Chapter 1

Introduction

Making the correct decision when facing a problem is a complicate task, yet a task of special importance in our society. In order to make a correct decision, it is necessary to perform an evaluation of the available data, extract information from experience and involve any people who could have an important contribution into the solution of the problem. Herbert A. Simon describes the context of decision making as follows:

The work of managers, of scientists, of engineers, of lawyers–the work that steers the course of society and its economic and governmental organizations–is largely work of making decisions and solving problems. It is work of choosing issues that require attention, setting goals, finding or designing suitable courses of action, and evaluating and choosing among alternative actions. The first three of these activities–fixing agendas, setting goals, and designing actions–are usually called problem solving; the last, evaluating and choosing, is usually called decision making.[24]

The complexity of decision making is inherent to the complexity of the problem at hand. Most problems contain an unknown number of variables and sometimes insufficient, sometimes overwhelming amount of data. It is such problems which present the most complexity when attempted to be solved. It is then when the use of tools such as computers becomes crucial in finding an useful solution.

Decision Making is a study area that involves disciplines such as math, economics, psychology and computer science. There is a broad spectrum of decision making problems, which in general involve deciding what the best option in a certain situation is. For instance, voting for a president in an election, estimating the prize of an item or simply deciding what group a certain item belongs to.

Among this range of problems, 2 types of problems stand out because of their importance and the potential use of software to help in making a decision. The first one, classification, can be defined as associating an item with its correct class. Deciding the type of an item represents a problem of variable complexity. Sorting objects by their color may be simple for both a person and a machine, but for instance, classifying documents based on their topic is a problem that has yet to be completely solved. Several problems that seem simple, such as classifying handwritten digits, pose machines great difficulty, while problems that may be impossible for humans, like classifying DNA sequences, may be simple for machines.

The second problem is prediction. Prediction is defined as a claim that a particular event will take place in the future [1]. Classic prediction problems take place at the stock market, where extensive investigation efforts have been performed to discover patterns and increase the confidence of different actions. Common computational prediction techniques make use of statistics extensively.

1.1. Multi-Agent Systems in Decision Making problems

One of the most important approaches into solving decision making problems has been multi-agent systems. Wooldrige defines an agent as:

Autonomous agents are computational systems that inhabit some complex dynamic environment, sense and act autonomously in this environment, and by doing so realize a set of goals or tasks for which they are designed [29]

In decision making problems, agents usually handle a broad domain of information and work with complex problems. This translates into agents based on intricate models which are hard to analyze and whose decisions are not always reliable. An approach for improving agent design has been the pre-partitioning of a problem, adopting a divide-and-conquer strategy to find an optimal decision. This opened the window to the possibility of using multiple agents, each one specialized in a certain problem partition, to reach an optimal solution. Thus, multi-agent systems were introduced.

Regarding multi-agent systems, Wooldrige offers the following definition:

A Multi-agent system is a loosely coupled network of problem solvers that interact to solve problems that are beyond the individual capabilities or knowledge of each problem solver [29]

1.2. Machine Learning and Ensemble Algorithms

One approach in the design and application of multi-agent systems is machine learning. Machine Learning deals with the design of programs which take advantage of data, examples and experience to improve accuracy or performance [16]. Specifically, the area of Machine Learning algorithms that deal with Multi-Agent systems is known as Ensemble Learning. Ensemble learning is based on the idea of having a set of weak learners (which can also be denoted as agents) which together build a strong learner through agreement mechanisms. Each of these learner's knowledge may be represented in different ways, but some of the most successful methods have been those that introduce machine learning techniques into the system [28].

