Knowledge Extraction based on Evolutionary Learning (KEEL): Analysis of Development Method, Genetic Fuzzy System

Manju
M.Tech. Student
MDU Rohtak

Pooja Mittal
Assistant Professor
MDU, Rohtak

ABSTRACT

The purpose of this paper is to describe some basic concepts about keel including evolutionary learning , keel structure with dataset and analysis of genetic fuzzy system. The aim of this section is to present the KEEL framework, describing some guidelines to help a potential developer to build new methods inside the KEEL environment. The next sections will deal with the formats of the configuration files of KEEL which include data sets files, method descriptions and so on. And , the last section describes the API dataset of KEEL, which is used to handle and check the dataset files.

Keywords

Data mining, Evolutionary algorithms, Genetic Fuzzy System, Knowledge extraction, Dataset

1. INTRODUCTION

KEEL (Knowledge Extraction based on Evolutionary Learning) tool, an open source software that supports data management and provides a platform for the analysis of evolutionary learning for Data Mining problems of different kinds including as regression, classification, unsupervised learning. It includes a big collection of evolutionary learning algorithms based on different approaches: Pittsburgh, Michigan. it empowers the user to perform complete analysis of any genetic fuzzy system in comparison to existing ones, with a statistical test module for comparison.

The presently available version of KEEL consists

of the following function blocks .

1) Data Management -- This part is composed of a set of tools that can be used to build new data, export and import data in other formats to the KEEL format, data edition and visualization, applying transformations and partitioning to data.

2) Design of Experiments -- The aim of this part is the design of the desired experimentation over the selected data sets. It provides many options to choose from: type of validation, type of learning (classification, regression, unsupervised learning), etc.

3) Educational Experiments -- With a similar structure to the previous part, it allows you to design an experiment which can be debugged step-by-step in order to use this as a guideline, to show the learning process of a certain model by using the platform for educational objectives.

According to the above function blocks, KEEL can be useful for different types of users, who expect to find determined features in Data Mining (DM) software.

KEEL pays special attention to the implementation of evolutionary learning and soft computing based techniques for Data Mining problems including regression, classification, clustering, pattern mining and so on.

The aim of this paper is to present three new aspects of KEEL: KEEL dataset , a data set repository which includes the data set partitions in the KEEL format and shows some results of algorithms in these data sets; some guidelines for including new algorithms in KEEL, helping the researchers to make their methods easily accessible to other authors and to compare the results of many approaches already included within the KEEL software; and a module of statistical procedures developed in order to provide to the researcher a suitable tool to contrast the results obtained in any experimental study.

Evolutionary Algorithms (EAs) are optimization algorithms based on natural evolution and genetic processes. In Artificial Intelligence (AI), EAs are one of the most successful search techniques for complex problems. The main motivation for applying EAs to knowledge extraction tasks is that they are robust and adaptive search methods that perform

a global search in place of candidate solutions (for instance, rules or other forms of knowledge representation). They have proven to be an important technique both for learning and knowledge extraction, making them a promising technique in DM. Recently EAs, particularly Genetic Algorithms (GAs) have proved to be an important technique for learning and knowledge extraction. This makes them also a promising tool in Data Mining .The idea of automatically discovering knowledge from databases is a very attractive and complex task. That’s why, there has been a growing interest in DM in various AI related areas, including EAs. The main objective for applying EAs to knowledge extraction tasks is that they are robust and adaptive search methods that perform a global search in place of candidate solutions . The use of EAs in problem solving such as image retrieval, the learning of controllers in robotics and the improvement of E-learning systems show their suitability.

One of the most popular approaches is the hybridization between Fuzzy Logic and GAs leading to genetic fuzzy systems (GFSs)

GFS is basically a fuzzy system augmented by a learning process based on a GA.GAs are search algorithms based on natural genetics that provide robust search capabilities in complex spaces, and thereby offer a valid approach to problems requiring efficient and effective search processes.

