Machine Learning Engineer Nanodegree

Capstone Proposal

Dominik Lindenberger November, 2018

Proposal

The goal of this project is to build a recommender system for climbing routes within a given climbing area. The system will take a climbers preference and his or her stats into consideration.

Domain Background

My Capstone Project is located in the world of **rock climbing**.

> *Rock climbing is an activity in which participants climb up, down or across natural rock formations or artificial rock walls. The goal is to reach the summit of a formation or the endpoint of a pre-defined route without falling.* [1]

Here is a video of a rock climber on a famous route called Action Directe.

In recent years rock climbing has gained a lot of traction and is becoming more popular as an outdoor sports for a larger number of people. As such, there exists a potentially large audience who could be interested in this project.

Unlike the climber from the video above most people do rock climbing as a leisure activity. Therefore, time they can spend to go climbing is limited. As a hobby climber planning a climbing trip it is quite common to have a time window of round about 1-3 weeks at the climbing destination of choice. The majority of climbing areas boast a large number of routes to climb, far greater than anyone can achieve within a single trip. For example, the climbing area 'Frankenjura', one of Germany's largest climbing area, contains more than 10,000 climbs. It is therefore in the interest of the climber to know and attempt those climbs that are most enjoyable to her or him and just skip the rest.

Often times there is a common agreement on what are the 'best' routes in a certain area and they are common knowledge. Most guide books contain some sort of list of the "Top climbs" in that region. However, climbers often differ in their personal preferences. Some prefer to climb on small crimpy holds, while others enjoy long dynamic movements on large holds.
Additionally, for some type of rock, the climbs deteriorate over time as heavy traffic 'polishes' holds with formerly good and important friction.

Therefore, climbers would benefit from a personal recommendation of the best climbs within a climbing

area, that match their climbing level as well as their personal preference.

Meanwhile there are many websites where climbers can track their climbs and ascents in a virtual logbook and also rate the quality of the route. Most famous websites of this kind are

- 8a.nu
- theCrag.com
- UKClimbing.com

With this project, I will explore the database of 8a.nu - published on Kaggle - to establish a personal recommendation system for climbers venturing into new climbing areas.

There has been quite a lot of academic research on the topic of recommender systems as well. Here is a short excerpt of publications:

- Su and Khoshgoftaar provide a good introduction into CF in their article A Survey of Collaborative Filtering Techniques
- In their 2018 paper Christakopoulou and Karypis provide an overview of Local Latent Space Models for Top-N Recommendation

As I am an avid climber myself, this project is something I take a strong personal interest in.

Problem Statement

The problem climbers face when first entering a new climbing area is which routes to go for. Typically there are a lot more climbs available than anybody can manage to climb in a typical climbing vacation. And not all climbs are worthwhile of doing (e.g. boring route, dangerous to climb, lot of dirt and vegetation, etc.). For any particular climber it would be very interesting to have a list of the best routes in each area. Ideally sorted by attractiveness and matched with the climbers skill level.

Hence the structure of the problem is that of **Ranking problem** or more precise a **Collaborative Filtering problem**, where we try to suggest climbs based on a climber's similarity to other climbers. During model exploration we may apply both supervised (k-nearest neighbor) as well as unsupervised (k-means) algorithms.

Since not all climbers enjoy the same type of climbing, such a list of attractive climbs would ideally be tailored to every individual climber. Such a system of personal recommendation does not exist so far.

Input to the problem will be

- list of users (climbers)
- list of climbs (ascents)
- list of user ratings per climb

As an output we expect to get

- list of top n recommendations per user (climber)

A note on collaborative filtering vs content based filtering. In our case content based filterin is not considered, because there is very limited information on the routes itself. This could have been different, had there been more route information such as rock type, height, climbing style, indicator for polishedness, etc.

Datasets and Inputs

On Kaggle a big data set of logged ascents from the popular website 8a.nu is available[2]. The data set contains *all of the publicly available information from http://www.8a.nu, the world's largest rock climbing logbook.* Climbers from all over the world have logged ascents in various climbing areas on this website.

In more detail the data set contains four tables with information as following.

Table	Description
ascent	Holds information about individual ascents (= completed climbs) climbers have made. Has fields such as • `crag` - the climbing area • `sector` - a specific sector within a climbin area • `name` - the name of a route in a particular sector • `rating` - user rating for the route
grade	Contains a list of all the different climbing grades, i.e. the difficulty of a climb. (The difficulty of the route "Action Directe" from the video above is 9a which is out of reach for 99% of all climbers.)
method	Defines the different styles of ascents. See this external article[3] if you are interested in more detail
user	Information about climbers such as `birth`, `country`, `height`, etc.

The original data set from Kaggle is 196 MB. For this project I will use only a subset, the area "Frankenjura" (crag = "Frankenjura") which is one of the largest and most famous climbing areas in Germany.

For this proposal I will provide a subset of the data and tables already joined.