Developing a multi-agent system based on ensemble methods requires deciding what

the data for the learning process will be. The data is then divided among agents, training each one of them and establishing a mechanism to reach an agreement. This process, based on our analysis, is presented in the following algorithm:

CREATEMULTIAGENTSYSTEM($ProblemData$)

1 $Agents \leftarrow \emptyset$

2 $ProblemDataSimplified \leftarrow SIMPLIFYPROBLEMDATA(ProblemData)$

3 $ProblemPartitions \leftarrow PARTITIONPROBLEM(ProblemDataSimplified)$

4 **for** each Partition in ProblemPartitions

5 **do** $Agents \leftarrow Agents \cup \{CREATEAGENT(Partition)\}$

6 **for** each Agent in Agents

7 **do** $TRAIN(AGENT)$

8 **return** $Agents$

Once having the multi-agent system built, in order to be able to test it the following method should be used.

GETDMPSOLUTION($Agents, ProblemInstance$)

1 $Solution \leftarrow GETSOLUTION(Agents, ProblemInstance)$

2 **return** $Solution$

In the previous algorithm, we may identify 4 different components:

- **Simplification** : Simplifying or reducing the dimensions of a problem is important in scenarios where the huge amount of data make it difficult to identify a

solution. Documented methods in this front include Principal Component Analysis (PCA) and Charting, among others [17]. The goal is achieving a simpler representation of a problem without compromising relevant information.

- **Partition** : The content and number of partitions affect directly the performance of the agents. It is important for us to find a partition such that it will optimize the reliability of the results. Partitions may be done in several fashions, from hard partitions in which there is no intersection between subproblems, to soft ones [26] where the subproblems solved by two different agents may be similar.

- **Learning** : Each agent must specialize in the solution of its corresponding subproblem. Different Machine Learning techniques are applicable, depending on the knowledge area we find ourselves in. It is not necessary for all agents to use homogeneous techniques. Thus, part of the analysis in this section is identifying which technique best fits the problem in hand [14] [26]

- **Integration**: It is necessary to identify a way in which the information provided by each of the agents can be combined for the generation of a solution. The technique to use is dependent of the type of problem we are handling. Simple approaches like weighted averaging or best solution selection are possibilities in this front. However, depending of the type of problem more complex methods may be convenient. [26] In our algorithm, GETSOLUTION uses the integration component by reaching an agreement among all agents and returning an answer.

It is clear that there are multiple options when considering a multi-agent system, which are dependent on the type of problem we are handling. It is thus hard to design an optimal multi-agent system for the solution of a specific decision making problem.

1.3. Objectives

1.3.1. General Objective

To design and implement a framework for Automated Multi-agent system creation, able to solve decision making problems, and through this framework, analyze the nature of Multi-agent systems and use the gathered knowledge in the solution of novel problems.

1.3.2. Specific Objectives

- Design and implement an easy to use, flexible and extensible API interface for the creation of multi-agent systems
- Design and implement a graphic application to work with the formentioned framework.
- Provide grounds for further development on multi-agent learning algorithms.
- Study the different stages in a multi-agent system, especially the simplification, partition and learning stages.
- Use the research results to design a multi-agent system to solve the HLA classification problem. [20].
- Analyze the results of the HLA classifying multi-agent and evaluate the implications on the dataset.

1.4. Research Scope

This thesis is focused on working with ensemble algorithms, based on the perception of them as multi-agent systems. Boosting [26] and Bagging [5] are the two families of algorithms considered. Thus, other possible approaches to the design of these systems were not considered. As well, we only focused on the solution of two specific types of problems: Prediction and Classification.

The research considered was regarding the selection of a combination of components (simplification, partition, training and post-processing components) which produce the best performance in the solution of a certain problem in a specific area of interest. We were specially focused on recent work by Sun and Ho[26] regarding the partition component, and work by Freund and Schapire [10] regarding the learning component.

The test scenarios for our system were on specific areas, and due to time constraints, we did not consider an extense amount of test scenarios. For the case of classification, we considered datasets on the area of computer vision and bioinformatics. In the case of prediction, we considered a dataset in the area of data analysis.

This document contains the context and results of our research, including the description of the developed software, the results of the multi-agent experimental analysis and the description of the characteristics of the multi-classifier built for the HLA [4] problem.