Analysis

Q1: Modifying Mamdani's Inference System to Prioritize "Golden Rules"

Experts have identified that some rules are significantly more important than others, often referred to as "golden rules." To address this issue in the Mamdani inference system, the concept of rule weights can be introduced. Rule weights allow us to assign different levels of importance to each rule. By assigning higher weights to golden rules, their impact on the final decision is increased, ensuring that these critical rules have a stronger influence on the output.

In practice, this can be implemented by multiplying the degree of fulfillment of each rule by its corresponding weight during the aggregation step. For instance, if a golden rule has a weight of 2 and another less important rule has a weight of 1, the system will prioritize the golden rule when calculating the aggregated fuzzy set. This approach helps in giving more significance to the critical rules without disregarding the less important ones completely.

Q2: Understanding the Lack of Extreme Risk Values

The observation that extreme risk values are never obtained could be due to several factors in the design of the Mamdani inference system:

1. Rule Design: The rules may not cover the scenarios that lead to extreme risk values. If no rules explicitly describe conditions leading to very high or very low risks, those extreme values will not be produced.
2. Membership Functions: The membership functions of the input variables and the output variable (Risk) might be designed in a way that they do not reach extreme values. For example, if the membership functions for the output variable "Risk" are primarily centered around medium values, the final risk scores will also tend to be medium.
3. Aggregation and Defuzzification: During the aggregation and defuzzification steps, the combined effect of the rules might be averaging out the results, leading to less extreme values.

To address this, we can ensure that the rules explicitly cover extreme cases. Additionally, adjusting the membership functions to have more pronounced peaks at the extreme values can help. The maximum value that can be obtained with the current system is determined by the range of the output variable "Risk" and how the rules and membership functions are defined. If the membership function for "High Risk" peaks at 100, then 100 is the maximum risk value that can be achieved.

Q3: Scaling the System to Handle More Requests

With Banco Pichin's acquisition by a larger bank, the system needs to handle hundreds of additional requests quickly without any hardware upgrades. Given that all possible software optimizations have already been implemented, one effective strategy is to parallelize the processing of requests.

1. Parallel Processing: Utilize multi-threading or multi-processing techniques to process multiple loan applications simultaneously. This can significantly reduce the time taken to handle a batch of requests.
2. Batch Processing: Instead of processing each request individually, group multiple requests and process them in batches. This can optimize the use of resources and reduce the overall processing time.
3. Asynchronous Processing: Implement asynchronous processing where the system can start processing the next request before the current one is fully completed. This can be achieved using asynchronous programming libraries available in Python, such as asyncio.
4. Load Balancing: Distribute the requests evenly across multiple instances of the inference system (if multiple instances are allowed within the same hardware). This ensures that no single instance becomes a bottleneck.

By implementing these strategies, the system can meet the increased demand and process the requests within the required timeframe, ensuring smooth operations even with the increased load.