Mert Türedioğlu

1856335

cogs516-literature-review

probabilistic model for difficulty assessment of climbing routes

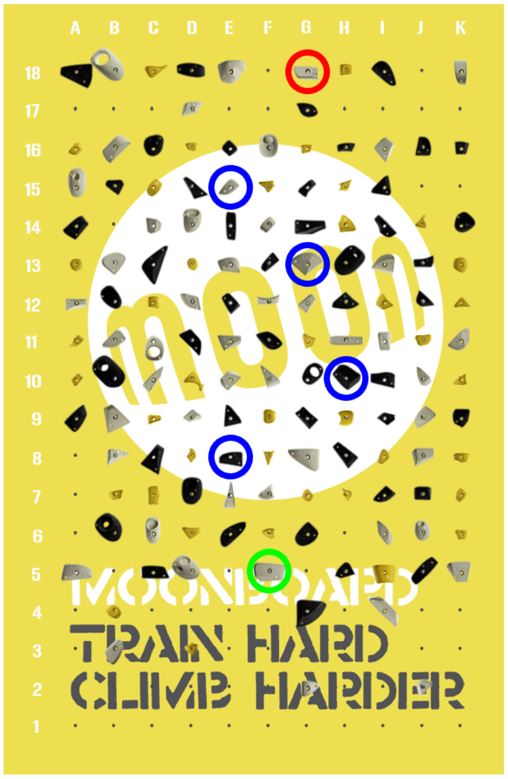
- ***Tai, C. H., Wu, A., & Hinojosa, R. (2020). Graph neural networks in classifying rock climbing difficulties. Technical report***

** Aim:*

Their main goal is to classify MoonBoard problems into difficulty categories. They hope to replace the heuristic-based classification of routes with their Machine Learning model. They aim to help climbers estimate their routes with the help of this model.

** Methods*

They were inspired by the NLP domain in which Graph Convolutional Network architectures have proven success in text classification. Like words constitute sentences, they conceived holds form MoonBoard routes. There are 140 different holds on MoonBoard 2016 layout. A problem consists of specific holds. The difficulty of a problem results from this selection of particular holds. An image of an example MoonBoard problem is below:

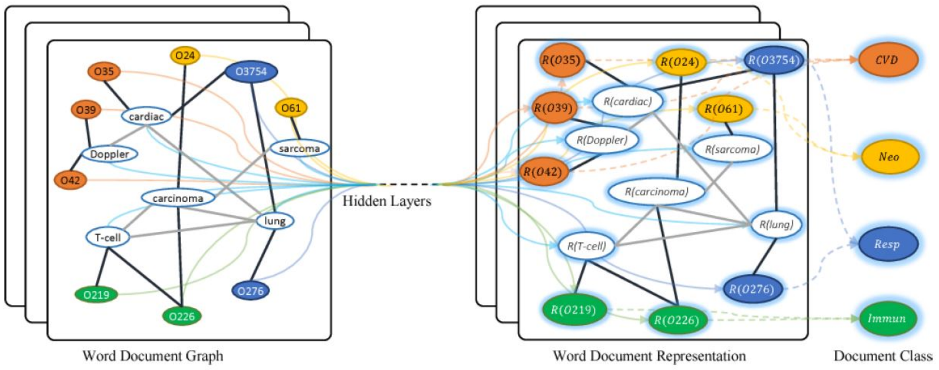


The green circles indicate the starting holds. The blue circles are intermediary ones. Lastly, the red circle designates the top(i.e. end) holds. You can use only these holds to claim an ascent of this specific problem..

Like I did in this project, they have scraped the MoonBoard website to create their problems dataset. Unlike me, they have scraped both benchmark and non-benchmark problems. I only scraped the benchmark ones. The reason for this, the benchmark routes are reliable concerning their assigned difficulty grade. Benchmark routes are the ones that MoonBoard crew evaluates and confirms their grades. There is an imbalance in the dataset. I think they needed much more routes to train their machine learning model. I believe that I do not need that many problems to fit my model. There are far more easier routes than difficult ones in the dataset. They have used a sampling method to overcome this imbalance. They have sampled 2000 routes from each category. They split problems of each category into 80-20 divisions. In the end, they got 14,080 problems in the training set and 3,520 problems in the test set.

