Building Java Intelligent Applications
Data Mining for Java Type-2 Fuzzy Inference Systems

Manuel Castañón-Puga¹, Josué Miguel Flores-Parra¹, Juan Ramón Castro¹,
Carelia Gaxiola-Pacheco¹, and Luis Enrique Palafox-Maestre¹

Autonomous University of Baja California,
Tijuana, Baja California, México
{puga, mflores31, jrcastror, cgaxiola, lepalafox}@uabc.edu.mx

Abstract
This paper introduces JT2FISClustering, a data mining extension for JT2FIS. JT2FIS is a Java
class library for building intelligent applications. This extension is used to extract information
from a data set and transform it into an Interval Type-2 Fuzzy Inference System in Java
applications. Mamdani and Takagi-Sugeno Fuzzy Inference Systems can be generated using
fuzzy c-means or subtractive data mining methods. We compare the outputs and performance
of Matlab R versus Java in order to validate the proposed extension.

Keywords: Data mining, Interval Type-2 Fuzzy Inference System, Java Intelligent Applications

1 Introduction

In recent years, the use of new information technologies has come to help in the handling of a
large amount of data. Data mining extraction is an evolved technology that allows representing
knowledge of data and the implicit storage.

Additionally, data mining contributes to the making of tactical and strategic decisions pro-
vided by power users because they can measure actions and results in the best way. It generates
descriptive models to explore and understand the data and identifies patterns, relationships
and dependencies that affect the final results. They can create predictive models [6] that allow
undiscovered relationship through the data mining process and explore possible business rules
[10].

JT2FIS¹ is a class library developed for Java applications [4]. The main purpose of JT2FIS is
to deploy a library to build interval Type-2 fuzzy inference system (FIS) with an object-oriented
programming language for Java developers.

Most of FIS used until now are FIS based on a Type-1 model [13], but lately, a Type-2
model has been developed and other applications are being extended with it [14]. This current

¹http://kiliwa.tij.uabc.mx/projects/jt2fis

Selection and peer-review under responsibility of the Scientific Programme Committee of ICCS 2015
© The Authors. Published by Elsevier B.V.
doi:10.1016/j.procs.2015.05.401
technique led us to Type-2 General Fuzzy Inference Model [7] that has been developed as a next step to design and have Fuzzy Inference Systems with more capability to model real-world things [5].

The purpose of this paper is to introduce JT2FISClustering\(^2\), a Java data mining extension for JT2FIS. We described the JT2FIS architecture, design, implementation, and Java programmers examples in detail in [4].

This paper uses the JT2FIS class library as a core Type-2 Fuzzy Inference System for the implementation of JT2FISClustering data mining methods.

1.1 Mamdani and Takagi-Sugeno Fuzzy Logic System

Mamdani and Takagi-Sugeno Fuzzy Logic Systems are popular FIS approaches and are formed by IF-THEN rules with the same antecedent structures. The difference between them is in the consequent structures. The consequent of a Mamdani rule is a fuzzy set while in Takagi-Sugeno is a function, so Takagi-Sugeno uses fewer fuzzy rules to represent a real system than Mamdani.

Takagi-Sugeno Fuzzy Inference System was proposed in an effort to develop a systematic approach for generating fuzzy rules from a given input-output data set. This model consists of rules with fuzzy antecedents and mathematical function in the consequent part. The antecedents divide the input space into a set of fuzzy regions while consequences describe the behavior of the system in those regions [9].

The Mamdani and Takagi-Sugeno process is divided into four parts: fuzzifier, rule base, fuzzy inference engine, and output processor. In type-2, a type reducer is needed in the output processor to derive a type-1 set from the type-2 set [3]. Figure 1 shows a block diagram of the classic structure of a Mamdani and Takagi-Sugeno Fuzzy Logic System.

![Figure 1: Type-2 fuzzy inference system block diagram.](image)

1.2 Fuzzy C-Means clustering algorithm

The fuzzy proposals represent a significant place in data mining providing intelligible results. One of these proposals is the Fuzzy C-Means clustering algorithm (FCM)\(^2\) [2] [1] that makes

\(^2\)http://kiliwa.tij.uabc.mx/projects/jt2fisclustering

2720
use of a membership function and centroid computation procedure iteratively to find the best centroid.

The FCM is one of the most popular clustering algorithms; the effectiveness of the clustering method relies on the distance measure. An FCM is the resulting combination of the c-means approach with the handling of fuzzy data. The result of this combination is sufficient because it considers the uncertainty presented in the data avoiding incorrect results and creating crisp partitions in the correct way [2]. Additionally the FCM is used to acquire the adequate levels of the set clustering parameters [12].

