Baseball-AI

- Real-Time winning rate computing

Abstract—Today baseball is becoming more and more popular and getting new fans. Those who just have put their first step into baseball find it hard to follow the rules and to understand the changes in the situation. Unlike our baseball league KBO, MLB already has many meaningful methods for in-game analyzing with real time statistics. So we are going to adopt those ideas into KBO league. Winning Probability Added, which is also known as WPA, is one of good factors which allows easy understanding of the situation and the impacts that pitcher or batter makes each time. Our project aims to show ‘Winning rate’ and ‘WPA factor’ in real-time, with the algorithm that fits KBO situation..

I. Introduction

Baseball, even though the myriad scandals, attracts more and more fans and is becoming more nationwide sports. There is a major problem that, though baseball fans are inflowing, it’s hard for beginners to understand the rules and catch situation of the match. Though baseball is much more a number-oriented sport than any others such as soccer or basketball, the classical stats do not fully reflect the match stream and evaluate the value of the players properly. However, in MLB, a lot of sabermetricians have already quantified vague situations and values in numerical way, and KBO tends to follow it. (i.e Babip, OPS, etc..). Korean baseball websites nowadays use the same WPA algorithm which had been used in earlier days of MLB, so the differences between two leagues are not considered. We wanted to solve this problem by creating our own WPA algorithm based on KBO, which not only calculates the stats suitable for KBO, but also analyzes winning possibilities of each team and the impacts each players make. We also focus on helping beginners to understand baseball.

We will make KBO winning rate DB and we are going to put that into our program. The program represents ongoing situation of the match and each team’s winning possibilities simultaneously. Also, by showing quantified stats such as batters’ WPA or pitchers’ WAR , it shows how much powerful the player is(or has been) throughout the game, seasons or his entire career. Whether baseball match is underway or not, the program will show player rank categorized by players, teams, or positions according to user’s request.

It’s not just ‘Baseball statistic calculator’, but also somewhat like websites such as Statiz or KBReport. The software indicates different types of statistical analysis, and shows them in visualized ways. We thought that current WPA algorithm used in korean web sites does not fit korean baseball situation. That’s why we’ve started this project.

This project is composed of 4 steps.

i) Crawl the data of last 10 seasons of KBO from baseball statistics websites.

ii) Compute the data to make some statistics.

iii) Apply the statistics to WPA algorithm.

iv) Real-time data capturing and showing the winning rate and WPA.

II. REQUIREMENTS

1) Data handling

a. Crawling

* Get every single raw data of KBO from baseball webpage to construct the root database
* Crawiling source : [Http://www.koreabaseball.com](http://www.koreabaseball.com)

b. Capturing

- Get real-time data when the match is underway.

c. Real-Time Mirroring

* Program should immediately renew the database according to the result of the match.

d. Computing

* Calculate the numerical data to make some meaningful statistics.
* Every single data has different weight.

i.e.) Hits at 1st inning have different value from those at 9th.

* Calculate numerical values including WPA

e. Data storage

* Save every single stats data
* Divide players into two tables. One table is for players who is in active service, the other for retired
* Table for players who is in active service needs to be updated constantly, and the other table doesn’t
* User who wants to conceal the program from screen can do that by clicking ‘window minimization’ button

2) Functions

a. EXCEL Compatibility

* User can export datas of specific player or stats to MS Excel files
* User can import fixed form of MS Excel file of specific game result to compute changes of KBO algorithm winning rate shown as image file which can also exported as jpeg, gif, png, or bmp.
* User can import fixed form of MS Excel file of specific league(fantasy or amatuer) data to compute WAR stats. This data can be exported as EXCEL file.

b. On-Board Posting

* Someone who wants to post any idea or thoughts can share theirs’
* Make another Q&A board so as to help beginners solve their curiosity

c. Board Log-in & Sign-out

* Log-in to or Sign-out from Board.
* User who logged-in the board can upload their post or reply to other users’

d. Stats Visualization

Showing the current state of game in a table.

