Soil Classification by Generating Fuzzy rules

P.Bhargavi, M.Sc., M.Tech.
Department of CSE
Madanapalli Institute of Technology and Science,Madanapalli.

Dr. S. Jyothi, M.Sc., M.S, Ph.D
Associate Professor & Head, Department of Computer Science
Sri Padmavathi Mahila ViswaVidyalayam,Tirupati.
(Womens University)

Abstract

Fuzzy classification rules are widely considered as a well-suited representation of classification knowledge, as they allow readable and interpretable rule bases. The goal of this paper is to discuss how fuzzy classification rules are generated for soil data. The most important task in the design of fuzzy classification systems is to find a set of fuzzy rules from training data to deal with a specific classification problem. In this paper, we generate fuzzy rules from training data to deal the soil data classification problem, by first defining the membership functions for the input attributes of the soil data, and then generating the initial fuzzy rules for the training data based on the member functions defined for the attributes of the soil data and then we merge these fuzzy rules in order to generate definitive fuzzy rules .

Keywords
Fuzzy classification systems, soil data, membership functions, fuzzy rules.

1. Introduction

Of soil characteristics, soil Classification is the most important one. It influences many other properties of great significance to land use and management.

Soil classification can effectively reduce the complexity of information and help us to understand the main features in soil groups. Although the process will lose some information, it provides a convenient means of information transfer. This paper tries to discuss and present a fuzzy classification method, which can reduce information and identify natural soil textures.

One of the important applications of fuzzy set theory [16] is in the fuzzy classification systems. Fuzzy classification rules are widely considered a well-suited representation of classification knowledge, as they allow readable and interpretable fuzzy rule bases [3]. Due to their abstraction from numbers to linguistic variables they resemble the way; humans would possibly formulate their knowledge [21]. Everybody can easily be convinced that the world is inherently fuzzy and that crisp thresholds can almost never be justified. Therefore, the use of fuzzy rules seems quite intuitive. Specifying fuzzy sets may seem easier than the specification of crisp intervals. One may hope that the inference system will 'somehow' deal with the uncertainties. Especially, when rule bases are automatically or half-automatically created from example data, this can hold true.

There are two approaches to obtain fuzzy rules for fuzzy classification systems. One of them is given directly by experts; the other is produced through an automatic learning process. In recent years, some methods have been presented to generate fuzzy rules from training instances.

In this paper, we have used triangular member ship function method to define the membership functions for the input attributes and we present a method to generate fuzzy rules from a set of training data to deal with the soil data [4] [5][6] classification problem. In section 2 we discuss about problem characteristics and in section 3 we have defined the membership functions for the input attributes and section 4 discusses with the algorithmic steps for fuzzy classification and in section 5 we present experimental results and section 6 presents conclusion.

2. Problem Characteristics

The Soil data contains 111 instances, with the 7 input attributes (i.e., Depth, Sand, Silt, Clay, Sandbysilt, Sandbyclay, Sandbysiltclay), and one output attribute (i.e,Textureclass). The characteristics of the input attributes of the Soil data are shown in Table 1.The output attribute of the soil data consists of ten types of output values, as shown in Table 2.

Type	Output Attribute
1	S
2	Sicl
3	Sic
4	C
5	Sl
6	Cl
7	Sil
8	L
9	Ls
10	Scl

Table 1. The characteristics of input attributes

Input attributes	**Minimal value (cm)**	**Maximal value (cm)**
Depth	0.1	2.08
Sand	16.81	97.8
Silt	1.2	56.53
Clay	1.0	58.3
SandbySilt	0.31	81.5
SiltbyClay	0.3	4.0
SandbySiltClay	0.2	44.45

Table 2. Output values of the output attributes

In order to clearly illustrate clearly the proposed fuzzy rules generation algorithm, we have chosen 2 to 5 instances for each type of output attribute from the Soil data.(i.e. s, sicl, sic, c, sl, cl, sil, l, ls, scl).

