



SCHOOL OF COMPUTATION, INFORMATION
AND TECHNOLOGY

DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics

**Exploring Fuzzy Tuning Technique for
Molecular Dynamics Simulations in
AutoPas**

Manuel Lerchner



SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY

DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics

Exploring Fuzzy Tuning Technique for Molecular Dynamics Simulations in AutoPas

Untersuchung von Fuzzy Tuning Verfahren für Molekulardynamik-Simulationen in AutoPas

Author: Manuel Lerchner
Supervisor: Univ.-Prof. Dr. Hans-Joachim Bungartz
Advisors: Manish Kumar Mishra, M.Sc.
Samuel Newcome, M.Sc.
Date: 10.08.2024

I confirm that this bachelor's thesis is my own work and I have documented all sources and material used.

Munich, 10.08.2024

Manuel Lerchner

Contents

Acknowledgements	vii
Abstract	ix
Zusammenfassung	xi
1 Introduction	1
1.1 A	1
2 Theoretical Background	2
2.1 A	2
3 Implementation	3
3.1 A	3
4 Proof of Concept	4
4.1 Creating the Knowledge Base	4
4.1.1 Decision Trees	4
4.1.2 Fuzzy Decision Trees	5
5 Comparison and Evaluation	5
5.1 A	5
6 Future Work	6
6.1 A	6
7 Conclusion	7
7.1 A	7
8 Demo	8
8.1 Tips	8
8.1.1 How to Describe	8
8.1.2 How to Quote	8
8.1.3 How to Math	8
8.2 Environments	9
8.2.1 How to Figure	9
8.2.2 How to Algorithm	9
8.2.3 How to Code	11
8.2.4 How to Table	11

A Some more stuff	12
Bibliography	15

4 Proof of Concept

In this chapter, we present a proof of concept for the fuzzy tuning technique. We will develop a set of linguistic variables and fuzzy rules to predict the optimal configuration parameters for `md_flexible` simulations.

4.1 Creating the Knowledge Base

One of the hardest parts of developing a fuzzy system is creating the knowledge base, as it typically requires a very deep understanding of the system to be able to create meaningful rules. However, there also exist methods to use a data-driven approach to create the knowledge base automatically. This is especially useful as those methods don't require any prior expert knowledge about the system. But regardless of the way the knowledge base is created, it is still possible to manually evaluate and adjust the rules to add manual expert knowledge to the system. Using such data-driven methods can be a good starting point for creating a fuzzy system, as it can provide a good initial set of rules that can be further refined by experts.

There are several methods to automatically create and tune fuzzy systems based on data. Some of the most common methods include genetic algorithms, particle swarm optimization, and decision trees. In this work, we will use a decision tree approach proposed by Crockett et al. [CBMO06] to create the knowledge base for the fuzzy system. This proposed method uses machine learning to first train a classical decision tree on the dataset and then converts the decision tree into a fuzzy decision tree which can then be used to extract the linguistic variables and fuzzy rules.

Add references to the methods

4.1.1 Decision Trees

Decision trees are very popular machine learning algorithms that are used for classification and regression tasks. They are defined by recursively partitioning the input using axis-parallel splits [Mur12]. There are several algorithms to train decision trees, such as ID3, C4.5, CART, and many others, but they all follow the same basic principle of splitting the input space in such a way that the resulting subsets are as pure as possible, thereby minimizing the impurity of the subsets. The impurity of a subset is typically measured using a metric such as the Gini impurity, entropy, or the mean squared error.

Check if this is correct

Decision trees are known for their interpretability and can be also be easily visualized. An example of a decision tree is shown in Figure 4.1. The interpretability of decision trees makes them a good choice for creating the initial knowledge base for a fuzzy system, as it is very easy to understand and refine the rules created by the decision tree with additional expert knowledge.

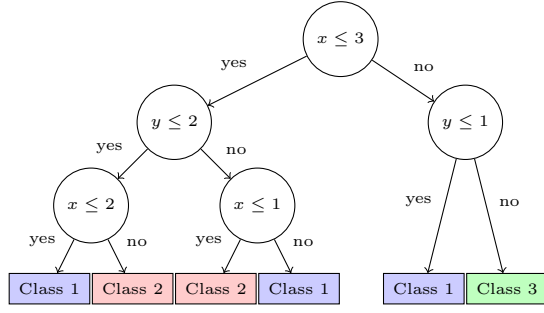


Figure 4.1: An example decision tree for a dataset with two features x and y . There are three distinct classes in the dataset

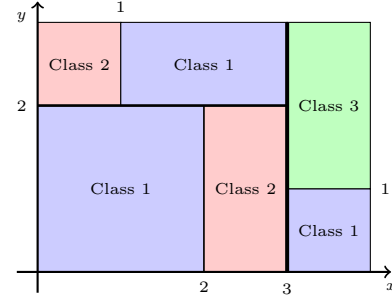


Figure 4.2: The decision boundary of the decision tree from Figure 4.1 on $\mathcal{D} = [0, 4] \times [0, 3]$.