Showing the current winning average of each team

Showing the current WPA stats of players

Showing players picture

3) User Interface

a. Window Minimization & Window Maximization

* Default size : 960\*540
* User who wants to see the program widely can do that by clicking ‘window maximization’ button

b. Program Turn On & Turn Off

* User can turn on the program by clicking ‘desktop icon’
* If user try to power on the program even if that is already turned on, terminate existing program and launch program again
* User can turn off the program by clicking ‘x button’ at the top-right corner of the program

c. Mouse Click Event

* Provide user with three options [To Home, Window Minimization, Termination] when user right-click any area within program
* If user double-clicks ‘search’ button, program shows last request

d. Player Stat Pop-Up

* When user clicks certain player, program shows his profile by generating a new pop-up
* If player is a pitcher, pop up list of the first string who has not on the match yet
* If player is taking the field, pop up his profile as batter
* Pitcher pop-up profile stats list : ERA(Earned Run Average) for applicable season, WPA, WAR, WHIP for last 5 matches, (K/BB 9), hyperlink connected to NAVER article about him
* Batter pop-up profile stats list : BA(Batting Average), WAR, WPA, OPS for last 5 matches, BABIP for applicable season, hyperlink connected to NAVER article about him
* The number of pop-up cannot be over two

e. Player Ranking

* Sort players by team, position, date and game with WPA stats

f. Data Searching

* Search option constitutes of ‘match schedule/ player and stats / player’
* If option ‘match schedule’ is chosen, program shows match schedule as a calendar
* If user click one of date, there are three cases. First one is ‘past match’, so program shows match log. Second one is ‘on-going match’, so program directs user to the match. And the last one is ‘coming match’, so program shows every details of the match including players, referees, park, appointed first thrower, weather forecast
* If option ‘player and stat’ is chosen, program shows the applicable stats separately by entire players, team, position, monthly
* If option ‘player’ is chosen, program shows every single stat of applicable player

g. Get Information real-time

* User can choose the way one gets some information(pop-up or push window)
* Pop-up is a kind of window, so when user have it on the screen, one cannot click main program
* Information could be as follows
* Agreed Decision : User can get information about agreed decision and its details
* Cancellation in case of rain : User can get information when the match is cancelled in case of rain by getting a pop-up or push window
* Player Substitute : In case of substituting player, User can get information why the player was substituted with other, and information about that ‘other’ player
* When option is ‘pop-up’, user can have additional function which is ‘multi-view’. By doing so, user can watch several matches simultaneously
* If there has multiple pop-ups, eliminate pop-up windows sequentially after checking them
* Refresh in every 30 seconds.

h. Sounds

* Error alert
* ‘Search’click

4) Development environment

a. IDE(Integrated Development Environment)

* Developing Language : Google-Go
* Developing Platform : GO
* Use NAVER weather API to get weather information
* Use NAVER time API to get time information

5) Execution Environment

a. Network Environment

* Accessing to network needs basically
* If network is off-line, user cannot launch program

b. Least Background Memory

* There should be enough available memory space to execute the software.

III. Development environment

A. Choice of software development platform

Which platform and why? (e.g., Windows, Linux, Web, or etc.)

Which programming language and why?

Provide a cost estimation for your built.

(including any purchase of software/hardware)

Provide clear information of your development environment.

(e.g., version of software, OS version, your computer resources)

Using any commercial cloud platform (e.g., Amazon’s EC2) is

definitely a BONUS.

B. Software in use

Any existing software or algorithm in use? (doing a similar task as

your proposal; provide a proper reference if there is any)

C. Task distribution (If you want, you can provide this later at the

next phase - design)

Which member is responsible for what?

IV. Specifications

For all requirements above, you should provide details of how you can

achieve each.

A. Requirement 1

Describe a detailed description of implementation process (e.g.,

pseudocode level) under your development environment. Any type

of graphical representation is also recommended.

B. Requirement 2

Like-wise

16.04.08 Feedback.

Requirement 구술이 많았으면 => 제목의 모호성을 제거해야(직관적으로 알 수 있게) ex - Minimization&Maximization => (Window)Minimization&Maximization

Applying 수식 같은것들 => 이건 requirement specification 단계에서 구현

Specification 단계(디자인 바로 직전단계)에서 모든 every detail 구현 a-z

Ex- requirement I/O에서 import(or export)하는 excel파일은 어떤 것?