They have preprocessed their problems into both one-hot and multi-hot encoded representations. Recall that there are 140 holds in MoonBoard 2016 layout. In their encoding, a problem is a 140-dimensional vector in which holds are either present or absent(i.e. 0 or 1).

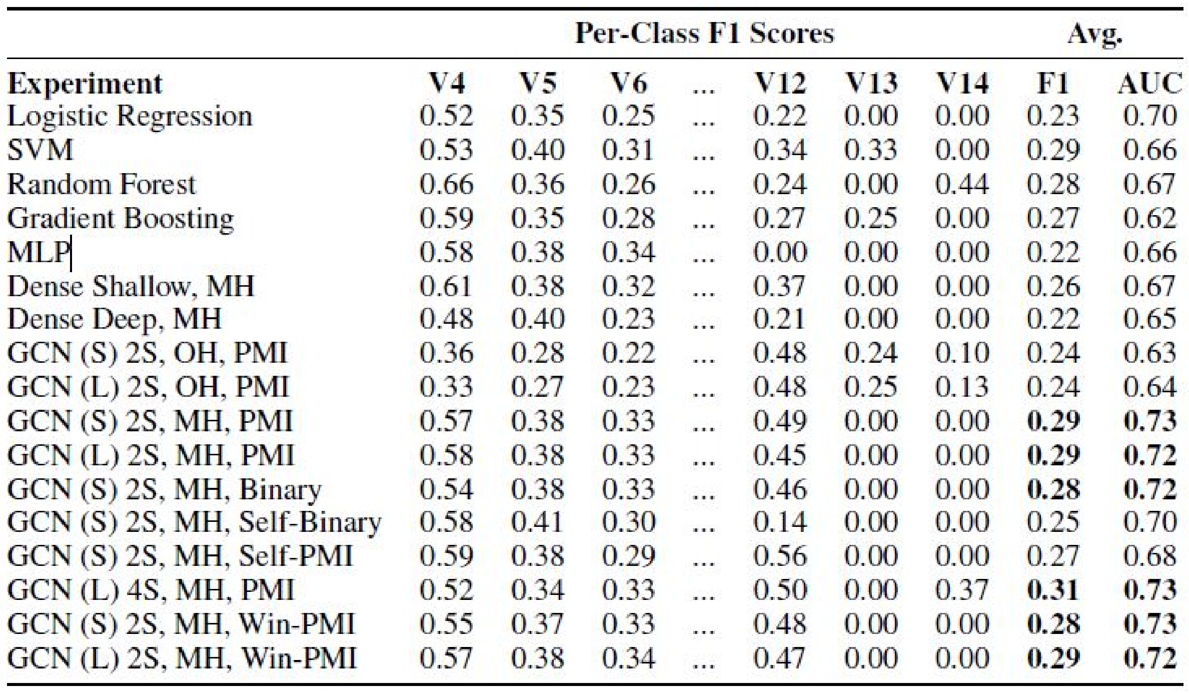
Heterogenous Corpus Graph: For text classification, heterogenous corpus graphs are used. In text GCN's, there are two kinds of nodes, words and documents. An example figure is below:



In their MoonBoard application of this technique, words are holds, and documents are problems. They expect that problems with similar holds have similar difficulty grades.

