A fair grade: assessing difficulty of climbing routes through machine learning

Lindsay Kempen
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
lindsaykempen@gmail.com

ABSTRACT

In the sport of climbing and bouldering, difficulty grades are important for skill assessment, enjoyment and safety. Although there are standardized scales to minimize inconsistency, the difficulty assessment is performed by humans and sensitive to subjectivity. Therefore, we have developed a tool that predicts the difficulty grades of climbing routes using machine-learning. The routes are described in a semi-natural Domain-Specific Language, which can be parsed into symbol sequences. Here, a symbol represents a climbing move. The symbol sequences are then used as inputs to a variable-order Markov models (VOMMs) based classifier. With the VOMM prediction algorithm Decomposed Context Tree Weighting (DE-CTW), we trained one VOMM on *Easy* climbing routes and one on *Hard* climbing routes. By calculating a test route's likelihood for both VOMMs, the *average log-loss*, we predict if a route is *Easy* or *Hard*. We have implemented six predictor variations to vary with interpretation detail in the symbolization process. After using 50-fold cross validation on 146 climbing routes, our best performing variation performed roughly as well as a trivial classifier. Still we believe this research's foundations are of interest for future research. We conclude with detailed explanations and proposed improvements.

Keywords

Climbing and bouldering, Sports technologies, Machine learning, Classification, Formalization, Variable-order Markov model, Markov chain, Domain-Specific Language, Binary classification

1. INTRODUCTION

The difficulty grades of climbing routes are subject to human evaluation and the accompanied subjectivity. In order to truly standardize these grades, we built a tool that attempts to predict difficulty grades.

1.1 Problem statement

The importance of a climbing route's difficulty grade is not to be overlooked. There are several different grade systems that help standardize grades. This not only helps climbers assess their skill level, it provides safety as well. Climbing routes outside one's comfortable grade range could cause severe injuries. In outdoor situations, injury even poses additional risks as less resources are available.

Route-setters assess the difficulty of the routes they create. A route-setter's level of strength or technique leads to personal bias however, meaning grades are subjective. A means of checking the actual consistency of these grades would be the first step towards true standardization. This helps one compare between different route-setters, climbing halls or even grading systems. With grade inconsistencies out of the way climbers will have a more accurate expectation of the route, which will naturally lead to a safer and more enjoyable experience.

The aim of this research, therefore, is to develop a tool that assesses the difficulty grade of a route.

1.2 Research questions

To check or achieve grade consistency, one would first need to investigate to what extent a climbing route grade can be assessed objectively. Therefore we drew up the following research question.

How can the difficulty of a climbing route be assessed by a computer?

This main research question is answered by means of three sub questions.

1. *How can a climbing route be characterized in a machine-interpretable language suitable for difficulty assessment?*
2. *What machine learning techniques are suitable for assessing the difficulty of a route specified in such a way?*
3. *How does the correctness of a difficulty grade classified by machine learning compare to the correctness of human-determined difficulties?*

1.3 Method and results

Section 3 describes literature research, based on which we decided to use the Decomposed Context Tree Weighting (DE-CTW) prediction algorithm. Section 4 explains how a tool was developed to convert relatively free-form input routes to a formal sequence of symbols and apply classification. Section 5 shows that the tool achieved an accuracy of 64.38% (95% confidence interval of [0.57, 0.72]), and thus barely outperforms a trivial classifier.

1.4 Implications

In section 6, we conclude that the question whether a computer can assess climbing route difficulty without sensors[1] is left unproven. Even though our tool was unable to correctly predict difficulty, we believe its foundations are

30th Twente Student Conference on IT Febr. 2nd, 2018, Enschede, The Netherlands.

still of important interest for future research. In section 7, we describe in great detail a multitude of factors that are believed to have negatively influenced the predictor's performance: the data quantity, the data quality and certain aspects of the formalization process (the conversion of climbing routes to a machine-interpretable form). We conclude that the concepts of formalizing climbing routes with a Domain-Specific Language, symbolizing them, and using Markov models-based machine learning are still promising.

2. BACKGROUND

2.1 Bouldering & climbing

Bouldering is a branch of climbing that is performed on relatively short and low routes, compared to regular climbing. It does not require belaying of the climber. Bouldering routes are known for having a puzzle-like nature. They are set with an intended strategy in mind, and figuring out this route information – called the beta – is deemed integral to the sport. Different difficulty grade systems are used for bouldering than for regular climbing. The popular American grade scale for bouldering is Hueco, and Yosemite Decimal Scale (YDS) for regular climbing. The first column in Table 2 shows what the grades on these scales look like.

Regular climbing and bouldering will be used interchangably during this research, but it is important to keep in mind these can be considered as different sports. The same holds for indoor and outdoor climbing – or bouldering, for that matter. Our research's methods concern indoor situations, but they are also applicable to outdoor contexts.

As explained, there are specific strategies to climbing routes. They depend heavily on the hand and foot holds, and the slope of the wall. Holds come in different types that require different approaches and they can be assembled in various positions and rotations. Besides, holds from the same type can differ considerably in facilitating the climber. All these factors influence a route's difficulty, which makes modelling a climbing route not a straightforward task. In this proposal, it is assumed the easiest strategy for a combination of holds (i.e. a climbing route) determines the difficulty grade.

