	**0.012(0.010)**	0.261(0.148)	0.357(0.241)	0.261(0.148)	0.625(0.357)
	96	0.279(0.153)	0.458(0.088)	0.303(0.158)	**0.012(0.011)**	0.264(0.148)	0.359(0.240)	0.265(0.149)	0.613(0.366)
RS	8	0.398(0.175)	0.598(0.107)	0.403(0.174)	0.268(0.347)	0.278(0.180)	0.165(0.375)	**0.162(0.116)**	0.123(0.757)
	16	0.408(0.177)	0.606(0.124)	0.279(0.150)	0.255(0.323)	0.329(0.190)	0.267(0.334)	0.232(0.142)	0.280(0.718)
	24	0.394(0.174)	0.608(0.114)	0.266(0.148)	0.352(0.376)	0.306(0.184)	**0.298(0.306)**	0.237(0.143)	0.235(0.724)
	32	0.389(0.173)	0.616(0.116)	0.266(0.147)	**0.370(0.383)**	0.267(0.167)	0.260(0.313)	0.229(0.140)	0.309(0.680)
	40	**0.355(0.166)**	**0.594(0.118)**	0.279(0.149)	0.322(0.362)	0.224(0.139)	0.256(0.315)	0.224(0.139)	**0.326(0.659)**
	48	0.372(0.169)	0.603(0.115)	0.239(0.140)	0.281(0.344)	**0.198(0.131)**	0.230(0.332)	0.211(0.134)	0.321(0.668)
	56	0.376(0.170)	0.608(0.114)	0.236(0.140)	0.263(0.337)	0.203(0.132)	0.232(0.337)	0.204(0.132)	0.282(0.680)
	64	0.405(0.175)	0.620(0.112)	**0.218(0.136)**	0.258(0.333)	0.210(0.134)	0.245(0.332)	0.203(0.132)	0.273(0.658)
	72	0.404(0.175)	0.617(0.110)	0.238(0.140)	0.246(0.327)	0.210(0.134)	0.256(0.330)	0.203(0.132)	0.275(0.661)
	80	0.400(0.174)	0.618(0.109)	0.262(0.145)	0.254(0.330)	0.211(0.134)	0.252(0.319)	0.212(0.134)	0.283(0.643)
	88	0.394(0.173)	0.612(0.106)	0.255(0.143)	0.242(0.323)	0.219(0.136)	0.246(0.314)	0.210(0.134)	0.281(0.655)
	96	0.381(0.170)	0.596(0.106)	0.250(0.143)	0.239(0.320)	0.212(0.134)	0.239(0.311)	0.213(0.134)	0.285(0.647)

Global Diversity among different diversity generation methods. In Table 3.7, the diversity generated by the different methods under different pairwise diversity measures is shown, where the nonpairwise measures has not standard deviation because it is a unique value. For the pairwise measures all the members generated diversity is given. In this case, we can also show the nonpairwise measures of diversity for all the members generated by each diversity generation method. For comparison purposes, the diversity measures obtained with the NCL method are introduced, which means to get the votes of the members of that ensemble. It is understood that the NCL makes a fusion of the continuous outputs of its members and not the votes of these, which means playing a slightly different experiment to the pure NCL. All the members generated with the NCL and value of $\lambda = 1.0$ was used to show the diversity measures.

Table 3.7: Average values of diversity measures among generated members in test set and its standard deviation between parentheses for the different diversity creation methods. Best module error (BME) and worst module error (WME) are also given with its standard deviation.

	DF	Dis	p	Q	KW	k	Ent	BME	WME
HNV	0.11(0.005)	0.22(0.33)	0.014(0.039)	0.014(0.039)	0.0002	-1365.222	0.031	0.000(0.000)	0.125(0.174)
WI	0.001(0.004)	0.195(0.311)	0.011(0.032)	0.012(0.034)	0.00001	-17527.5	0.027	0.000(0.000)	0.417(0.305)
Bgg	0.370(0.110)	0.260(0.240)	0.280(0.320)	0.450(0.520)	0.0002	-2736.436	0.501	0.250(0.242)	2.625(0.475)
RS	0.331(0.286)	0.286(0.314)	0.452(0.318)	0.319(0.018)	0.0002	-6214.71	0.533	0.000(0.000)	1.791(0.497)
NCL	0.242(0.033)	0.040(0.131)	0.920(0.144)	0.965(0.129)	0.00002	-920.41	0.034	0.000(0.000)	0.500(0.330)

