Machine Learning Methods for Climbing Route Classification

A. Dobles, J.C. Sarmiento & P. Satterthwaite

adobles@stanford.edu jcs10@stanford.edu psatt99@stanford.edu

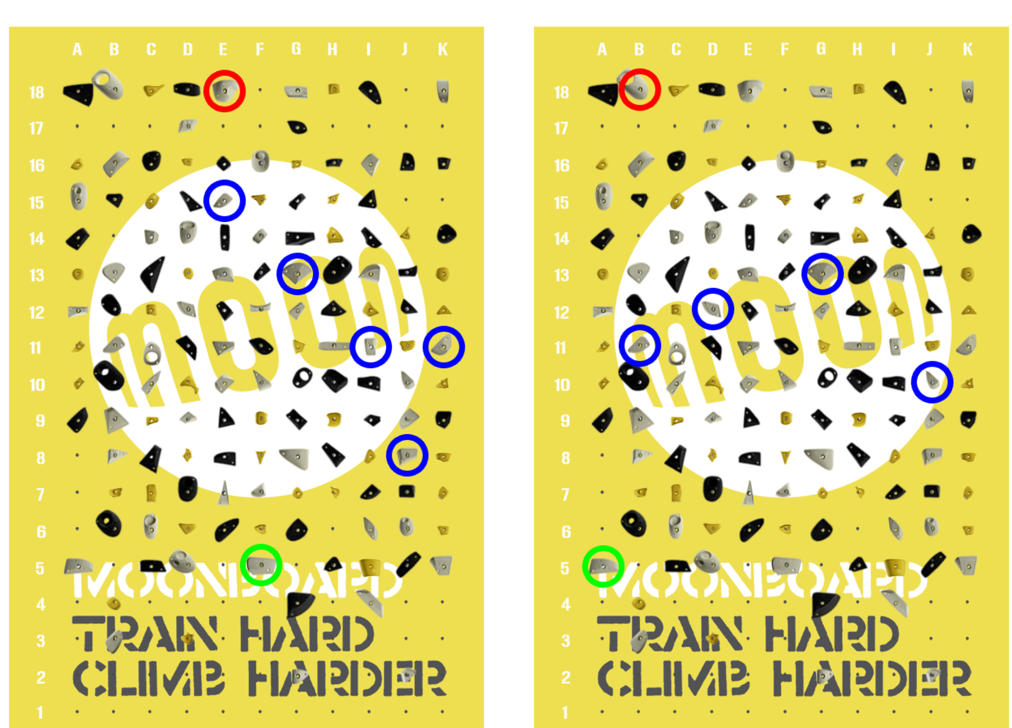
CS 229 Final Project – Stanford University

Abstract

Classifying the difficulty of climbing routes is a challenging and subjective task, even for experienced climbers. This project builds on previous work [1] and seeks to create an algorithm to aid in this process. Data was sourced from a standard climbing wall (the Moonboard) found in gyms around the world. Three different models were explored for classification: Naïve Bayes, Softmax Regression, and a Convolutional Neural-network (CNN) with an ordinal output. The CNN was found to have the best combination of accuracy and accurate reproduction of underlying distribution of the three classifiers, though all models lag behind human classification performance.

