**2019 电子科技大学美国数学建模竞赛模拟赛**

**承 诺 书**

#### 我们仔细阅读了数学建模竞赛的竞赛规则.

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮件、网上咨询等）与队外的任何人（包括指导教师）研究、讨论与赛题有关的问题。

我们知道，抄袭别人的成果是违反竞赛规则的, 如果引用别人的成果或其他公开的资料（包括网上查到的资料），必须按照规定的参考文献的表述方式在正文引用处和参考文献中明确列出。

我们郑重承诺，严格遵守竞赛规则，以保证竞赛的公正、公平性。如有违反竞赛规则的行为，我们将受到严肃处理。

我们的题目编号是（填写：A 或者 B 或者 C）： A

我们报名队伍号是： H170

参赛队员姓名学号(中文填写打印并签名):1. 马宗扬

1. 蒲正瑶
2. 孙宇 指导教师或指导教师组负责人 (有的话填写)：

是否愿意参加 2019 年美国赛（是，否）： 是

日期： 2018 年 11 月 15 日

Team # H170 Page -99 of 17

# Summary

This paper is aim to solve the problem of roller coaster evaluation. Based on KNN, Decision Rree, Multiple Linear Regression, EWM and TOPSIS evaluation method, an objective roller coaster rating algorithm in the absence of data is established. And applying this algorithm, an app owning “Search“ and “Recommendation“ functions to help potential riders find what they want is desinged.

Firstly, we correct the error data and fill in missing data by KNN Algorithm, Regression De- cision Tree and Multiple Linear Regression Algorithm, achieving good results. As for missing data, the prediction results using Decision Tree and Multiple Linear Regression Algorithm are more than 95% similar to the original data. Then, according to the influence of each index on the excitability, the positive correlation index is screened and reconstructed. An objective weighting method: Entropy Weight Method(EWM) is used to empower each index. Finally, the roller coaster rating algorithm can be obtained by applying TOPSIS comprehensive evaluation method.

Implementing our Roller Coaster Rating Algorithm, we get a “Top 10 Roller Coasters in the World“ list and a “Roller Coaster Score Sheet“. Comparing our ranking and grading results with the results of the “captaincoaster“ and “coasterbuzz“ websites, we discover that there are 3 and 2 repetitions in the “Top Ten“. Apart from the individual roller coasters, the rest of the roller coaster rankings are mostly in our ranking results. Both are located before 10%, and a small part is before 20%. Therefore, we prove that our algorithm is consistent with the subjective feelings of roller coaster “experts“. In addition, we also analyze the reasons for the difference from four perspectives: “Object Sample Size“, “Rating Criteria“, “Indicator Selection“ and “Weighting Evaluation methods“.

In order to help a potential app user find the roller coaster what they want to ride, we conceive the design of the app from two aspects. On the one hand, the app can automatically recom- mend roller coasters based on the information of the users‘ registration information and the preferences‘ similarity of other app users. This function is built in two steps.

Predict the potential users‘ scores on roller coasters using Granular Recommendation Algorithm.

*•*

Decide whether to recommend a roller coaster to the user based on Three-branch Gran- ular Recommendation Model.

*•*

On the other hand, the user can Retrieving the desired roller coaster for their own needs, search- ing according to the roller coaster “Thrill“ level, the comprehensive evaluation or the single index. This function can be implemented based on our proposed roller coaster rating algorith-

m. In addition, the function that users could search for roller coasters within a certain distance range is also considered in our app. Therefore, we realize the reasonable packaging of the app information to the user and satisfy the user-friendly design concept.

**Keywords**: Roller Coaster Ranking EWM TOPSIS Three-branch Granular Recom- mendation Algorithm

# Contents

1. [Introduction 1](#_TOC_250032)

[1.1 Problem Background . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1](#_TOC_250031)

[1.2 Restatement of the problem . . . . . . . . . . . . . . . . . . . . . . . . . . 1](#_TOC_250030)

[1.3 Analysis of the Problem . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1](#_TOC_250029)

[1.3.1 Develop Rating Algorithm . . . . . . . . . . . . . . . . . . . . . . 2](#_TOC_250028)

[1.3.2 Develop Top 10 list . . . . . . . . . . . . . . . . . . . . . . . . . . . 2](#_TOC_250027)

[1.3.3 Design a user-friendly app . . . . . . . . . . . . . . . . . . . . . . 2](#_TOC_250026)

1. [Assumptions and Notations 2](#_TOC_250025)

[2.1 Assumptions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2](#_TOC_250024)

[2.2 Notations . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2](#_TOC_250023)

1. [Preparation of Data 3](#_TOC_250022)

[3.1 Revise erroneous data . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3](#_TOC_250021)

[3.2 Fill in missing data . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3](#_TOC_250020)

[3.2.1 “Height““Speed““Length“ . . . . . . . . . . . . . . . . . . . . . . 3](#_TOC_250019)

3.2.2 “Duration“ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

[3.2.3 “Drop“ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5](#_TOC_250018)

[3.3 Deleting unavailable data . . . . . . . . . . . . . . . . . . . . . . . . . . . 5](#_TOC_250017)

[3.4 Convert words to data . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6](#_TOC_250016)

[3.5 Define average speed . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6](#_TOC_250015)