To conclude something about NCL is not trivial because of the elements mentioned. The method that create less diversity is the WI, followed by the HNV and then, those who modify the dataset to create diversity, as Partridge and Yates stated in different works [Partridge and Yates, 1996, Yates and Partridge, 1996]. Opitz proved that WI is the least effective mechanism to reach a good diversity [Opitz, 1999]. [Sharkey et al., 1995] discovered that multiple ANN trained with backpropagation starting from random points converge to the same or very similar local optima. Despite that, the non pairwise diversity measure *k* gives to the WI method the greatest diversity, being the only one which does this. In the other cases, the *k* measure has a similar behaviour to the others measures. The Bgg and the RS methods obtained higher diversity values, depending on the measure selected. The RS diversity is slightly higher than the Bgg, specially in non pairwise measures, and also presents better module errors. The NCL methods is so unstable on the measures given, but presents the higher diversity in the p and the Q measure. The non pairwise measure shows the NCL as a worst method than WI or the HNV. This last one presents the best modules error, followed by the WI.

Chapter 4

Conclusions and Future Work

4.1 Conclusions

Studies and developments along this project have helped to achieve high interest conclusions both in the field of computation as in the detection of pollutants. We now present the main findings of this project:

1. We have made a deep review of the diversity generation methods, ensemble methods, combination rules and diversity measures.

2. We have compared different methods of diversity creation from the 3 main possibilities in diversity creation: a) Starting Point in The Hipotesis Space b) Set of Accessible Hypothesis c) Traversal of Hypothesis Space; giving a wide exploration of the problem.

3. We have compared the Bagging method, the Random Subspaces method, the Negative Correlation Learning, the hidden nodes variation and the weights initialization variation.

4. We have compared different pairwise diversity measures in the member selection strategy. Also the selection of members with higher accuracy was compared.

5. We have compared different combination rules on the ensemble decision strategy, both from class label fusion and continuous outputs fusion.The combination of continuous outputs provide more and better information to make up the ensemble decision. Specifically, the mean rule produces better results than SMV in identifying pesticides in complex fluorescence mixtures. Also the Max rule produces interesting results in this information context.

6. We have developed and completed the software tool MULLPY to work with complex ensemble learning, diversity creation methods, different combination rules, many kind of new parameters and new presentation methods.

7. There is no diversity measure that could find the best combination for an ensemble in this problem, as pointed out [Kuncheva and Whitaker, 2003] and other authors in their respective studies to other data set context. The diversity is a key factor in finding the best possible ensemble, but we also found that it is needed a compromise, still indefinitely, between accuracy of the members created and its diversity among them.

8. The modification of the feature space provide much more and better diversity in classification context than instances, weights initialization or the number of hidden nodes does.

9. The weights initialization and the hidden nodes variation does not provide any diversity at all. Its are methods to select the best configuration of the architecture, not to generate diversity.

10. The NCL method is more simple to use and requires less computational resources than bagging and random subspaces method.

11. We have developed a variation of the NCL process in the combination stage, after the learning process. In this problem, the Max rule provides better results than the Mean rule, the original method of combination. The mean rule provides more stability in the decision making than others.

12. Diversity measures, by themselves, are unable to find a solution to the problem of low concentrations of fungicides on fluorescence spectra. Despite that, with the diversity generated is now possible to find an optimal combination of classifiers to improve the results obtained so far. Problem that can be addressed by optimization (Evolutive algorithms) or by finding a compromise between accuracy and diversity.

Future work

1. The multilabel method is easier to work with the technique of One-Versus-All, dividing the problem into *N* binary classifiers, which not only accelerates the convergence in slow algorithms as backpropagation, but also allows the construction of hybrid ensembles, due to the limitation of many machine learning methods to work with more than 2 classes simultaneously. Multilabel classification process is then carried out through the ensemble of those members.

2. The use of evolutionary algorithms (EA) is not only interesting for the learning of individuals or ensembles, attending more possibilities, but for the most diverse selection members within multiple combinations of the individuals.

3. The use of hybrid ensembles seems to provide higher classification accuracies. To extend the study to other machine learning paradigms like decision trees, Naive Bayes, Support vector machines and also regression procedures may provide better and more consistent results.

4. To use the original dataset without preprocessing, in order to produce more diversity. The 401 features available could provide more diversity than this 14 does.

5. To study the separation of the classes MBC and BM with diversity creation methods and explore its limits.

6. To study more complex rules of selecting diverse but also accurate members in a static manner. The EA could provide better solutions, but the static study could provide us a better knowing about the problem in order to design dynamic processes.

7. To work with the NCL in an iterative manner, adding new diverse members after the ensemble learning process.

8. To substitute the Backpropagation algorithm with a resilient propagation learning algorithm (RPROP), which is much faster and accurate. To study also the possibility of having a NCL with the RPROP algorithm.

