What is a utility-based agent?

Utility-based agents, defined

A utility-based agent is an intelligent system that uses a utility function to make rational decisions by maximizing the expected utility of possible outcomes. The utility function mathematically predicts utility of all the potential actions the artificial intelligence (AI) agent can take.

The goal of a utility-based agent is to maximize the utility function with each action. To further its goal, the AI agent uses the results of the utility function to choose the next most beneficial action.

Utility-based agent components

The primary components of a utility-based agent are:

- Utility function
- Sensors
- Internal model
- Action selection mechanism
- Actuators

Utility function

A utility function is a mathematical equation that represents how the agent should assess the benefit of any possible actions it can take. It is essentially the values system of the agent and represents how the agent prioritizes relevant factors when making choices.

Utility functions assign a numerical value to each outcome of a potential action, quantifying the preferences that the agent should maintain. Utility-based agents use the utility function to negotiate complex environments, weigh tradeoffs and maximize the utility of their choices.

A good utility function factors multiple considerations, such as safety, efficiency, resource allocation and multi-objective opportunity costs. The utility function is the keystone of a utility-based agent and is what separates them from different types of AI agents.

Sensors

Utility-based agents use sensors to perceive their real-world environments. Sensors can be physical, such as cameras and thermometers, or digital, such as API connections and simulations. Agentic AI perception uses complex algorithms to filter environmental data and isolate the most important and relevant data points for informed decisions.

Internal model

Utility-based agents maintain a simplified internal model of their real-world environment. The model is created and updated based on the data perceived by the agent's sensors. By tracking environmental data over time, the internal model can also infer unobservable data about the agent's environment.

State-transition models

Many utility-based agents use a state-transition model of the world, which establishes the possible states for the environment and the criteria for when and how it changes from one to another. State-transition models showcase how a system or dynamic environment can change over time. More advanced state-transition models calculate the probability of the environment changing its current state at any time.

The utility function assigns a value to each state, and the agent aims to move the environment into the future state with the highest utility. State-transition models are especially useful in stochastic or dynamic environments, where the agent must reason about probabilities rather than certainties.

Action selection mechanism

The action selection mechanism is the AI decision-making component of the agent. Based on the current state of the internal model, the agent generates a list of all potential actions it can take. Action selection algorithms use the utility function to evaluate all the different actions and optimize the agent's choice for maximum overall benefit.

In some modern implementations, large language models (LLMs) are used to reason about complex, high-level goals or interpret ambiguous inputs before converting them into structured utility calculations.

Actuators

Actuators, or performance elements, allow the agent to act on its environment. Physical actuators can be a robotic arm on a manufacturing line, a thermostat controlling the temperature of a smart home or an entire autonomous vehicle. Virtual or digital actuators can be an API connection, chatbot interface or software output.

The utility-based agent workflow

Utility-based agents share a standard internal workflow that guides their behavior:

1. Perception

2. Internal modeling

3. Action generation

4. Outcome prediction

5. Utility assessment

6. Action selection

7. Action

1. Perception

The agent uses its sensors to perceive its environment and collect data. This data is used to inform the agent as to its own state and the current state of its environment.

2. Internal modeling

Using the current perceptions from its sensors, the agent updates its internal model of its environment in real time. This gives the agent an understanding of its surroundings and any relevant factors that affect its decision-making process.

3. Action generation

The agent uses search and optimization algorithms to generate a list of potential actions it can take, based on the state of its internal model. Action generation and selection techniques encourage the agent to consider novel ideas and referencing past experiences with proven results to maintain reliable performance.

In practice, many agents don’t explicitly generate a full list of possible actions. Instead, they use optimization or reinforcement methods to evaluate the likely best actions within a continuous action space.

4. Outcome prediction

For each action generated in the previous step, the agent uses its state-transition model to predict the expected outcome. The model calculates the probability of a certain state being reached when the agent takes a specific action.

5. Utility assessment

The agent's action selection mechanism applies the utility function to each generated action and associated probable outcome. The function returns a numerical utility score for each possible choice. Higher scores represent greater overall utility.

6. Action selection

The agent selects the action leading to the outcome with the greatest overall benefit, as determined by the preferences of the utility function. Because the agent's goal is to maximize its utility function, the action selection process leads the agent to act in a way the furthers the goals of the AI system in which the agent is being used.

