Math 301 (Game Theory) - Final Project

The Use of Utility Theory in Artificial Intelligence

December 05, 2023

Dhruv Patel (301385233)

Vraj Patel (301384968)

Simon Fraser University

Table of Contents

1. Introduction to Utility Theory

Artificial Intelligence is widely used in our daily lives, from our smart phones to complex decision-making systems used in various technological industries. An important concept that plays a huge role in making decisions is by the use of utility theory in AI. It provides a mathematical background to understand how AI systems can make choices based on multiple options. In this report, we look at how it works and its significant importance in decision making by providing a few practical examples.

1.1. What is Utility Theory?

Utility Theory in AI is a framework that is used to model decision making under uncertainty. An individual ranks their choices depending on their needs or preferences. This approach explains how an agent's preferences are made by selecting from a set of various available options when an agent faces uncertainty about which way they should follow (Yang & Liu, 2023). At the minimum, an agent has a specific goal that it needs to maximize, known as the **utility** (Yang & Liu, 2023).

1.2. What is a Utility Function?

In Game theory, a utility function for a given player assigns a number of every possible outcomes with the probability that a higher number suggests the outcome that is preferred.

In AI, the concept is similar, an agent's preferences are captured by a Utility function where a single number is assigned to express the desired outcome of a state.

If there is a level of uncertainty in the preferences made by the agent, the utility is known as the **expected value** of its utility function. From this, the relationship between an agent (decision maker) and its preference, the agent would rather choose a more preferred course of action or strategy than one than is less preferred.

Another concept is the **maximum expected utility** (MEU) that states a rational agent should choose a method that gives the maximum outcome based on the agent's expected utility (Russell & Norvig, 2021, p. 611).

1.3 Lottery

To understand the concept of the use of Utility theory in AI, let's take a look at an example of a simple lottery.

Suppose we have two choices:

1) $1000

2) 50% chance of $3000

Which option would we choose?

Our decision would be based on risk, finance and a personal preference. This is the same way an agent uses to base it's options mathematically using utility theory.

Researchers say that the satisfaction from winning $3000 is less than the one from $1000. We can construct a scale to understand this. Let's say we assign 0 to winning nothing, and 100 to winning $3000. What number would be given to $1000? Suppose we choose 80. This indicates that the difference between what we have now and $1000 is four times greater than the difference between $1000 and $3000. We can express this as utilities instead of dollars. For option A, 80% utilities is guaranteed. For option B, it 50% chance of the 100 utilities. From this we can calculate the expected utility which is the probability times the utilties. Based on the expected utility, we tend to choose option A (Calculemus, n.d.).

A lottery L with possible outcomes S_1, ... S_n. with probabilities p_1, ... p_n is represented as $L = [p_1, S_1; p_2, S_2; .... ; p_n, S_n]$ (Russell & Norvig, 2021, p. 612)

2. Mathematical Framework

2.1. Von Neumann-Morgenstern Utility Theorem

According to Von Neumann & Morgenstern, people's preferences for different outcomes are not just about which outcomes they prefer but also about how much more they prefer one outcome over another, such as the choice between the certainty of one thing or a lottery with the possibility of getting something better. The VNM theorem states that decision-makers choosing between risky options will behave as if they are maximizing an underlying utility function, and there are four conditions that must be met for this to be valid (von Neumann & Morgenstern, 2007). These conditions include:

i) Completeness: For all outcomes, A & B players have preferences i.e.,

$$U_i(A) > U_i(B),\ U_i(A) = U_i(B),\ \text{or}\ U_i(A) < U_i(B)$$

ii) Transitivity: Is the idea that

$$\text{If } U_i(A) > U_i(B) \text{ and } U_i(B) > U_i(C), \text{ then } U_i(A) > U_i(C)$$

iii) Continuity: Suggests that value cannot change suddenly; there must always be some probability p, a chance of getting one outcome and a probability of getting another, that makes them indifferent to the choice. For all A, B, C, preferred in that order, there must be a probability p (between 0 and 1) such that

$$U_i(B) = p * U_i(A) + (1-p)*U_i(C)$$

iv) Independence (of Irrelevant Alternatives): Preferences over lotteries must remain consistent regardless of the choices. If one is indifferent over A or 50% Chance of B, and indifferen between B and C, then the player should be indifferent over lottery of A or 50% chance of C.