**** Results***

To make a comparison between different techniques, they have experimented with 17 different machine learning techniques. They grouped them under three categories: baselines, dense, and GCN. The best performing models are the Graph Convolutional Networks. With my limited knowledge of Machine Learning models, I think the reason behind this is their one-hot representation of MoonBoard problems. Not all the models are suitable for one-hot encoding. Anyway, the classes are difficulty grades: from V1 to V14. And experiment column are the techniques they used. The F1 table per class is below:



	Per-Class F1 Scores							Avg.	
Experiment	**V4**	**V5**	**V6**	...	**V12**	**V13**	**V14**	**F1**	**AUC**
Logistic Regression	0.52	0.35	0.25	...	0.22	0.00	0.00	0.23	0.70
SVM	0.53	0.40	0.31	...	0.34	0.33	0.00	0.29	0.66
Random Forest	0.66	0.36	0.26	...	0.24	0.00	0.44	0.28	0.67
Gradient Boosting	0.59	0.35	0.28	...	0.27	0.25	0.00	0.27	0.62
MLP	0.58	0.38	0.34	...	0.00	0.00	0.00	0.22	0.66
Dense Shallow, MH	0.61	0.38	0.32	...	0.37	0.00	0.00	0.26	0.67
Dense Deep, MH	0.48	0.40	0.23	...	0.21	0.00	0.00	0.22	0.65
GCN (S) 2S, OH, PMI	0.36	0.28	0.22	...	0.48	0.24	0.10	0.24	0.63
GCN (L) 2S, OH, PMI	0.33	0.27	0.23	...	0.48	0.25	0.13	0.24	0.64
GCN (S) 2S, MH, PMI	0.57	0.38	0.33	...	0.49	0.00	0.00	**0.29**	**0.73**
GCN (L) 2S, MH, PMI	0.58	0.38	0.33	...	0.45	0.00	0.00	**0.29**	**0.72**
GCN (S) 2S, MH, Binary	0.54	0.38	0.33	...	0.46	0.00	0.00	**0.28**	**0.72**
GCN (S) 2S, MH, Self-Binary	0.58	0.41	0.30	...	0.14	0.00	0.00	0.25	0.70
GCN (S) 2S, MH, Self-PMI	0.59	0.38	0.29	...	0.56	0.00	0.00	0.27	0.68
GCN (L) 4S, MH, PMI	0.52	0.34	0.33	...	0.50	0.00	0.37	**0.31**	**0.73**
GCN (S) 2S, MH, Win-PMI	0.55	0.37	0.33	...	0.48	0.00	0.00	**0.28**	**0.73**
GCN (L) 2S, MH, Win-PMI	0.57	0.38	0.34	...	0.47	0.00	0.00	**0.29**	**0.72**

They found the 4-step GCN model best performing model. They reported that the 4-step GCN performed better with restricted datasets as they expected. They claim that the optimal number of graph convolutions are domain-specific and four graph convolutions worked well in this problem.

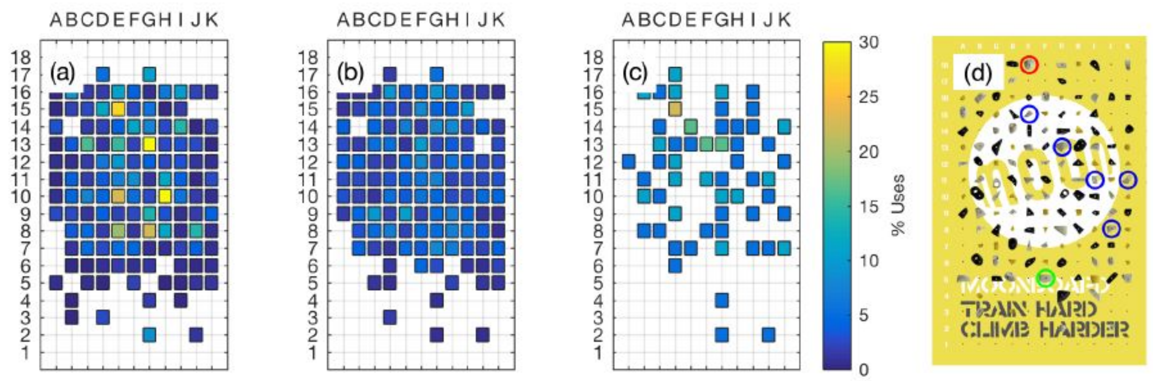
- ***Dobles, A., Sarmiento, J. C., & Satterthwaite, P. (2017). Machine learning methods for climbing route classification.***

**** Aim:***

Like the paper above, their main goal is to classify MoonBoard problems into difficulty categories. They want to eliminate the subjective aspect of classifying problems' difficulties and make this classification task accurate and consistent. This work is published before the one above.