3. Member ship functions

There exist numerous types of membership functions, the most commonly used in practice are triangles, trapezoids, bell curves, Gaussian and sigmoidal functions. In this paper we have used triangle method to find the membership function. The triangular membership function is specified by three parameters [a,b,c] as follows:

$$\text{Triangle}(x:a,b,c)=\begin{cases} 0 & x < a \\ (x-a)/(b-a) & a \leq x \leq b \\ (c-x)/(c-b) & b \leq x \leq c \\ 0 & x > c \end{cases}$$

We assume that the number of labels for each input attribute is 4, i.e., ZE, PL, PM, PH. In [20], the membership functions of the input attributes depth, Sand, Silt, Clay, Sandbysilt, Sandbyclay, Sandbysiltclay, can be defined as shown in Figure 1, Figure 2, Figure 3,Figure 4, Figure 5,Figure 6 and Figure 7, respectively.

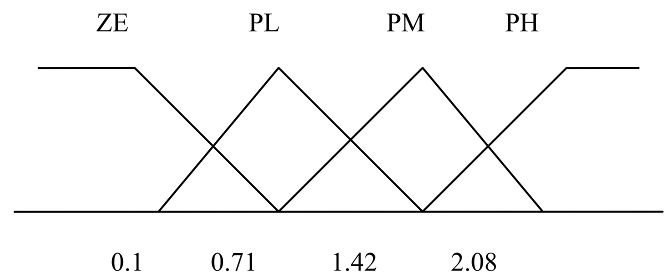

Figure 1: Membership functions of the input attribute Depth.

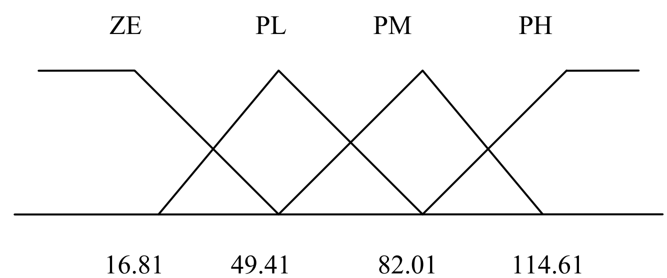

Figure 2: Membership functions of the input attribute Sand.

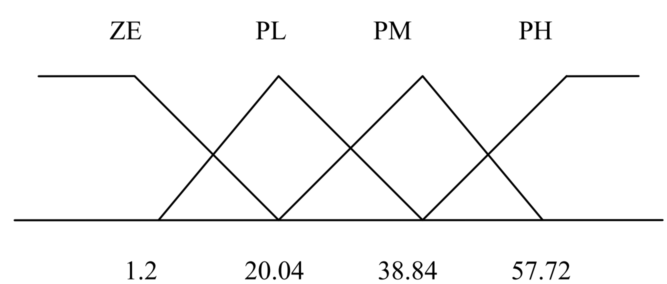

Figure 3: Membership functions of the input attribute Silt.

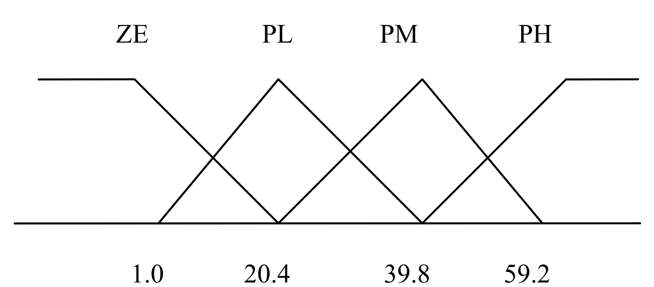

Figure 4: Membership functions of the input attribute Clay.

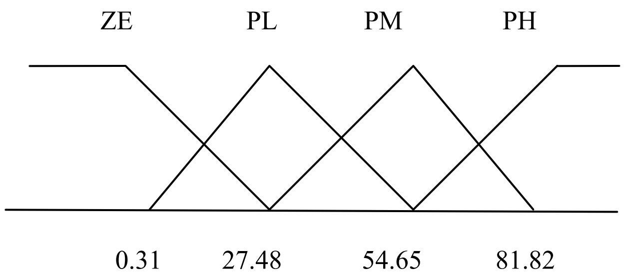

Figure 5: Membership functions of the input attribute SandbySilt.

