DAT410 Module 1 Assignment 1 – Group 26

Yahui Wu (MPMOB) (15 hrs)
yahuiw@chalmers.se
Personal number: 000617-3918
Tianshuo Xiao (MPMOB) (15 hrs)
tianshuo@chalmers.se
Personal number: 000922-7950

January 27, 2023

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

Problem analysis

The current stage of weather and temperature prediction relies on numerical weather forecasting. It is a science that uses high performance computers to calculate the future weather. Because atmospheric motion always follows certain physical laws, people write the laws of atmospheric motion changes into a series of mathematical equations, and then use high performance computers to calculate, and get the future weather development change conditions. These results are used to predict the weather temperature and weather conditions.

For the next weekend

For short-term temperature forecasting, we can use numerical forecasting methods for prediction. First, we select data for a certain time period and weight the data according to different independent variables to obtain the processed data. The results are then scored and those with high scores are selected. We perform a linear regression analysis on the prediction results to get the temperature prediction for the next week.

For the same date as today but next year

When we predict the weather temperature for a long period of time, the results obtained by using regression analysis are no longer accurate, so we can use neural networks for modeling analysis.
First, a large amount of weather data is collected and the feature values are sorted. Then it is separated into a training set and a test set. We use DNNRegressor for analysis[1] and define the following parameters: $feature\_columns, hidden\_units, optimizer, activation\_fn, model\_dir$
Then we set up a two-layer deep neural network and build an input function to feed our test set and training set. Finally, the prediction results are periodically scored and evaluated. From this, we can obtain the temperature for a certain day of the next year.

Bingo lottery problem

The rules of the Bingo Lottery

In the game, each lottery ticket consists of a 5x5 square where each number 1 to 25 is randomly placed. Players scratch the numbers in the Caller's Card area (5x5 squares) and then mark the corresponding numbers in the Card area. When the corresponding numbers line up, either horizontally, vertically or in diagonal rows, the player is awarded the winning combination.

Assumptions about the model

1.We assume that the accuracy of the computer-generated random numbers is sufficient for the experiment and consider that the way the computer generates random numbers simulates the way the numbers are printed.
2.We assume that the figures are assumed to be independent of each other.
3.We assume that the three ways to win are the same prize.
For this question, we can start with a statistical approach. Let's start with a mathematical analysis. There are 25 natural numbers in total, from 1 to 25. We assign them randomly to a 5*5 matrix. These data (1-25) are discrete data and we have to assign the values of the matrix, i.e. perform a no-release sampling.

Modeling

First we had the idea of doing the calculation by statistics. 25 numbers were randomly assigned to a 5*5 matrix, and since this was sorted sequentially, we used permutations to calculate all possible scenarios.
The equations are as follows:
$A5^5 * (C_5^{25} * C_1^5 + C_5^{20} * C_1^4 + C_5^{15} * C_1^3 + C_5^{10} * C_1^2 + 1)$
Then we set the numbers for the lucky ticket: $A25^{25}$
Finally, we trained the model to a correct rate of around 12% by means of Constraint satisfaction, using a vertical traversal, a horizontal traversal and a diagonal traversal for all cases.
X: set of variables (all sorting of numbers)
D: set of value fields, each variable has its own value field (winning numbers)
C: set of constraints describing the values of the variables (horizontal correspondence, vertical correspondence, diagonal correspondence)
For this example we use the algorithm of a random forest to implement it. We first randomly select five numbers as the lucky numbers. We then classify the 1000 tickets in the previously set up lottery by boosting. In boosting, each tree is built using $\alpha * n$ samples, and each time the tree is built by randomly drawing so many samples from the full number of samples. The α is usually 0.5 0.8. For each round, the data is not repeated. The classified data is trained and the final expectation is compared with 12%, and the classification is taken to be close to 12%.