2.2 Variable-order Markov models

Our tool makes use of Decomposed Context Tree Weighting (DE-CTW). DE-CTW is a general-purpose prediction algorithm that makes use of variable-order Markov models (VOMMs) [1]. A Markov model can be seen as a state-machine that has probabilistic transitions from one symbol to another. For a VOMM, the transition probability does not only rely on the new symbol and just one previous symbol, but the D previous symbols – called the *context*. D is the depth. There is no universally ideal value of D; it must be selected based on the purposes for which one's using the VOMM.

Markov models are widely used, as they can model any sequential concept and useful calculations can be made with them. Examples are sequence prediction and sequence classification [1].

2.3 Decomposed Context Tree Weighting

The DE-CTW algorithm can essentially predict VOMMs; it can assess the probability of a symbol succeeding a certain context, and it can assess the probability of a complete sequence by repeatedly using the prior calculation. The *average log-loss* is a measure of a sequence's likelihood. The lower it is, the more probable the test sequence is. The average log-loss $\ell(\hat{P}, x_1^T)$ for a test sequence $x_1^T = x_1 x_2 \cdots x_T$ and (the trained) model $\hat{P}$ is calculated as follows [1].

$$\ell(\hat{P}, x_1^T) = -\frac{1}{T}\sum_{i=1}^{T} log\hat{P}(x_i|x_1 \cdots x_{i-1})$$

Here, $\hat{P}(x_i|x_1 \cdots x_{i-1})$ is the probability that the symbol x_i would succeed the context $x_1 \cdots x_{i-1}$, according to the Markov model $\hat{P}$. DE-CTW learns these probabilities for its Markov model from training sequences.

The DE-CTW algorithm has a depth parameter. When learning with a depth of 5, the algorithm would ignore the first five symbols of the training route. DE-CTW will update the probabilities by looking at the remaining symbols, while taking the previous five symbols as context into account every time.

3. RELATED WORK

3.1 Other predictors

Ebert et al. made a very successful attempt at difficulty calculation in [3]. 98% of the tested routes were classified correctly through sensor data and machine learning. From sensors placed on the climber's limbs, they extracted information that modelled the features *control*, *stability*, *speed*, and *economical use of strength*. There are two critical remarks to be made on the applicability of the results however. The input data only contained routes of difficulty level 1 through 4 on the Fontainebleau scale[2]. These easiest levels might not be representative for the majority of the bouldering problems - bouldering halls might even lack the first two levels. Moreover, eleven of the eighteen climbers in the experiment have never climbed before, which results in a sheer lack of technique. We believe that makes them considerably less representative for the vast majority of climbers. The noticed data bias still leaves the prediction problem in a broader difficulty context open. Besides, the necessity of specific sensors complicates an approachable and wide-spread adaptation of the grade prediction tool, as is required for enabling overall grade consistency.

By contrast, [2] details a machine learning approach without sensors. They collected routes that have been set for the Moonboard, a standardized bouldering wall. This wall has instances across the world, all having an identical hold layout. In the software implementation holds could be recognized as being the same one, allowing the algorithm to match parts of routes. However, it still meant that a hold's type and rotation were completely neglected, while those properties are significant for climbing difficulty. This might have been the determining factor for the tool's classification accuracy of roughly 35%.

3.2 Rich climbing toolset

In [4], Philips et al. showcase a climbing route generator called Strange Beta. The outputted routes were determined as good or even better than human-built. Here, routes were expressed in the domain-specific language CRDL (Climbing Route Description Language). CRDL has been created for the purpose of generating routes. Herein, routes are ordered lists of moves, a move being a hold type and a suggestion to use one's left or right hand. Foot holds were

[1][3] was able to successfully predict climbing route difficulty in a limited context, using sensors.

[2]The Fontainebleau grade system ranks climbing routes on a scale from 1 to 9, where 1 is considered easiest.

neglected with the consideration that these only exist to enable the moves a climber's hands make. In order to reduce complexity, hold positioning was neglected as well. Through the later publicly available Strange Beta implementation, user-entered routes were gathered. These did not strictly comply to the CRDL syntax. Philips et al. used a self-built domain-specific parser to extract route information from these routes. This allowed extra information to be included, such as the quality of a hold and the size of a move. Though not fully formally tested, they have trained a Variable-Order Markov Model (VOMM) using DE-CTW [1] on these user-entered routes, of which the preliminary results are said to look promising. The VOMM was used for interpolation in the generated climbing routes, by adding moves to smooth over seemingly less natural transitions.

3.3 Conclusion

We have chosen to expand our research on [4]. As the authors suggested, we used a VOMM in combination with the richer CRDL notation for climbing route recognition. This way, we could calculate the likelihood of a route having been created by a certain VOMM. We simply compared these likelihoods (the log-loss scores) of VOMMs that have been trained on subsets of varying difficulty grades. This way, we could predict a route's grade. This implementation is usable without special sensor equipment, while trying to limit the loss of important domain-specific data. That is, features that directly influence a route's difficulty should still be included in the prediction, as opposed to [2].

4. METHODOLOGY

We have implemented a difficulty predictor in hopes of proving that a machine can assess a route's difficulty. This predictor will base its decisions on existing climbing routes and their human-determined grades by trying to find patterns in that knowledge base.

4.1 Data collection

As mentioned in section 3.3, we have determined an appropriate data format for capturing climbing routes and their relevant characteristics: Climbing Route Description Language (CRDL) [4]. In CRDL, one describes climbing routes as sequences of moves. A move is specified as a certain type of climbing hold being grabbed by a left or right hand. In special cases, a move refers to a different type of event, such as a 'match' – joining hands on one grip – or a 'toehook'[3].