**** Methods***

They have scraped 13,871 total problems. There were fewer problems when they conducted this research. Again, the dataset is imbalanced. The dataset is skewed towers the easier side. They randomly divided the dataset into 11,095 training, 1,388 validation, and 1,388 test. Yet, the division is not fully randomized; they kept the distribution of grades in each set. In their research, they provided the below figures. In these figures, the first one represents easier routes, the middle one is the intermediate problems, and lastly, the harder routes. From these figures, they found to classify intermediate routes was harder than the other two groups since all the holds were more or less used with the same frequencies. In contrast, there are specific holds used in easier and harder routes.



For CNN They represented MoonBoard as

$$\mathbb{x}^{(i)} \in 0, 1^{18x11}$$

For the Naive Bayes and softmax classifiers each MoonBoard problem turned into a vector and used as feature space as follows:

$$\mathbb{x}^{(i)} \in 0, 1^{n}$$

They have experimented with three different classifiers: Naive Bayes classifier, softmax regression classifier, and convolutional neural network classifier. They also used SVM, but they did not report it because of its poor performance on the validation set.

They have used a Naive Bayes classifier using the Bernoulli event model, and its input problem matrix is like the previous paper. The values are {0, 1} where 0 is absence and 1 is the presence of a hold in a problem. Yet, they thought that the assumption of the Naive Bayes classifier is violated since they believe that holds in a route are not independent of each other. .

A softmax classifier were also used hold vectors of

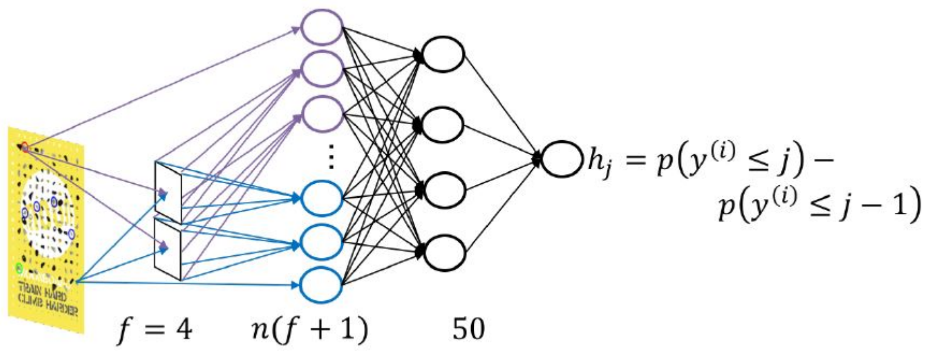
$$(\mathbb{0}, \mathbb{1})^{n}$$

. They used Batch gradient descent algorithm. In the first implementation, the model had a tendency to towards to common labels when predicting the routes' difficulties. They added the penalizing weighting below to overcome this issue:

$$w^{(i)} = a\frac{m}{\sum_{j=1}^{m} 1\{y^{(i)} = y^{(j)}\}} + 1 - a$$

Yet, it did not help. This time model started to overfit to less common categories.

They treated this problem as an image problem. Apart from the baseline models above, they also used Convolutional Neural Network classifier to assess the difficulty grades of routes. Their diagram of implementation is below:



Hypothesis function:

$$h(x^{(i)})_j = p(y^{(i)} \le j) - p(y^{(i)} \le j-1)$$

Loss function:

$$L(w,b) = -\sum_{i=1}^{m} w^{(i)} \sum_{j=1}^{k} \mathbf{1}\{y^{(i)} \le j\} \log p(y^{(i)} \le j | x^{(i)}; w, b)$$