1.3 Subtractive clustering algorithm

Subtractive Clustering operates by finding the optimal data point to define a cluster centre based on the density of surrounding data points. It reduces the computational complexities and gives a better distribution of cluster centres in comparison with other clustering algorithms [9].

This method considers each point as a potential centre, and based on mathematical approximations, calculates the best choice of centre. Each cluster centre can be seen as a fuzzy rule of the system, and the identified group represents the antecedent of this rule. The measure of potential for data is estimated based on the distance of this data point from all other data points [11].

The identification of a Takagi-Sugeno using clustering involves formation of clusters in the data space and translation of these clusters into Takagi-Sugeno rules such that the model obtained is close to the system to be identified.

2 JT2FISCLUSTERING DATA MINING EXTENSION FOR JT2FIS

JT2FISClustering is a class library developed in Java. The main purpose of this library is data mining large data sets to set up automatically a Type 2 Inference System applying the paradigm of object-oriented programming relying on JT2FIS[4] class library.

JT2FISClustering is a structure in packets containing a collection of classes. The content and organization of these packages is shown in Table 1.

<table>
<thead>
<tr>
<th>Package JT2FISClustering</th>
<th>Package Clustering</th>
<th>Class Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Class FuzzyCMMeans</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Class Subtractive</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Package Generate</th>
<th>Class GenerateSugenoFis</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Class GenerateMamdaniFis</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Package util</th>
<th>Class MersenneTwister</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Class Utilities</td>
</tr>
</tbody>
</table>

Figure 2 shows a diagram packages UML of class library JT2FISClustering. The class collection is organized in code packages that encapsulate its behavior.

One advantage of this library is the heritage of the capabilities of an object-oriented programming paradigm to integrate new clustering methods. Figure 3 shows the abstract class
Clustering that allows us to extend clustering methods.

The current version of JT2FISClustering has three different membership functions (see Table 2). Fuzzy c-Means clustering is the default method.

<table>
<thead>
<tr>
<th>Type of Clustering</th>
<th>Type 2 Membership Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuzzy c-Means, Subtractive</td>
<td>GaussCutMemberFunction</td>
</tr>
<tr>
<td></td>
<td>Params=[inputs outputs uncertainty]</td>
</tr>
<tr>
<td>Fuzzy c-Means</td>
<td>GaussUncertaintyMeanMemberFunction</td>
</tr>
<tr>
<td></td>
<td>Params=[inputs outputs]</td>
</tr>
<tr>
<td>Fuzzy c-Means, Subtractive</td>
<td>GaussUncertaintyMeanMemberFunction</td>
</tr>
<tr>
<td></td>
<td>Params=[inputs outputs uncertainty]</td>
</tr>
<tr>
<td>Fuzzy c-Means</td>
<td>GaussUncertaintyStandardDesviationMemberFunction</td>
</tr>
<tr>
<td></td>
<td>Params=[inputs outputs]</td>
</tr>
<tr>
<td>Fuzzy c-Means, Subtractive</td>
<td>GaussUncertaintyStandardDesviationMemberFunction</td>
</tr>
<tr>
<td></td>
<td>Params=[inputs outputs uncertainty]</td>
</tr>
</tbody>
</table>
2.1 JT2FIS Fuzzy C-Means Data Mining Operation

In order to exemplify how to use JT2FISClustering to create an FIS using Fuzzy C-Means method, we are going to follow the next steps:

1. Create a new instance of the class FuzzyCMeans.
2. Create a new instance of the class GenerateMamdaniFis.
3. Create a list with all the dataset for inputs and outputs.
4. Generate FIS, selecting the required membership function.

2.1.1 Creating a new instance of the class FuzzyCMeans.

First, we must create a new instance of the class FuzzyCMeans and define the number of clusters that you want to generate. Listing 1 shows a Java code example.

```
Listing 1: Creating a New Instance FuzzyCMeans

// Number of clusters required.
int numberCluster=3;
FuzzyCMeans fcm=new FuzzyCMeans(numberCluster);
```

2.1.2 Creating a new instance of the class GenerateMamdaniFis

To build a FIS, we must create a new instance of the class GenerateMamdaniFis. This instance receives an object as a parameter that represents the type of clustering method you want to use. Listing 2 shows a Java code example.

```
Listing 2: Creating a New Instance GenerateMamdaniFis

//fcm is a instance of the class FuzzyCMeans.
GenerateMamdaniFis gMamFis=new GenerateMamdaniFis(fcm);
```

2.1.3 Data sets for inputs and outputs

A dataset (ArrayList) that contains all data from each of the inputs and outputs must be used to build the FIS. In order to show this particular example, we are going to use a 2 inputs array and 2 outputs array.