Fortunately, we have been provided with user-supplied data that is collected within the Strange Beta application [4] as well as its source code. Of the climbing routes, after filtering those that did not include a difficulty grade or proper descriptions, 146 are useful for the classifier. The routes are of mixed disciplines; we used 74 bouldering routes and 72 regular climbing routes. This has two implications: the context of the routes are different, and the assigned grades are from two different grading scales. Section 4.3.3 describes how the classifier handles this separation in data.

4.2 Data processing

[3]The consideration of a climb being a simple sequence of hand movements neglects foot coordination. Even though CRDL usually assumes feet are not a distinctive factor, it still includes extraordinary feet strategies. For example, a route might require the climber to hook their toe behind a grip or other surface to exert additional force.

4.2.1 Symbolization

While all moves in the route transcriptions were sequential and included a left or right tag, their descriptions did not adhere to a strict formal form. An example is given in Listing 1, showing inconsistent syntax and various types of detail.

The provided Phoenix [5] parser is however able to extract the relevant information from the natural language snippets. Listing 2 shows a general idea of what is recognized within such a move description. The full grammar specification can be found in appendix A. Based on the parse tree, every move is assigned a symbol. For example, the programme would determine the fourth and fifth move in Listing 1 as semantically equal and thus the moves will lead to the same symbol.

```
1 R Jug
2 L reach to jug
3 R match
4 R to good sidepull
5 L good sidepull
6 R to finger pocket
7 L go to flat edge crimp
8 R big cross through to jug
9 L moves upright to angled gaston crimp
10 R toe hook
```

Listing 1. Example of a parsable user-transcribed climbing route (fabricated). A line describes one move by its sequence number, a left or right hand tag, and a free-form description.

```
[Move]
    ([Action] [Hold])
    ([Hold] [Action])
    ([Hold])
    ([Match])
```

Listing 2. Top node in context-free grammar specification for a climbing move, as implemented in Philips et al.'s parser. [4]

4.2.2 Symbol sets

The set of all symbols used is called the *symbol set*. An important question is which details should be incorporated in a symbol. Descriptions can have a varying level and type of detail. Inclusion of detail is a trade-off with direct consequences for a classifier's performance.

For example, if only hold types are considered in the case of Listing 1, the same symbol would be assigned to moves 1, 2 and 8. This would neglect the move difficulty that is implied by the texts "reach to" and "big cross". On the other hand, if different symbols are assigned to moves 1, 2 and 8, it is no longer retraceable that they all refer to the same type of hold: a jug. A jug is characterized as easy and sturdy to grab. Such a hold would not only facilitate the current move, but the next one as well. Having one hand on a jug, the climber benefits from high stability. In the context of predicting routes this would mean difficult moves are likely to be preceded by a jug. It is however harder to recognize such patterns when jugs are split into different symbols. For every jug variation, the programme would need enough data to effectively learn this pattern.

The consequences of the detail trade-off heavily depends on the available data. We have not been able to concretely research this relation, but we believe there are several factors: the data quantity, consistency of detail and the type

Table 1. Comparison of the six symbol sets.

$\sum$	Description	$\lvert\sum\rvert$	Example symbol
1	Hold types	97	`sidepull`
2	Hold types with precise descriptors	244	`sidepull deep angled`
3	Hold types with quality booleans	163	`sidepull (big move) (good hold)`
4	Hold types with precise descriptors and quality booleans	301	`sidepull deep angled (big move) (good hold)`
2*	Hold types with generic descriptors	171	`sidepull big good`
4*	Hold types with generic descriptors and quality booleans	229	`sidepull big good (big move) (good hold)`

of routes described. Our approach to this issue is simple; the application builds multiple symbol sets that represent different types of detail [4]. Table 1 shows an overview of six symbol sets, which are an expansion from Philips et al.'s implementation. We denote a symbol set by $\sum$, and its size with $|\sum|$. The example symbols show how the input "throw to a deep angled sidepull" is symbolized differently for every set.

Symbol set 1 through 4 are taken from [4], with a slight modification in implementation[4]. Sets 2* and 4* are our modifications from sets 2 and 4 respectively. In these six symbol sets, we apply more detail with the following properties.

- The descriptor refers to the shape and size of a hold. The precise descriptor specifies between kinds of shapes and sizes.
- The generic descriptor aggregates these kinds of shapes and sizes. The hold shape is incorporated by appending `bad`, `good` or nothing to the hold type. The same goes for the hold size, resulting in a `big`, a `small` or no postfix.
- A set of quality booleans is determined per move. For every quality found to be true, its shorthand is appended.
  - `big move` – Set to true if either the action verb or action size in the description points to a strenuous move. Not all move descriptions mention actions, as they may only describe the climbing holds. Examples of action verbs and action sizes can be found in our grammar specification (appendix A).
  - `hold goodness` – Set to true if there are more positive hold shape and hold size descriptors than negative ones.
  - `cross` – Set to true if the action mentions a cross: the act of crossing one's limbs.

4.3 Classification

4.3.1 Models in the process

Our predictor is a classifier; it predicts the class of the input, in this case the difficulty class of a climbing route. Notably, the classifier uses variable-order Markov models under the hood. For making a prediction the program goes through the following steps. Note that this procedure is undertaken for every symbol set, meaning we have six predictions: one for every symbol set.

1. Train the models which represent different classes.
2. Test the input on every model. For each one, this yields a log-loss score.
3. Choose the model with the smallest log-loss. This is the predicted class.