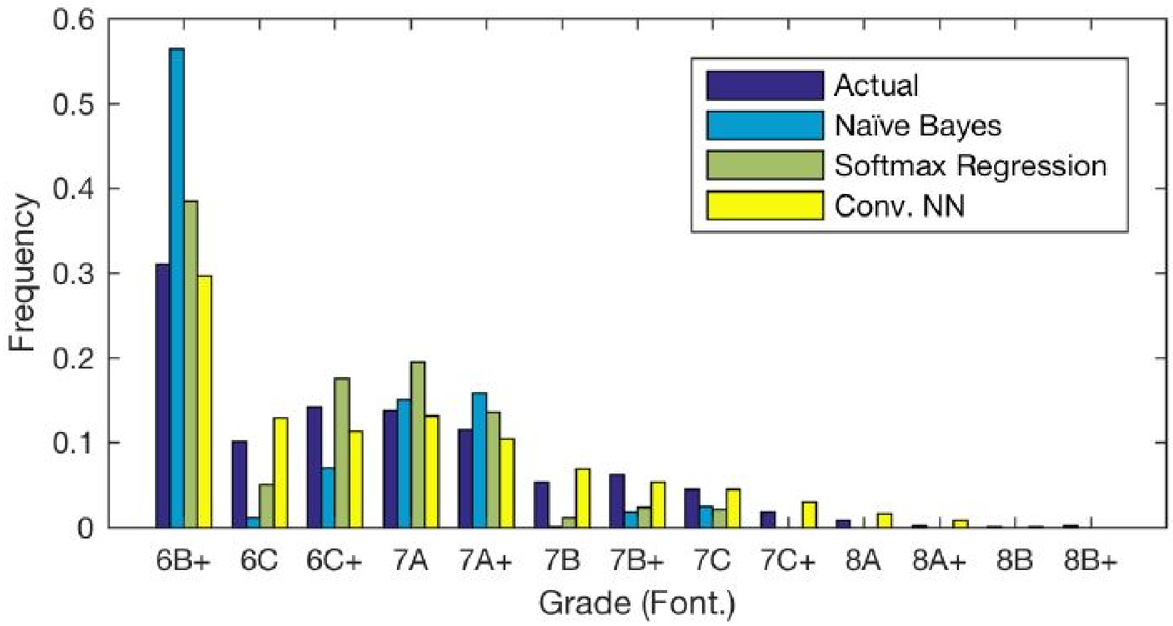
**** Results***

They tested performance of their three classifiers against humans. There are three metrics: accuracy, MAE(mean absolute error) and KL-divergence(Kullback-Leibler divergence). The results on validation is below:

	User Ratings	Naive Bayes	Softmax	CNN
Accuracy	93.4%	34.0%	36.5%	34.0%
MAE	0.12	1.73	1.37	1.40
KL-divergence	0.0071	0.41	0.078	0.020

Their performance is poor against human performance. However, they think that the reason behind this is that people did not grade the routes by just looking at the image of a route. People climb it, and it allows them to make accurate assessments.

The table below shows the comparison of three classifiers with respect to actual categories:



Note that both baseline models did not label any route with a grade 8A or harder. According to the authors, the reason for this is that both classifiers tend to predict common categories and the data is skewed towards the easier side. They also noted that more complex CNN was not possible since the dataset is relatively small.

Discussion:

I believe that a probabilistic model with a well-featured dataset will outperform the models of both papers. One treated the problem as an NLP problem and the other as an image problem. Both were far from representing the causal relations between holds when assessing the difficulty of a problem. The first paper was able to describe the relation between holds and problems to a certain extent. Their model reflected their assumption that similar problems have similar holds. Yet, it was not enough to have a model with high accuracy. I am aware that the subjective nature of assessing the difficulty grade of a route makes the problem difficult. Moreover, although the grading scale is discrete, to a certain extent, there may be differences between the routes in the same difficulty class. Some routes are closer to one grade harder class; the others may be closer to one grade easier class. Because of this, I believe that the performances of classifiers would significantly get better if we test them in +-1 classes.