2.1.4 Generate FIS using Fuzzy C-Means

Once you have created the inputs and outputs data sets, the FIS can be generated. Listing 3 shows how to generate a FIS using “GaussUncertaintyMeanMemberFunction” membership function.

```
Listing 3: Generating FIS Type Mamdani.

//Variable to establish uncertainty
double uncertainty=0.8;
//Calling the method to generate Mamdani FIS type membership function "GaussUncertaintyMeanMemberFunction" choosing uncertainty.
```
Mamdani fis = MamFis.generateMamdaniFisGauss
    UncertaintyMeanMemberFunction(inputList, outputList, uncertainty);
System.out.println(fis.toString());

You can use different Member Function options to obtain the desired FIS. The generated
FIS object contains a full Mamdani or Takagi-Sugeno Fuzzy Logic System functionality and
can be used into Java code to build JT2FIS intelligent applications.

2.2 JT2FIS Subtractive Data Mining Operation

In order to exemplify how to use JT2FISClustering to create an FIS using Fuzzy Subtractive
method, we are going to follow the next steps:

1. Create a new instance of the class Subtractive.
2. Create a new instance of the class GenerateSugenoFis.
3. Create a list with all the dataset for inputs and outputs.
4. Generate FIS, selecting the required membership function.

2.2.1 Creating a new instance of the class Subtractive.

First, we must create a new instance of the class Subtractive and define the range of influence
of the cluster for each input and output dimension that you want to generate. Listing 4 shows
Java code example.

RADII specifies the range of influence of the cluster centre for each input and output di-
mension.

Listing 4: Creating a New Instance Subtractive

// Range of influence of the clusters
double radii = 0.5;
Subtractive sbt = new Subtractive(radii);

2.2.2 Creating a new instance of the class GenerateSugenoFis

To build a FIS, we must create a new instance of the class GenerateSugenoFis. This instance
receives an object as a parameter that represents the type of clustering method you want to
use. Listing 5 shows a Java code example.

Listing 5: Creating a New Instance GenerateSugenoFis

// fcm is a instance of the class Subtractive
GenerateSugenoFis gSugenoFis = new GenerateSugenoFis(sbt);

2.2.3 Data sets for inputs and outputs

A dataset (ArrayList) that contains all data from each of the inputs and outputs must be used
to build the FIS. In order to show this particular example, we are going to use a 2 inputs array
and 2 outputs array.

2724
2.2.4 Generate FIS using Subtractive

Once you have created the inputs and outputs data sets, the FIS can be generated. Listing 6 shows how to generate a FIS using “GaussUncertaintyMeanMemberFunction” membership function.

Listing 6: Generating FIS Type Sugeno.

```java
//Variable to establish uncertainty
double uncertainty = 0.9;

//Calling the method to generate Sugeno FIS type membership function “GaussUncertaintyMeanMemberFunction” choosing uncertainty.
Sugeno fis = gSugenoFis.generateSugenoFisGaussUncertaintyMeanMemberFunction(inputList, outputList, uncertainty);
System.out.println(fis.toString());
```

You can use different Member Function options to obtain the desired FIS. The generated FIS object contains a full Mamdani or Takagi-Sugeno Fuzzy Logic System functionality and can be used into Java code to build JT2FIS intelligent applications.

3 VALIDATING JT2FISCLUSTERING

One of the main motivations of JT2FIS and JT2FISClustering is not to substitute or displace Matlab® libraries to analyze data, but to create a Java library for building object oriented Java applications.

To validate the proposed library, we compared JT2FISClustering against the methods provided by Matlab®, genfis3 and genfis2. We expected that JT2FISClustering were able to generate the same outputs that the Matlab® libraries in order to validate accuracy and performance.

So first, we generated a data set randomly created using the Mersenne Twister pseudo-random numbers generation method[8] in order to use it as inputs and output in JT2FISClustering as well as Matlab®.

3.1 JT2FIS Fuzzy C-Means Data Mining Test

For this first test case, we used two inputs, two outputs and three rules (numbers of clusters) for 100, 1,000 and 10,000 dataset. We selected the Gaussian membership function with uncertainty in the middle (“GaussUncertaintyMeanMemberFunction”) for inputs and outputs.

In Table 3 the time that each method took to generate the inference system is shown. We can see that the JT2FISClustering library is as fast as the genfis3 method provided by Matlab®.