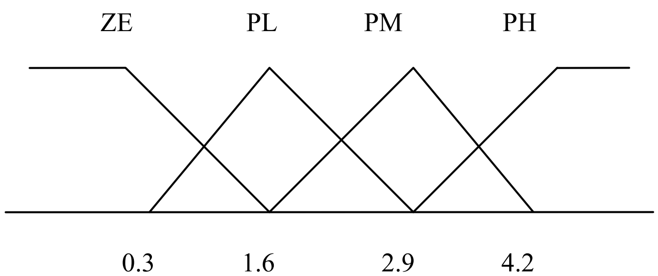

Figure 6: Membership functions of the input attribute SandbyClay.

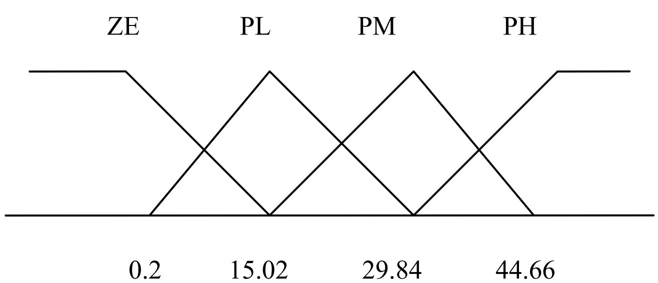

Figure 7: Membership functions of the input attribute SandbySiltClay.

4. Algorithm

In the following section, we present an algorithm to generate fuzzy rules from a set of training data. The algorithm is now presented as follows:

Step 1: Convert each training data in the initial training data set into a fuzzy rule and put them into the set of initial rules.

Step 2: If the set of initial rules is empty or all of the fuzzy rules in the set of initial rules have been taken **then Stop**;
else take a fuzzy rule *R* form the set of initial rules.

Step 3: If the set of definitive rules is empty or all of the fuzzy rules which are in the set of definitive rules that have the same output with fuzzy rule *R* have attempted to merge with fuzzy rule *R* **then** fuzzy rule *R* becomes one member of the set of definitive rules;
else go to Step 4.

Step 4: Take a fuzzy rule *R'*which has the same output with fuzzy rule *R* and has not attempted to merge with fuzzy rule *R* from the set of definitive rules;
If the merge of fuzzy rule *R* with fuzzy rule *R'*is allowed **then** merge them into fuzzy rule *R"*and replace *R'*by fuzzy rule *R"*, and go to Step 2
else go to Step 3.

The set of definitive rules is produced by the above four steps. Now, we use those fuzzy rules that are in the set of definitive rules, for classification. First, we convert the testing datum into labels, and then we test if it is subsumed in the fuzzy rules that are in the set of definitive rules.

5. Experimental Results

In this section, we have chosen the defined membership functions for the input attributes and we convert each training datum into a fuzzy rule, respectively which is implemented in programming language C. We proceed by taking a training datum ((0.17, 91.4, 6.85, 1.75, 13.34, 3.91, 10.62)) from initial training data. The value of the attribute Depth is 0.17, we map the value "0.17" into the membership function for the input attribute depth and check with the label PL we can see that the membership value is 1; similarly we map the value "0.17"into the membership ZE, PM and PH respectively, and we can see that the membership values are 0, respectively. We can see that when we map the value "0.17"into the membership function PL, we get the largest membership value, so we convert the value "0.17" into "PL". In the same way, we can convert ((0.17, 91.4, 6.85, 1.75, 13.34, 3.91, 10.62), 1) into (({ZE}, {PH}, {ZE}, {ZE}, {ZE}, {PH}, {PL}), 1). Repeating the above steps, we can convert the initial training data into the set of initial rules as shown in Table 3.

Type 1
{{ZE}{PH}{ZE}{ZE}{ZE}{PH}{PM}{PL},1}
{{PL}{PH}{ZE}{ZE}{PM}{PM}{PM},1}
{{PM}{PH}{ZE}{ZE}{PM}{PM}{PH},1}
{{PM}{PH}{ZE}{ZE}{PM}{PM}{PH},1}
{{PH}{PM}{PH}{ZE}{ZE}{PH}{PM}{PH},1}
Type 2
{{ZE}{PL}{PM}{PM}{ZE}{ZE}{ZE},2}
{{PM}{PL}{PM}{PM}{ZE}{PL}{ZE},2}
{{PL}{PL}{PM}{PM}{ZE}{PL}{ZE},2}
{{PM}{PL}{PM}{PM}{ZE}{PL}{ZE},2}
{{PM}{PL}{PM}{PM}{ZE}{PL}{ZE},2}
Type 3
{{PL}{PL}{PM}{PH}{ZE}{ZE}{ZE},3}