9. To eliminate the restriction of the making up ensembles with all the spectra, specially in the NCL, building diverse ensembles for each kind of spectra.

Bibliography

[Albani, 2008] Albani, J. (2008). *Principles and Applications of Fluorescence Spectroscopy*. Wiley.

[Almhdi et al., 2007] Almhdi, K. M., Valigi, P., Gulbinas, V., Westphal, R., and Reuter, R. (2007). Classification with artificial neural networks and support vector machines: application to oil fluorescence spectra. *EARSeL eProceedings*, 6(2):115–129.

[Álvarez Romero et al., 2013] Álvarez Romero, Y., García Báez, P., and Suárez Araujo, C. P. (2013). Computational study based on supervised neural architectures for fluorescence detection of fungicides. In *IWANN*, pages 114–123.

[Bachiller et al., 1998] Bachiller, P., Pérez, R., Martínez, P., Aguilar, P., and Merchán, A. (1998). Optimize on the size of a backpropagation nn solving the mixture problem. In *Proceedings Int. ICSC Symp. on Eng. of Int. Syst.*

[Blum, 1962] Blum, M. (1962). Properties of a neuron with many inputs. In *Principles of self-organization*, International tracts in computer science and technology and their application, pages 95–120. Symposium Publications, Pergamon Press.

[Breiman, 1996] Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.

[Breiman, 1998] Breiman, L. (1998). Arcing classifiers. *Annals of Statistics*, 26(3):801–849. cited By (since 1996)494.

[Breiman, 1999] Breiman, L. (1999). Pasting small votes for classification in large databases and on-line. *Machine Learning*, 36(1):85–103. cited By (since 1996)80.

[Breiman, 2000] Breiman, L. (2000). Randomizing outputs to increase prediction accuracy. *Machine Learning*, 40(3):229–242.

[Breiman, 2001] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

[Brown, 2004] Brown, G. (2004). *Diversity in Neural Network Ensembles*. PhD thesis, University of Birmingham.

[Brown and Kuncheva, 2010] Brown, G. and Kuncheva, L. (2010). Good and bad diversity in majority vote ensembles. *Lecture Notes in Computer Science*, 5997 LNCS:124–133.

[Brown et al., 2005] Brown, G., Wyatt, J., Harris, R., and Yao, X. (2005). Diversity creation methods: A survey and categorisation. *Information Fusion*, 6(1):5–20.

[Cabaniss, 1991] Cabaniss, S. (1991). Theory of variable-angle synchronous fluorescence spectra. *Analytical Chemistry*, 63(13):1323–1327.

[Chandra and Yao, 2006] Chandra, A. and Yao, X. (2006). Ensemble learning using multi-objective evolutionary algorithms. *Journal of Mathematical Modelling and Algorithms*, 5(4):417–445.

[Chowdary et al., 2009] Chowdary, M., Mahato, K., Kumar, K., Mathew, S., Rao, L., Krishna, C., and Kurien, J. (2009). Autofluorescence of breast tissues: Evaluation of discriminating algorithms for diagnosis of normal, benign, and malignant conditions. *Photomedicine and Laser Surgery*, 27(2):241–252. cited By (since 1996) 2.

[Chudoba et al., 2013] Chudoba, R., Sadílek, V., Rypl, R., and Vorechovsky, M. (2013). Using python for scientific computing: Efficient and flexible evaluation of the statistical characteristics of functions with multivariate random inputs. *Computer Physics Communications*, 184(2):414–427. cited By (since 1996) 0.

[Clarke, 2008] Clarke, C. (2008). *Development of an automated identification system for nanocrystal encoded microspheres in flow cytometry*. PhD thesis, Cranfield University.

[Danaher et al., 2007] Danaher, M., Ruyck, H. D., Crooks, S. R., Dowling, G., and O'Keeffe, M. (2007). Review of methodology for the determination of benzimidazole residues in biological matrices. *Journal of Chromatography B*, 845(1):1 – 37.

[Dietterich, 2000] Dietterich, T. (2000). Ensemble methods in machine learning. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1857 LNCS:1–15.

[Dietterich and Bakiri, 1991] Dietterich, T. G. and Bakiri, G. (1991). Error-correcting output codes: A general method for improving multiclass inductive learning programs. In *IN PROCEEDINGS OF AAAI-91*, pages 572–577. AAAI Press.

[Dorney et al., 2012] Dorney, J., Bonnier, F., Garcia, A., Casey, A., Chambers, G., and Byrne, H. (2012). Identifying and localizing intracellular nanoparticles using raman spectroscopy. *Analyst*, 137(5):1111–1119. cited By (since 1996) 0.