Fuzzy systems are one of the most important areas for the application of the Fuzzy Set Theory. Fuzzy systems have been successfully applied to solve different kinds of problems in various application domains. In this contribution we introduce a non-commercial Java software tool named KEEL (Knowledge Extraction based on Evolutionary Learning) . This tool empowers the user to assess the behavior of EAs for different kinds of Data Mining problems: regression, classification, clustering, pattern mining, etc. Consequently, the application of EAs for learning fuzzy systems is also included in KEEL, including a representative set of GFSs. It allows us to perform a complete analysis of any genetic fuzzy system in comparison to existing ones, including a statistical test module for comparison.

KEELS divide the GFS approaches into two processes, tuning and learning:

Genetic tuning: If there exists a knowledge base (KB), We apply a genetic tuning process for improving the FRBS performance.

Genetic learning:The second possibility is to learn KB Components (where we can even include an adaptive Inference engine).

This tool can offer several advantages:

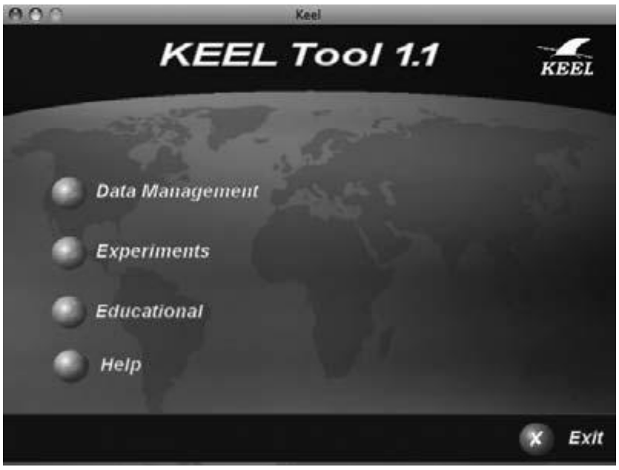
1) It reduces programming work. It includes a library with evolutionary learning algorithms based on different paradigms (Pittsburgh, Michigan and IRL) and simplifies the integration of evolutionary learning algorithms with different pre-processing techniques. It can alleviate the work of programming and enable researchers to focus on the analysis of their new learning models in comparison with the existing ones.

2) Due to the use of a strict object-oriented approach for the library and software tool, these can be used on any machine with Java. As a result any researcher can use KEEL on his or hers machine, independently of the operating system.

3) It extends the range of possible users applying evolutionary learning algorithms. Researchers with less knowledge, would be able to successfully apply these algorithms to their problems.

2. KEEL DESCRIPTION

KEEL is a software tool to assess EAs for DM problems including regression, classification, clustering, pattern mining and so on. Presently available version of keel consists of the following function blocks(see below mentioned fig.)



The main features of KEEL are Describe below.

It provides a user-friendly interface, oriented to the analysis of algorithms.

• The software is aimed at creating experiments containing multiple data sets and algorithms connected among them selves to obtain an expected results. Experiments are independently script-generated from the user interface for an off-line run in the same or other machines.

• KEEL also allows the creation of experiments in on-line mode, aiming to provide an educational support in order to learn the operation of the algorithm included.

• It includes data pre-processing algorithms .

• It has a statistical library to analyze results of algorithms. It comprises a set of statistical tests for analyzing the suitability of the results and performing parametric and non-parametric comparisons between the algorithms.

• Some algorithms have been developed using Java Class Library for Evolutionary Computation (JCLEC).

3. METHODS IN KEEL

Every method in KEEL has assigned a XML file which describes its main characteristics. This file will be employed by the KEEL GUI to allow the user to select the values of the parameters of any execution of the method.The KEEL Method Description files are located under the ../dist/algorithm directory, inside of the folder where its associated .JAR file is generated (e.g.,../dist/algorithm/methods). Each Method Description file is an XML composed by a unique root node, <algorithm _specification> . This node is divided into two parts:

<algorithm_specification>

Header

Parameters

</ algorithm_specification >

Header : Basic information about the method.

