Module 1 - AI Problem Solving

DAT410
24-01-2023
Group 8

Dylan Osolian Lisa Samuelsson

We hereby declare that we have both actively participated in solving every exercise. All solutions are entirely our own work, without having taken part of other solutions.

Predict the temperature

- For the next weekend
- For the same date as today but next year

We would model this problem as a function, where the function would take a number of parameters and give us a predicted temperature. The parameters could be many different things but could for example be: the current month, current temperature, which location we want to predict the temperature in, what time of the day we want to predict for and also some data about the temperature of previous years. All of these parameters contain information which could help give an accurate prediction of the temperature. We therefore think that the problem could be modeled as a function and be solved with statistics/machine learning. Since there is statistics and data with these parameters a model could be trained to give good predictions based on the data.

When we think about the problem in general we recognize that the complexity of the model can be different, there could be both very simple models and more complex models. A simple model for predicting the temperature for the same date as today but next year could be just predicting the same temperature as it was today whereas a more complex model would consider more parameters.

Thinking about the characteristics of the problem we realized that it can be solved by hand. Most of the problems in the assignments are hard to solve for a human however we think that humans can in most cases give a satisfactory prediction of the temperature, although a complex model might be more accurate. We also understand that this problem, as most problems, can be modeled in many different ways. We thought about how the weather system could be modeled as a dynamic system and with the help of simulation the temperature for next weekend could be predicted. However this would only work for predicting the temperature in the near future.

Bingo lottery problem

A characteristic of this problem is that it has an optimal solution, and that this solution would be extremely difficult for a human to find. We believe that this problem could be modeled as a constraint satisfaction problem. There are a number of “rules” for the problems, which are the constraints. We for example know how many tickets there are, that no ticket should give more than one win, and how a winning combination is received. We also have a lot of data in the form of knowledge of all lottery tickets.

Ideally, 120 tickets are given a win, so we want to find a solution as close to this as possible. To solve this problem it should be possible to use constraint programming, where the rules of the systems are the constraints.

Public transport departure forecast

We think that this problem could be modeled as a function. The next departure time depends on a number of parameters, such as the current location of the vehicle, the traffic situation and the weather. These parameters could be inserted into a function to give a prediction of the next departure time. This is a form of regression, where you try to predict a value based on a set of parameters.

To find a departure time you could also look at previous situations with similar conditions, so we think that you could use statistics and machine learning to solve this problem. Given current conditions, as well as information about the past, you could calculate a probable departure time. For example, there might be data showing that snowy weather leads to later departure times than planned. If it is currently snowing, then you could combine this information about the past with the current state to predict the next departure time. A machine learning model that is trained on data from past situations could be used. If you have data on the current parameters (such as weather, position and traffic situation), you could insert this into the model to make a prediction.

This problem is one where you want to make a prediction about the future. A human who also has information about for example the planned departure time, the traffic situation, and the weather could also make a guess at the next departure time. However, using a computer, it is possible to use a very large amount of data and perhaps find patterns that humans can't identify.

Film festival problem

This problem can be modeled as a discrete algorithmic problem, more precisely we believe that this can be modeled as an interval scheduling problem. The films would be the tasks/intervals in the model and the selected priority for a film from the user would give the intervals a specific weight. The problem would then be solved and would give the user an optimized schedule. If the problem is modeled in this way we know it can be solved in polynomial time with dynamic programming since it is a weighted interval scheduling problem. The solution is guaranteed to be optimal based on the users priority of the films.

We think that an interesting characteristic of this problem is that for small inputs this problem can be solved by humans. If we for instance have a very small film festival with only a couple of films it is not too hard to optimize our schedule to be able to see our favorite films, however when the input becomes large it becomes a much harder task. In that case, it is beneficial to use a computer to solve the problem.

Product rating in consumer test

In this problem, we assume that we have a number of parameters that we consider when we should determine a score for some product. If we for example want to determine the score of a dishwasher, things that affect the score could be how loud the dishwasher is, how long a program takes, how clean the dishes get, etc. These different factors should then be combined into a single score. What we need here is some way to weigh the different factors to determine the final score. If some factor is more important than the other, then it should have a larger impact on the score.

One characteristic of this problem is that there is no clear “correct” solution. There are numerous ways to select this score, using a variety of different strategies. We think that a way to “solve” this problem is by prescriptive modeling. The problem could be modeled as a function, using prescriptive modeling. We can compare this to how BMI is calculated, where the two parameters weight and height are combined to get a numerical value. For this product rating, we could define a formula that calculates a final score, and simply input the values for the different parameters we have measured in the test.

Constraint satisfaction and constraint programming

Briefly explain the main concepts.

Constraint satisfaction problems are problems which have a set of variables, a set of domains for the variables and a set of constraints. The goal is to find an assignment for every variable that satisfies all constraints. There are many areas where you can model problems as constraint satisfaction problems, and one common example is map coloring. If you have a map with a number of districts or countries, and you want to color the map so that no neighboring countries have the same color, then this is a constraint satisfaction problem. In this example the countries of the map would be the variables, the domain would be the different colors and we would have the constraint that no neighboring country should have the same color. The goal is then to give each variable a color.

In constraint satisfaction the idea is to check for feasibility rather than optimality, meaning that we want to find a solution that satisfies all constraints but not necessarily an optimal solution. Coming back to the example of map coloring, there could be multiple ways of coloring the map, but the importance lies in finding any feasible solution. You can use constraint programming to solve constraint satisfaction problems. To do so, you define the variables, their domains and the set of constraints. Constraint programming can use different techniques to solve the problems, such as backtracking, dynamic programming and local search.

Summary - Intro AI problem solving, Lisa Samuelsson

The first lecture of the course started with an introduction to artificial intelligence. In the first part of this introduction we got a historical review of the topic, starting with discussions of the Turing machine. The Turing test and an important quote from Turing himself was mentioned, as well as a quote from computer scientist John McCarthy. McCarthy had a very positive outlook on the future of artificial intelligence, and stated in 1956 that “significant advance” could be made in a single summer. However, these types of swift advances proved to be difficult to obtain, and in the lecture we learned that there instead were big successes in other areas of computing, such as in hardware.

The lecture further discussed the differences between basic computing and advanced computing. Different types of AI methods as categorized in a book by Russell and Norvig were shown. This provided an overview of some common methods, many of which I have heard of and some that I have used in previous courses. It also showed how many things are actually included in the topic of AI. Finally, we got some examples of areas where AI can be applied to solve problems.

After this part of the lecture came an introduction to the course itself, that contained an overview of the different modules. We also got information about how a module is structured. The lecture ended with a brief introduction about the first module.

Summary - Intro AI problem solving, Dylan Osolian

The first lecture of the course contained practical information about the course, and course staff but also an introduction to AI and its history. The story of Alan Turing and the Turing machine was told. The Turing machine can be seen as the first model of a computer and is the first mechanism which can do all kinds of computing. It is often seen as the base or introduction to computer science and AI. The lecture also described problem solving with different kinds of computing, starting from basic computing to advanced computing. How problems can be modeled and then solved by different levels of computing.

The first lecture discussed what intelligence is and also mentioned different areas where AI is used, some of which we will discover during the course.

Mainly the first lecture was an introduction to the course and some historical background of the topic.