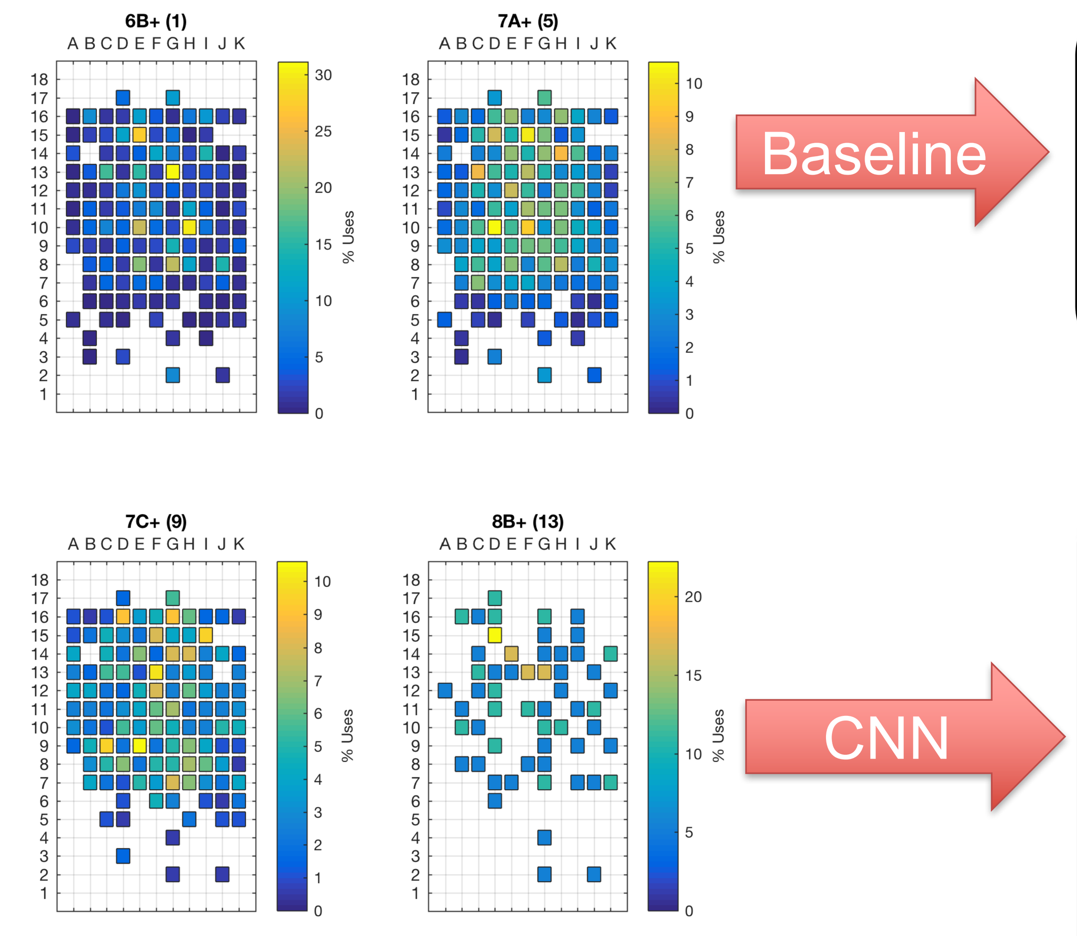
Data

- Source: Moonboard Website (https://www.moonboard.com/)
- 14,000+ different routes have been set by the community
- 142 holds arranged on an 18 x 11 array
- Users pick a subset of holds to define a route, including start and end holds
- 13 difficulty categories (6B+ to 8B+ on Fontainebleau scale)
- Difficulty assigned by setter, 42% of routes also have grade determined by community



6B+ (1) Route 8B+ (13) Route

Models



Input data $x^{(i)} \in \{0,1\}^{142}$

$m = 11{,}095$

(I) Naïve Bayes $\theta \in \mathbb{R}^{nxk}$

- Event model
- Ignores:
  - Climbing movements create correlations between holds

(II) Softmax Regression $\theta \in \mathbb{R}^{nxk}$

- Binary data limits feature space
- Ignores:
  - Spatial correlations
  - Ordering of categories

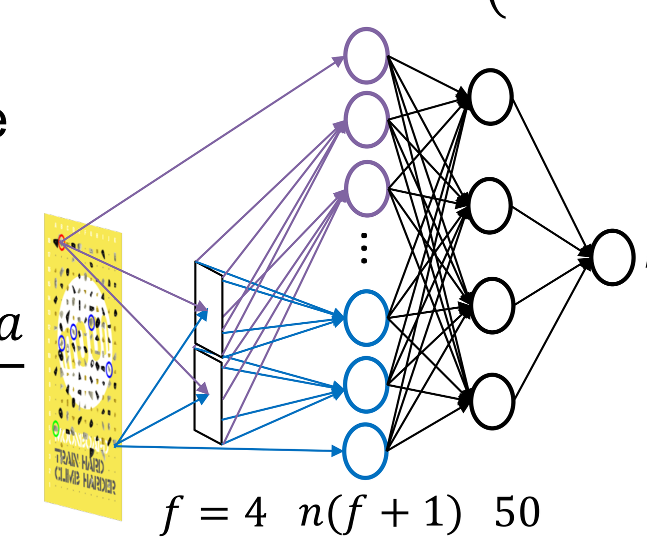
- Ordinal regression on output neuron [2]
  - Reduces parameter space ($\theta^{[l]} \in \mathbb{R}^{n^{[l-1]}+k}$) and uses ordering of data

$$L^{(i)} = -w^{(i)} \sum_{j=1}^{k} 1\{y^{(i)} \leq j\} \log p(y^{(i)} \leq j \mid x^{(i)}; w, b)$$