Parameters : A list of parameters of the method.

Method Configuration files

In KEEL, every method uses a configuration file to extract the values of the parameters which will be employed during its execution. Although it is generated automatically by the KEEL GUI (by using the information contained in the corresponding method description file, and the values of the parameters specified by the user), it is important to fully describe its structure because any KEEL method must be able to completely read it, in order to get the values of its parameters specified in each execution.

Each configuration file has the following structure:

algorithm : Name of the method.

inputData : A list with the input data files of the method.

outputData : A list with the output data files of the method.

parameters : A list of parameters of the method, containing the name of each parameter and its value.

Developing a new Method in Keel:

Before to start the task of developing a new method inside of KEEL environment, some operations have to be performed in order to fully integrate it. By following these guidelines, a developer can left all the input/output operations to be accomplished by KEEL environment, focusing its efforts in the construction of the method itself. The steps needed to complete the integration of a new method in KEEL are:

--Reading of the configuration file.

-- Development of the method.

-- Writing the output files.

-- Registering the method in KEEL.

-- Making the use case files.

-- Building the executables of the method.

Data files

In KEEL, the data sets are managed by plain ASCII text files, with the .dat extension. Usually, they are located under the ../dist/data directory, each one in its own folder (which also should contains the partitions created from the whole data set). In addition, preprocess methods will also create data files as its output, which will be placed on the ../datasets directory of its experiment. This section describes the format employed to define them . Each KEEL data file is composed by 2 sections:

Header : Basic metadata describing the data set.

Data : Content of the dataset.

In both sections it is possible to insert comments, by employing the

“%”character.

Data: The data instances are represented as rows of comma separated values, where each value corresponds to one attribute, in the order defined by the header. Missing or null values are defined as <null> or ? .If the dataset corresponds to a classification problem, the output type must be nominal:

Output files: Every method in KEEL must produce at least two output files: A train results file (marked with the extension .tra) and a test results file (marked with the extension .tst). Although the method can employ additional output files to show more information about the process performed, those additional files must be handled entirely by the method. Thus, KEEL will only handle the two standards output files. Both output files share the same structure: They are composite by the same header of the data employed as input of the method, and a set of rows (one for each instance of the data set) describing the expected outputs and the outputs obtained by the application of the method. Thus, they are structured as follows:

<Expected1,1 > . . . <Expected1,n ><Method1,1 > . . . <Method1,n >

<Expected2,1 > . . . <Expected2,n ><Method2,1 > . . . <Method2,n >

Use Case files

The use case files of KEEL provides valuable information to understanding every of the methods which are available to use. They are XML files, located in the ../dist/help directory.

Each KEEL use case file is composed by 4 sections:

Name : The name of the method.

Reference : A list of references associated with the method.

General Description : Generic information about the method.

Example : A example about the use of the method.

4. KEEL DATA SET

The KEEL-dataset repository is devoted to the data sets in KEEL format which can be used with the software and provides:

A detailed categorization of the considered data sets and a description of their characteristics. Tables for the data sets in each category have been also created. A descriptions of the papers which have used the partitions of data sets available in the KEEL-dataset repository. These descriptions include results tables, the algorithms used and additional material.

KEEL-dataset contains two main sections according to the previous two points. In the first part, the data sets of the repository are presented. They have been organized in several categories and sub-categories arranging them in tables. Each data set has a dedicated webpage. These webpages also provide the complete data set and the partitions ready to download. On the other hand, the experimental studies section is a novel approach in this type of repositories. It provides a series of webpages for each experimental study with the data sets used and their results in different formats as well, ready to perform a direct comparison.

