Design of AI Systems (DAT410)
Assignment 1: AI Problem Solving

Luca Modica
Hugo Manuel Alves Henriques e Silva

January 21, 2024

1 Predict the temperature

The task is to predict the temperature for the next weekend, and for exactly one year off today. The first problem can be seen as a short-term forecasting one whereas the second is a long-term forecasting problem. Both of them can be modelled as a **regression problem**, since our goal is to predict continuous numerical values (the temperature). In stating the solution to the task we also need to consider the following factors, which can influence the model and its accuracy.

- This specific type of predictions heavily relies on historical data, leading to a relevant *data dependency*.
- An important factor to be taken into account while creating the model is how many patterns of weather data can present. An example is the seasonality of the weather.
- We need consider external factors related to the geographical location, from climate change to local weather phenomena.

1.1 Possible Ways to Solve

Statistics and/or Machine Learning

To accomplish the required task, machine learning models such as Random Forest or Gradient Boosting machines can be good choices for 2 main reasons:

- trade off between flexibility and interpretability, along with less cases of overfitting;
- being able to capture complex data patterns.

Also considering the time series related to weather forecast data, good alternatives are Neural Networks and in particular LSTM (Long Short Term Memory) networks; the last model mentioned is based on a deep learning architecture and is particularly useful in handling sequences, such the time series data in this context. In general, neural networks represent a solution to capture even more complex patterns of data, but with a greater overfitting risk.

Simulation

Furthermore, the simulation of weather patterns according historical data and laws related to weather systems can be a good approach. Through simulations we can sample a temperature value at a certain time, as well as it could be a range of values.

Prescriptive Modelling

Even though this type of solution is not directly applicable to weather forecasting, It can be used to understand the impact both real-time and historical data have to an actual prediction, made for example using the machine learning models mentioned above. Moreover, through processes of data collection and analysis, we can also improve a model even further:

- model calibration to represent relationships and the dynamics of a weather system;
- optimize the model for the goal for which we are predicting the temperature;
- supporting the model, by making a decision not only based on the prediction but also on the understanding of the variable used.

2 Bingo Lottery Problem

In the context of modelling a Bingo Lottery, we can represent the task as an optimization problem with specific constraints (the rules of the game). In particular it is a Constraint Satisfaction Problem (CSP): the game rules can be considered as a set of finite constraints over variables, which can be solved by constraint satisfaction methods. Depending on how complex the problem can become to be solved, we can also consider it as combinatorial optimization: instead of using exhaustive search, which can be intractable, specialized algorithms will instead rule out large parts of the search space and approximate the solution. Moreover, in modeling possible ways to solve the task several factors should be taken into account:

- the *uniqueness* of the constraints (for example, each ticket is unique in terms of ordering);
- as mentioned before, the *complexity* due to large number of possible combinations and constraints to satisfy.

2.1 Possible Ways to Solve

Constraint Programming

For a problem where we need to find a solution that meets a number of constraints, Constraint Programming is among the best way to solve it. In particular, we can use constraint solvers to explore and find a possible arrangements of numbers for the tickets.

Optimization Techniques

A good alternative to constraint programming is using optimization techniques, especially integer linear programming. Given an objective function, which in this case is to minimize the deviation from 120 winning tickets while meeting specific constraints, algorithms based on heuristic methods can be used to search for a feasible solution.

Simulation and Random Sampling

A relatively easier approaching to simulate the game by generating ticket combinations, followed by assessing their outcomes against the constraints. However, It can bes less efficient compared to the first 2 methods described.

Algorithm Design

For this particular task, techniques such as backtracking (to search in space of possible solutions) and greedy algorithms (for keeping track of the best solutions) are helpful to list a set of possible solutions that also meet the constraints. Moreover, due to the possible high complexity, dynamic programming can also help in break down the problem in smaller ones and avoid solving possible overlapping sub problems; this results in an increased efficiency.

3 Public transport departure forecast

The task of predicting the next departure time for a means of public transport requires the estimation of arrival times based on different factors. This problem can be tackled as a regression problem (the time until the next departure can be considered as continuous values), but it can also involve characteristics of a dynamic system due to the changes in real time and uncertainties within that public transport network. Moreover, in creating a model to solve this task the following relevant factors should be considered.

- As mentioned before, public transport schedules can be heavily influenced by different factors such as weather, the state of the track or roads they take, or maybe even holidays or when workers go on strikes.
- Departure times also have patterns which they take over time.
- These times depend on real-time data: the position of the vehicle, how much traffic there is or also frequent delays.

3.1 Possible Ways to Solve

Statistics and/or Machine Learning

Machine learning algorithms such as Random Forests and Neural Networks can be the most suitable models for tackling this type of task: this mainly because they can capture complex patterns in the data and lead us to reasonable predictions. In addition, as mentioned in the temperature prediction problem, deep learning models such as LSTM (Long Short-Term Memory) networks and Recurrent Neural Networks are useful also to handle data like time series.

Simulations

Simulations of traffic patterns according to historical data and also different conditions of the transport network incorporating real time traffic data could lead us to good and informed predictions.

Optimization Techniques

Optimizing the network by calculating alternative routes when events, instead of the fastest path a bus can take will provide us with optimized schedules that can make it easier to predict the arrival of public transports with better accuracy.