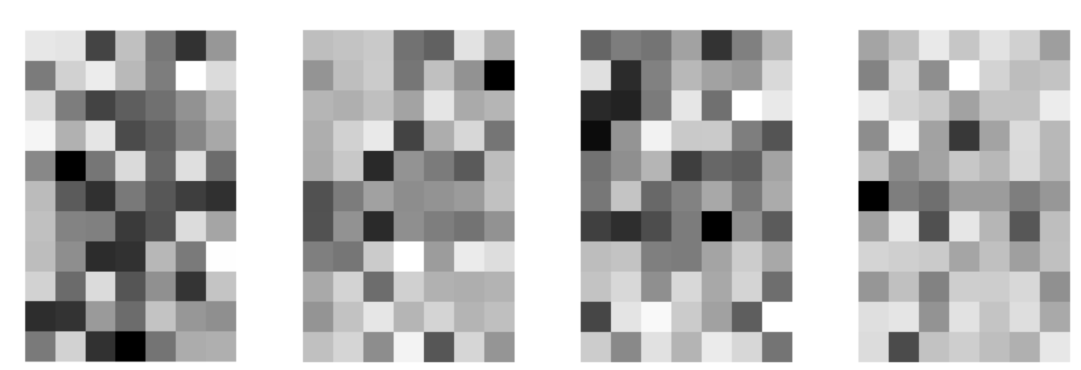
$$p(y^{(i)} \leq j | x^{(i)}; w, b) = \begin{cases} 0 & j = 0 \\ \sigma(b_j - w^T x^{(i)}) & 0 < j < k \\ 1 & j = k \end{cases}$$

- Weighted so that the NN is more penalized for getting less common categories wrong

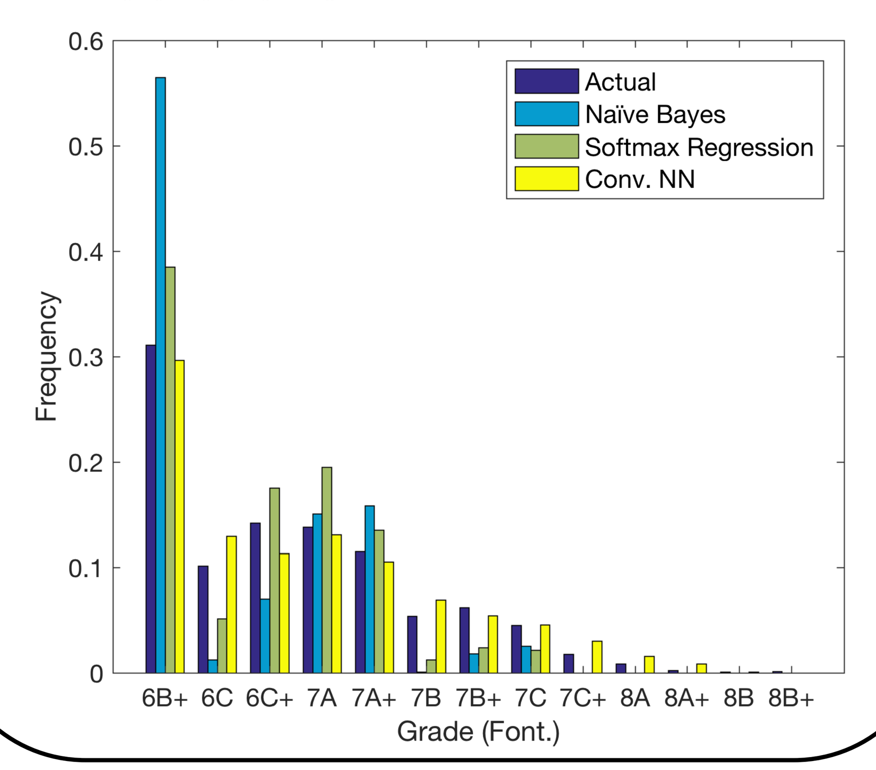
$$w^{(i)} = \frac{a \sum_{i=1}^{m} 1\{y^{(j)} = y^{(i)}\}}{m} + \frac{1-a}{m}$$

