Mert Türedioğlu

1856335

cogs514-literature-review

machine learning model for difficulty assessment of climbing routes

- ***Stapel, F. (2020) A Heuristic Approach to Indoor Rock Climbing Route Generation***

**** Aim:***

The paper's main is to generate novel boulder problems(climbing routes). The achieve this goal; they asked the questions:

- To generate a new boulder problem, what heuristics one may follow?
    - How can one classify climbing holds?
    - How can one classify climbing moves?
    - How do particular heuristics alter the difficulty grade of a boulder problem?
- With the help of these heuristics, how can one set a new boulder problem?
- What standard one may follow to assess the quality of a boulder problem?

The training board they used in this research is MoonBoard 2017, which is a 40˚ overhang artificial climbing wall.

It may seem that my project's aim and this paper do not overlap. However, to achieve their goal, they assessed the difficulty grades of climbing routes. The overlap is here. Although their main goal does not correnspond to mine, this assessment makes it so. In addition to this, their heuristic approach is valuable here. This heuristic approach will help me to have a good feature set.

**** Methods***

They used greedy tree algorithms to generate a tree that consists of novel boulder problems. In the literature, people used machine learning models to assess the difficulty of a problem. In this research, they think that a heuristic approach will be better since they found previous work unsuccessful because their models couldn't assess the difficulty of holds as people do.

To grow trees, they classified climbing holds and climbing moves and studied the influence of these heuristics on the grade of a problem.

Classification of holds:

There are 198 different holds on this board. The classified these holds regarding these aspects:

- A difficulty rating of gripping a particular hold. They gave a questionnaire consisting of verbal phrases to experienced climbers to rate how difficult to grab a specific hold on a scale of 1 to 5.
- The hold's rotational angle to the ground. They defined eight angles: east, west, south, north, north-west, north-east, etc.
- The primary type of the hold: edge, sloper, pinch, etc.

- The secondary type of the hold.

Classification of moves:

After analyzing the videos of people climbing MoonBoard problems, they listed the moves made in climbing MoonBoard problems: using a side pull, 'normal' climbing move, dead-pointing, dyno-ing, heel hook, etc.

Influence of heuristics on the difficulty of a problem

They listed the heuristics below to assess the difficulty grade of a problem:

- Longer move, harder problem
- Harder hold, harder problem
- Fewer footholds, harder problem
- More special climbing moves, harder problem

Score of a hold =

Regarding classification of holds, experienced climbers agreed upon the difficulty rate of the hard and easy holds, yet their ratings did not overlap regarding holds neither easy nor hard.

Score of a hold = (2 * difficulty score + 1 * hold score + 3 * move score + 3 * rotation score) / 9. Each score has own equation which can be seen in the paper.

Tree algorithm:

Algorithm 2 A greedy algorithm for route generation using trees

```
Data: Desired difficulty, hold types, length modifier,
      number of leafs
Result: A list of lists of coordinates forming climbing
        routes
calculate score of each hold
routes = [[best hold], [second best hold], ...]
final routes = []
while routes is not empty do
    for route in routes do
        calculate the score of each hold
        add best scoring reachable hold to route
        for 5 best holds do
            if new hold is not in row 18 then
                add the new route with the new hold to
                routes
            else
                add the new route with the new hold to
                final routes
```

**** Results***

They asked experienced climbers to evaluate the routes generated and benchmark MoonBoard routes: %80 of the benchmark routes have good flow, yet it was %45 for the routes generated by the algorithm. %75 of the benchmark routes are enjoyable, whereas %40 of the generated routes are enjoyable. The main reason for this is that the algorithm does not consider the full state climbers on routes on each move. What I mean by the full state of the climbers is that the algorithm does not keep track of the positions of the limbs of the climbers when climbing the route. This leads to awkward moves in generated routes.
They compared their grades with people's opinions on the routes generated. These were pretty close. They were successful in estimating the grades of the routes.