Every difficulty class is represented by one Markov model. This is achieved by training the model on only that class' data.

In our classifier, testing a route means that we calculate the probability the route would exist given a certain model, i.e. knowledge base. The result is a log-loss score. A lower log-loss score indicates a higher probability of the model generating the input entry.

This is why we choose the model that returns the lowest log-loss score. This model is the most likely to generate the input. The difficulty class associated with the model is then the predicted class.

4.3.2 Model implementation

For the model training and testing we use the DE-CTW implementation from [1]. This was already included in the delivered code of [4] for climbing route interpolation purposes. As for training, we feed the DE-CTW algorithm symbol sequences that result from the parser. This determines the transition probabilities within the variable-order Markov model. With a built model, the DE-CTW implementation can calculate the average log-loss for a given test sequence. By trial and error, we have found that there is not a preferred value for DE-CTW's depth parameter for this project. The predictions and log-loss scores were barely different when varying with the depth value. Having to choose any value for depth, we picked five, as [1] showed seven served as a good value for other applications while a value of seven slowed our process significantly down.

4.3.3 Choosing classes

As can be derived from section 4.3.1, models play a crucial role in the classifier. An important choice is which classes these models should represent.

The grade scales used in the data set are Hueco (bouldering) and Yosemite Decimal Scale (regular climbing). Ideally, the grade predictor would be able to determine an exact grade, such as 'V4' or '5.11'. A considerable number of Markov models would need to be trained; one for every possible grade. We do not possess sufficient data entries per grade in order to do this, as can be seen in Table 2. Models based on just several routes would not be representative. They would be overfitted to the unvaried training data. Therefore, the classifier predicts a grade range instead. This reduces the number of classes and thus increases the amount of data per class.

Further reduction in classes can be achieved by merging the grade scales, but this comes at a cost. Bouldering and regular climbing are considered different disciplines, which means conversion is neither straightforward or without loss of quality. For example, endurance plays a significantly larger role in regular climbing than bouldering. YDS grades might reflect that factor while Hueco grades

[4]Even though the original parser of [4] recognizes action verbs as well as action sizes, their implementation of 'big move' did not involve the action size.

Table 2. Grade distribution of the data set.

Grade	Count
Hueco	
V0	5
V1	2
V2	7
V3	13
V4	23
V5	10
V6	3
V7	5
V8	4
V9	1
V10	1
Total	74
YDS	
5.7	2
5.8	5
5.9	0
5.10	44
5.11	12
5.12	9
Total	72

might not. As no conversion could be found in climbing documentation, we built one based on closely similar descriptions provided by several individuals[5].

The first two columns of Table 3 presents the result. This is by no means a strict or official conversion, but that is not considered an issue for this project. The data is split in just two classes, *Easy* and *Hard*, minimizing the number of 'edge cases', i.e. grades of which the class is debatable. The rightmost column shows the Easy class has considerably more entries than the Hard class. A more even class assignment was however not feasible with the given data set and conversion chart.

4.4 Analysis

4.4.1 Cross validation

The predictor's performance is evaluated by k-fold cross validation. This means we split our data into k groups. For k rounds, we select one group as test set while the remaining groups serve as training set. A k of 50 yields 50 groups of three or two climbing routes. That means the training set size will be 143 or 144 during every round. By evaluating every climbing route on almost the whole data set, we estimate as accurately as nearly possible how well our predictor performs. It's important to note our form of cross validation is not *stratified.* In other words, the groups do not have a fixed ratio of *Easy* and *Hard* routes.

The result of the cross validation is a confusion matrix that contains every tested route. To compare varying levels of detail, we do the prediction process for every one

Table 3. Class distribution of the data set.[6]

Grade		Class	
Hueco	**YDS**	**Name**	**Size**
V0-	5.7, 5.8	*Easy*	90
V0	5.9		
V0+	5.10		
V1	5.11		
V2			
V3			
V4	5.12	*Hard*	56
V5			
V6			
V7	5.13		
V8			
V9			
V10	5.14		
V11			
V12			
V13			
V14	5.15		

of our symbol sets. This yields six complete confusion matrices, each showing predictions done with a different symbolization approach.

4.4.2 Matthews correlation coefficient

We evaluate the predictor performance with the Matthews correlation coefficient (MCC), calculated from the confusion matrix. The MCC ranges between -1 (completely wrong) and 1 (completely right). A random classifier would on average score 0. As opposed to the more common classification measure accuracy, the MCC is resistant to significantly unequal class sizes.

In binary classification, the accuracy and the MCC are calculated as follows.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Here, TP is the number of true positives, or correctly classified *Easy* routes; TN the number of true negatives, or correctly classified *Hard* routes; FP the number of false positives, or incorrectly classified *Hard* routes; and FN the number of false negatives, or incorrectly classified *Easy* routes.

The *Easy* class counts 90 instances and the *Hard* class 56. As you can see, a totally uninformed classifier would still reach an accuracy of around 62% by always classifying *Easy*, whereas the MCC would not even be computable due to division by zero. The nearly identical case of $TP = 89$, $TN = 1$, $FP = 55$ and $FN = 1$ yields an accuracy of around 62% and a MCC of around 0.03. While the accuracy seems optimistic, the MCC shows this hypothetical classifier barely outperforms random classification.

We quantify the predictor's performance as the highest MCC of the six predictor variations.