[Duin and Tax, 2000] Duin, R. and Tax, D. (2000). Experiments with classifier combining rules. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1857 LNCS:16–29.

[Font Fernández et al., 2009] Font Fernández, J. M., Manrique Gamo, D., and Ríos Carrión, J. (2009). *Redes de neuronas artificiales y computación evolutiva*. Universidad Politécnica de Madrid.

[Forcada, 2003] Forcada, V. R. (2003). *Clasificación supervisada basada en redes bayesianas. Aplicación en biología computacional*. PhD thesis, Universidad politécnica de Madrid, Madrid.

[Freeman and Skapura, 1991] Freeman, J. and Skapura, D. (1991). *Neural networks: algorithms, applications, and programming techniques*. Computation and neural systems series. Addison-Wesley.

[Freund and Schapire, 1997] Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119 – 139.

[Fumera et al., 2008] Fumera, G., Roli, F., and Serrau, A. (2008). A theoretical analysis of bagging as a linear combination of classifiers. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(7):1293–1299.

[García Báez, 2005] García Báez, P. (2005). *HUMANN: Una nueva red neuronal artificial adaptativa, no supervisada, modular y jerárquica. Aplicaciones en neurociencia y medioambiente.* PhD thesis, ULPGC.

[García Báez et al., 2012] García Báez, P., Álvarez Romero, Y., and Suárez Araujo, C. P. (2012). A computational study on supervised and unsupervised neural architectures with data fusion for fluorescence detection of fungicides. *Luminescence*, 27:534–572.

[González Pérez and Hernández Hernández, 2002] González Pérez, C. and Hernández Hernández, L. (2002). *Introducción al análisis instrumental*. Editorial Ariel S.A.

[Gutta and Wechsler, 1996] Gutta, S. and Wechsler, H. (1996). Face recognition using hybrid classifier systems. In *ICNN96, Washington, DC*, pages 1017–1022. IEEE Computer Society Press, Los Alamitos. CA.

[Hansen and Salamon, 1990] Hansen, L. K. and Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001.

[Haykin, 1994] Haykin, S. (1994). *Neural Networks: A Comprehensive Foundation*. Macmillan, New York.

[Hebb, 1949] Hebb, D. O. (1949). *The Organization of Behavior*. John Wiley, New York.

[Hecht-Nielsen, 1990] Hecht-Nielsen, R. (1990). *Neurocomputing*. New Horizons in Technology Series. Addison-Wesley Publishing Company.

[Hinton and Sejnowski, 1986] Hinton, G. E. and Sejnowski, T. J. (1986). Explorations in the microstructure of cognition. In Rumelhart, D. E. and McClelland, J. L., editors, *Parallel distributed processing*, volume 1, chapter Learning and relearning in Boltzmann machines, pages 282–317. MIT Press, Cambridge, MA, USA.

[Ho, 1998] Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(8):832–844.

[Holland, 1992] Holland, J. H. (1992). *Adaptation in natural and artificial systems*. MIT Press, Cambridge, MA, USA.

[Islam et al., 2003] Islam, M., Yao, X., and Murase, K. (2003). A constructive algorithm for training cooperative neural network ensembles. *Neural Networks, IEEE Transactions on*, 14(4):820–834.

[Jain et al., 1996] Jain, A. K., Mao, J., and Mohiuddin, K. (1996). Artificial neural networks: A tutorial. *Computer*, 29:31–44.

[Johansson et al., 2007] Johansson, U., Lofstrom, T., and Niklasson, L. (2007). The importance of diversity in neural network ensembles - an empirical investigation. In *Neural Networks, 2007. IJCNN 2007. International Joint Conference on*, pages 661–666.

[Jones et al., 2013] Jones, E., Oliphant, T., Peterson, P., et al. (2001–2013). Scipy: Open source scientific tools for python.

[Judd, 1990] Judd, J. (1990). *Neural Network Design and the Complexity of Learning*. Neural Network Modeling and Connectionism. MIT Press.

[Junfei et al., 2010] Junfei, C., Qingfeng, W., and Huailin, D. (2010). A random feature selection approach for neural network ensembles: Considering diversity. In *Computational Intelligence and Software Engineering (CiSE), 2010 International Conference on*, pages 1–4.

[Kim and Cho, 2008] Kim, K.-J. and Cho, S.-B. (2008). Evolutionary ensemble of diverse artificial neural networks using speciation. *Neurocomputing*, 71(7–9):1604 – 1618.

[Kittler et al., 1998] Kittler, J., Hatef, M., Duin, R., and Matas, J. (1998). On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239.

[Ko and Sabourin, 2013] Ko, A.-R. and Sabourin, R. (2013). Single classifier-based multiple classification scheme for weak classifiers: An experimental comparison. *Expert Systems with Applications*, 40(9):3606–3622.

