CS230

Graph-Based Classification of Rock Climbing Difficulties

Rafael Hinojosa
Microsoft Corporation
Stanford University
rahinojo@stanford.edu

Cheng-Hao Tai
Microsoft Corporation
Stanford University
c2tai@stanford.edu

Aaron Wu
Evisort Inc.
-
aaron@evisort.com

1 Introduction

Given the considerable increase in popularity of rock climbing in recent years, we want to create a neural network that can automatically classify a given climbing route (a.k.a. problem) into a suitable difficulty category. Such a tool could either speed up or replace the traditional heuristic-based, community-agreement approach to ranking climbing problems, thereby lowering the barrier of entry for anyone seeking to create their own routes. We plan to solve this problem using a classifier built for the "Moonboard" apparatus — a modular climbing wall of fixed dimensions and pre-set hold configurations (Figure 1).

2 Dataset

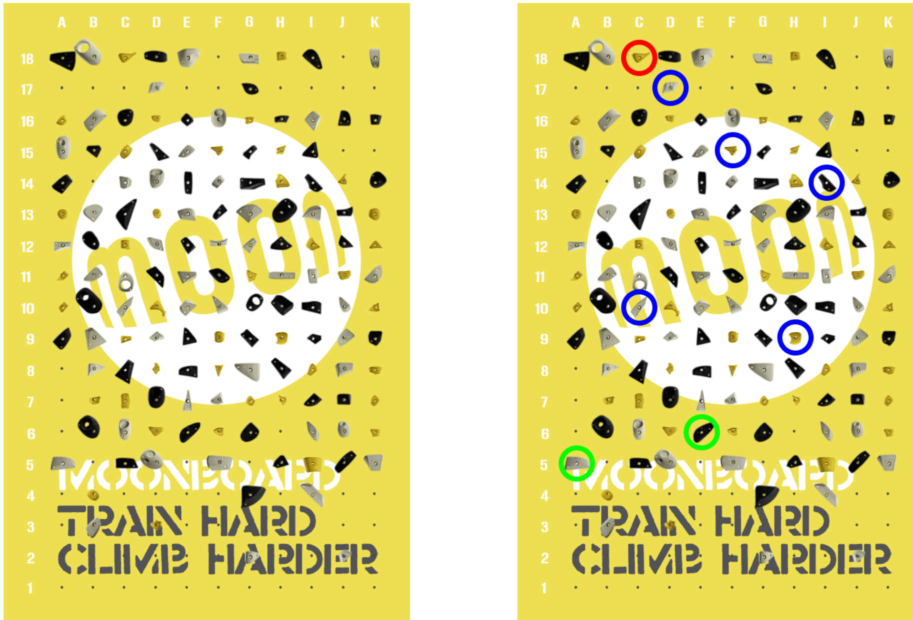
On their website, Moonboard hosts a database of Moonboard climbing routes (see example in Figure 1b) that have been created and ranked by the global climbing community. For a given hold configuration (i.e. Moonboard 2016), routes are denoted using multi-colored circles superimposed on a stock Moonboard image — specifying a hold set which is what defines a problem and motivates its difficulty. Every Moonboard problem comes with metadata tags such as a suggested difficulty level, quality rating, and the number of people who have completed the route.

3 Preprocessing and Learning Method

We plan to mine this data from the Moonboard webpage and structure it into a graph upon which we will apply ideas drawn from [3] which solves text classification tasks using a Text Graph Convolutional Network (GCN) (Figure 2). While the authors of [3] organized their heterogeneous corpus graph using documents and words as nodes and global co-occurrence / TFIDF scores as edge weights, we will adapt this to a Moonboard scope. In our application, document nodes are replaced by problem nodes, word nodes by hold nodes. The idea is that just as documents are composed of words, Moonboard problems are composed of holds. Tangentially, we'd also like to implement [2] which presents a simplified version of the Text GCN that actually performed better on the same text classification task as [3].

4 Evaluation

To evaluate our model, we plan to use binary classification metrics (i.e. accuracy, precision, recall, and F-score) extended to a multi-class classification use case. Since there are many distinct Moonboard difficulty ratings (around 10), we plan to perform model evaluation on each of these difficulty settings so that we're able to easily diagnose class-dependent model issues (i.e. data imbalance, overfitting



(a) Stock Moonboard 2016 configuration

(b) A specific Moonboard problem

Figure 1: Moonboard training apparatus showing a specific Moonboard problem

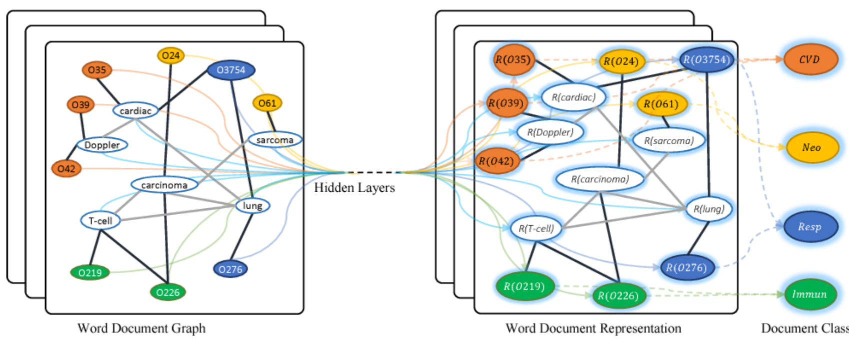


Figure 2: A heterogenous corpus graph for document classification using a Text Graph Convolutional Network. Nodes come in 2 types: document entities and word entities. The different colors correspond to different node categories (notice only document nodes are colored). A direct analogy to Moonboard problems can be made by substituting documents for problems and words for nodes.

one class). For the Moonboard 2016 hold configuration, there are approximately 28,000 unique problems on Moonboard's server which will correspond to a total dataset size of 28,000 samples. We plan to use a 80-20 split to create our training and test datasets and use hold-out subsets of the training set to evaluate model performance before final validation on the test set.

5 Additional Challenges

Finally, if our classifier succeeds, we wish to extend our project in one of two ways: (1) a generative model [1] that can produce a new Moonboard problem, given a specific difficulty and (2) experiment with ways to overcome a significant GCN limitation (needing to retrain entire graph for every new inference sample).

References

[1] Mirza, M. & Osindero, S. Conditional Generative Adversarial Nets *arXiv:1411.1784* (2014)

[2] Wu, F., Zhang, T., de Souza Jr., A., Fifty, C., Yu, T., & Weinberger, K. Simplifying Graph Convolutional Networks *Proceedings of the 36th International Conference on Machine Learning* (2019)

[3] Yao, L., Mao, C. & Luo, Y. Graph Convolutional Networks for Text Classification *33rd AAAI Conference on Artificial Intelligence (AAAI-19)*, 7370-7377 (2018)