4.4.3 Confidence interval

We also evaluate the predictor performance with a 95% confidence interval on its accuracy, by considering our predictor as a binomial distribution. We calculate this as

[5]For the sources, we used charts of the following authors: the bouldering and climbing gym Planet Granite in San Francisco (`https://www.do-not-panic.com/2012/03/v-scale-and-yds-conversion-chart.html`); Jon McCartie (`https://gist.github.com/jmccartie/891748`); Reddit forum user DCBarefootRun (`https://www.reddit.com/r/climbing/comments/3vun4x/`); and Mountain Equipment Co-op (`https://www.mec.ca/en/explore/climbing-grade-conversion`). All information was retrieved from the links on the 21st of February, 2019.

[6]A minus or plus suffix for a grade indicates that a route is on the respectively easier or harder side of the grade.

Table 4. Predictor results for varying levels of interpretation detail.

$\sum_1$		Predicted	
		Easy	Hard
Act.	Easy	76	14
	Hard	47	9

$\sum_2$		Predicted	
		Easy	Hard
Act.	Easy	81	9
	Hard	45	11

$\sum_3$		Predicted	
		Easy	Hard
Act.	Easy	66	24
	Hard	35	21

$\sum_4$		Predicted	
		Easy	Hard
Act.	Easy	76	14
	Hard	38	18

$\sum_{2*}$		Predicted	
		Easy	Hard
Act.	Easy	80	10
	Hard	47	9

$\sum_{4*}$		Predicted	
		Easy	Hard
Act.	Easy	75	15
	Hard	40	16

Table 5. Predictor performance for varying levels of interpretation detail.

$\sum$	**Accuracy**	**CI-95%**	**MCC**
1	58.22%	$[0.50, 0.66]$	0.01
2	63.01%	$[0.55, 0.71]$	0.14
3	59.59%	$[0.52, 0.68]$	0.11
4	64.38%	$[0.57, 0.72]$	0.19
2*	60.96%	$[0.53, 0.69]$	0.07
4*	62.33%	$[0.54, 0.70]$	0.14

follows, with the normal approximation interval.

$$CI = [\hat{p} - z\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}, \hat{p} + z\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}]$$

In our situation, $n = TP + TN + FP + FN$ and $\hat{p} = \frac{TP+TN}{n}$. With a confidence of 95%, we use $z = 1.96$.

We also quantify the predictor's performance as the highest confidence interval of the six predictor variations.

5. RESULTS

Table 4 shows our predictor's results. Every confusion matrix summarizes the prediction outcomes of one 50-fold cross validation experiment. The six experiments only differ in which symbol set has been used, allowing the tool to evaluate climbing moves in varying levels of detail. The accuracy and Matthew correlation coefficient that accompany these confusion matrices are given in Table 5.

A complete random classifier would have an average accuracy of 50%, and a simple classifier that always determines the largest class (*Easy*) would be right around 62% of the time. Looking at the confidence intervals, we see that with 95% confidence every predictor variation outperforms a random classifier, except for the predictor using symbol set 1 where there is a chance of performing equally well. However, we are not able to state with 95% confidence that any of our predictors outperforms the aforementioned simple classifier.

The Matthews Correlation Coefficient points towards a similar verdict. All values are considered close to 0, which is the average MCC for a random classifier. The scores are however above 0. This is, although small, still a positive indicator that some correlation is possibly found, i.e. the classifier might have made slightly informed decisions.

6. CONCLUSION

In this section, we draw conclusions and summarize the research by answering the research questions as stated in section 1.2.

6.1 Formalization

How can a climbing route be characterized in a machine-interpretable language suitable for difficulty assessment?

We were able to formalize climbing routes with CRDL [4] and symbolize them for varying interpretation levels of detail. This formalization was applied to 146 user-entered climbing routes [4], using a Phoenix [5] parser to handle the not strictly formal nature of the data. The grammar specification (see appendix A) highlights the relevant climbing aspects, which allows for easy symbolization of every climbing move.

6.2 Machine learning

What machine learning techniques are suitable for assessing the difficulty of a route specified in such a way?

We have found that, when its moves are symbolized, climbing routes can be used for machine learning. In this case they were fed to the general-purpose prediction algorithm DE-CTW [1]. DE-CTW uses variable-order Markov models (VOMMs), which can be considered as probabilistic state machines. We used DE-CTW to train an *Easy* VOMM and a *Hard* VOMM. The VOMM with the lowest log-loss score for the test route was the predicted grade class.

6.3 Correctness

How does the correctness of a difficulty grade classified by machine learning compare to the correctness of human-determined difficulties?

We made six predictor variations by using six different symbolization methods, each distinguishing climbing moves with another level of detail. We used 50-fold cross validation and the measures of the Matthew Correlation Coefficient and the binomial proportion 95% confidence interval to evaluate the predictor performance. Although our methods seemed adequate beforehand and deliberate thought has been put into correctly modelling the domain of climbing, the predictor disappointed. The best predictor variation had an accuracy of 64.38%, a CI-95% of [0.57, 0.72] and a MCC of 0.19. With 95% confidence, we can state our classifier outperforms a random classifier ($accuracy = 50\%$), but not a simple classifier that always predicts the *Easy* class ($accuracy \approx 62\%$ – the classes are uneven). The MCC shows the classifier barely outperforms a random classifier.

6.4 Main question

How can the difficulty of a climbing route be assessed by a computer?