[Kohavi and John, 1997] Kohavi, R. and John, G. H. (1997). Wrappers for feature subset selection. *ARTIFICIAL INTELLIGENCE*, 97(1):273–324.

[Kohonen, 1988] Kohonen, T. (1988). *Self-organization and associative memory*. Springer series in information sciences. Springer-Verlag.

[Krogh and Vedelsby, 1995] Krogh, A. and Vedelsby, J. (1995). Neural network ensembles, cross validation, and active learning. In *Advances in Neural Information Processing Systems*, pages 231–238. MIT Press.

[Kuncheva et al., 2002] Kuncheva, L., Skurichina, M., and Duin, R. (2002). An experimental study on diversity for bagging and boosting with linear classifiers. *Information Fusion*, 3(4):245–258.

[Kuncheva et al., 2001] Kuncheva, L. I., Bezdek, J. C., and Duin, R. P. (2001). Decision templates for multiple classifier fusion: an experimental comparison. *Pattern Recognition*, 34(2):299 – 314.

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

[Langdon et al., 2002] Langdon, W., Barrett, S., and Buxton, B. (2002). Combining decision trees and neural networks for drug discovery. In Foster, J., Lutton, E., Miller, J., Ryan, C., and Tettamanzi, A., editors, *Genetic Programming*, volume 2278 of *Lecture Notes in Computer Science*, pages 60–70. Springer Berlin Heidelberg.

[Lashley, 1950] Lashley, K. (1950). *In search of the engram. Symposia of the Society for Experimental Biology*, volume 4. Society for Experimental Biology.

[Lawton and Martin, 1985] Lawton, W. and Martin, M. (1985). The advance mixture problem-principles algorithms. Technical report, Jet Propulsion Laboratory IOM384, California Institute of Technology.

[Lippmann, 1994] Lippmann, R. (1994). Neural networks, bayesian a posteriori probabilities, and pattern classification. In Cherkassky, V., Friedman, J., and Wechsler, H., editors, *From Statistics to Neural Networks*, volume 136 of *NATO ASI Series*, pages 83–104. Springer Berlin Heidelberg.

[Liu and Higuchi, 2003] Liu, Y. Yao, X. and Higuchi, T. (2003). *Designing Neural Network Ensembles by Minimising Mutual Information. Masoud Mohammadian, Rahul A. Sarker, and Xin Yao (Eds.). Computational Intelligence in Control*, pages 1–21. Hershey : Idea Group Pub, USA & London (UK).

[Liu, 1998] Liu, Y. (1998). *Negative correlation learning and evolutionary neural network ensembles*. PhD thesis, The University of New South Wales.

[Liu and Yao, 1999] Liu, Y. and Yao, X. (1999). Ensemble learning via negative correlation. *Neural Networks*, 12:1399–1404.

[Liu et al., 2000] Liu, Y., Yao, X., and Higuchi, T. (2000). Evolutionary ensembles with negative correlation learning. *IEEE Transactions on Evolutionary Computation*, 4:380–387.

[Lloyd and Evett, 1977] Lloyd, J. and Evett, I. (1977). Prediction of peak wavelengths and intensities in synchronously excited fluorescence. *Anal. Chemistry*, 49(12):1710–1715.

[Loewy, 2000] Loewy, R. (2000). Plaguicidas en aguas subterráneas del alto valle de río negro neuquén. tesis de maestría en ciencias químicas. Master's thesis, Universidad Nacional de Comahue, Argentina.

[Lofstrom et al., 2010] Lofstrom, T., Johansson, U., and Bostrom, H. (2010). Comparing methods for generating diverse ensembles of artificial neural networks. cited By (since 1996)0.

[Lv and Gu, 2012] Lv, J. and Gu, Q. (2012). The type detection of mineral oil fluorescence spectroscopy in water based on the kpca and cca-svm. *Advances in Intelligent and Soft Computing*, 160 AISC(VOL. 2):19–22. cited By (since 1996) 0.

[Maclin and Shavlik, 1995] Maclin, R. and Shavlik, J. W. (1995). Combining the predictions of multiple classifiers: using competitive learning to initialize neural networks. In *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 1*, IJCAI'95, pages 524–530, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[McCulloch, 1959] McCulloch, P. (1959). *Agathe tyche: of nervous net – the lucky reckoners. Mechanization of thought Processes*. H.M. Stationery Office.

[McCulloch and Pitts, 1943] McCulloch, W. and Pitts, W. (1943). *A logical calculus of the ideas inmanent in nervous activity*, chapter 5, pages 115–133. Bulletin of mathematical biophysics.