Dynamic System Analysis

As this type of task deals with data that is constantly changing over time, models such as queuing theory or dynamic network flow models could help predicting the behaviour of transport systems. Models such as these can capture variations in passengers demands during on and off peak hours, as well as they can take into consideration different routing strategies to improve the efficiency of the transport service. They also take into account traffic jams or congestion in parts of the network.

Historical Data Analysis

Analyzing historical data is one useful way of getting an insight on how things have happened in the past. Realizing the repetition of patterns such as delays allows for adjustments for predictions, especially in real time.

4 Film festival problem

This problem can be viewed as an optimization problem, with focus on scheduling in the best way possible for each customer. In particular, it can be viewed as a constraint satisfaction problem, where the constraints can include the length of films, the priority each person gives them and their availability. To further analyze a problem of this nature, the following points are essential.

- We need to consider the fact that some films overlap each other and the impossibility of watching more than one film at once.
- Despite that, every film is shown multiple times during the week so the overlapping at a certain time should be fixed.
- As stated, we also need to take into account the preferences and priorities given by the customers.
- Lastly, we must be sure the capacity of every film room is not exceeded.

4.1 Possible Ways to Solve

Constraint Programming

With a similar approach as the one used in the Bingo Lottery Problem, also in this case Constraint Programming represent a suited choice: constraint solvers can be used to handle multiple constraint to find a schedule to maximize the number of high-priority films, considering screening times and overlaps.

Optimization Techniques

Defining an objective function that maximize the weighted sum of attended films (based on user preferences) and is subjected to the screening times constraints, the problem can be considered as an optimization one. Integer Linear programming, used along with an Heuristic algorithms, can solve the optimization problem and search for a good-enough solution in a reasonable amount of time.

Algorithm Design

We can use the following algorithm design techniques as approach to this problem:

- Greedy algorithm to pick the best options at each step, providing optimal solution really quickly even though not always optimal;
- dynamic programming to break down the scheduling problem into stages and consider one time slot at a time.

Moreover, we can model this problem as a graph: each films are nodes, while the edges represent the conflicts between screening times. We can then apply graph algorithms to find the optimal path.

5 Product rating in consumer test

The problem of rating a product in a consumer test can be viewed as a multivariate analysis problem, where many features of the product are taken into account. Also, to give an overall score based on different ratings and features that are described, we can quantify and combine these features which would make this a prescriptive modelling problem. To then reach an overall judgment, we would need to assign different weights to ratings and combining different criteria; this leads to include aspects of decision analysis.

Regardless of the type of the problem, in this context we also need to pay attention to the following factors.

- We should take into consideration the different aspects and features that can be evaluated in the product. Depending on the product this could, for instance, be efficiency or cost.
- As previously mentioned, we need to consider both the objectiveness and non-objectiveness (user-friendliness) of the features.
- In such problems we need to deal with both objective and subjective criteria which can make them complex.

5.1 Possible Ways to Solve

Statistical Analysis and Machine Learning

In these type of problems we are dealing with multiple features: models such as multivariate regression can show us which of these contribute more or less to the overall satisfaction of the customers. Moreover, it might also be useful to reduce the dimensionality of the data set with algorithms such as PCA (Principal Component Analysis) to maintain more meaningful properties of the original dataset.

Prescriptive Modelling

Developing a scoring system based on the features and the importance of each of the criteria we are taking into account could be a way to approach this problem. For this task we need to assign relative importance to each of the features that we are considering and how they influence the score measure.

Consumer Surveys and Feedback Analysis

The feedback provided by customers is crucial to understand which preferences they have and how happy they are.

5.1.1 Standardized Testing

It is essential to perform tests to our product regarding its features such as efficiency or durability, with the goal of assessing the product performance in objectively way.

Benchmarking

It is also useful to compare our product and its relative performance score with others products in the market, making sure that this scoring system is transparent and impartial.

6 Constraint satisfaction problems and constraint programming

a)

In a broad sense, a **Constraint Satisfaction Problem (CSP)** can be considered as mathematical question that involves a set of object, whose state must satify a number of constraints in order to find feasible solutions; in other words, the goal is to find an assignment for all values that doesn't violates any CSP constraints. Usually, this kind of problem is formed by 3 main components:

- a set of variables;
- as set of constraints;
- optionally and based on the context of the problem, an objective function that minimizes or maximizes a certain value.

A CSP represents an important subject of research in the field of computer science, artificial intelligence and operational research; it's also used to model a wide range of problems: planning, logic puzzles, scheduling and optimization problems. Usually, a CSP is solved in a form of a search; some of the most common algorithm in this context are variants *backtracking*, *constraint propagation* and *local search*. In all cases, the steps that are usually involved in the algorithm are the assignment of a value in turn and the consistency of the partial assignment with the constraints of the problem.

Above the possible ways to solve a Constraint Satisfaction Problem and paradigm that can be applied, **Constraint Programming (CP)** is one of the most popular. It is a paradigm that states in a declarative way constraints on the feasible solutions for the set of decision variables, always using the method of backtracking (to explore for potential solutions) and constraint propagation (to infer new constraints and reduce the search space) to find then the solution. The interesting feature of CP is its flexibility: new custom type of solutions can indeed be implemented with code, like a problem-specific branching heuristic. In Constraint Programming, the predominant programming paradigm is logic programming: implementations of this type of solution are usually developed in programming languages such as Prolog, CLP(R) and CHIP, which allows to embed constraints into a logic program.