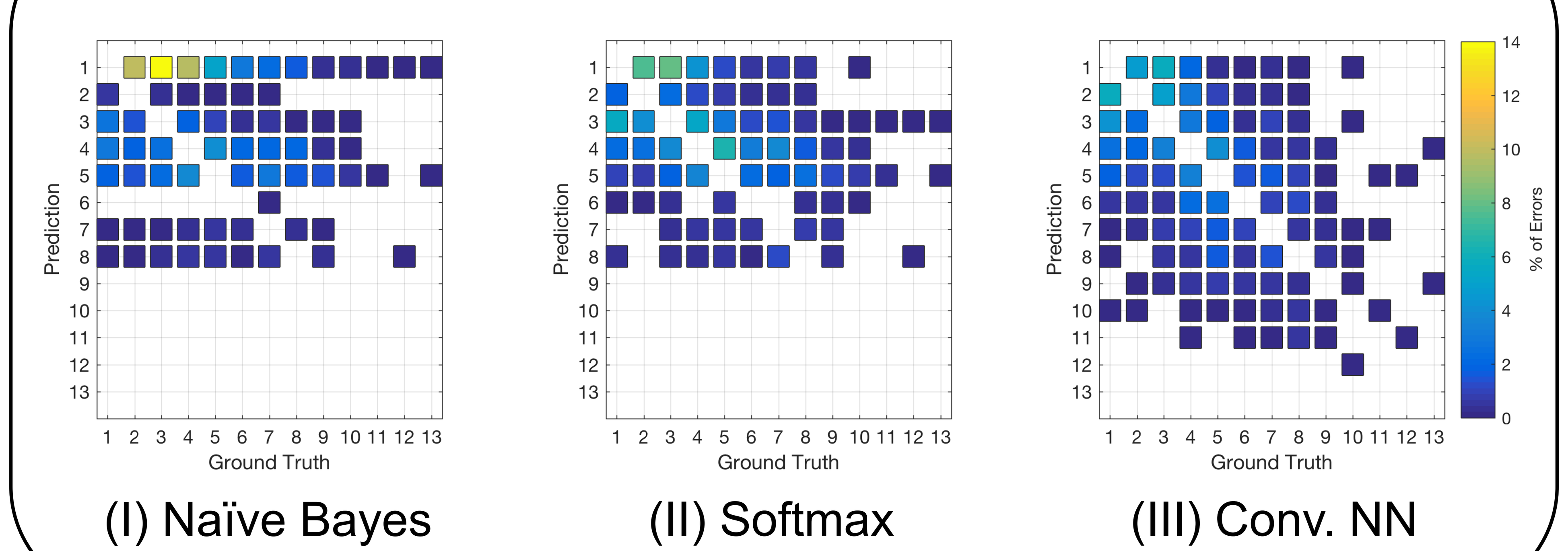


Results



Dev. Set Statistics	User Ratings	Naïve 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	0.0779	0.0199





Discussion

None of the tested models were able to rival human classification accuracy (93.4%). This discrepancy can be partially accounted for by the ability of humans to actually climb the problem to asses its difficulty. Though the Naïve Bayes, Softmax and CNN classifiers all had comparable accuracy and mean absolute error (MAE), the CNN had a significantly lower Kullback-Leibler (KL) divergence relative to the underlying distribution. This improvement can be accounted for by: (1) the convolutional filters mapping the input features from $\{0,1\} \rightarrow \mathbb{R}$ and (2) the use of an ordinal regression and weighting the loss function to discourage the algorithm from always picking the most common categories. This significant improvement in the distribution of predictions, can be seen in the confusion matrices and prediction histograms. Implementation of a more complex CNN was limited by the size of the training set, the implemented CNN had ~50,000 parameters and required dropout [3] to converge.

Future Work

- Data augmentation
  - Human-generated information on holds
  - Graphical models of climbing routes
- Exploration of alternative cost functions
- Implementing Generative Adversarial Networks (GANs) for climbing route generation

References

[1] Phillips, C., Becker, L., & Bradley, E. (2012). Strange Beta: An Assistance System for Indoor Rock Climbing Route Setting. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, *22*(1), 013130.

[2] Winship, C., & Mare, R. D. (1984). Regression Models with Ordinal Variables. *American sociological review*, 512-525.

[3] Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, *15*(1), 1929-1958.