1. [Roller Coaster Ranking Algorithm 7](#_TOC_250014)
   1. [Calculating Weights using EWM . . . . . . . . . . . . . . . . . . . . . . . 7](#_TOC_250013)
   2. [Rating Algorithm based on TOPSIS . . . . . . . . . . . . . . . . . . . . . . 8](#_TOC_250012)
2. [Discussion of Roller Coaster Rating Sites 9](#_TOC_250011)
   1. [Roller Coaster Rating Results 10](#_TOC_250010)
   2. [Comparison and Discussion 11](#_TOC_250009)
      1. [Comparison 11](#_TOC_250008)
      2. Discussion 11
3. [Design User-friendly App 12](#_TOC_250007)
   1. [Concepts and Designs 12](#_TOC_250006)
   2. [Three-way Granular Recommendation Algorithm 13](#_TOC_250005)
4. [Non-Technical News Release 15](#_TOC_250004)
5. [Strengths and Weaknesses 16](#_TOC_250003)
   1. [Strengths 16](#_TOC_250002)
   2. [Weaknesses 16](#_TOC_250001)

[References 16](#_TOC_250000)

# Introduction

## Problem Background

In the early 1800s, the first roller coasters, where the train was attached to a wooden track, first appeared in France. Although wooden roller coasters are still being produced, steel roller coasters are more common and can be found on every continent except Antarctica.

Ranked by height, speed, length, inversions, and steepness, roller coasters are also rated through public opinion polls. Amusement parks often compete to build the tallest, fastest, and longest rides to attract thrill seekers and boost overall park attendance. However, many records do not usually last long. When Magnum XL-200, the first complete-circuit roller coaster built over 200 feet, opened in 1989, it began a new era of roller coasters and increased competition a- mong parks to set new world records. Also, to cater preferences of different consumer groups, abundant roller coaster ranking websites come into being, whose judgement criterion is the subjective feelings of roller coaster players rather than objective measures such as using statis- tic data. Therefore, these online ranking sites are not so precise because feelings always varied in different people.

In a nutshell, building an objective roller coaster rating system based only on statistic data is a necessity. Applying this system, we could develop characteristic ranking list to meet different demands. Also, a user-friendly app aiming to help potential roller coaster riders to find what they want could be exploited upon this system.

## Restatement of the problem

We are required to create an objective quantitative algorithm to establish a descriptive roller coaster rating system with the roller coaster numerical and descriptive specification data. Then we need to develop “a Top 10 Roller Coasters in the World“ list based on our algorithm, and then compare and discuss the ranking results and descriptions with two other rating systems online. Afterwards, we are required to describe the concept and design for a user-friendly app using our algorithms to recommend a potential roller coaster rider what she or he would want to ride. A non-technical News Release is also needed to describ our new algorithm, results, and app.

In order to solve those problems, we will proceed as follows:

* + - Build an algorithm to develop a “Top 10 Roller Coasters in the World“ list.
    - Compare and discuss our results with other two ranking sites found online.
    - Give the description of the concept and design for the app.
    - Write the non-technical News Release to explain our algorithm and app.

## Analysis of the Problem

The main research of this paper is to exploit a roller coaster ranting algorithm based on the original data and develop a “Top 10 Roller Coasters in the World“ list using this algorithm. And then compare and discuss this results with two existing roller coaster rating sites online.

Applying our new rating algorithm, design a user-friendly app in order to satisfy demands of roller coaster enthusiasts.

### Develop Rating Algorithm

Before creating the roller coaster ranking algorithm, we should preprocess original data first involving correcting fault data, deleting unavailable data and filling in the missing data. Some methods like k-NearestNeighbor, Decision Tree and linear Regression are considered. Given that these two descriptive specification date “Construction“ and “Type“ cannot be used di- rectly, we consider replacing the descriptive words with their self-information numerical fig- ure.Based on these processed data, we plan to use EWM to assign weights for every specifi- cation data. Afterwards, apply these weights in TOPSIS to determine our roller coaster rating algorithm.

### Develop Top 10 list

Use the established roller coaster rating system, combined with the already completed roller coaster indicator data, to comprehensively rank the roller coasters listed in the RollerCoast- erData.xlsx, generating our “Top 10 Roller Coasters in the World“ list.Compare the list of the world‘s top ten roller coasters with the other two online roller coaster ranking sites and find out differences between our lists and the other two online sites. From the perspective of ranking, sample size, etc., analyze the reasons for the differences.

### Design a user-friendly app

Aimed to design a user-friendly app, the user‘s needs should be clarified. As for potential roller coaster, when using the App to find the desired roller coaster, they may not only want the app to recommend some roller coasters to them, but hope that the app can find the roller coaster according to the specified conditions they input. Therefore, on the one hand, we use the Three-particle Recommendation Algorithm to recommend some roller coasters to APP users. On the other hand, we use the designed roller coaster sorting algorithm to help users realize the targeted roller coaster search function.

# Assumptions and Notations

## Assumptions

* + - Assume that the data used is true and valid.
    - Assume that the roller coasters‘ data indicators do not change over time.

## Notations

The primary notations used in this paper are listed in **Table 1**.

Table 1: Notations

#### Symbol Definition

*H* Height of roller coaster

*v* Largest speed of roller coaster

*v* Average speed of roller coaster

*D* Largest drop of roller coaster

*L* Length of roller coaster

*wn* Weight of n-th indicator

*T* Total duration of roller coaster

*E* Entropy of indicators

# Preparation of Data

As the RollerCoasterData.xlsx have some missing, noisy, or inconsistent data, we preprocess these data to facilitate the establishment of the roller coaster rating algorithm later. The data preprocessing includes the following points:

## Revise erroneous data

Correct one “Wood“ and two “Steel“ in specification data “Type“ into “Sit down“, since “Wood“ and “Steel“ belong to “Construction“ and we search this roller coaster “Happy Angel“ online, finding that this roller coaster is a sit-down roller coaster.