Public transport departure forecast

Characteristics of the problem

This problem is to predict the departure time for the public transport line. Take buses for an example, there is a screen at the bus station showing the arriving time of the next bus. For other public transports like trains and subways, it's the departure time on the screen as useful information for passengers to pan their trip. The departure prediction affected by several factors such as which station bus or train arrives, at what period time of the day it will arrive and the weather condition etc. All the factors and the departure time are collected from the historical data. The departure time of the next train is somehow affected by the previous one. In other words, it's a prediction related to the time sequential. Meanwhile, the prediction result is continuous rather than discrete. To achieve the prediction, LinearRegression is selected to be the algorithm we would like to use.

Solution

Linear Regression

1. For a bus/train running on the same route, we suppose the feature to be considered is x_k^m where m is refers to which bus it is and k refers to which station we are focusing on. To simplify the problem, we focus on one specific bus/train and station. Then the feature set could be [s h w] where s is the arriving station (it suggests the current traffic condition), h is the period of time the prediction system operation and w is the weather condition. The features and the departure time are collected from historical data and the dataset is split into train set and test set.
2. LinearRgression imported from scikit-learn is used to train the dataset.
3. *cross_val_score* function can be used to evaluate the performance and tune the model. When the model is tunned, fit the train set and do the predictions.
4. Use *accuracy_score* to give the resuts a score.

Film festival problem

Problem statement

At the time of the festival, a system can automatically create a schedule for the audience. All films will be scheduled to daily and weekly. So there will be a film schedule. But this does not ensure that the schedules of all the films that the audience wants to see do not conflict. The aim is to get the largest number of films to the audience, so if there is a day of the week when the audience does not see the film they want , we can schedule that film for another day of the week.

Problem analysis and modelling

For this problem, we can consider it as a discrete algorithm problem and model it with a greedy algorithm.
First we pick out the films that film enthusiasts like first and schedule them for the whole week. Then, we take out the first day alone and sort the showtimes for that day. Next, we compare the start time of the next film with the end time of the previous film, sorting them in ascending order by the end point, and thus arranging the maximum number of films to be screened on that day.
we have to schedule films that are shown less regularly.When dividing the movies to be played for each day of the week, we can use the dynamic programming(find the optimal subsolution), which divides this whole big problem into 7 smaller problems(7 days), each with a greedy algorithm.
It is possible to plan a weekly optimal solution according to the optimal solution for each day.
Finally, we score the number of films available to film enthusiasts and the arrangement with the highest score is the optimal solution.

Modelling ideas

1.Describe the structure of the optimal solution. We can schedule the first day of movie playback by recursively using a greedy algorithm.
2.Calculate the value of the optimal solution in a bottom-up manner.
3.From the calculated results, construct an optimal solution, which is the week's schedule.

Product rating in consumer test

Characteristics of the problem

Considering the shopping on most widely used e-commerce website, when products are sold to consumers, the consumers are allowed to give reviews for their purchase and product ratings appearing as 1-5 stars and according to the reviews and ratings, the rating algorithm will give a final score shown in the form of 1-5 stars. Basically, there are two factors affecting the final ratings of a product. One of the factors is consumers' individual ratings and the other one is consumers' reviews. It is not fair to rate a product by taking a mean value of all consumers' ratings since the number of ratings and reviews of different s may vary with the sales so it's important to weight consumer ratings in a appropriate statistical method. Viewing a mass of reviews to filter useful ones could take a lot of time for a censor. However, the censoring time can be save by handing the task over to some machine learning solutions.

Solution

1. Consumers' individual ratings

In case of some products with few ratings but high rating level to create a bias and dominate the rating system, the algorithm is required to add a base number to each item involved in the rating system to eliminate the bias. Bayes' weighted-rank algorithm could meet the requirements for creating a fair outcome.The algorithm is based on the equation below:

$$WR = (V \div (V + M)) \times R + (M \div (V + M)) \times C$$

In this equation, V is how many consumers are involved in the rating. M is how many ratings from consumers the product needs so that the system will give a final rating for it. R is the mean score of this product. C is the mean score across the whole stock of products. To avoid the some sellers or e-commerce operators from misleading consummers by manipulating user comments and ratings, a well-designed filter which takes aspects related to consumers' profiles into account can be applied to the rating algorithm.