3. Utility Theory in AI Applications

Utility theory has found widespread applications in various AI domains, including:

3.1. Self-Driving Cars: Navigating the Road with Utility-Driven Decisions

- Self-driving cars have emerged as a groundbreaking technology with the potential to revolutionize the transportation landscape. At the core of their sophisticated decision-making capabilities lies utility theory, a framework that enables these vehicles to make safe and efficient choices in dynamic and complex environments (Dazeley et al., 2021, p. 3).
- Self-driving cars employ utility functions to evaluate the potential consequences of various actions, such as changing lanes, overtaking, or maintaining a steady speed. These utility functions carefully consider factors such as the risk of collisions, the time it takes to reach the destination, and the comfort of passengers (Dazeley et al., 2021, p. 12). By maximizing the expected utility, self-driving cars can navigate roads safely and efficiently, reducing accidents and improving traffic flow.

- For instance, when approaching an intersection, a self-driving car may assess the probability of encountering other vehicles, the time required to clear the intersection safely, and the potential delays caused by waiting. By weighing these factors, the car can make an informed decision about whether to proceed, slow down, or come to a complete stop (Dazeley et al., 2021, p. 12).

3.2. Recommendation Systems: Personalized Suggestions Guided by Utility

- Recommendation systems have become ubiquitous in our digital lives, suggesting products, movies, music, and other items that align with our preferences. Utility theory plays a crucial role in these systems, enabling them to personalize recommendations and enhance user experiences (Tansuchat & Kosheleva, 2022).
- Recommendation systems utilize utility functions to quantify the desirability of different items for each user. These utility functions consider factors such as the user's past purchases, ratings, and browsing history, as well as the popularity of the items and their relevance to the user's current interests. By maximizing the expected utility, recommendation systems can provide personalized suggestions that are likely to appeal to each user's unique tastes and preferences. For example, an e-commerce website might recommend products that a user has previously shown interest in, while a music streaming service might suggest songs that are similar to those the user has listened to in the past (Tansuchat & Kosheleva, 2022).

3.3. Reinforcement Learning: Learning Optimal Strategies through Utility Maximization

- Reinforcement learning is a powerful machine learning technique that enables AI agents to learn how to make decisions through trial and error. Utility theory plays a central role in reinforcement learning, providing a framework for the agent to evaluate the outcomes of its actions and optimize its behavior (Dazeley et al., 2021, p. 9-14).
- Reinforcement learning algorithms use utility functions to guide the agent's interactions with its environment. The agent receives rewards or penalties for its actions, and over time, it learns to select actions that maximize its expected cumulative reward (Dazeley et al., 2021, p. 9-14).
- For instance, a reinforcement learning algorithm might be used to train a robotic arm to perform a complex task, such as assembling a product. The agent would

learn to manipulate the arm's movements to achieve the desired outcome, guided by the utility function that represents the successful completion of the task (Dazeley et al., 2021, p. 9-14).

4. Practical Implementation

To demonstrate the use of Utility Theory in AI, we decided to come up with a practical example that shows how decision-making is made based on various preferences that need to be selected to maximize the outcome. In this example, we provide a simple code written in Python by us specifically for choosing the best route amongst several options. It incorporates different utility functions (linear and logarithmic) to evaluate the desirability of routes based on factors like distance and road quality. We first calculate the expected utility under varying traffic conditions and visually compare the utilities (Code snippet in Appendix on page 10).

4.1. Brief Descriptions of Functions

- utility_function_linear(distance, road_quality): Calculates the utility of a route based on distance and road quality using a linear scale. It reflects a direct, proportional impact of these factors on utility.
- utility_function_logarithmic(distance, road_quality): Determines the utility using a logarithmic scale, giving more significance to shorter distances. This function is sensitive to smaller changes in distance.
- calculate_expected_utility(route, utility_function): Computes the expected utility of a route by accounting for different traffic conditions and their probabilities. It demonstrates how an AI system evaluates options under uncertainty.
- choose_best_route(routes, utility_function): Identifies the route with the highest utility from a set of options, using a specified utility function. It showcases decision-making based on utility maximization.
- plot_utilities(routes, utility_functions): Creates a bar graph to visually compare the utilities of routes under different utility functions. This function aids in understanding how utility assessments vary with different evaluation criteria.

