Grundlagen der Wissensverarbeitung
Blatt 1

Daniel Speck, Lena Niermeyer, Florian Klückman

16.10.2015

Aufgabe 1.1 (Application Scenarios for Artificial Intelligence)

1. Navigating through a labyrinth

Task: An agent has to find a way through a labyrinth.
Result: The found solution should be as efficient as possible, so the found path through the labyrinth has to be as short as possible (either in terms of the time it takes to finish or the distance of the path).
Examples: Pacman[1], a classic computer game, could be considered as a sufficient example, because Pacman has to find a path through a labyrinth, avoid the ghosts and get all coins as fast as possible (to achieve a high score). Therefore his paths have to be as efficient as possible.
Computational solutions: A computational solution would be fine as long as finding the shortest path is the only objective. A computational solution would be hard to implement or fail, if many secondary objectives (such as a high score, avoid ghosts, ...) come into play. Additionally a classic computational solution has a certain space and time complexity, so it would always fail if there is not enough memory or time for computing the whole algorithm. Potentially an artificial intelligence could estimate if there are enough resources (time/memory) for finding an optimal solution or if a less optimal solution has to be found because the decision (e.g. where to go? stop now? go now?) has to be made right now.

2. Calibration of sensor data

Task: Real world data recorded from sensors often contains noise and other distortions of input data. A supervised agent could be trained to find patterns in those distortions and deliver more distortion-free data.
Result: Ideally the agent learns if any calibration issues exist, if a typical noise on data can be found and so on in order to deliver input data which is as distortion-free as possible.
Examples: A robot with a camera to perceive its environment could have a sensor reporting the angle of a rotatable substructure for aligning the camera. If the sensor is not calibrated correctly it may always report 5 degrees too much or too less. Furthermore the glass of the camera protecting its image sensor could be polluted so that the view of the camera gets blurry or contains

[1] `https://de.wikipedia.org/wiki/Pac-Man`

contortions. Those abnormalities could be learned by an artificial intelligence for trying to compensate the error. A supervised artificial neural network would be one possible solution.
Computational solutions: A fix, computational solution could most likely not be used in this scenario, because the hassle to implement such a solution by hand would result in a high cost (time, money, ...). An intelligent agent which is capable of re-adapting if the measurement error for example of the sensor of the rotatable substructure changes is an appropriate solution.

3. Image classification

Task: Much information is meaningless if it is not accessible easily for humans or computers. Large databases of images (e.g. Google images) have to be categorized correctly to build up a useful database. An agent could be used to classify this data automatically.
Result: If the image of an object, e.g. "a dog", is searched, the agent should optimally only show images of dogs. If the search is being refined, e.g. "a white dog", the agent has to be able to connect those information with each other, interpret the search query and narrow down the results so that only white dogs are shown.
Examples: Google images, Bing images, Yahoo Image Search, ...
Computational solutions: It is nearly impossible to use classic computational solutions for building up clusters of images or even narrowing down search results dependent on visual information. The immense number of possible shapes, lighting, angles, ... renders this task very challenging. A computational solution would be far to static to adapt to real world images.

Aufgabe 1.2 (AI Terminology)

Information is a subset of *knowledge* created as well as spread via sharing. Machines and computer programs may be senders or receivers of information. The world wide web contains information in the form of messages, texts, encyclopedias, blogs, libraries and so forth. Not all information is *explicit knowledge*, but *explicit knowledge* consists of "saved", well formulated information.

Implicit knowledge is unconsciously, it can not be or has not been articulated. An individual "carries" *implicit knowledge* with its skills and abilities. It is (logically) deducible from observable information or behavior and it implies itself with the existence of such information, behavior, skills, abilities, ...
A popular German saying is "zwischen den Zeilen lesen" which means "read between the lines", this saying indicates the existence of hidden knowledge, *implicit knowledge*, which has to be discovered by observation and/or deductions. Examples for skills and abilities would be bicycling, swimming or socialization in communities. *Implicit knowledge* can also be more technical, e.g. in the form of data in data warehouses or databases of companies. "Here" the *implicit knowledge* would not be the data itself (which would be considered *explicit knowledge*) but conclusions which could be drawn out of this information. A concrete example could be the logging of each access of a specific information with the lack of further interpretations of this logged data (so it would remain a simple counter).

Explicit knowledge is articulated information in a well formulated structure which could be texts, articles, formulas, algorithms, computer programs, rules, laws, diagrams and other forms of held knowledge. A book of mathematics for example holds formulas and therefore unambiguous information.

Fully observable versus *partially observable*:
In a fully observable environment the agent is always informed sufficiently to make an optimal decision. In other environments agents must rely on experience or memories in order to make the best decision. Checkers e.g. is fully observable, while poker is only partially observable. If you want to design an algorithm for these Games for avoiding intuitive decisions and make logical ones, then in checkers for each position on the board (which are visible at all times) a reaction by the algorithm is / can be clearly defined. In poker the hand/cards of other players are unknown and because of this lack of information no optimal solution can be found. Reason, why the difference might be important when designing AI applications for the given environment: If there are unobserved states, it is much harder to learn a good policy. Unobserved states are a lot more difficult to learn a good policy.

Discrete versus *continuous*:
Discrete means that different values have different meanings. In chess, where you have an 8x8 array representing different values: 1 = pawn, 2 = rook, 3 = bishop .... 2 then is something else than 1 and does not mean 2x pawns. Continuous means that the values have the same meaning, but different amounts. For example, pixel values (RGB): 0 means "no light" (black) and 255 means "maximum amount of light" (white). The problem in a discrete environment is that you have to think abstract, but most things in nature are continuous (temperature, health, etc.) and not as simple. On the other hand, abstraction allows manageability of things (light switch: on, off is simple, dimmable would be a continuous solution and requires further decisions making everything more complex). Reason, why the difference might be important when designing AI applications for the given environment: The difference is important as many statistical and data mining algorithms can handle one type but not the other. For example in regular regression, the Y must be continuous. In logistic regression the Y is discrete.

Deterministic versus *stochastic*:
Deterministic describes that the same input results in the same output every time. The outcome is defined by mathematical laws. In stochastic environments probability values are part of the calculation, so that the same input can result in different outputs. Crosswords e.g. are deterministic, while medical diagnoses are stochastically. Problem with determ. Environment: inflexible and further calculation with wrong results, because the system is so rigid. Problem at Stoch. Environment: ambiguous. For planning and follow-up steps precise values/information is helpful, but through probability values much data can not be used for further processing. Reason, why the difference might be important when designing AI applications for the given environment: Modeling and simulating a stochastic process is more difficult and in complex systems you may have to neglect some variables to be able to use the model.