2. Consumers' reviews

Only using statistics method to give a product a final rating is still not fair enough for the rating system as we want it to be more intelligent. Some machine learning algorithms can be of great use to make the system smarter.
The reviews for each product by users is important since it can be taken as a weighting coefficient to avoid $click farming$.
Machine learning can learn the frequency of words consumers tend to use when they give a review which can be used as a reference in our rating system. We can label such reviews and train the dataset which is the existing reviews. After training, the algorithm predicts the rating of the product by viewing the reviews and it can be added to the final rating multiplied with a appropriate weighting coefficient.

Constraint satisfaction and constraint programming (read and explain)

Main concepts

Constraint satisfaction problem:
In CSP, there is a set of variables and each variable has its own value range. The problem is solved when each variable has its own value and all the constraints on the variable are satisfied. A brief explanation is to give you a few constraints and then give you a few systems of equations and give me the solution that satisfies the system of equations under the conditions.
CSP takes advantage of the state structure and uses generic strategies rather than problem specialisation heuristics to solve complex problems. It is the rapid elimination of large scale search spaces by identifying combinations of variables/values that breach constraints.
Constraint programming:
Constraint propagation or constraint programming is the process of finding a feasible solution from a very large set of candidates where the problem can be modelled with arbitrary constraints.
CP is based on feasibility (finding a feasible solution) rather than optimality (finding an optimal solution), and it is concerned with constraints and variables rather than objective functions. In fact, constraint programming problems may not have an objective function - the goal may simply be to narrow down the various possible solutions to a more manageable subset by adding constraints to the problem.

Constraint programming (and constraint propagation) to numbers

When each letter represents a number, we find that there is no objective function that needs to be optimised, so constraint programming is used to solve for it. First, build the model.[2]

```
model = cp_model.CpModel()
kBase = 10
# Creates the variables.
s = model.NewIntVar(1, kBase - 1, 'S');
e = model.NewIntVar(0, kBase - 1, 'E');
n = model.NewIntVar(0, kBase - 1, 'N');
d = model.NewIntVar(0, kBase - 1, 'D');
m = model.NewIntVar(1, kBase - 1, 'M');
o = model.NewIntVar(0, kBase - 1, 'O');
r = model.NewIntVar(0, kBase - 1, 'R');
y = model.NewIntVar(0, kBase - 1, 'Y');
letters = [s,e,n,d,m,o,r,y]

# Creates the constraints.
model.AddAllDifferent(letters)
model.Add(d  + e + kBase * (n+r) + kBase * kBase * (e+o) + kBase * kBase * kBase * (s+
    m) ==
          y + kBase * e + kBase * kBase * n + kBase * kBase * kBase * o + kBase *
    kBase * kBase * kBase * m)
# Creates a solver and solves the model.
solver = cp_model.CpSolver()
```

Listing 1: Model

and then output all solutions(use *cp_model.CpSolver.SearchForAllSolutions*)

```
class VarArraySolutionPrinter(cp_model.CpSolverSolutionCallback):
    """Print intermediate solutions."""
```

```
    def __init__(self, variables):
        cp_model.CpSolverSolutionCallback.__init__(self)
        self.__variables = variables
        self.__solution_count = 0

    def OnSolutionCallback(self):
        self.__solution_count += 1
        for v in self.__variables:
            print('%s=%i' % (v, self.Value(v)), end=' ')
        print()

    def SolutionCount(self):
        return self.__solution_count

solution_printer = VarArraySolutionPrinter(letters)
status = solver.SearchForAllSolutions(model, solution_printer)

print('Status = %s' % solver.StatusName(status))
print('Number of solutions found: %i' % solution_printer.SolutionCount())
>>S=9 E=5 N=6 D=7 M=1 O=0 R=8 Y=2
>>Status = FEASIBLE
>>Number of solutions found: 1
```