## Fill in missing data

### 3.2.1 “Height““Speed““Length“

The missing data in specification data “Height“, “Speed“ and “Length“ is filled in using KNN methods since these three types of data not misses a lot.

Using Euclidean Distance to calculate the distance between other specification data and “Height“ “Speed“ and “Length“, respectively. And then obtain *k* smallest distances, using the average data of these *k* specification data of “Height“ “Speed“ and “Length“ to fill in missing data.

The formula of Euclidean Distance is :

*n*

「|艺

U

*d* =

*i*=1

(*xi − yi*)2

The resultant Roller Coaster Data “Height“ of “Happy Angel“ is 141.7.

The resultant Roller Coaster Data “Speed“ after filling in is shown in Table 2. The resultant Roller Coaster Data “Length“ after filling in is shown in Table 3.

Table 2: Filling Data “Speed“

|  |  |
| --- | --- |
| **Name** | **Speed(mph)** |
| Crazy Bird | 38.5 |
| Journey to Atlantis | 53.7 |
| Montana Rusa(Salitre) | 37.3 |
| Montana Rusa(VulQano) | 46.7 |
| Table 3: Filling Data “Length“ | |

|  |  |  |
| --- | --- | --- |
|  | **Name** | **Length(feet)** |
| Bullet Coaster | 3298 |
| Happy Angel | 1757 |
| Journey to Atlantis | 2695 |
| OCT Thrust SSC1000 | 3288 |
| Timber Drop | 2846 |
| **3.2.2 “Duration“** |  |  |

The missing data in specification data “Duration“ has a poor linear correlation with other spec- ification data. After Comparing five methods: “BP Neural Network“, “SVM“, “Decision Tree“, “Multiple Linear Regression“ and “Multiple Nonlinear Regression“, we find that the correla- tion coefficient of “Decision Tree“ is the largest which means using this method, the correlation is the best. So Nonlinear Regression Decision Tree is applied to fill in these missing data.

The correlation coefficients of these five methods are shown in Table 4. The filling in results in Table 4: Correlation Coefficient

|  |  |
| --- | --- |
| **Name of Methods** | **Correlation Coefficient** |
| BP Neural Network | 0.932 |
| SVM | 0.904 |
| Decision Tree | 0.958 |
| Multiple Linear Regression | 0.533 |
| Multiple Nonlinear Regression | 0.584 |

“Duration“ is shown in Figure 1.

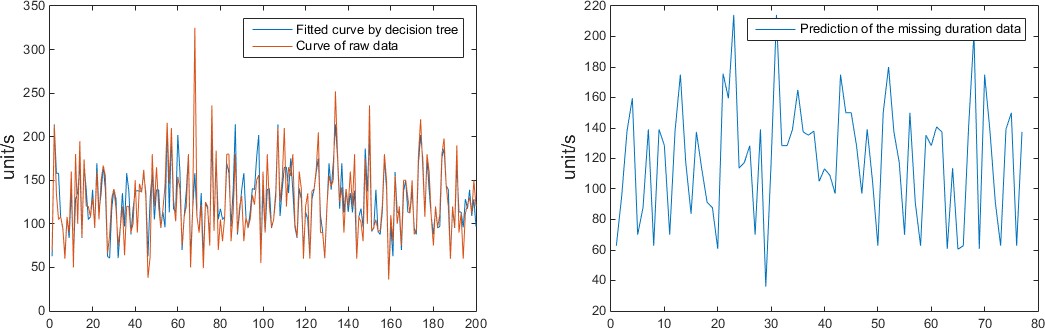


Figure 1: Filling Data in “Duration“

The orange line in left picture represents the “Duration“ of raw data while the blue line repre- sents the “Duration“ data after fitting with “Decision Tree“; The right picture symbolizes the predicted “Duration“ data of original filling data using “Decision Tree“.

### 3.2.3 “Drop“

The data in “Drop“ have a good linear correlation with “Speed“ and “Height“, which is 0.95 and 0.96, respectively. Therefore, we use Multiple Linear Regression to fill in missing data in type “Drop“.

The Multiple Linear Regression Model is:

*f* (*x*) = *w*1*H* + *w*2*v*

*H* represents the data “Height“ and *v* represents the data “Speed“. The weights *w* is calculated to 0.37 and 2.83. So the resultant Multiple Linear Regression formula is:

*Drop* = 0*.*37*H* + 2*.*83*v −* 88*.*71

The linear correlation coefficient of “Drop“ and other specification data and the filling in results are shown in Figure 2.

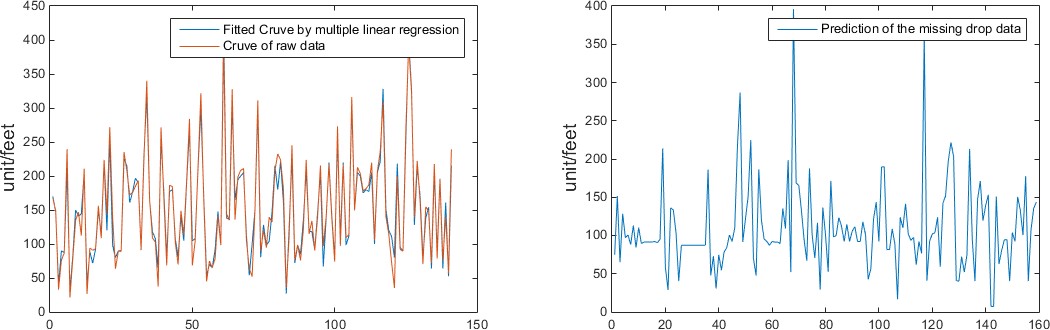


Figure 2: Filling Data in “Drop“

The orange line in left picture represents the “Drop“ of raw data while the blue line represents the “Drop“ data after fitting with “Multiple Linear Regression“; The right picture symbolizes the predicted “Drop“ data of original filling data using “Multiple Linear Regression“.