[Mitchell, 1998] Mitchell, M. (1998). *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA, USA.

[Márquez, 2008] Márquez, L. (2008). *Buenas prácticas agrícolas en la aplicación de los fitosanitarios*, pages 3–5. Ministerio de Medio Ambiente y Medio Rural y Marino.

[Nabavi-Kerizi et al., 2010] Nabavi-Kerizi, S., Abadi, M., and Kabir, E. (2010). A pso-based weighting method for linear combination of neural networks. *Computers and Electrical Engineering*, 36(5):886–894.

[Oliphant, 2007] Oliphant, T. E. (2007). Python for scientific computing. *Computing in Science Engineering*, 9(3):10–20.

[Opitz and Maclin, 1999] Opitz, D. and Maclin, R. (1999). Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 11:169–198.

[Opitz and Shavlik, 1996a] Opitz, D. and Shavlik, J. (1996a). Actively searching for an effective neural network ensemble. *Connection Science*, 8:337–353.

[Opitz, 1999] Opitz, D. W. (1999). Feature selection for ensembles. In *Proceedings of the National Conference on Artificial Intelligence*, pages 379–384.

[Opitz and Maclin, 1997] Opitz, D. W. and Maclin, R. E. (1997). Empirical evaluation of bagging and boosting for artificial neural networks. In *Neural Networks,1997., International Conference on*, volume 3, pages 1401–1405.

[Opitz and Shavlik, 1996b] Opitz, D. W. and Shavlik, J. W. (1996b). Generating accurate and diverse members of a neural-network ensemble. In *Advances in Neural Information Processing Systems*, pages 535–541. MIT Press.

[Oza and Tumer, 2001] Oza, N. C. and Tumer, K. (2001). Input decimated ensembles: Decorrelation through dimensionality reduction. In Kittler, J. and Roli, F., editors, *Proceedings of the Second International Workshop on Multiple Classifier Systems*, pages 238–249, Cambridge, UK. Springe-Verlagr.

[Park and Sandberg, 1991] Park, J. and Sandberg, I. W. (1991). *Universal approximation using radial-basis-function networks. Neural Computation*, chapter 3(2), pages 246–257. Neural Computation.

[Parmanto et al., 1995] Parmanto, B., Munro, P. W., and Doyle, H. R. (1995). Improving committee diagnosis with resampling techniques. In Touretzky, D. S., Mozer, M., and Hasselmo, M. E., editors, *Advances in Neural Information Processing Systems 8, NIPS, Denver, CO, November 27-30, 1995*, pages 882–888. MIT Press.

[Partridge, 1996] Partridge, D. (1996). Network generalization differences quantified. *Neural Netw.*, 9(2):263–271.

[Partridge and Yates, 1996] Partridge, D. and Yates, W. B. (1996). Engineering multiversion neural-net systems. *Neural Comput.*, 8(4):869–893.

[Parvin et al., 2013] Parvin, H., Ansari, S., and Parvin, S. (2013). A diversity production approach in ensemble of base classifiers. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7629 LNAI(PART 1):51–60.

[Polikar, 2006] Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, 6(3):21–45.

[Prechelt, 2000] Prechelt, L. (2000). An empirical comparison of c, c++, java, perl, python, rexx, and tcl. Technical report, Fakultat for Informatik, Universitat Karlsruhe. Germany.

[Rahman et al., 2010] Rahman, A., Verma, B., and Yao, X. (2010). Non–uniform layered clustering for ensemble classifier generation and optimality. In Wong, K., Mendis, B., and Bouzerdoum, A., editors, *Neural Information Processing. Theory and Algorithms*, volume 6443 of *Lecture Notes in Computer Science*, pages 551–558. Springer Berlin Heidelberg.

[Raviv and Intrator, 1996] Raviv, Y. and Intrator, N. (1996). Bootstrapping with noise: An effective regularization technique. *Connection Science*, 8:355–372.

[Rifkin and Klautau, 2004] Rifkin, R. and Klautau, A. (2004). In defense of one-vs-all classification. *The Journal of Machine Learning Research*, 5:101–141.

[Rodríguez et al., 2006] Rodríguez, J. J., Kuncheva, L. I., and Alonso, C. J. (2006). Rotation forest: A new classifier ensemble method. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(10):1619–1630.

[Rodríguez et al., 2010] Rodríguez, J. J. S., Padrón, M. E. T., Aufartová, J., and Ferrera, Z. S. (2010). *Benzimidazole Fungicides in Environmental Samples: Extraction and Determination Procedures*. InTech.

[Rojas, 1996] Rojas, R. (1996). *Neural Networks: A Systematic Introduction*. Springer-Verlag.