3.2 JT2FIS Subtractive Data Mining Test

For this second test case, we used two inputs, two outputs for 100, 1,000 and 10,000 dataset. We selected the Gaussian membership function with uncertainty in the middle (“GaussUncertaintyMeanMemberFunction”) for inputs and outputs and the cluster influence range was 0.9.

In Table 4 the time that each method took to generate the inference system is shown. We can see that the JT2FISClustering library is as fast as the genfis3 method provided by Matlab®.
Table 3: Comparing times (ms) of FuzzyCMeans JT2FISClustering to genfis3 Matlab.

<table>
<thead>
<tr>
<th>Number of Datas</th>
<th>JT2FISClustering FuzzyCMeans Time (ms)</th>
<th>Matlab genfis3 FuzzyCMeans Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>14.5</td>
<td>16.18</td>
</tr>
<tr>
<td>1000</td>
<td>94.86</td>
<td>172.81</td>
</tr>
<tr>
<td>10000</td>
<td>197.21</td>
<td>277.84</td>
</tr>
</tbody>
</table>

Table 4: Comparing times (ms) of Subtractive JT2FISClustering to genfis2 Matlab.

<table>
<thead>
<tr>
<th>Number of Datas</th>
<th>JT2FISClustering Subtractive Time (ms)</th>
<th>Matlab genfis2 Subtractive Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>81</td>
<td>90</td>
</tr>
<tr>
<td>1000</td>
<td>270</td>
<td>280</td>
</tr>
<tr>
<td>10000</td>
<td>4012</td>
<td>6492</td>
</tr>
</tbody>
</table>

3.3 JT2FISClustering Versus Matlab®

In this study case the cancer dataset was taken proportionately for Matlab®. This dataset contains 9 inputs and 2 outputs. Inputs are a 9x699 matrix defining nine attributes of 699 biopsies, and outputs are a 2x966 matrix where each column indicates a correct category with a one in either element 1 or element 2.

It was compared the genfis2 and genfis3 methods of MatLab® versus the JT2FISClustering Methods with uncertainty 0. In case of genfis2 the radio was 0.3 and in case of genfis3 the type selected was Mamdani and the numbers of clusters was 3. The results were equal for this case, the same number of rules, with the same values for parameters of each member function of FIS. This is seen in the Table 5.

Table 5: Comparing Results of Methods JT2FISClustering to Methods MatLab.

<table>
<thead>
<tr>
<th>Method</th>
<th>JT2FISClustering</th>
<th>MatLab</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subtractive</td>
<td>2 rules</td>
<td>2 rules</td>
</tr>
<tr>
<td></td>
<td>2 members functions for each Input</td>
<td>2 members functions for each Input</td>
</tr>
<tr>
<td></td>
<td>2 members functions for each Output</td>
<td>2 members functions for each Output</td>
</tr>
<tr>
<td>Fuzzy C-Means</td>
<td>3 rules</td>
<td>3 rules</td>
</tr>
<tr>
<td></td>
<td>3 members functions for each Input</td>
<td>3 members functions for each Input</td>
</tr>
<tr>
<td></td>
<td>3 members functions for each Output</td>
<td>3 members functions for each Output</td>
</tr>
</tbody>
</table>

Figure 4 and 5 show values for an input for each method of MatLab® and JT2FISClustering.

4 Conclusion and future work

On this paper, we introduced JT2FISClustering, a data mining extension for JT2FIS class library. The proposed extension library includes the basic fuzzy c-means and subtractive data mining methods to extract information from the data set and transform it into an Interval Fuzzy Inference System. The methods help to configure Mamdani and Takagi-Sugeno Java
FLSs in JT2FIS Java applications. First, we presented a brief description of the library. Some UML diagrams were depicted to show its components in a software point of view. Then, a code example was listed to help to understand how to use it by Java programmers. Finally, a set of performance tests was applied to compare data mining methods with Matlab.

The JT2FISClustering library operations showed the comparison times (ms) of FuzzyCMeans JT2FISClustering versus genfis3 and Subtractive versus genfis2 in Matlab; in both cases JT2FISClustering proved to be as fast as Matlab.

As future work, we will continue working on improving performance and accuracy applying more tests, and planning continues refactoring and adding more data mining methods for Fuzzy Logic Systems to the library.

JT2FIS and JT2FISClustering is available for academic purposes in http://kiliwa.tij.uabc.mx

Acknowledgments

This work was supported in part by the Internal Fund for Research Projects (Grant No. 300.6.C.135.17) of the Autonomous University of Baja California, México.
References