## Deleting unavailable data

The missing data in “G Force“ and “Vertical Angle“ is excessive, which will cause a great many errors after filling in new data, so we delete these data.

## Convert words to data

The specification data “Construction“ and “Type“ contains only descriptive words which could not be used to create our rating algorithm. So we convert these words into numerical figures applying the concept of self-information.

The definition of self-information:

*I* (*x*) = log

1

*p* (*x*)

Usually, *I(*x*)* is called the self-information of message *x*, which has the nature of random vari- ables. When the probability *P* is smaller, the probability that the message *x* appears is smaller, and the amount of information obtained is larger.

Similarly, we could regard the descriptive words such as “Steel“ “Wood“ and “Sit Down“ as specific information contained in this type, thus, the self-information could objectively repre- sent that information.

The inverted results of “Construction“ are shown in Table 5.

Table 5: Filling Data “Construction“

### Name Construction

#### Steel 0.19

Wood 0.81

The inverted results of “Type“ are shown in Table 6.

Table 6: Filling Data “Type“

### Name Type

#### Flying 0.21

Inverted 0.10

Sit Down 0.02

Stand Up 0.25

Suspended 0.23

Wing 0.19

## Define average speed

Considering that the data in “Length“ and “Duration“ are not directly correlate to roller coaster riders‘ excitement, so we define a new variable: average speed.

*v* = *L/D*

This new type “Average Speed“ will be used to create ranking algorithm later together with other six types “Construction“ “Type“ “Height“ “Speed“ “Number of Inversion“ and “Drop“.

# Roller Coaster Ranking Algorithm

## Calculating Weights using EWM

The entropy weight method is an objective weighting method. In the specific using process, the entropy weight method uses the information entropy to calculate the entropy weight of each index according to the degree of variation of each index, and then corrects the weight of each index through the entropy weight, thus obtaining a more objective index weight.

As the entropy of discrete data is relatively large, the weights calculated by EWM will be re- markably large, so we weak this influence by dividing each index by their standard deviation. The calculation formula is:

*σ* = 「|U 1

*N*

*N*

艺*i*=1

(*xi−*2*µ*)

If the system may be in many different states. When the probability of occurrence of each state is *pi* (i=1,2, *· · ·* ,m), then the entropy of the system is defined as:

*m*

艺

*E* = *− pi* ln *pi*

*i*=1

Obviously, when *pi* = 1 (i=1,2, *· · ·* ,m), that is, the probability of every states appearing is

*m*

equal, the entropy achieves highest, which is:

*E* = ln *m*

And now, we have 300 kinds of different roller coasters and 7 types of specification data, which is “Construction“ “Type“ “Height“ “Speed“ “Number of Inversion“ “Drop“ and “Average Speed“. Thus, we form a ranking matrix *R* = (*rij*)300*×*7,which is:

 

*r*11 *r*12 *... r*1*n r*21 *r*22 *... r*2*n*

 

*... ... ... ...*

*rm*1 *rm*2 *rm*3 *...*

For a certain type *rj*, the information entropy is:

*m*

艺

*Ej* = *− pij* ln *pij*

*i*=1

The process of calculating weights is shown as follows:

1. Calculate the probability *pij* of i-th roller coaster in j-th specification type:

*pij*

*rij*

= *m*

2

*rij*

*i*=1

1. Calculate the entropy *Ej* of j-th specification:

*m*

艺

*Ej* = *−k pij* ln *pij*

*i*=1

1. Calculate the entropy weight *wj* of j-th specification:

*n*

艺

*wj* = (1 *− Ej*) */* (1 *− Ej*)

*j*=1

The results of all entropy weights are shown in Figure 3.

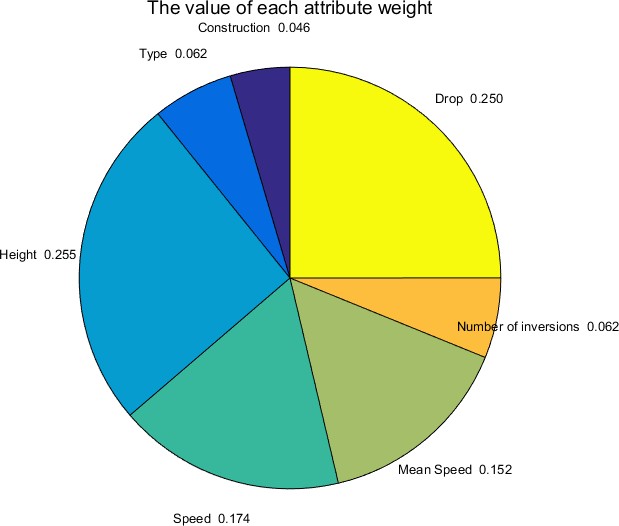


Figure 3: Entropy Weights

## Rating Algorithm based on TOPSIS

The basic idea of the TOPSIS method is to define the ideal solution and the negative ideal solution of the decision problem, and then find a solution in the feasible solution, which is closest to the ideal solution and farthest from the negative ideal solution. The ideal solution is generally the best solution, and the corresponding attributes correspond to at least the best values in each plan; the negative ideal solution is assumed to be the worst solution, and its corresponding attributes are at least not better than the worst value.