[Rosen, 1996] Rosen, B. (1996). Ensemble learning using decorrelated neural networks. *Connection Science*, 8:373–384.

[Rosenblatt, 1962] Rosenblatt, F. (1962). *Principles of neurodynamics: perceptrons and the theory of brain mechanisms*. Report (Cornell Aeronautical Laboratory). Spartan Books.

[Rumelhart et al., 1987] Rumelhart, D. E., Hinton, G. E., and McClelland, J. L. (1987). A general framework for parallel distributed processing. In Rumelhart, D. E., McClelland, J. L., et al., editors, *Parallel Distributed Processing: Volume 1: Foundations*, pages 45–76. MIT Press, Cambridge.

[Rumelhart et al., 1986] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Parallel distributed processing: explorations in the microstructure of cognition, vol. 1. chapter Learning internal representations by error propagation, pages 318–362. MIT Press, Cambridge, MA, USA.

[Ruta and Gabrys, 2001] Ruta, D. and Gabrys, B. (2001). Analysis of the correlation between majority voting error and the diversity measures in multiple classifier systems.

[Ruta and Gabrys, 2005] Ruta, D. and Gabrys, B. (2005). Classifier selection for majority voting. *International Journal of Information Fusion*, pages 63–81.

[Römer et al., 2011] Römer, C., Bürling, K., Hunsche, M., Rumpf, T., Noga, G., and Plümer, L. (2011). Robust fitting of fluorescence spectra for pre-symptomatic wheat leaf rust detection with support vector machines. *Computers and Electronics in Agriculture*, 79(2):180–188.

[Sabik and Jeannot, 1998] Sabik, H. and Jeannot, R. (1998). Determination of organonitrogen pesticides in large volumes of surface water by liquid–liquid and solid-phase extraction using gas chromatography with nitrogen–phosphorus detection and liquid chromatography with atmospheric pressure chemical ionization mass spectrometry. *Journal of Chromatography A*, 818(2):197 – 207.

[Sahin, 1997] Sahin, F. (1997). A radial basis function approach to a color image classification problema in real time industrial application. Master's thesis, Polytecnic insititue of Virginia. Electronic Engineering.

[Schapire, 1990] Schapire, R. (1990). The strength of weak learnability. *Machine Learning*, 5(2):197–227. cited By (since 1996)1115.

[Schiffmann et al., 1994] Schiffmann, W., Joost, M., and Werner, R. (1994). Optimization of the backpropagation algorithm for training multilayer perceptrons. Technical report.

[Sesmero et al., 2012] Sesmero, M. P., Alonso-Weber, J. M., Gutiérrez, G., Ledezma, A., and Sanchís, A. (2012). A new artificial neural network ensemble based on feature selection and class recoding. *Neural Computing and Applications*, 21(4):771–783.

[Sharkey, 1999] Sharkey, A. (1999). *Combining artificial neural nets: ensemble and modular multi-net systems*. Perspectives in neural computing. Springer.

[Sharkey et al., 2000] Sharkey, A., Sharkey, N., Gerecke, U., and Chandroth, G. (2000). The "test and select" approach to ensemble combination. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1857 LNCS:30–44.

[Sharkey et al., 1997] Sharkey, A. J., Sharkey, A. J. C., and Sharkey, N. E. (1997). Diversity, selection, and ensembles of artificial neural nets. In *In Proceedings of Third International Conference on Neural Networks and their Applications. IUSPIM, University of Aix-Marseille III*, pages 205–212.

[Sharkey et al., 1995] Sharkey, N., Neary, J., and Sharkey, A. (1995). Searching weight space for backpropagation solution types. In *Current Trends in Connectionism: Swedish Conference on Connectionism.*, pages 103–120.

[Suárez Araujo et al., 2013] Suárez Araujo, C. P., Álvarez Romero, Y., García Báez, P., Sánchez Rodríguez, A., and Santana Rodríguez, J. J. (2013). Fluorescence identification of benzimidazole fungicides: An intelligente computation solution. In *8th International Conference: Instrumenal Methods of Analysis-Modern Trends and Applications, Thessaloniki, 15-19 September*.

[Suarez Araujo et al., 2010] Suarez Araujo, C. P., García Báez, P., and Hernández Trujillo, Y. (2010). *Neural Computation Methods in the Determination of Fungicides, Fungicides. Odile Carisse (Ed.)*, chapter 23. Intech.

[Suárez Araujo, 1996] Suárez Araujo, C. P. (1996). *Neurociencia y Computación Neuronal*. Universidad de Las Palmas de Gran Canaria.