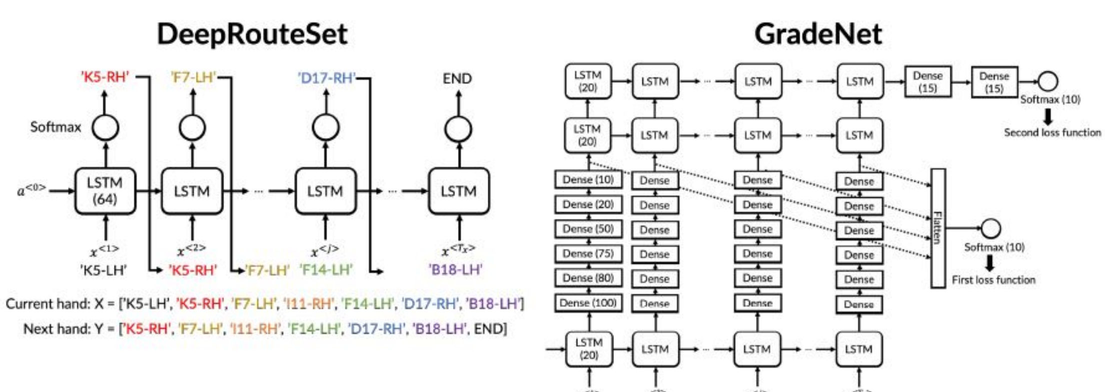
- ***Duh, Y. S., & Chang, R. (2021). Recurrent neural network for MoonBoard climbing route classification and generation***

**** Aim:***

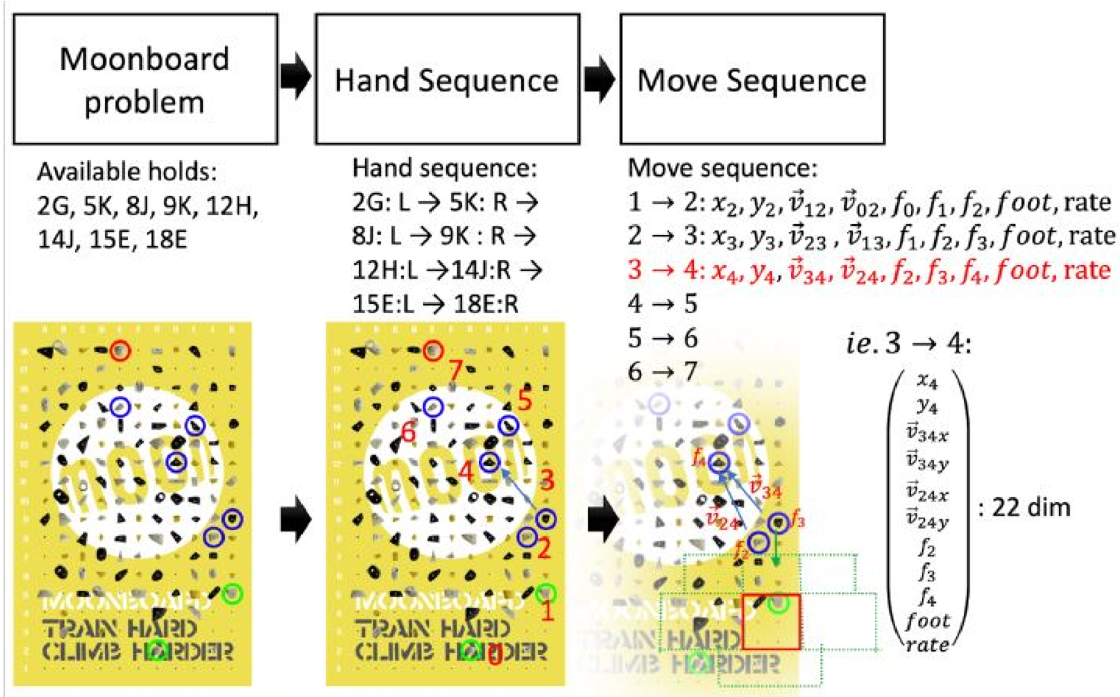
Their main aim is to develop a model that generates MoonBoard problems, DeepRouteSet. To achieve this, they develop a pipeline, BetaMove, that returns the hand sequence of a given route. This sequence output fed into another model that returns the difficulty grade of a given route, GradeNet.