In this problem, we have 300 decision objectives (300 roller coasters) *fj* (*j* = 1*,* 2*, , m*) *m* =

*· · ·*

300. Set that we have n feasible solutions *Zi* = (*Zi*1*, Zi*2*, , Zim*) (*i* = 1*,* 2*, , n*); and the ideal solution for the normalized weighted goal of the problem is *Z∗*, involving

*· · · · · ·*

*Z*+ = `*Z*+*, Z*+*, · · ·, Z*+＼

` ＼

1

2

*m*

*Z−* = *Z*1*−, Z*2*−, · · ·, Zm−*

Define the distance from arbitrary feasible solution *Zi* to *Z*+ is:

「|艺*m*

*i*

*j*

*S*+ = U

(*Zij − Z*+)2 *i* = 1*,* 2*, · · ·, n*

*j*=1

In this formula, *Zij* is the normalized weights of j-th roller coaster using i-th solution plan.

Similarly, we define the distance from *Zi* to *Z−* is:

「|艺*m*

*Si−* = U

(*Zij − Zj−*)2 *i* = 1*,* 2*, · · ·, n*

*j*=1

So, the relative proximity of a feasible solution to an ideal solution is defined as:

*C* = *Si−*

*i S−* + *S*+

*i i*

The calculating process of TOPSIS is as follows:

1. Supposing that a certain decision problem’s decision matrix is *A*, which could form a normalized decision matrix *ZI* . Its elements are *ZI* , and

*ij*

*I fij*

2

*Zij* = *n*

2

*f*

*ij*

*i*=1

*i* = 1*,* 2*, · · ·, n*; *j* = 1*,* 2*, · · ·, m*

And *fij* is defined by decision matrix *A*:

 *... ...* 

*f*11 *f*12 *... f*1*m* 

|  |  |  |  |
| --- | --- | --- | --- |
| *f*21 | *f*22  *...* | *...*  *...* | *f*2*m* |
| *fn*1 | *fn*2 | *...* | *fnm* |

1. Construct a normalized weighted decision matrix *Z*, its elements *Zij* is:

*Zij* = *WjZij i* = 1*,* 2*, · · ·, n*; *j* = 1*,* 2*, · · ·, m Wj* is the weight of j-th roller coaster.

1. Determining the ideal solution and the negative ideal solution: the larger the elements

*Zij* in decision matrix, the better this decision plan. And,

*Z*+ = `*Z*+*, Z*+*, · · ·, Z*+＼ = *{*max *Zij|j* = 1*,* 2*, · · ·, m}*

` ＼

1

2

*m*

*Z−* = *Z*1*−, Z*2*−, · · ·, Zm−* = *{*min *Zij|j* = 1*,* 2*, · · ·, m}*

1. Calculate the distance *Si* from each solution to the ideal point and the distance *Si−* to the negative ideal point.
2. Calculating *Ci* and sort it according to each solutions‘ Relative Proximity, then discover the best solution.

# Discussion of Roller Coaster Rating Sites

Implementing our Roller Coaster Rating Algorithm, we get a “Top 10 Roller Coasters in the World“ list and a “Roller Coaster Score Sheet“,as well as “Roller Coaster Rating System“ in which we classify the “Thrill“ level in 4 degrees: “Slightly“, “Moderately“, “Greatly“ and “Ex- tremely“. To verify whether our Roller Coaster Rating Algorithm is confidential or not, we compare our ranking and grading results with the results of the “captaincoaster“ and “coast- erbuzz“ websites. As for some differences between our rating results and that demonstrate on the Interner, we also analyze the reasons from four perspectives: “Object Sample Size“, “Rating Criteria“, “Indicator Selection“ and “Weighting Evaluation methods“.

## Roller Coaster Rating Results

The roller coaster rating system is shown in Figure 4.

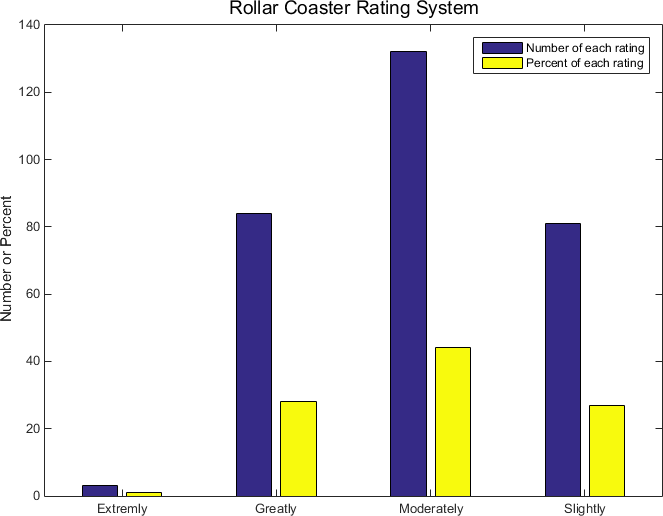


Figure 4: Roller Coster Rating System

The “Top Ten Roller Coaster Score Sheet“ is shown in Table 7 and Figure 5.