4.2. How it helps in AI

- Expected Utility Calculation: This feature illustrates how AI systems can make informed decisions under uncertainty, a key aspect of utility theory in AI.

4.3. The concept of Linear vs Logarithmic Utility Function

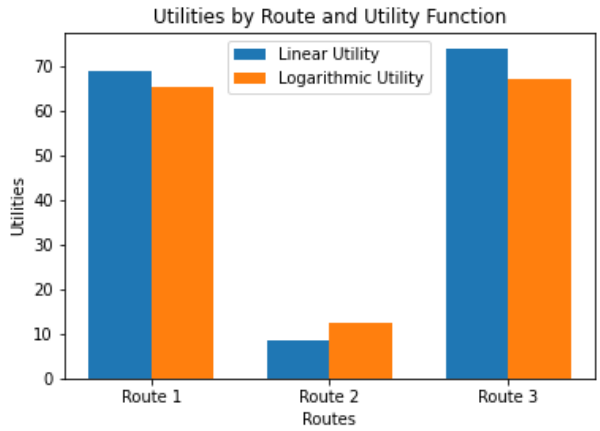
In a linear utility model, changes in input (like distance or road quality) have a constant effect on utility. For example, each additional kilometer reduces utility by the same amount, regardless of the total distance.

In a logarithmic model, the utility decreases more rapidly for initial increases in distance and then levels off. This means the difference in utility between 1 and 2 kilometers is greater than between 45 and 46 kilometers.

In decision-making, the choice between these two depends on how the decision-maker values changes in the attributes (Russell & Norvig, 2021). For instance, if small changes in distance at lower values are significantly more critical (e.g., in dense urban areas where the difference between a 5 km and a 6 km route is significant), a logarithmic function is more appropriate. In contrast, for scenarios where each unit change has a consistent impact, a linear function is more suitable.

The AI's choice of route can vary significantly depending on which utility function it uses. The linear function might prefer longer routes with better road quality, while the logarithmic function might favor shorter routes even if they have slightly worse road conditions.

The graph below is generated from our code, where we defined 3 different routes. We can see how different routes have different utilities based on various conditions. The bar chart shows the linear utility function consistently assigns higher utility values compared to the logarithmic utility function, illustrating the logarithmic function's greater sensitivity to shorter distances.



In summary, the linear utility function implies a consistent valuation of changes, while the logarithmic function implies a diminishing sensitivity to changes as values increase.

5. Summary

Utility theory is a powerful tool for understanding and modeling decision-making under uncertainty. It involves assigning numerical values, called utilities, to different outcomes and selecting the option that maximizes the expected utility. It has become an essential tool in AI, enabling the development of intelligent systems that can make informed decisions in complex and uncertain environments. Utility theory has found widespread applications in various AI domains, including self-driving cars, recommendation systems, and reinforcement learning. As AI continues to evolve, utility theory will play an increasingly important role in ensuring that AI systems make decisions that are aligned with our values, objectives, and ethical considerations.

6. References

Brilliant. (n.d.). *Utility functions*. https://brilliant.org/wiki/utility-functions/

Calculemus. (n.d.). *Expected value, expected utility and multi-attribute utility theory*. https://www.calculemus.org/lect/L-I-MNS/06/exp-ut1.html#:~:text=The%20solution%3A%20Expected%20utility%20theory,quantify%20the%20amount%20of

Dazeley, R., Vamplew, P., Foale, C., Young, C., Aryal, S., & Cruz, F. (2021). Levels of explainable artificial intelligence for human-aligned conversational explanations. *Artificial Intelligence*, *299*, 103525. https://doi.org/10.1016/j.artint.2021.103525

Eintalu, J. (2019, December 9). *Some simple utility functions*. Medium. https://eintalu.medium.com/some-simple-utility-functions-cc04c545ce02

Martins, T. G. (2014, January 22). *Declining marginal utility and the logarithmic utility function*. Wordpress. https://tgmstat.wordpress.com/2014/01/22/declining-marginal-utility-and-the-logarithmic-utility-function/#:~:text=,notion%20of%20declining%20marginal%20utility

Russell, S., & Norvig, P. (2021). *Artificial intelligence: A modern approach*. Pearson Higher Ed.

