Module 1

Group: 12
Anton Claesson, 971104-3217, MPDSC, canton@student.chalmers.se, ~8h
Lukas Martinsson : 980203-9678, MPDSC, malukas@student.chalmers.se, ~8h

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:
We assume the objective of this question is to predict the temperature in terms of the local weather. We note that temperature and weather is based on physical properties of the real world. In the near term, temperature could be assumed to depend on properties such as wind direction and strength, air pressure, humidity, season, composition of the ground, closeness to the sea and much more. This is a dynamic system. We believe it is likely that this problem can be solved by measuring current conditions in the local and surrounding area, after which the near-time weather/temperature might be predicted using a mathematical model of the dynamic system. This is most likely done using computer simulation as there are extremely many parameters which might influence the weather that needs to be considered for an accurate result.
Prediction using computer simulation of the modeled dynamic system only works in the near-term, as errors will be propagated and grow exponentially the further into the future we want to simulate/predict and uncertainty grows.
Note that we in this case have quite a fuzzy line between a dynamic system modeling-approach and a data science approach. Partly because any model we create probably needs to be back-tested on historical data to verify that it is a somewhat reasonable model. Parameters of the models also need to be inferred through prior knowledge and/or historical data. We also require measurements of current conditions.
Alternatively, a deep learning model for time-series data such as satellite imagery over time might be possible to use.

For the same date as today but next year
As mentioned previously, we can't use a model of a dynamic system this far into the future as the uncertainty will be too large. It seems impossible to predict the temperature of a certain day with high accuracy this far into the future. The best approach would probably be to utilize a statistical approach to model the different weather outcomes based on historical data points, for example taking the average temperature of that week over the last 10 years. One could use a Bayesian approach to introduce uncertainty into the model. A more advanced system would probably not do much good in this case as there is simply no way to include all the required information to for example make an accurate simulation based on a dynamic system model. However, one problem that should be considered is that the weather is evolving, and thus data from too far back is either not relevant or only marginally relevant for the prediction of the data today if measurements are assumed to be independent.

For this reason, another model that could be considered is observing trends in the weather, for example if historically a cold summer is usually followed by a warmer winter or even more short-term, if rainfall of a certain amount during the spring affects the summer in any way. This could possibly be statistically modeled by some sort of markov chain.

Bingo lottery problem

We notice that we know all the 1000 lottery tickets, but still we can’t be sure that there will be exactly 120 tickets that satisfy the constraints for any combination of numbers we pick due to the random placement. In fact, to illustrate this, note that since the placement is random in theory all the 1000 tickets might be equal. We instead need a solution which comes as close as possible to the goal of 120 winning tickets.
This means the problem could probably be solved by modeling it as a constrained optimization problem and solving it with integer programming. However, modeling the problem as such might be quite hard.

A more suitable approach if the optimal solution is required in an optimal amount of time would be to come up with a clever discrete algorithmic solution. This algorithm would probably be required to “look ahead” to determine how many winners there would be when picking the next number and combining this knowledge with looking ahead to see how many tickets are close to winning (i.e. a certain number of drawn numbers from winning). It seems plausible a dynamic programming algorithm might be able to solve this problem, but perhaps there is a much simpler algorithmic approach as well.

Public transport departure forecast

It could be assumed that some form of scheduling of departures has already been created, so the bus/train won’t leave earlier than scheduled. The problem would therefore be equivalent to predicting the expected delay of departure. If a data-driven approach is taken, it seems as if this problem can be modeled as a regression problem. Assuming historical data is available, we might be able to solve this regression problem through means of statistical models or machine learning models.
Another option is to create a dynamic model of the traffic/transportation system, after which departure can be forecasted through simulation. However, this is computationally expensive and needs to be done every time we want to make a forecast. It might not make much sense unless the transport system we are trying to forecast is so complex that other approaches are unavailable.
One could also create a very simple probabilistic model dependent on the departure time at the previous stop and historical data and hope it will be good on average. However such a simulation would only be reasonable to implement on smaller scale transports, such as at an airport or train station (by smaller we mean fewer departures) since it would probably be too computationally heavy to implement on every bus stop for every bus for example.

Film festival problem

We assume the goal of this problem is to automatically produce a schedule for when the different films should be shown, such that an individual has the possibility to go to a single sequence of movies during the week that together have the highest possible average rating (note: not the individual's own rating). We can assume the films have a maximum number of repeated viewings and that there is a maximum number of films that can be shown at any given time.
If we consider each film’s runtime as an interval placed during the week, and if each interval has a weight equal to the average rating of the film, then the problem occurs to be very similar to a discrete algorithmic problem formulation that is called weighted interval scheduling.
One might also imagine including more constraints into the problem such that the viewers want to watch a minimum number of films, in which case the weight of a sequence of viewings should be adjusted according to how many different films there are in the sequence.
The ordinary weighted interval scheduling problem can be solved using a dynamic programming approach, which is why we initially would try to use a similar approach for solving this problem as well.