Table 7: “Top Ten“ Score Sheet

### Name Score

#### Kingda Ka 0.637

#### Top Thrill Dragster 0.608

#### Superman: Escape from Krypton 0.510

#### Tower of Terror II 0.495

#### Red Force 0.474

#### Formula Rossa 0.463

#### Millennium Force 0.458

#### Fury 325 0.449

#### Steel Dragon 2000 0.444

#### Eejanaika 0.436

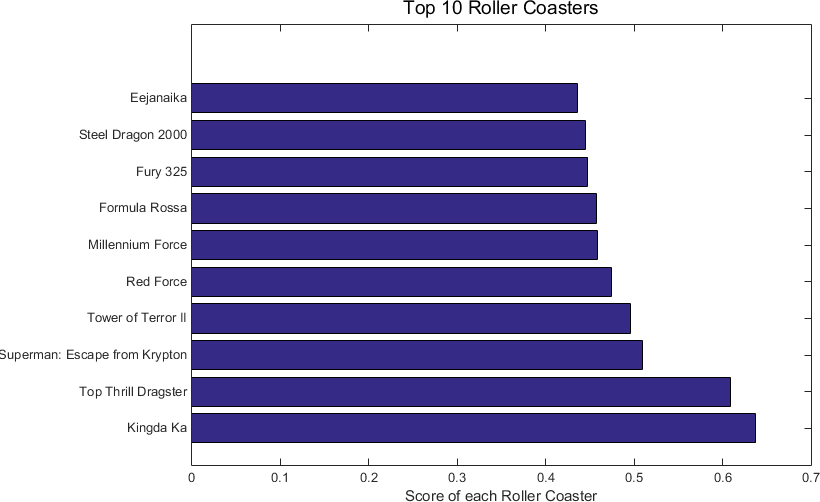


Figure 5: Score Sheet

## Comparison and Discussion

### Comparison

Compare the number and name of the duplicates of the two top ten roller coaster lists in the world, and list the top ten roller coasters of other sites in our forecasting system. (“no“ means this roller coaster is not in our original data list)

The comparison results is shown in Table 8.

Table 8: Comparison Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | “captaincoaster“ |  | “coasterbuzz“ |  |
| Steel Vengeance | 29 | Steel Vengeance | 29 |
| Fury 325 | 8 | Fury 325 | 8 |
| Maverick | 211 | El Toro | 40 |
| Twisted Timbers | no | Lightning Rod | no |
| El Toro | 40 | Twisted Timbers | no |
| Wildfire | 52 | Millennium Force | 6 |
| Eejanaika | 10 | Twisted Colossus | 33 |
| Lightning Rod | no | Boulder Dash | 147 |
| Twisted Colossus | 33 | Wicked Cyclon | 16 |
| Shambhala | 21 | Voyage | 80 |
| **5.2.2 Discussion** |  |  |  |  |

The reasons why the “Top 10 Roller Coasters in the World“ list produced by our rating algo- rithm is different from online ranking results are shown as follows:

* + - 1. The rating object sample size is not exactly the same:

Some of the top roller coaster rankings in the other roller coaster ranking systems are not available in our roller coaster database. Similarly, some of the top ten roller coasters in our roller coaster rating system are not available in other ranking systems.

For example, The total sample size of our roller coaster is 300, while the total number of roller coaster samples in the “captaincoaster“ website is 754. The “Top 10“ roller coaster “Twisted Timbers“ and “Lightning Rod“ in its website cannot be found in our roller coaster sample library.

* + - 1. Ratings are based on different criteria:

These two online rating systems are ranked based on users‘ comments and preferences. Our roller coaster rating algorithm is based on the objective indicators and statistic data and does not introduce subjective feelings. The users‘ preferences are not only related to the objective indicators of the roller coaster, but also related to the fare , the configuration of the playground facilities where the roller coaster is located, and the surrounding en- vironment such as transportation and accommodation. Therefore, the evaluation results are different.

For example, The roller coaster ranking lists in “captaincoaster“ website is based on a comprehensive rating of 30.3 thousand users. The “coasterbuzz“ website is also based on users’ comments and grades. But our rating criteria are based only on objective indicators of roller coasters such as speed, height, etc.

* + - 1. Difference in objective indicator selection:

Even if they are evaluated by objective indicators, there are differences between the indi- cators selected by other online websites and the indicators we select. Different websites will choose different indicators to be included in the evaluation system.

* + - 1. Differences between weighting methods and evaluation methods:

In the case where the selected indicators are the same, because of the differences between the weighting method and the comprehensive evaluation method, the roller coaster rank- ing lists will be different in various algorithm.

# Design User-friendly App

## Concepts and Designs

The concepts of our app is user-friendly. To achieve this target, we set two functions: Search and Recommendation. To realise “Search“ function, we first develop a Database for Roller Coaster and then process these data by correcting fault data, filling in missing data and delet- ing useless data. Afterwards, applying our roller coaster rating algorithm, we generate two ranking list, one is the overall ranking list, another is based on single indicator. Thus, using these two ranking lists to fulfill “Search“ function. To realise “Recommendation“ function, we first develop a Database for users and then create the Three-way Granular Recommendation Model based on Granular Recommendation Algorithm. Besides, when new users register in our app, we ask them to input some information such as their locations. And then ask them whether they would like to search roller coasters they like within some distance range.

The Design of this app is shown in the following Flow Chart:

|  |  |  |
| --- | --- | --- |
|  | Recommend Roller Coaster | |
|  |
| Develop user’s  Database  **Granular Recommendation Algorithm** | | |
|  | **Three-branch**  **Granular Recommendation**  **Model** | |
|  | |  |

Whether to search within a No

certain distance

Judge users’ input

Ranking by

Thrill Level

Single Index Rating

Overall

Rating

Thrill Level Rating

Overall

Rating

Thrill Level Rating

Screening Roller

Coasters within

Distance Range

Ranking By Single Index

Overall Ranking By Rating System

Single Index Rating

Calculating Distances between Users and

Roller Coaster

Data Correcting

Deleting and Filling

Yes

Develop roller

coaster’s Database

Registration Information of New

Users

Figure 6: Design Flow Chart

## Three-way Granular Recommendation Algorithm

Three-way Granular Recommendation Algorithm consists of two processes. One is the gran- ularity recommendation algorithm, which is used to predict users‘s rating for a roller coaster, the other one is the three-grain recommendation model, which is used to recommend the desire roller coaster for the user.