7. Action

The agent selects the action leading to the outcome with the greatest overall benefit, as determined by the preferences of the utility function. Because the agent's goal is to maximize its utility function, the agent acts in a way that furthers the goals of the AI system in which the agent is being used.

Utility-based agent use cases

Utility-based agents are ideal for complex tasks with multiple competing directives. These can include:

- **Smart homes:** Utility-based agents can power the intelligent systems in a smart home, weighing priorities such as comfort, energy costs and sustainability.
- **Self-driving cars:** Autonomous vehicles pose a range of complex problems for machine learning engineers. Agent-controlled cars must deal with human drivers, pedestrians, obstacles, weather, road closures and many other conditions in a dynamic environment. Problem-solving in this setting requires a well-crafted utility function.
- **Healthcare:** Because they can juggle various considerations in pursuit of maximum benefit, utility-based agents might be able to help with formulating treatment plans and managing costs.
- **Robotics:** Robots also need to weigh various factors in pursuit of maximum benefit. Delivery bots have many of the same considerations as autonomous vehicles.
- **Recommendation and pricing systems:** The utility function lets the agent weigh factors such as the user's preferences, time of day and year and larger trends to keep the user entertained. In generative AI systems, a utility-based approach can guide the generation of content that best matches user intent, context and long-term engagement goals.
- **Pricing systems:** Similarly, utility-based agents can manage dynamic pricing systems to maximize purchases and revenues for a business.
- **Logistics and supply chain automation:** Complex supply chains must balance factors such as efficiency, costs, risk, quality and more. Enterprises can tailor a logistics agent's utility function to prioritize the factors most important to their business and create a scalable system.

Utility-based agents versus goal-based agents

Utility-based agents and goal-based agents are both useful in situations where the agent must work toward a long-term outcome. However, the difference is that while utility-based agents seek to maximize the utility of their choices, goal-based agents pursue specific goals. They are motivated by goal achievement.

While a goal-based agent treats all goal-achieving states as equally desirable, a utility-based agent can differentiate between them by degree, allowing for more nuanced decision-making. Utility-based agents can manage multiple conflicting goals and maintain performance even in the face of uncertain outcomes.

Benefits of utility-based agents

Utility-based agents are resilient, able to navigate complex problems and changing environments while delivering consistent results. The benefits of utility-based agents include:

- **Adaptability:** Utility-based agents use fluid utility functions rather than fixed rule-based systems such as condition-action rules. They can adapt to changing conditions and new tasks where lower-level simple reflex agents and model-based reflex agents might struggle against their rigid rule-based programming.

- **Flexibility:** Utility-based agents can successfully juggle competing priorities to make decisions that still lead to good outcomes. Goal-based agents are singularly focused on one specific goal and might struggle to consider other directives.

- **Reliability:** Utility functions lead utility-based agents to make rational decisions even when outcomes are uncertain. The decisions these agents make are likely to be the ones that lead to more beneficial long-term outcomes.

Utility-based agent limitations

While utility-based agents are capable in many settings, they aren’t always the best choice. Their limitations include:

- **Rigidity:** Without a learning element, utility-based agents cannot learn from their actions and update their utility functions and state-transition models autonomously. Adding a learning element allows them to improve through reinforcement learning, but defines them more formally as hybrid agents or even learning agents.

- **Computational demands:** Utility functions are complex algorithms, and running them continuously requires significant compute and energy. Without sufficient compute, utility-based agents might be too slow for real-time use in time-sensitive situations.

- **Complexity:** Effective utility functions are difficult to design, and a utility-based agent is only as effective as its utility function. Machine learning engineers must successfully

convert their values system into a numerical equation that forces the agent to make appropriate choices.

- **Ethical considerations:** Utility-based choice poses an ethical concern. Who determines the values system for the agent—especially, as with self-driving cars—if it is an agent with the ability to potentially harm humans? As intelligent agents become more autonomous and pervasive, it is critical to define who determines their underlying value systems, and how those values align with societal ethics.

To overcome some of these challenges, utility-based agents are often integrated into multi-agent systems, where several specialized agents collaborate, share information and balance competing objectives. In such architectures, each agent's utility function contributes to the system's collective optimization strategy.