One of the main components of KEEL is the API Dataset. It manages the entire process of acquisition, processing and validation of the data files, offering the data sets to the developer in a suitable way, freeing him from the task of acquiring the data needed to perform any experiment. This section describes three key concepts of the API Dataset:

Data files grammar: The grammar employed to define the data files. Any file generated by this grammar will be a valid data file.

Semantic restrictions of the data files: Apart from the syntax restrictions, some semantic verifications are performed by the API Dataset over the data files.

Description of the classes: To close this section, the main public classes of the API Dataset are described.

Data files grammar

In this subsection is shown the grammar which describes the format of the KEEL data files. The final tokens of the grammar are:

_ fg. Denotes the void production. It is also known as l or e.

_ IDENT. Denotes an identifier (IDENT = (’A’ -’Z’ , ’a’-’z’ , ’0’-’9’)_).

_ INTEGER. Is an integer value (INTEGER = (0 . 9)+).

_ REAL . Is a real value (REAL = (0-9)+[.(0-9)_]).

principal -> Relation

-> Attributes

-> Inputs

-> Outputs

-> Data

Relation -> "@relation" IDENT

Attributes -> "@attribute" IDENT attributeType Attributes

-> {}

attributeType -> "integer" intBoundaries

-> "real" realBoundaries

-> "{" IDENT idList "}"

intBoundaries -> "[" INTEGER "," INTEGER "]"

-> {}

realBoundaries -> "[" REAL "," REAL "]"

-> {}

idList -> "," IDENT idList

-> {}

Inputs -> "@inputs" IDENT idList

-> {}

Outputs -> "@outputs" IDENT idList

-> {}

Data -> @data dataList

dataList -> lineData dataList

-> {}

lineData -> IDENT lineDataCont

lineDataCont -> "," IDENT lineDataCont2

lineDataCont2 -> "," IDENT lineDataCont2

Semantic restrictions of the data files

Attributes

Description of the classes

The API Dataset is composed by four main classes:

InstanceSet: This class contains a complete set of instances defining a data base.

Instance: This class represents a single instance.

Attributes: This static class contains definitions about every attribute of the data contained in the Instance set.

Attribute: This class contains relevant information about a single attribute.

5. CONCLUSION

In this work, we have described KEEL, a software tool to Assess EAs for DM problems, paying special attention to its structure including with the concept of evolutionary leaning. It relieves researchers of much technical work and allows them to focus on the Analysis of their new GFS algorithms in comparison with the Existing ones. Moreover, the tool enables researchers with a Basic knowledge of fuzzy logic and evolutionary computation to apply GFSs to their work. We have shown how keel used on different Datasets. The KEEL software tool is being continuously updated and improved. At the moment, we are developing a new set of GFSs and a test tool that will allow us to apply parametric and non-parametric tests on any set of data. We are also developing data visualization tools for the on-line and offline modules.

6. REFERENCES

[1] D.E. Goldberg, Genetic algorithms in search, optimization, and machine learning, Addison-Wesley Professional, Canada (1989).

[2] J.H. Holland, Adaptation in natural and artificial systems, Ann Arbor: University of Michigan Press (1975).

[3] O. Cord´on, F. Herrera, F. Hoffmann and L. Magdalena, Genetic fuzzy systems Evolutionary tuning and learning of fuzzy knowledge bases, World Scientific, Singapore (2001).

[4] A.E. Eiben and J.E. Smith, Introduction to Evolutionary Computing,

[5] S. Smith, A learning system based on genetic algorithms, Ph.D. thesis, Unversity of Pittsburgh (1980).

[6] K.A. De Jong, W.M. Spears and D.F. Gordon, Using genetic algorithms for concept learning, Machine Learning 13 (1993) 161-188.

[7] S.W. Wilson, Classifier Fitness Based on Accuracy, Evolutionary Computation 3:2 (1995) 149-175.

[8] E. Bernad´o-Mansilla and J.M. Garrell, Accuracy-Based Learning Classifier