* + - Granularity Recommendation Algorithm:

Three-way granular recommendation algorithm is used to calculate the predicted score by the collaborative filtering based on granularity recommendation algorithm. The core of granularity recommendation algorithm is the user-item granularity rating matrix. Gran- ularity recommendation algorithm judge the similarity between the users by the degree of there preference for item granularity. The specific steps are as follows:

1. Calculate the person coefficient between users, person coefficient varies from -1 to 1, the closer it is to 1, the more similar the users‘ preferences. It can be calculated by the following formula:
   * 2*ki,j⊂K*(*u*)*∩K*(*v*) `*bu,i,j − bu*＼ `*bv,i,j − bv* ＼

*ki,j⊂K*(*u*)*∩K*(*v*)

*bu,i,j − bu*

*ki,j⊂K*(*u*)*∩K*(*v*)

*bv,i,j − bv*

*Sim* (*u, v*) =

2

`

＼2✓2

`

＼2

Where, *K* (*u*) and *K* (*v*) respectively represent the set of granulometric score of *u* and *v*, *ki,j⊂K*(*u*)*∩K*(*v*) represents the common granulometric score items, and *bu,i,j* represents the user u‘s score to the j-th granularity of the i-th roller coaster, *bu* rep- resents user u‘s average score for all items‘ granularity, *Sim* (*u, v*) represents the

person coefficient between user *u* and *v*.

1. Choose nearest neighbors. Select *K* users with highest person coefficients as the nearest neighbors according to the person coefficient between users.
2. Predict user‘s rating for unrated objects. Calculate the possible scores of current users for unrated object based on the person coefficient between the neighborhood users, the current users and the previous score record of the neighborhood users. Use user‘s average score as the benchmark and similarity as weight to calculate the predicting score of target user. The specific formula is as follows:

*p*  2*v∈k Sim* (*u, v*) *∗* `*r*(*v,i*) *− rv*＼

*Ru,i* = *r* +

2*v∈k*

`*r*(*v,i*)

* *rv*＼

Where *Rp* represents the user u‘s predictive rating for roller coaster *i*, *r*(*v,i*) rep-

*u,i*

resents the first *K* nearest neighbors of user *u* that have rated *i*. *rv* represents the average score for all roller coasters of user *v*.

* + Three Granularity Recommendation Model:

In order to reduce the cost of recommendation in the process, Three Granularity Recom- mendation Model is build based on Granularity Recommendation Algorithm, it is the core idea to decide whether to recommend, not recommend or delayed recommend in the recommendation process according to the predicted score. Recommendation cost main- ly includes misclassification cost and learning cost of delayed recommendation in the decision-making process. The specific recommended costs are shown in Table 9. Where

Table 9: Recommendation Cost Matrix

|  |  |  |
| --- | --- | --- |
| **Decision Rules** | **Like** | **Dislike** |
| Recommend (P) | *λPL* | *λPD* |
| Delayed Recommend (B) | *λBL* | *λBD* |
| Not Recommend (N) | *λNL* | *λND* |

*λpl*, *λbl* and *λnl* respectively represent the cost of recommending, delayed recommend- ing and no recommending roller coasters to users what they like, and *λpd*, *λbd* and *λnd* respectively represent the cost of recommending, delayed recommending and not rec- ommending roller coasters to users whta they dislike. When given the cost matrix, we determine the threshold *α* and *β* of the three-branch decision. The rules are shown as follows:

*TC* (*α, β*) = *Cp* (*α, β*) + *CB* (*α, β*) + *CN* (*α, β*)

Where, *TC* (*α, β*) represents the total cost when the threshold is *α* and *β*, *Cp* (*α, β*) rep- resents the cost of recommendation, *CB* (*α, β*) represent the cost of delayed recommen- dation, *CN* (*α, β*) represents the cost of no recommendation. They can be calculated as following formula:



*Cp* (*α, β*) = *λP LNP L* + *λP DNP D CB* (*α, β*) = *λBLNBL* + *λBDNBD CN* (*α, β*) = *λNLNNL* + *λNDNND*



Where, *Npl*, *Nbl* and *Nnl* respectively represent the number of roller coasters which users like and are recommended, delayed recommended, not recommended. *Npd*, *Nbd* and *Nnd* respectively represent the number of roller coasters which users dislike and are recom- mended, delayed recommended, not recommended.

In the recommending process, the cost of misclassification is usually higher than the cost in learning process of the delay recommendation, also the learning cost is usually higher than the cost generated by correcting recommendation. So we proposed the following hypothesis based above situation.

*λpl ≤ λbl ≤ λnl, λpd ≤ λbd ≤ λnd*

According to the Bayes Decision Theory, we can get the most optimal *α* and *β* according to the following formula:

f

*α* = *λpd−λbd*

*λpd−λbd*+*λbl−λpl β* = *λbd−λnd λbd−λnd*+*λnl−λbl*

Meanwhile, in order to measure the domain of users‘ preferences, we also need to deter- mine the users‘ preferences for a certain roller coaster, that is *Pu,i*:

*p*

 *u,i*

 1 *R*

*R*

*p*

0 *R*

*> Ru,max*

*< Ru,min*

*p*

*u,i*

*−Ru,min*

*Ru,min ≤ R*

*≤ Ru,max*

 *Ru,max−Ru,min*

*p*

*u,i*

*u,i*

Where, *Ru,max* represents the highest score for all roller coasters of user *u*, and *Ru,min*





represents the lowest score for all roller coasters of user *u*.