Product rating in consumer test

The rating of the dishwashers might be obtained by statistical evaluation of consumer surveys. For example, there could be different categories such as price, satisfaction, loudness etc that the consumers get to rate numerically if they own the dishwasher. If the same approach is taken for different dishwashers the scores could be compared against each other. Unfortunately, it might be hard to remove biases in the data among the consumers which answer the surveys. Also the number of consumers that answer the survey could also vary, which could influence how well represented the opinions on each dishwasher are.
One could also imagine that historical data is available that contains information about each dishwasher such as components, features, price etc. as well as the corresponding consumer rating of the dishwasher, obtained by an approach similar to above.
A machine learning model would then possibly be able to predict the rating of a new line of dishwashers using this historical data if we model it as a classification problem.

Constraint satisfaction and constraint programming

Main concepts:
Constraint satisfaction problems (CSPs) are a set of problems where the goal is to identify feasible solutions on a set of decision variables where the constraints are satisfied[1]. The goal could either be to find a solution, find all solutions or prove that there are no feasible solutions.
Exhaustive search or brute force solutions to CSPs often exhibit exponential time complexity (or worse) since CSPs are usually combinatorial problems, which results in combinatorial explosion[2] of computing time. Therefore these types of problems often require special methods and heuristic methods to find a solution more efficiently.

CSPs are generally solved by means of constraint programming, an area of research that is specifically focused on solving these types of problems[3]. There are three main techniques for solving CSP problems:

- Backtracking[4]
- Dynamic programming[5]
- Local search[6]

The backtracking algorithm could be regarded as the simplest to understand intuitively. It chooses the first possible value of the first variables and continues to the next variable until a constraint is broken or it has found a feasible solution. It is broken, it backtracks and chooses the next possible value. The flaw with this algorithm is that the time it takes to find the solution is quite significant compared to other more sophisticated algorithms.
From our understanding Backtracking and Dynamic programming are quite similar, as both search for solutions using recursion as means of graph/tree-traversal. One major difference is that Dynamic programming recursively solves smaller subproblems for which the solution is memoized, such that if the subproblem were to be encountered again the algorithm can re-use the already computed answer. Due to this property Dynamic programming often yields solutions of lower complexity, even though memory usage will be higher than for Backtracking-solutions.
Local search is a quite different method in comparison. It is based on iteratively improving the solution until all constraints are satisfied. As its name implies, this method might get stuck in local optima.

Generalized constraint programming

---

1 https://en.wikipedia.org/wiki/Constraint_satisfaction_problem
2 https://en.wikipedia.org/wiki/Combinatorial_explosion
3 https://en.wikipedia.org/wiki/Constraint_programming
4 https://en.wikipedia.org/wiki/Backtracking
5 https://en.wikipedia.org/wiki/Dynamic_programming
6 https://en.wikipedia.org/wiki/Local_search_(constraint_satisfaction)

As mentioned on the Wikipedia page for constraint programming, the constraints are not just defined on boolean domains. Examples of other domains are integer and interval domains.

Anton Claesson

Summary: Intro to AI problem solving (18-01-2022)

The first lecture briefly covers the history of AI, computation and statistics and how our understanding of what AI is has changed over time.
Notably, one might categorize different programs and problems into two main categories:

- Basic computing: Basic computing is software that is not typically “smart” in the sense that the software needs to perform some kind of reasoning. It does not attempt to intelligently solve a complex task. It is software that performs traditionally basic things, even though the technology and hardware used is advanced. One example is text processors such as Word where we can edit and write documents with human input.
- Advanced computing: Advanced computing are programs which have a higher degree of complexity, such as simulation, control and robotics, or construction of efficient algorithms and optimization programs for hard problems such as graph traversals, scheduling and so on.It also includes solving problems such as image classification where human intuition and specified rules might not be usable for a computer program, but rather requires the program to learn to perform this task on its own. Classic rule-based AI can also be considered to be part of the advanced computing area and might excel in specific tasks.

Nowadays when we talk about AI, we often refer to a broad range of advanced computing areas. A large part of today's AI systems are based on data rather than human intuition and rules. Such systems are usually classified as machine learning as these types of models learn from the data. Since these systems are working with large quantities of data, much knowledge has been transferred from the data-science and statistics domains which can oftentimes lead to a lot of confusion in regards to the terminology.

Lukas martinsson

Summary: Intro to AI problem solving (18-01-2022)

In the first lecture of the course design of AI systems we got a brief introduction to the topic as a whole. Although the lecture did not delve deep into it, the part of what defines intelligence I found particularly interesting. Through a test called the Turing test a so-called AI could be “fairly” tested compared to an ordinary human to see if it has achieved some definition of intelligence. Further the lecture covered how “AI” which in itself is hard to define can be magnitudes more efficient at solving certain tasks then humans, however at some aspects it can struggle to solve what would be a trivial task for humans, e.g determining if there is a dog in the picture.

The lecture also gave an introduction to basic and advanced computing and how they differ. Basic computing solves tasks that would be considered basic and not require any smart approaches to it. It would for example to handle the textinput in the word program. However, the word auto-correct function is probably an example of advanced computing since it, while theoretically can be based on an enormous dataset, could be more efficient through some sort of algorithm that would qualify as advanced computing.

Furthermore, the lecture explained the subtle but important difference between data sciences and AI, and how closely they correlate. Data science is focused on the prediction based on the data given while AI is instead a sort of system that solves a task efficiently. Therefore data sciences can make use of AI algorithms to make more efficient and precise predictions while AI for example makes use of data sciences to train and validate it’s models.

Lastly the lecture briefly covered different “AI methods” in different areas and gave a short introduction of what the courses as a whole will cover