The question whether a computer can assess climbing route difficulty without sensors[7] is left unproven, as our tool was unable to correctly predict difficulty. The predictor's performance is not sufficient for setting new consistency standards. We however believe its foundations are still of important interest for future research. A multitude of factors (mentioned in section 7) are believed to have negatively influenced the predictor's performance: the data quantity, the data quality and certain aspects of the formalization process. Crucial aspects of this research such as the formalization or machine learning approach could still be useful for future work with only minor adjustments. The CRDL notation and symbolization, for example, is a good starting point for further research in formalizing climbing routes. It raises the question what exactly is important within a route.

7. DISCUSSION

In this section, we try to critically explain the results found. The following factors are thought to have negatively influenced the predictor's performance.

7.1 Data quantity

The size of the data set, 146 climbing routes, is decidedly small. Admittedly, the DE-CTW algorithm significantly increases the number of data points. Using a depth of five, it considers every symbol after the first five in a sequence as a data point. Still, the limited data does not help recognizing patterns. As an example we take the very common hold type sidepull. In the data set there are 147 sidepulls. 74 of them are however joined with another hold type, yielding another symbol, such as `pocket sidepull`. Thus we have 73 `sidepull` symbols. These are then split over the two classes, roughly halving the available data for both models. As you can see in Table 1, we have 97 hold types. This means a lot of combinations of a sidepull with a few preceding hold type symbols are possible. We believe many patterns that may occur naturally are most probably underrepresented, because of this symbol set size. Additionally, 61 of the hold types are in fact 'composite', such as `pocket sidepull`. This has two consequences: these composite types have a considerably low occurrence making it hard to recognize patterns, and it takes away data from its more popular components, such as `sidepull`.

Another consequence of the small data set is that we needed to merge two not officially compatible grading scales. As Hueco and YDS are for different disciplines, no official conversion chart could be found. We made our own, which is prone to errors. To add, any inter-discipline grade conversion is thought to have (inherently) brought about noise to the data anyway.

7.2 Data quality

Several aspects might negatively affect the data set's quality. The climbing routes have been transcribed by multitude of users on [4]'s website. This means there is no enforced consistency in the determined difficulty grade, the style of the climbing routes and in the notation style.

Firstly, as difficulty grades are inconsistent, it is in fact not ideal to compare our classifier's predictions with the grades determined by humans. The 'true grades' in our data sets are not guaranteed consistent with each other. It might be the case that our classifier has actually predicted every grade in a consistent manner, which satisfies the eventual goal, but the results do not match the inconsistent 'true grades'.

Secondly, the style of climbing routes poses a problem when there's an imbalance of styles; the training data might lack routes of an underrepresented style in order to correctly classify certain routes.

Thirdly, notation style plays a huge role in building a useful knowledge base. The freedom of the CRDL notation allows the same route to be transcribed in various ways. One could leave out details about the hold size or shape, or what kind of action is performed in order to grab the hold (e.g. "toe hook to jug"). Holds come in many unlabeled forms, and it is up to the transcriber's beliefs what aspect of a hold is relevant, e.g. what hold type it is in the route's context. Moreover, one could mention multiple hold types for one move; this yields a composite hold type symbol. A climber may find a certain hold good or a certain action strenuous while another doesn't, purely based on their individual experiences and climbing qualities. Ironically, the bias problem that raised our research question has appeared in our tool.

Another problem is that the transcriber may note the wrong approach (beta) for a climbing route. A climbing grade reflects the *best* way to pass the route. If the transcriber does not use the same beta as the route-setter did when grading the route, there's a high probability their noted approach is less efficient. As CRDL is actually a record of a route's beta, this could result in a CRDL entry that has been labeled with a too low grade.

7.3 Formalization

The placement of all limbs are relevant during climbing. The CRDL notation however leaves out foot placement, while it can make a significant difference. Most climbing techniques revolve around strategically placing one's feet to keep balance or exert power. One cannot note in CRDL how much and how exactly a climber is relying on their feet. Taken to the extreme, they might need to campus (climb without using feet) a part for a certain route. This is considered very exhausting and advanced. The route's transcription would however not mention it, as campusing is not part of the syntax. We could add 'campus' as a type of action to the vocabulary, but CRDL would still lack clear space to specify more nuanced feet placement.

Arm placement is less neglected but lacks important detail as well. While CRDL requires every move to be tagged with a choice of hand, left or right, this is not incorporated into the symbolization process. This is already a problem in the following simple example. A sidepull is a hold type that is best gripped with a specific hand while leaning into the opposite direction. A climber might need to grab a left-facing sidepull with their right hand, while the hold is normally best grabbed with the left one. The second example shows a more generic issue: it could be that two routes have the same type of holds in the same order and the same type of actions required for reaching every hold, but the routes are still climbed completely different. The spatial placement of a hold in one route might require a climber to use a different hand than they would in the other route. Maybe the climber needs to move the same hand consecutively before the other one – not a rare occurrence in our data set. However, none of this is retraceable in the formalized climbing routes.

CRDL leaves out a considerable amount of spatial detail. Firstly, it does not log the amount of overhang a route has. Overhang is the angle the climbing wall is offset from a regular vertical wall. More overhang makes a route considerably more difficult, forcing a climber to rely less on gravity and more on their arm strength. Strategies exist to mitigate these effects, but the need to apply these is an indicator of a route's difficulty. Secondly, there is no spatial information on the climbing holds used. The exact angle and location of the whole hold, or of a feature on a hold can be of important difference. The climber might not be able to grab a hold in an straightforward manner, and they would need to apply certain techniques, extra strength or balance. Thirdly, climbing walls are oftentimes not a flat surface. Outdoor climbers mount rocks that are have arbitrary natural forms, and indoor climbers can come across modules. These are extruding wall parts, attached to the wall. Holds can be attached to modules as well. Logically,

[7][3] was able to successfully predict climbing route difficulty in a limited context, using sensors.

the existence of a module can greatly influence a route's difficulty. This can however not be noted in CRDL.