So we can determine whether to recommend, delayed recommend or not recommend

the roller coaster *i* to user *u* according to the following formulas:

*Pu,i < α notrecommend*

*α ≤ Pu,i ≤ β delayedrecommend*

*α < Pu,i recommend*

# Non-Technical News Release

**New Roller Coaster Ranking Algorithm**

Roller Coasters‘ “Thrill“ level evaluation is a complicated task. Although there are several rat- ing sites on the internet have rated some roller coasters, most of them do not provide objective and precise ranking results since their judgement criterion is heavily dependent on subjective feelings of “professional“ roller coaster riders such as their an “excitement“ or “experience“. In fact, people‘s subjective feelings are always related to many factors, and there are volatility and instability under different external stimuli. In order to improve these evaluation system‘s blemish, and to achieve a confidential and objective reflection for the degree of “Thrill“ of the roller coaster, a new roller coaster ranking algorithm is proposed in this paper.

The new roller coaster rating algorithm has got rid of the drawbacks of the previous website that rely too much on users‘ feelings, it uses a series of real data of roller coaster as the basis for rating. The algorithm uses the objective weight assignment method to calculate the weight of each roller coaster attribute based on the existing data, then we can get the comprehensive score of the roller coaster by the comprehensive rating algorithm. The comprehensive score can reflect the degree of thrill of the roller coaster, the higher the score, the thriller the roller coaster. We sort the existing roller coaster according to the comprehensive score, and then take the top ten with the highest score as the “The world‘s top ten roller coasters“.

We can compare our list of the world‘s top 10 roller coasters with those in the “coasterbuzz“ site and the “captaincoaster“ site. It is easy to find that three or two of the top 10 roller coasters on those sites are also appearance in our list about “top 10 roller coasters in the world“, and most of the others on those sites are in our top ten percent. Therefore, the results obtained by our objective evaluation method are highly consistent with the ranking based on users‘ subjective feelings, but there are still some differences. Therefore, we try to analyze the reasons for ranking difference from four different perspectives.

Based on our algorithm, we hope to develop a user-friendly app to help a potential riders to find the roller coaster which they want to ride. On the one hand, the app can automatically recommend roller coasters, which other users owing the same private information. On the other hand, users can select the roller coasters by their own demands. For example, the roller coasters can be ranked by its “Thrill“ level or single index. In addition, users can also select all roller coasters within a certain distance range from users.

In a nutshell, our roller coaster rating algorithm could reproduce “Roller Coaster Ranking List in the World“ based on input data and generate the “Roller Coaster Rating System“. Our new roller coaster app maintains the following functions:

Help potential roller coaster riders search roller coasters what they would like to ride within a distance range.

*•*

Recommend roller coasters to app users based on their registration information and the similar preferences between the target use and other app users.

*•*

Allow app users to search their favorite roller coasters within these three ranking lists: “Overall Ranking List“, “Single Index Ranking List“ (like “Height“ “Speed“) and “Thrill Ranking List“.

*•*

# Strengths and Weaknesses

## Strengths

The paper which is based on the reasonable and clear analysis of the problem, makes use of the data about roller coaster in the attachment and adopts an objective method to establish a ranking model on roller coaster.

*•*

The model is simple and feasible, and is able to apply to the comprehensive evaluation of roller coaster in real life.

*•*

## Weaknesses

To convince others of the reliability of the ranking model, we removed some about roller coaster which loses too much data,such as G force, Vertical Angle and so on. It is possible to lead to the loss of some effective information.

*•*

The roller coaster ranking model refers to several data type about roller coaster. some roller coasters lacking data type cannot be included in the ranking system.

*•*

# References

1. YE Xiao-qing, LIU Dun, LIANG De-cui. Three-way Granular Recommendation Algorithm Based on Collaborative Filtering. *COMPUTER SCIENCE*, Vol.45, No.1, pp.90-96,2018
2. CSDN:K-NearestNeighbor.2018.11.13 https://blog.csdn.net/saltriver/arti cle/details/52502253
3. cnblog: Decision tree.2018.11.13 [http://www.cnblogs.com/NeilZhang/p/ 9216354.html](http://www.cnblogs.com/NeilZhang/p/9216354.html)
4. cnblog: Multiple Linear Regression.2018.11.13 [https://www.cnblogs.com/pengyi ngzhi/p/5383801.html](https://www.cnblogs.com/pengyingzhi/p/5383801.html)
5. wenku.baidu: EWM.2018.11.13 [https://wenku.baidu.com/view/b54c63cfd1d 233d4b14e852458fb770bf68a3b71.html?tdsourcetag=s\_pcqq\_aiomsg&qq-p f-to=pcqq.group](https://wenku.baidu.com/view/b54c63cfd1d233d4b14e852458fb770bf68a3b71.html?tdsourcetag=s_pcqq_aiomsg&qq-pf-to=pcqq.group)
6. wenku.baidu: TOPSIS.2018.11.13 [https://wenku.baidu.com/view/3bb2e8e f0975f46527d3e12e.html?tdsourcetag=s\_pcqq\_aiomsg&qq-pf-to=pcqq.g roup](https://wenku.baidu.com/view/3bb2e8ef0975f46527d3e12e.html?tdsourcetag=s_pcqq_aiomsg&qq-pf-to=pcqq.group)
7. CAPTAIN COASTER: Ranking of the best coasters - November 2018.2018.11.14 [https:](https://captaincoaster.com/en/ranking/)

[//captaincoaster.com/en/ranking/](https://captaincoaster.com/en/ranking/)

1. CoasterBuzz: Top 100 Roller Coasters: The CoasterBuzz 100.2018.11.14 [https://coas terbuzz.com/rollercoasters/top100](https://coasterbuzz.com/rollercoasters/top100)