{{PM}{ZE}{PM}{PH}{ZE}{ZE}{ZE},3}

Type 4

{{PM}{PL}{PM}{PM}{ZE}{ZE}{ZE},4}
{{PL}{ZE}{PM}{PH}{ZE}{ZE}{ZE},4}
{{PL}{ZE}{PM}{PH}{ZE}{ZE}{ZE},4}
{{PM}{PL}{PM}{PH}{ZE}{ZE}{ZE},4}
{{PH}{ZE}{PL}{PH}{PM}{ZE}{ZE}{ZE},4}

Type 5

{{ZE}{PM}{ZE}{PL}{ZE}{ZE}{ZE},5}
{{PM}{PM}{PM}{PL}{ZE}{PL}{ZE},5}
{{PM}{PM}{PM}{PL}{ZE}{PL}{ZE},5}
{{PH}{PM}{PM}{PL}{ZE}{PL}{ZE},5}
{{PH}{PM}{PM}{PL}{ZE}{PL}{ZE},5}

Type 6

{{PM}{PL}{PM}{PM}{ZE}{ZE}{ZE},6}
{{PH}{PL}{PM}{PM}{ZE}{ZE}{ZE},6}
{{ZE}{PL}{PM}{PM}{ZE}{PL}{ZE},6}
{{PH}{ZE}{PM}{PM}{ZE}{ZE}{ZE},6}
{{PH}{PM}{ZE}{PH}{PM}{PM}{ZE}{PM}{ZE},6}

Type 7

{{PL}{PM}{PM}{PM}{ZE}{PL}{ZE},7}
{{PL}{PL}{PM}{PM}{ZE}{PL}{ZE},7}
{{ZE}{ZE}{PH}{PM}{PM}{ZE}{PM}{ZE},7}

Type 8

{{PL}{PM}{PM}{PM}{ZE}{ZE}{ZE},8}
{{ZE}{PM}{PM}{PM}{ZE}{PL}{ZE},8}
{{PL}{PM}{PL}{PL}{ZE}{ZE}{ZE},8}
{{ZE}{PL}{PM}{PM}{ZE}{PL}{ZE},8}
{{ZE}{PL}{PM}{PM}{ZE}{PL}{ZE},8}

Type 9

{{PH}{PM}{PM}{PL}{ZE}{ZE}{PH}{ZE},9}
{{ZE}{PH}{PL}{ZE}{ZE}{PM}{ZE},9}
{{PL}{PH}{ZE}{ZE}{ZE}{PL}{ZE},9}
{{PL}{PM}{PL}{ZE}{ZE}{PH}{PM}{ZE},9}
{{ZE}{PH}{PL}{ZE}{ZE}{PM}{ZE},9}

Type 10

{{ZE}{PM}{PM}{PM}{ZE}{PL}{ZE},10}
{{ZE}{PM}{PM}{PL}{ZE}{PL}{ZE},10}
{{PL}{PM}{PL}{PM}{ZE}{ZE}{ZE},10}
{{ZE}{PM}{PL}{PM}{ZE}{ZE}{ZE},10}
{{ZE}{PM}{PL}{PM}{ZE}{ZE}{ZE},10}

Table 3. Fuzzy rules in the set of initial rules.

Then, we take the fuzzy rules sequentially from the set of initial rules shown in Table 5 and perform the merging process. Finally, we can get a set of definitive fuzzy rules from the set of initial rules. There are 10 fuzzy rules in the set of definitive rules, shown as follows:

R0:
{{ZE,PL,PM,PH,}{PH,PM,}{ZE,PH,}{ZE,}{ZE,PM,}{PH, PM,}{PL,PM,PH,},1}
R1:
{{ZE,PM,PL,}{PL,}{PM,}{PM,}{ZE,}{ZE,PL,}{ZE,},2}
R2: {{PL,PM,}{PL,ZE,}{PM,}{PH,}{ZE,}{ZE,}{ZE,},3}
R3:
{{PM,PL,PH,}{PL,ZE,}{PM,PL,}{PM,PH,}{ZE,PM,}{ZE,}{ZE,},4}
R4:
{{ZE,PM,PH,}{PM,}{ZE,PM,}{PL,}{ZE,}{ZE,PL,}{ZE,},5}
R5:
{{PM,PH,ZE,}{PL,ZE,PM,}{PM,ZE,}{PM,PH,}{ZE,PM,}{ZE,PL,PM,}{ZE,PL,PM,},6}
R6:
{{PL,ZE,}{PM,PL,ZE,}{PM,PH,}{PM,}{ZE,PM,}{PL,ZE,}{ZE,PM,},7}
R7:
{{PL,ZE,}{PM,PL,}{PM,PL,}{PM,PL,}{ZE,}{ZE,PL,}{ZE,},8}
R8:
{{PH,ZE,PL,}{PM,PH,}{PM,PL,ZE,}{PL,ZE,}{ZE,}{ZE,PM,PL,PH,}{PH,ZE,PM,},9}
R9:
{{ZE,PL,}{PM,}{PM,PL,}{PM,PL,}{ZE,}{PL,ZE,}{ZE,},10}

Now, we apply those fuzzy rules to deal with the classification. We use an instance (0.73, 37.15, 18.25, 44.0, 2.07, 0.41, and 0.6) of the Soil data as a testing datum to illustrate the classification process. First, we convert this testing datum into ({PL}, {PL}, {PL}, {PM}, {ZE}, {ZE}, {ZE}). We take this converted datum into all of the generated fuzzy rules, respectively, and then we can see that only "R3: {{PM,PL,PH,}{PL,ZE,}{PM,PL,}{PM,PH,}{ZE,PM,}{ZE,}{ZE,},4}" can subsume in it (because {PL} ϕ{PM,PL,PH},{PL}ϕ{PL,ZE}, {PL}ϕ{PM,PL}, {PM}ϕ {PM,PH},{ZE}ϕ{ZE,PM}, {ZE}ϕ{ZE}, and {ZE}ϕ{ZE}). Thus, the classification result indicates that it is belonging to Type 4, i.e., Soil Texture is C (Clay). From Table 1, we can see that it is a correct classification.

Further Modification was done to the same program which could accept input attributes and generates fuzzy rule that specifies the type of the texture class also.

6. Conclusion

In this paper, we have presented a method to generate fuzzy rules from input attributes to deal with the

Soil data classification problem. We have implemented this in Turbo C Build Version 3.0.

The implementation was done with two approaches.

In the First approach, we convert the training data into initial set of fuzzy rules, and then we merged those initially generated fuzzy rules sequentially one after the other in order to reduce the number of fuzzy rules. Then finally testing datum can be taken to test the generated fuzzy rules.

In the second approach, we have modified the first program in such a way that it accepts input attributes and generates the final rule that also states the type of the texture class.

The second approach is more effective than the first approach, as first approach works well only for predefined set of training data and generates initial set of rules only for that data and again a program is to be run to merge these rules whereas the second approach eliminates the need of generating initial rules and merging the initial rules and also eliminates the need of mapping to which texture class it belongs to and further more to say that the program could be run to accept input attributes any number of times and immediately generates the fuzzy rule with the type of texture class.

7. References:

[1] S. M. Chen, S. H. Lee, and C. H. Lee, A new method for generating fuzzy rules fromnumerical data for handling classification problems Appl. Art. Intell. vol. 15 no. 7 pp. 645 – 664, 2001.

[2] Wang, L. X. and Mendel, J. M. 1992. Generating fuzzy rules by learning from examples. IEEE Transactions on Systems, Man, and Cybernetics, 22, 6: 1414-1427.

[3] Yaochu Jin, Member, IEEE, Werner von Seelen, and Bernhard Sendhoff, On generating FC Fuzzy Rule Systems from Data Using Evolution Strategies, IEEE Transactions on Systems, Man, and Cybernetics,- Part B: Cybernetics, Vol. 29, No. 6, December 1999.

[4] A Thesis by D.Basavaraju Characterisation and classification of soils in Chandragiri mandal of Chittoor district,Andhra Pradesh.