The symbolization process does not consider the action verbs or action sizes in a route on a detail-level. The action verbs and action sizes are only incorporated in the `big move` quality boolean present in symbol sets 3, 4 and 4*. Although not enough data is currently present to justify distinguishing between different types of action, the specific are relevant for a route's difficulty. One might have specified a move is a 'dyno', which means that the climber needs to climb dynamically and use momentum. This technique is considered advanced, but our predictor wouldn't be able to pick on that.

7.4 Modeling the crux

We model climbing routes as a sequence for Markov models, because of their evident sequence structure and their further workings: for every climbing move, the preceding ones influence the difficulty. We expect a similar difficulty when we take the same climbing move with context to another climbing route. There is one inherent problem in mapping difficulty grades to sequences. Routes can have a 'crux', a part that is particularly hard. The crux would determine the route's grade, while the remainder of the route is below that level. In the training phase, our predictor would incorrectly learn these simpler moves as more advanced ones.

8. FUTURE WORK

There are still many questions to investigate in the field of climbing. We believe the best continuation of our work requires the disciplines of climbing and computer science (or mathematics) to join hands.

8.1 Predictor expansion

We see potential in our predictor, but it requires tweaking in various ways. One can tackle the issues mentioned in section 7 in the following ways. The tasks deemed most important are marked with an asterisk.

- (*) More basic validation of the classifier could further demonstrate the correctness or potential of the classifier, e.g. by testing on (fabricated) routes that are clearly *Easy* or *Hard*.
- (*) Increase the amount of CRDL data in general, and make sure there is sufficient data representing different climbing styles and difficulty grades.
- (*) Let one route-setter regrade the CRDL climbing routes, ensuring grade consistency.
- In the same light, further research the issue of routes possibly having cruxes (a considerably harder part in the route).
- Approach composite hold types differently. They could be symbolized differently by 'downgrading' to a single hold type; e.g. `jug sidepull` could be converted to `jug`. Alternatively, the VOMM could repeat the same operation for every component of a composite hold type. For example, instead of learning `[crimp, jug, jug sidepull]`, it would learn both `[crimp, jug, jug]` and `[crimp, jug, sidepull]`.
- The symbol sets and grammar can be further tweaked to improve the formalization process. Currently, it does not regard the left/right hand tag that every climbing move has.
- (*) The CRDL notation could be expanded or replaced, in order to include relevant information such as foot placement and spatial information.
- Research machine learning alternatives to DE-CTW.
- Focus on only classifying between lower grades (easier routes), as the differences between lower grades are more noticeable than in higher grades. In more difficult routes, the difference between grades are oftentimes more (spatially) nuanced, concerning weird angles or positioning.

8.2 Formalization of climbing domain

We see this project as a starting point for research in purely the climbing domain as well. Formalization of climbing routes is interesting for many purposes: the storing and sharing of climbing routes, structuring climbing-centered science and finding the essence of climbing.

8.3 Beta calculation & Image recognition

Looking a bit further, there are several related computer science projects to take on. Our project retrieves the beta (best approach) of a route by letting the transcriber write this down in CRDL. This brings some form of subjectivity in our predictor's data. An ambitious project would be to calculate a route's beta. The input would need to capture all the available material for a route: every detail of the wall's surface and of every climbing hold. Extracting this information from, for example, several pictures or a video as well as efficiently storing this (inevitably 3D) information would already pose a tremendously difficult task. If this project would succeed, climbers could share and analyze climbing routes by the touch of a button. If a successful predictor that works similarly as ours – beta as input, difficulty grade as output – has been implemented, these projects could be connected. The user would no longer need to perform the somewhat subjective and pesky task of transcribing the test route. Alternatively, the intermediate step of finding the beta might not be necessary for finding a route's difficulty grade from images. One could attempt to apply machine learning with graded images as input.

9. ACKNOWLEDGEMENTS

I would like to thank my supervisor Enno Ruijters and my track chair Arend Rensink for supervising the project and its process. My appreciation goes out to their open and supportive attitude, which lowered the bar for treading the less explored field of climbing in computer science, and for helping me throughout the research.

I am certainly thankful for Caleb Philips, author of [4], who has been very helpful when I reached out to him. Without his contribution of data and code, the project could not have been anywhere near what it is.

I am glad with the advice of Frank Stapel, a climbing fanatic and computer science student, during the research's initial and final phases. This helped greatly with applying the field of computer science to climbing.

Lastly, I thank my friends for their neverending support and advice.

References

[1] R. Begleiter, R. El-Yaniv, and G. Yona. On prediction using variable order markov models. *Journal of Artificial Intelligence Research*, 22:385–421, 2004. `https://doi.org/10.1613/jair.1491`.

[2] A. Dobles, J. C. Sarmiento, and P. Satterthwaite. Machine learning methods for climbing route classification. *Student project report, CS 229 Machine Learning, University of Stanford*, 2017. `http://cs229.stanford.edu/proj2017/final-reports/5232206.pdf`. Accessed 2nd of December 2018.