[Suárez Araujo et al., 2009] Suárez Araujo, C. P., García Báez, P., Sánchez Rodríguez, A., and Santana Rodríguez, J. J. (2009). Humann-based system to identify benzimidazole fungicides using multi-synchronous fluorescence spectra: An ensemble approach. *Analytical and Bioanalytical Chemistry*, 394(4):1059–1072.

[Swann and Allinson, 1998] Swann, A. and Allinson, N. (1998). Fast committee learning: preliminary results. *Electronics Letters*, 34(14):1408–1410.

[Tumer and Oza, 2003] Tumer, K. and Oza, N. C. (2003). Input decimated ensembles. *Pattern Analysis and Applications*, 6(1):65–77.

[Vasilescu et al., 2011] Vasilescu, J., Marmureanu, L., and Carstea, E. (2011). Analysis of seawater pollution using neural networks and channels relationship algorithms. *Romanian Journal of Physics*, 56(3-4):530–539.

[Vassilakis et al., 1998] Vassilakis, I., Tsipi, D., and Scoullos, M. (1998). Determination of a variety of chemical classes of pesticides in surface and ground waters by off-line solid-phase extraction, gas chromatography with electron-capture and nitrogen-phosphorus detection, and high-performance liquid chromatography with post-column derivatization and fluorescence detection. *J Chromatogr A*, 823(1-2):49–58.

[Villasana, 2010] Villasana, P. M. (2010). *Introducción a las redes neuronales artificiales.*

[Wang et al., 2000] Wang, W., Jones, P., and Partridge, D. (2000). Diversity between neural networks and decision trees for building multiple classifier systems. In *Proceedings of the First International Workshop on Multiple Classifier Systems*, MCS '00, pages 240–249, London, UK, UK. Springer-Verlag.

[Werbos, 1974] Werbos, P. (1974). *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. PhD thesis, Harvard University.

[Widrow, 1962] Widrow, B. (1962). Generalization and information storage in networks of adaline 'neurons'. In *Self-organizing systems*, pages 96–104. Spartan Books.

[Wilbers et al.,] Wilbers, I. M., Langtangen, H. P., and Odegard, A. Using cython to speed up numerical python programs. Technical report, Department of Informatics, University of Oslo.

[Wirth, 2001] Wirth, M. (2001). *A nongrid approach to medical image registration: Matching images of the breast.* PhD thesis, University RMIT, Melbourne, Australia.

[Wolpert, 2002] Wolpert, D. (2002). The supervised learning no-free-lunch theorems. In Roy, R., Köppen, M., Ovaska, S., Furuhashi, T., and Hoffmann, F., editors, *Soft Computing and Industry*, pages 25–42. Springer London.

[Wolpert and Macready, 1995] Wolpert, D. and Macready, W. (1995). No free lunch theorems for search. Technical Report SFI-TR-95-02-010, Santa Fe Institute.

[Woods et al., 1997] Woods, K., Kegelmeyer, W.P., J., and Bowyer, K. (1997). Combination of multiple classifiers using local accuracy estimates. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(4):405–410.

[Wozniak et al., 2014] Wozniak, M., Graña, M., and Corchado, E. (2014). A survey of multiple classifier systems as hybrid systems. *Information Fusion*, 16(0):3 – 17. Special Issue on Information Fusion in Hybrid Intelligent Fusion Systems.

[Wu et al., 2009] Wu, Q., Li, Y., Wang, C., Liu, Z., Zang, X., Zhou, X., and Wang, Z. (2009). *Dispersive liquid-liquid microextraction combined with high performance liquid chromatography-fluorescence detection for the determination of carbendazim and thiabendazole in environmental samples*, chapter 638.2, pages 139–145. ISSN 0003-2670. Analytica Chimica Acta.

[Yao et al., 1998] Yao, X., (smieee, X. Y., and Liu, Y. (1998). Making use of population information in evolutionary artificial neural networks.

[Yates and Partridge, 1996] Yates, W. and Partridge, D. (1996). Use of methodological diversity to improve neural network generalisation. *Neural Computing & Applications*, 4(2):114–128.

[Zenobi and Cunningham, 2001] Zenobi, G. and Cunningham, P. (2001). Using diversity in preparing ensembles of classifiers based on different feature seature subsets to minimize generalization error. In *Lecture Notes in Computer Science*, pages 576–587. Springer Verlag.

[Zhou et al., 2002] Zhou, Z.-H., Wu, J., and Tang, W. (2002). Ensembling neural networks: Many could be better than all. *Artificial Intelligence*, 137(1–2):239 – 263.

[Álvarez Romero, 2012] Álvarez Romero, Y. (2012). Estudio de sistemas de detección de contaminantes basados en redes neuronales artificiales: Análisis y diseño. Technical report.