DIT728 Assignment 1

Group 64
Module 1

Jonatan Hellgren, 960727-7010
Matematikprogrammet GU
gusheljot@student.gu.se

Jens Ifver, 920528-2552
Matematikprogrammet GU
gusifvje@student.gu.se

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

January 2021

Hours spent working

Jonatan	Jens
10	10

Remark about Jens Ifver from Jens Ifver

I'm currently in contact with CSE student office and are (hopefully) soon registered on the course. I've been instructed to participate in the course meanwhile, until I get my verdict.

1 Predict the temperature

1.1 For the next week

A function that takes into account the temperature for the last couple of months and use statistics to construct a linear regression model that predict the temperature. This is particularly a good choice if you where to predict the average temperature of the days next week. Since the days closest in time probably say more about the temperature next week, it could be beneficial to increase the weight for these observations.

1.2 For the same date as today but next year

Similar to the solution for the next week, but the data that is used should include historical temperatures of the date where the most recent years will have higher weights. A good idea is to not only include the date we are trying to predict but also include some data from the days close to the given date, for example all the temperatures in a ten day interval around the date we are trying to predict.

2 Bingo lottery problem

By dividing the numbers in to three categories: drawn, not drawn and not available we can construct an algorithm that sample one number from a list of the not drawn numbers and places it in the list of drawn numbers. When a number is drawn there are two constraints that need to be satisfied for the number to be acceptable. First is that the sum of the number of bingo generated from the drawn number and the previously generated number of bingo are less than 120. Second is that the drawn number will not generate a double bingo. If one of the numbers in not drawn violates one of these constraint it will be placed in the not available list, where it will not be drawn. We note that it this method will not be certain to give exactly 120 wins. If you want exactly 120 wins one could iterate the method until you get exactly 120 wins and use the numbers obtained.

So the solution becomes a mixture between a algorithm and a constraint satisfaction problem.

3 Public transport departure forecast

A function could be a model for estimating the time until departure. One possibility is to have the whole city divided into segments where every route consists of a collection of segments. Estimated time will be a sum of the estimated time of each segment leading up to the stop. The function will take into account the gps-location of the vehicle, the route, the specific stop and the time of day.

Based on historical data one can use a regression model for each segment that estimates the time to travel that segment. This model could take into account the parameters described above.

4 Film festival problem

This could be seen as a constrained optimization problem, where we try to minimize an objective function describing a sum of missed movies multiplied with a penalty. The penalty comes from each individuals priority list, where a higher priority gives a higher penalty. Problem could be solved with linear programming. Examples of constraints could be: all movies needs to be shown at least once and a movie can only be shown at one place at a time.

5 Product rating in consumer test

A simple model that computes the mean of consumers rating for each product. If some consumers are considered more reliable based on a more extensive knowledge in a related subject, weights could be used to accentuate these ratings.

6 Constrain satisfaction and constraint programming

6.1 Briefly explain the main concepts

A constrain satisfaction problem consists of a set of variables which can take on certain values. These variables can then be constrained in different ways to take on a subset of values that is specified in the problem. If every variable has taken on a value that does not violate any constraints, these values are considered a solution to the problem.

An example of this is the map coloring problem, where each country can be colored a certain way from a limited set of colors. A valid solution is one where two adjacent countries doesn't have the same color. Here the variables become the countries and the color their value, the constraint is that two adjacent countries can't have the same color.

Summary: Design of AI-system Introduction, by Jens

In the lecture on the 18th of January, Dag Wendelin started by presenting a historical overview of the concept of AI. Wendelin used the Turing machine as a starting point but mentions that the mathematics behind had already been discovered a while before. The Turing machine was a simple machine that was believed to have no limitations in what it could calculate. Since this machine could compute almost anything it was thought that it could also be or become intelligent. The inventor of the machine Alan Turing though that it would be consensus to think of machines as a thinking entity at the end of the century. It seemed to be a general trend among the people involved in the first computers, to think that you can, with good enough scientist, define any kind of problem detailed enough to be able to program a computer to solve it. Today we know that this is not exactly the path that the evolution of AI took.

Wendelin further explains that what went wrong was that the problem solving relied a lot more on the programmers insight then originally thought and definition of problems appeared to be harder than expected. The branch of problems that AI did succeed in was game playing. In game playing the rules and logic needed to describe the game was simple enough for programmers to apply them in their algorithms and derive results that was ground braking. Wendelin lifts an example of image recognition of dogs, pointing out that it is very hard to define the characteristics of a dog in a way a computer will be able to distinguish a dog from for example a cat. These kinds of obstacles where not expected in the early days.

Later in the lecture Wendelin discuss different types of advanced computing and divide them into three categories. In the first one containing methods like algorithms and rule-based AI, and is characterized by heavy reliance on human insight and very little or no use of data. Second containing methods like simulation where there is a more balanced mix between the human knowledge and the use of data. The third category is characterized by very little human reliance which is compensated by a heavy use of data. In this third category methods of machine learning are included.

Through out the lecture Wendelin emphasize the importance of choosing the right method for the problem at hand and that different problems need different approaches. The most complicated method is not always the best method, even if it may look good or is more intriguing to apply. Wendelin therefore advice that you should never forget the simpler methods and a good way of understanding a problem is to try out the simpler methods first. There is generally nothing to loose from this as you may already then have found the way to work with the problem or you at least know that a simple method is not working.

Summary, design of AI-system introduction by Jonatan

The lecture was aimed at giving an perspective on AI by talking about the history of the subject. It also gave an introduction about the course, this I will not write a summary about.

To give an perspective of the subject Dag talked about the history of AI. He began talking about the early computers, mainly Turing machines. An interesting thing he brought up was that before the computer was invented humans likened their brains to a telephone switch, and now after we usually liken it to a computer. This suggest that we see things in the patterns we already know, that the brain tries to make sense of things by relating it to something similar, even the view of it self. Which is ofcourse similar to how machine learning methods work, they try to predict what is happening by analysing previous knowledge or "data".

Later he bagan to talk a bit more of the early version of AI in the 1950s, he brought up a quote from John McCarthy at the Darthmouth Workshop 1956. In the quote John was very optimistic and thought the scientist wold be able to make progress in the field if a carefully selected group of scientists would work on in together for a summer. What they managed to accomplish was mostly rule-based AI, they didn't manage to handle task such as language. The main reason for their failure was that it was mostly based on human insight on how the problem should be solved, this became an limiting factor since humans aren't that great at handling numbers.

So AI got less attention and more work was focused on other areas of computing, which we did tremendous progress in. With these advances computers became more reliable at doing basic tasks such as writing, storing and sending data. When these basic things became more reliable there was a higher incentive to make further progress in the field of AI. It doesn't make much sense to make advanced forms of AI if we aren't even able to distribute it.

In the later parts of the lecture Dag talked about different methods to solve intelligent task and that there isn't a well defined line between what is deemed to be intelligent or not. Advanced computing is mostly about writing the algorithms to solve certain tasks, but without a computer to run them on it would be near impossible to solve them. My view on the matter is that computers are able to compliment humans with the lack of logical thinking and computing power our brain offer. This mainly since the most successful versions of AI is a symbiosis between human insight and computer power.