[3] A. Ebert, K. Schmid, C. Marouane, and C. Linnhoff-Popien. Automated recognition and difficulty assessment of boulder routes. In *International Conference on IoT Technologies for HealthCare*, pages 62–68. Springer, 2017. `https://doi.org/10.1007/978-3-319-76213-5_9`.

[4] C. Phillips, L. Becker, and E. Bradley. Strange beta: An assistance system for indoor rock climbing route setting. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(1):013130, 2012. `https://doi.org/10.1063/1.3693047`.

[5] W. Ward. Extracting information in spontaneous speech. In *Third International Conference on Spoken Language Processing*, 1994.

.

APPENDIX

A. GRAMMAR SPECIFICATION

Below is the context-free grammar specification for a climbing move, which we have used to parse CRDL routes. It is a slight modification from [4]'s grammar.

```
[Move]
        ([Action] [Hold])
        ([Hold] [Action])
        ([Hold])
        ([Match])
;
[Match]
        (match)
;
[Hold]
        ([HoldSize] [HoldShape] [HoldType])
        ([HoldShape] [HoldSize] [HoldType])
        ([HoldShape] [HoldType])
        ([HoldSize] [HoldType])
        ([HoldType])
;
[HoldSize]
        ([HoldSizeBig])
        ([HoldSizeSmall])
;
[HoldSizeBig]
        ([HoldSizeBigT])
        ([Not] [HoldSizeBig])
;
[HoldSizeBigT]
        (big)
        (good)
        (manageable)
        (managable)
        (deep)
        (positive)
        (goodish)
        (okay)
        (ok)
        (solid)
        (decent)
;
[HoldSizeSmall]
        ([HoldSizeSmallT])
        ([Not] [HoldSizeBig])
;
[HoldSizeSmallT]
        (mini)
        (shallow)
        (small)
        (bad)
        (razor)
        (shitty)
        (tiny)
        (micro-dick)
        (transition)
        (nonexistent)
;
[HoldShape]
        ([HoldShapeGood])
        ([HoldShapeBad])
;
[HoldShapeGood]
        ([HoldShapeGoodT])
        ([Not] [HoldShapeBad])
;
[HoldShapeGoodT]
```

```
        (starting)
        (vertical)
        (bulbous)
        (angle)
        (sideways)
        (double sided)
        (right angle)
        (left angle)
;
[Not]
        (not)
        (no)
;
[HoldShapeBad]
        (roof)
        (slopey)
        (sloping)
        (vertical)
        (finger)
        (diagonal)
        (angled)
        (gaston)
        (flat)
        (downward)
        (down ward)
        (open hand)
        (openhand)
        (reachy)
;
[HoldType]
        ([HoldTypeT])
        ([UnderCling])
        ([SidePull])
        ([DownPull])
        ([FootHook])
        ([GenericHold])
        ([Layback])
        ([Mantle])
        ([Jib])
;
[HoldTypeT]
        (jug)
        (pocket)
        (crimp)
        (edge)
        (sloper)
        (cobble)
        (crimper)
        (crimpbeam)
        (beam)
        (layback)
        (horn)
        (ball)
        (boobies)
        (slope)
        (pinch)
        (bucket)
        (rail)
        (ear)
        (cup)
        (flake)
        (thumb catch)
        (slot)
        (gaston)
        (dish)
        (ledge)
        (incut)
        (teeth)
```

```
        (arete)
        (tufa)
        (hand jam)
        (fist jam)
        (finger jam)
        (mono)
        (offwidth)
        (chicken head)
        (knob)
        (handle)
;
[Mantle]
        (topout)
        (top out)
        (mantle)
        (finishing hold)
        (top)
        (finish)
;
[GenericHold]
        (hold)
        (hand)
        (feature)
        (grip)
        (start)
;
[Layback]
        (lay back)
        (layback)
        (lie back)
        (lieback)
;
[Jib]
        (jib)
        (gib)
        (churd)
;
[SidePull]
        (sidepull)
        (side pull)
;
[DownPull]
        (downpull)
        (down pull)
;
[UnderCling]
        (undercling)
        (under cling)
        (cling)
;
[FootHook]
        (heel hook)
        (heelhook)
        (toe hook)
        (toehook)
        (bicycle)
;
[Action]
        ([ActionSize] [ActionVerb])
        ([ActionVerb])
;
[ActionVerb]
        ([ActionVerbBig])
        ([ActionVerbSmall])
;
[ActionVerbSmall]
        ([ActionVerbSmallT])
        ([FootHook])
```

```
          ([Layback])
          ([Cross])
;
[ActionVerbSmallT]
          (bump)
          (out)
          (up)
          (left)
          (right)
          (fondle)
          (grab)
          (roll)
          (over)
          (diagonal)
          (slide)
          (grab)
          (go again)
          (go)
          (move)
;
[Cross]
          (cross over)
          (cross under)
          (crossover)
          (crossunder)
          (cross)
;
[ActionVerbBig]
          (throw)
          (dyno)
          (dynamic move)
          (reach)
          (fall into)
          (huck)
          (deadpoint)
          (rock)
          (dead point)
          (drop knee)
;
[ActionSize]
          ([ActionSizeBig])
          ([ActionSizeSmall])
;
[ActionSizeBig]
          (big)
          (huge)
          (far)
;
[ActionSizeSmall]
          (small)
;
```

Listing 3. Context-free grammar specification for a climbing move