**** Methods***

As I briefly mentioned in the aim part, the BetaMove pipeline is developed. It is used both to train GradeNet, and DeepRouteSet. BetaMove generates many possible hand sequences when given a route and returns the one with the best scores. They used a beam search algorithm to find the one with the best score. With these hand sequence data and grade labels, they trained both GradeNet, a model that returns the difficulty grade of a route, and DeepRouteSet, which generates novel routes with grade labels. They treated this problem of classifying and generating routes as an NLP problem. Because of the sequential nature of the climbing routes, they preferred RNN algorithm. Like music generator models, DeepRouteSet learns the patterns between adjoint moves and uses these patterns to create new routes. As it is done in sentiment analysis, they used sequantial data of hand sequences mimics a textual data and grade information like sentiments. The architectures of DeepRouteSet and GradeNet are below:



I also added their figure for BetaMove below. Yet, unfortunately, I did not understand how they compose their 22-dimensional vectors representing moves. Maybe analyzing their source code could clarify that 22-dimensional vector.



**** Results***

DeepRouteSet: The generated routes have a good flow. Yet, most generated routes, at least those uploaded to their GitHub repository, are not legitimate MoonBoard routes. Since it violates the MoonBoard rule, 'All start holds must be on row 6 or lower'. They asked an expert to make a quality analysis of their generated routes, and the latest 50 benchmark MoonBoard problems. Then, they have compared the results of the expert and another quality analysis of routes generated by another machine learning model MoonBoard. Their generated routes have the highest scores. The results table they provided is below:

	total problems evaluated	redundant/ holds	weird sequence/ ugly moves/ lack of flow	reasonable problems	high quality problems
Latest MoonBoard problems	50	10%	6%	84%	60%
Deep RouteSet	40	0%	5%	95%	80%
Model Ref[3]	40	35%	35%	40%	20%

GradeNet: They have reported their accuracy regarding grades as below. They have lower accuracy in the exact estimation, yet they have also reported their accuracy +-1 grade of actual grade. The results table they provided is below:

	HLP	GradeNet Training	GradeNet Dev set	GradeNet Test set	Naive RNN Dev set	Ref[1] CNN	Ref[2] Graph NN	Ref[3] MLP
Accuracy	45.0%	64.3%	47.5%	46.7%	34.7%	34.0%	Not reported	35.6%
±1 accuracy	87.5%	91.3%	84.8%	84.7%	Not evaluated	Not reported	Not reported	74.5%
F1 score	-	0.506	0.242	0.255	0.165	Not reported	0.310	Not reported
AUC	-	0.898	0.764	0.773	Not evaluated	Not reported	0.73	Not reported

----

Discussion:

The popularity of sport climbing increased after it became an Olympic sport in Tokyo 2020. Moreover, as MoonBoard became a globally used training board for climbers, research papers on it started to show up. The two papers I briefly reviewed are the ones I found the most relevant to my aim in this project. Although the primary goal of the two papers is different from my project goal, both papers' interest is to generate new moonboard problems; their process includes grade estimation of a MoonBoard problem. There are various reasons why I chose these two papers.

First, I chose the first one because of its heuristic approach. They have a heuristic approach that reflects insights regarding the nature of climbing. This inspired me to create a dataset with the insights they provided. This dataset will have features that grasps the nature of climbing. However, they used tree algorithms to generate new routes; I'm planning to use tree algorithms to estimate the grade of a route instead of generating new routes.

Second, I chose the other article because of their choice of a machine learning algorithm for the problem. Again, their main goal was to generate new routes, yet, they have a model named GradeNet, which predicts the difficulty grade of a problem. They have used Recurrent Neural Network algorithm to estimate the grade of a route. They have represented the routes as sequential data. I am not sure if it is necessary to have a sequential kind of data to estimate grade of a route. I plan to have a non-sequential dataset with features inspired by the first article.