4.1.2 Fuzzy Decision Trees

Fuzzy decision trees are a generalization of classical decision trees that allow for fuzzy logic to be used in the decision-making process. This extension allows to eliminate the crisp decision boundaries of classical decision trees and instead use fuzzy sets at each node of the tree to calculate the contribution of each branch to the final decision. This allows for a more flexible decision-making process that can take into account the uncertainty of the input data and the splits.

Contrary to classical decision trees, which follow a single path from the root to a leaf node, fuzzy decision trees explore all possible paths at the same time and make a final decision by aggregating the results of all paths using fuzzy logic. This is possible, as each node in a fuzzy decision tree can fuzzily assign how much each of its children should contribute to the final decision.

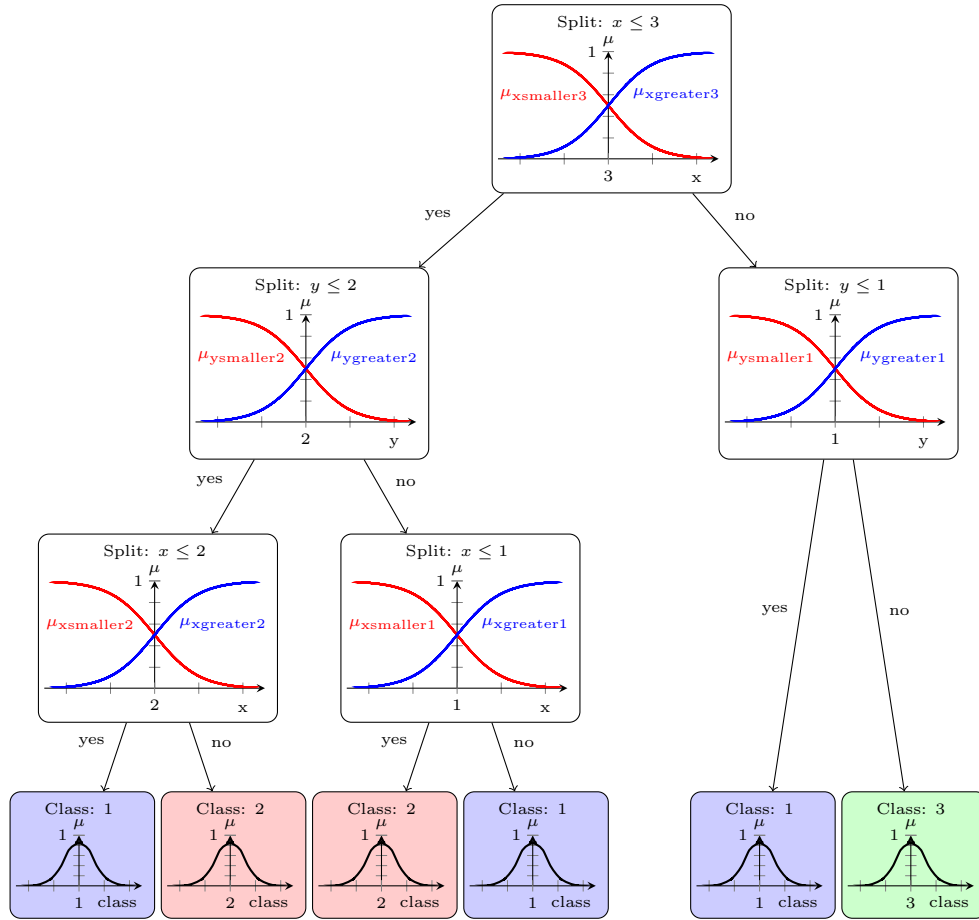


Figure 4.3: The fuzzy decision tree corresponding created from the decision tree in Figure 4.1.

Each internal node in the fuzzy decision tree uses two **sigmoid** membership functions (μ_{smaller} and μ_{greater} with appropriate parameters) to specify to which degree the comparison is true or false. The leaf nodes use different **gaussian** membership functions centered around the class value.

Note that the individual membership functions can be chosen arbitrarily and don't have to be restricted to the ones shown here.

Rule Extraction

Once the fuzzy decision tree has been created, the next step is to extract the fuzzy rules from the tree. This can be done by traversing the tree in a depth-first manner and collecting the correct membership functions for the path along the way. Each connection between two internal nodes in the tree corresponds to a **AND** operation, while each final connection between an internal node and a leaf node corresponds to an **IMPLIES** operation.

List of Figures

4.1	Example Decision Tree	5
4.2	Decision Boundary	5
4.3	Example Fuzzy Decision Tree	6
8.1	Example Figure	9
8.2	Figure with tikz	9
8.3	One caption to describe them all.	9
8.4	some description what is happening	10

List of Tables

8.1 Some Table 11

Bibliography

- [CBMO06] Keeley Crockett, Zuhair Bandar, David Mclean, and James O'Shea. On constructing a fuzzy inference framework using crisp decision trees. *Fuzzy Sets and Systems*, 157(21):2809–2832, 2006.
- [Gra17] Fabio Alexander Gratl. Task based parallelization of the fast multipole method implementation of ls1-mardyn via quicksched. Master's thesis, Institut für Informatik 5, Technische Universität München, Garching, November 2017.
- [Mur12] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.