[5] A thesis titled “Characterization, Classification and Evaluation of Soil Resources in Sivagiri Micro- Water Shed of Pichatur Mandal, Chittoor District, Andhra Pradesh”.

[6] A thesis titled “Morphology and Taxonomy of Soils in Ramachandrapuram Mandal of Chittoor District, Andhra Pradesh”.

[7] Bart Kosko. Neural Networks and Fuzzy Systems. A Dynamical Systems Approach to Machine Intelligence, PHI.

[8] Castro, J. L., Castro-Schez, J. J., and Zurita, J. M. 1999. Learning maximal structure rules in fuzzy logic for knowledge acquisition in expert systems. Fuzzy Sets and Systems, 101, 3:331-342.

[9] F. Höppner, F. Klawonn, R. Kruse, and T. Runkler, Fuzzy Cluster Analysis, Wiley, Chichester, 1999.

[10] George J. Klir and Bo Yan. Fuzzy Sets and Fuzzy Logic. Theory and Applications, PHI.

[11] Hayushi, Y. 1992. Fuzzy neural expert system with automated extraction of fuzzy if-then rules from a trained neural network. In “Analysis and Management of Uncertainty: Theory and Applications” (edited by Ayyub, B. M., Gupta, M. M., and Kanal, L. N.), North-Holland, Amsterdam: 171-181.

[12] John Yen and Reza Langari. Fuzzy Logic Intelligence, Control, and Information, Pearson Education Asia.
No. 18.

[13] R. Kruse, J. Gebhardt, and F. Klawonn, Foundations of Fuzzy Systems, Wiley, 1994.

[14] Rajasekaran, S. and Vijayalakshmi Pai, G. A. Neural Networks,Fuzzy Logic, and Genetic Algorithms. Synthesis and Applications, PHI.

[15] Soil Survey Staff 1951, Soil Survey Manual. US Department of Agricultural Hand book

[16] Soil Survey Staff 1998, Keys to soil taxonomy. Eight Edition, Natural Resource Conservation Services, USDA, Blacksburg, Virginia.

[17] Yuan, Y. and Shaw, M. J. 1995. Induction of fuzzy decision trees. Fuzzy Sets andSystems, 69, 2: 125- 139.

[18] Yuan, Y. and Zhuang, H. 1996. A genetic algorithm for generating fuzzy classification rules. Fuzzy Sets and Systems, 84, 1: 1-19.

[19] Zadeh, L. A. 1965. Fuzzy sets. Information and Control, 8: 338-353.

[20] Chen, Y. C. and Chen, S. M. 2000. A new method to generate fuzzy rules for fuzzy classification systems. Proceedings of the 2000 Eighth National Conference on Fuzzy Theory and Its Applications, Taipei, Taiwan, Republic of China.

[21] H. L. Lin and S. M. Chen, Generating weighted fuzzy rules from training data for handling fuzzy classification problems Proc. 2000 Int. Computer Symp.: Workshop ArtificialIntelligence pp. 11 - 18, 2000.

Bhargavi Peyakunta is working as Associate Professor in the Department of Computer Science and Engineering, Madanapalle Institute of Technology and Science, Madanapalle, Andhra Pradesh. **Educational Qualifications:** M.Sc in Computer Science from Sri Krishnadevaraya University, Anantapur and M.Tech degree from Sri Vinayaka missions University, Salem, India. **Teaching & Research Experience:** 12 years of teaching experience & 5 years of research experience. **Current Research Interests:** Data Mining, Fuzzy Systems, Genetic Algorithms and GIS.

Jyothi Singaraju is working as Head & Associate Professor in the Department of Computer Science, Sri Padmavathi Mahila Visvavidyalayam(SPMVV), Tirupati. **Educational Qualifications:** M.Sc in Applied Mathematics from S.V.University, Tirupati , M.S in Software Systems from BITS, Pilani, & Ph.D in Theoritical Computer Science from S.V.University, Tirupati. **Teaching & Research Experience:** 16 years teaching experience & 24 years research experience. **Current Research Interests:** Fuzzy Systems, Neural Networks, Data Mining, Data Base Management Systems, Genetic Algorithms, Bioinformatics and GIS.

This page is intentionally left blank