Our data is very high dimensional. On the one hand side we have a 'small' number of users (around 3.300) and a large number of logged climbs (close to 115.000). About half of the climbs (approx 47%) are rated.

The number of logged climbs is also the total number of examples, i.e. data points in the dataset.

Labels in this data set are basically the user ratings of climbs. Since only about 47% of climbs are rated, we need to maintain that ratio of rated vs unrated climbs when doing train-test-validation splits.

For the training set all items are submitted to the model. For validation and testing I need split ratings per user into

- Observation subset - ratings submitted to the model
- Testing subset - ratings used to evaluate predictions.

Solution Statement

My goal is to create a recommender system for climbers that can suggest a list of *n* climbing routes within a particular climbing area. For an individual climber the recommender should base recommendations on the types of routes the climber liked before and based on what other climbers with a comparable climbing history have liked. The recommended climbs should be ordered by their attractiveness to the climber.

Although, our solution presents the top *n* climbs, the problem is basically a rating prediction problem. We want to predict the rating a climber would give an unrated route. Then the top n rated climbs would form the top n recommendations.

For climbers without any personal climbing record, a simple recommendation should give a list of the top n climbs within the area based on average ratings from all climbers.

In particular I will explore **collaborative filtering** techniques such as

- k nearest neighbor (KNN)
- k means clustering
- item based collaborative filtering
- matrix factorization (SVD)

Benchmark Model

There are no benchmark models for personal recommendations as this has never been done before.

We will therefore compare our model to the results of

- Random model - model makes random recommendations
- Top average model - model recommends top rated climbs based on mean average of all ratings
- Magazine rated - model pulls recommendations from the Top 100 climbs of Frankenjura as chosen by klettern.de [4], a well known German climbing magazine.

Evaluation Metrics

Basically, our problem is a ranking problem. We want to show the most attractive and suitable climbs to a user. However, we'll frame it as a rating prediction problem.
Using different models we will predict ratings of so far unrated routes. For this purpose we will use **RMSE** as our evaluation metric.

The project model as well as all of the benchmark models can be evaluated using RMSE, thus making this metric the favorable choice.

Project Design

When executing this project, I will follow those main steps

1 - Data exploration
I will start with some data exploration to gain insights about data distribution (e.g. how many climbs are rated vs unrated, number of ratings per user), outliers (e.g. climbs in very low or very high grades) and missing values (e.g. route name missing or nonsense/ cannot be matched with any actual route).

During data exploration I want to quantify how bad the situation is around misspelling in route names, e.g. misspellings of route *Baggi ned* (engl. *Won't manage*) such as *BAGGI AND*, *Baggi Nad*, *Baggi Net*.

2 - Data preparation
During preparation I will try to extract a list of unique routes. In the dataset a record contains currently the route that was climbed and if the climber provided it, also a rating. As such there will be many duplicate routes. String similarity measures like Sørensen–Dice coefficient or Levensthein distance should help here. Getting to a unique list of routes is important for creating a *climber X routes* matrix containing ratings.

In case a rating cannot be attributed to any route, e.g. in case route name is missing, I will drop those items.

3 - Model exploration
Durin model exploration I will try out a number of alternatives to predict ratings of unrated climbs. (Hence RMSE, as mentioned in an earlier paragraph) is a suitable metric to evaluate all of the approaches).

Starting out with a user-based approach which is a sort of k-nearest neighbors algorithm.

Set up *n X m* matrix consisting of the ratings of *n* climbers and *m* routes. Each element *(i, j)* represents the rating climber *i* rated route *j*.

For each route *j* climber *i* has not climbed yet, we find similar climbers to climber *i* (e.g. Cosine, Pearson) that have rated route *j*. Then I will calculate a rating based on rating of k nearest neighbors. We recommend the *n* top rated routes to the climber.

As an alternative I will be exploring data with **K-Means** algorithm and try to find natural clusters of climbers. The *Elbow method* should help finding a good number of clusters. The elbow method works by plotting the ascending values of k versus the total error calculated using that k.
For users within a cluser, I will recommend routes on other climbers'

As a third potential option I may look into **Item-based collaborative filtering**, which is a model-based algorithm that recommends items based on their similarity. Again this uses similarity functions such as Cosine and Pearson.

If it turns out that the above three methods do not yield satisfactory results, e.g. due to sparsity in data, I may look into a fourth option:

Apply SVD algorithm, e.g. using `surprise` package[5].

4 - Fine-tuning of the model Once the most promising model is found, I will continue to fine-tune the model to improve results. At this stage I will also employ evaluation metrics as mentioned in an earlier paragraph.

5 - Presentation of the solution
Finally, I will present results of the entire project and especially the solution to the problem.

---

1. What is Rock Climbing? ↩
2. 8a.nu Climbing Logbook ↩
3. Styles of ascent in sport climbing ↩
4. klettern.de's Top 100 Routes ↩
5. surpriselib.com/ ↩