StudySmarter. (2023). *Utility functions*. https://www.studysmarter.co.uk/explanations/microeconomics/consumer-choice/utility-functions

Tansuchat, R., & Kosheleva, O. (2022). How to make recommendation systems fair: An adequate utility-based approach. *Asian Journal of Economics and Banking*, *6*(3), 308-313. https://www.emerald.com/insight/content/doi/10.1108/AJEB-03-2022-0031/full/pdf

Von Neumann, J., & Morgenstern, O. (2007). Theory of games and economic behavior (60th Anniversary Commemorative Edition). Princeton university press.

Yang, Q., & Liu, R. (2023). Understanding the application of utility theory in robotics and artificial. https://arxiv.org/pdf/2306.09445.pdf

7. Appendix

```python
import matplotlib.pyplot as plt
import numpy as np
def utility_function_linear(distance, road_quality):
    # Linear utility function
    return 100 - distance - (50 if road_quality == 'poor' else 0)
def utility_function_logarithmic(distance, road_quality):
    # Logarithmic utility function for more emphasis on lower distances
    return 100 - np.log(distance + 1) * 10 - (50 if road_quality == 'poor' else 0)
def calculate_expected_utility(route, utility_function):
    # Assuming hypothetical probabilities for different traffic conditions
    traffic_conditions = {"light": 0.3, "moderate": 0.5, "heavy": 0.2}
    expected_utility = 0
    for condition, probability in traffic_conditions.items():
        # Adjusting utility based on traffic condition (less utility in heavy traffic)
        adjusted_distance = route["distance"] * (1.2 if condition == "heavy" else 1)
        expected_utility += probability * utility_function(adjusted_distance, route["road_quality"])
    return expected_utility
def choose_best_route(routes, utility_function):
    best_route = None
    max_utility = -float('inf')
    for route_name, route_info in routes.items():
        utility = calculate_expected_utility(route_info, utility_function)
        if utility > max_utility:
            max_utility = utility
            best_route = route_name
    return best_route, max_utility
def choose_best_route_and_print(routes, utility_function, utility_function_name):
    best_route, max_utility = choose_best_route(routes, utility_function)
    print(f"Best route using utility function ({utility_function_name}): {best_route} with utility
{max_utility}")

def plot_utilities(routes, utility_functions, filename="route_utilities.png"):
    labels = list(routes.keys())
    linear_utilities = [calculate_expected_utility(routes[route], utility_functions[0]) for route in
routes]
    logarithmic_utilities = [calculate_expected_utility(routes[route], utility_functions[1]) for route
in routes]
    x = np.arange(len(labels))
    width = 0.35
    fig, ax = plt.subplots()
    rects1 = ax.bar(x - width/2, linear_utilities, width, label='Linear Utility')
    rects2 = ax.bar(x + width/2, logarithmic_utilities, width, label='Logarithmic Utility')
    ax.set_xlabel('Routes')
    ax.set_ylabel('Utilities')
    ax.set_title('Utilities by Route and Utility Function')
    ax.set_xticks(x)
    ax.set_xticklabels(labels)
    ax.legend()
    plt.savefig(filename)
    plt.show()
# Define routes
routes = {
    "Route 1": {"distance": 30, "road_quality": "good"},
    "Route 2": {"distance": 40, "road_quality": "poor"},
    "Route 3": {"distance": 25, "road_quality": "good"}
}
# Utility functions
utility_functions = [utility_function_linear, utility_function_logarithmic]
# Display process and results
print("Calculating best routes using utility theory...")
choose_best_route_and_print(routes, utility_function_linear, "Linear")
choose_best_route_and_print(routes, utility_function_logarithmic, "Logarithmic")
# Plot utilities
plot_utilities(routes, utility_functions, filename="route_utilities.png")
```