Listing 2: Outputs

For the end conditions, we can also set, for example, a number of limits on the number of solutions or a time limit.
In addition, we can use the Lagrange Multiplier Method, the basic idea of which is to transform a constrained optimisation problem with n variables and k constraints into an unconstrained optimisation problem with (n+k) variables by introducing Lagrange multipliers.
By introducing Lagrange multipliers to establish polar conditions, the partial derivatives of each of the n variables correspond to n equations, and then k constraints (corresponding to k Lagrange multipliers) are added together to form a system of equations problem containing (n+k) equations for (n+k) variables, which can then be solved according to the method of solving systems of equations.
Our basic idea is as follows:
min/maxf(x,y,z)
s.t.g(x,y,z)=0

References

[1] Neural network weather forecasting. Neural network weather forecasting_lichji2016-CSDN blog. (n.d.). Retrieved January 24, 2023, from https://blog.csdn.net/lichji2016/article/details/120069791

[2] Constraint programming and python solving. Constraint programming and python solving-IE06-CSDN blog. (n.d.). Retrieved January 24, 2023, from https://blog.csdn.net/kittyzc/article/details/84260593

Summary of lectures

Yahui Wu

Design of AI systems-introduction
01/17/2023
Teacher: Ashkan Panahi

Lecture 1 provides a brief history of AI and many forms of computing. Turin raised an idea of Turin machine in 1936 which is the main insight of computer science. In 1950, Turin test was introduced by Turin in his paper $ComputingMachineryandIntelligence$. The test is aimed to determine whether a machine can behave as intelligently as a human. The definition of AI was founded at Dartmouth Workshop in 1956. John McCarthy who initiated the workshop demonstrated his optimism about the future of AI. He believed that a significant advance could be made in one or more of the problems reserved for humans in AI field if a carefully selected group of scientists worked on it for a summer.
Although AI at that time gained success with games and some other things, it couldn't handle many non-numerical problems requiring rules and logic. Meanwhile, there is tremendous success in other areas of computing such as basic computing which is achieved by sophisticated technology and advanced computing. There are many forms of advanced computing like control, signal processing, simulation, etc. These forms need mix of human insight and data and a lot of math is unavoidable. Machine learning-based AI is now being widely used. It captures high-dimensional statistics and needs less insight and more data. Data science also uses machine learning techniques and it can make predictions by analyzing existing data.
Traditionally speaking, AI methods should have the ability to solve problems, acquire knowledge or uncertain knowledge and reason it, learn from examples or models and communicate to perceive and act. For many different applications of AI in reality, they all begin with creating models based on the data and analysis of the specific problems.

Tianshuo Xiao

Design of Al systems - introduction
01/17/2023
Teacher: Ashkan Panahi
This lecture is an introduction to AI, including the history of AI and computation and statistics. Turing believed that people would be able to speak of machines thinking without expecting to be contradicted. The invention of the Turing machine by Turing in 1936 took computer science to a new level. Later, people worked on enabling machines to use language, to form abstract concepts, to solve the various problems now left to humans and improve themselves. John McCarthy believed that every aspect of learning or features of intelligence, could be described precisely. He thought through the research of a group of scientists, they could make significant progress on these issues. But the development of artificial intelligence has not developed as smoothly as expected.
In the process of research, people discovered that AI is programmed according to human logic and rules, which means mostly they are non-numerical.This makes it difficult to solve abstract problems, for example they can distinguish between a cat and a dog, but they cannot separate which species.In contrast, people have had great success in other areas of computing.
Through basic computer, people can accomplish basic things with sophisticated technology, which makes it easier to understand complex things. Through advanced computer, it mix of human insight and data which includes a lot of math. Besides, advanced computers can perform simulation, control, algorithms and optimization and operations research.
Machine learning and AI can capture high-dimensional statistical patterns to solve problems, which means they require fewer insights but more data. Some traditional approaches to artificial intelli-

gence such as "Problem-solving ", "Knowledge, reasoning, planning " do not require data. However, "Uncertain knowledge and reasoning ", "Learning " and "Communicating, perceiving, acting " need algorithms and data to process.
When we build an AI model, we first need to collect and analyse the data, then select the mode of processing the data, and choose the appropriate system. Finally, we need to simulate and validate the model.