Requirement 순서는 상관없음

DB - 우리(그냥 간단하게 생각)랑 겨수(직접 구현…)님 생각이 좀 다름

개발 언어 - 구글 고로 하는게 괜찮(ㄴㄴ 구글고로 해야 함) -> 취직 시 드러낼 수 있음. 3 - 오픈소스가 많으므로 해볼만한 싸움

PT - 3번(나, 영재, 지암) => 처음(4/11일 or 18일 - 영재) 중간(없을 수도-지암) 최종(나) // 이건 다시 정합세다.

-------------------------------------------------------------------------------------------

Crawling - 오픈소스

Computing - algorithm? 테이블로 어떻게 받을 건지 알아야 할 수 있음

Searching

정해야 될것 role: 유저(준), 커스터머(승면,), 디벨로퍼(영재) 매니져(지암)

언어-google go? 어떤 펑션이 들어간다.

DB-mysql? 테이블 구조.

시즌 테이블 : 매치 아이디,(001-1, 001-2, ... , 002-1, .... 144-5), 각 날짜, 날짜별 게임, 매치랑 릴레이션.

Match table,

날짜,

(1회무사~9회2사)

시즌 별 State table, state -> --5, -4, .... , +4,++5;

Player table,

Wpa계산 참조 -> Ex) 1:0상황&2회1사&결과가 승리/ 1:0상황&2회1사

s(구동 환경) - windows 10, 최소 램, 최소 인터넷 속도,

크롤을 30초에 한번 (실시간 게임 기준), 2분동안 응답이 없을 경우 (에러) , 단 합의 판정 등, exception handling 미리 만들어야 할듯.

가로세로 픽셀 크기 : 960\*540

팝업창 사이즈

효과음

Example of WPA (home team)

Inning 6 :top

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1B** | **2B** | **3B** | **Out** | **-4** | **-3** | **-2** | **-1** | **Tie** | **1** | **2** | **3** | **4** |
| — | — | — | 0 | 0.089 | 0.146 | 0.23 | 0.348 | 0.5 | 0.651 | 0.769 | 0.854 | 0.911 |
| — | — | — | 1 | 0.097 | 0.158 | 0.249 | 0.375 | 0.534 | 0.69 | 0.802 | 0.879 | 0.929 |
| — | — | — | 2 | 0.103 | 0.167 | 0.263 | 0.394 | 0.56 | 0.717 | 0.825 | 0.896 | 0.941 |
| 1B | — | — | 0 | 0.078 | 0.128 | 0.204 | 0.31 | 0.448 | 0.594 | 0.717 | 0.812 | 0.881 |
| 1B | — | — | 1 | 0.089 | 0.145 | 0.23 | 0.347 | 0.498 | 0.649 | 0.766 | 0.852 | 0.91 |
| 1B | — | — | 2 | 0.099 | 0.161 | 0.253 | 0.38 | 0.542 | 0.697 | 0.808 | 0.884 | 0.932 |
| — | 2B | — | 0 | 0.069 | **0.114** | 0.182 | 0.28 | 0.41 | 0.557 | 0.689 | 0.793 | 0.868 |
| — | 2B | — | 1 | 0.083 | 0.136 | 0.216 | 0.327 | 0.473 | 0.625 | 0.749 | 0.84 | 0.902 |
| — | 2B | — | 2 | 0.095 | 0.155 | 0.244 | 0.368 | 0.526 | 0.682 | 0.797 | 0.876 | 0.928 |
| — | — | 3B | 0 | 0.058 | 0.098 | 0.158 | 0.247 | 0.369 | 0.517 | 0.662 | 0.774 | 0.856 |
| — | — | 3B | 1 | 0.071 | 0.118 | 0.189 | 0.291 | 0.427 | 0.582 | 0.719 | 0.82 | 0.889 |
| — | — | 3B | 2 | 0.093 | 0.152 | 0.24 | 0.362 | 0.519 | 0.675 | 0.793 | 0.873 | 0.926 |
| 1B | 2B | — | 0 | 0.062 | 0.102 | 0.164 | 0.253 | 0.372 | 0.506 | 0.633 | 0.742 | 0.827 |
| 1B | 2B | — | 1 | 0.078 | 0.127 | 0.202 | 0.308 | 0.445 | 0.59 | 0.711 | 0.806 | 0.877 |
| 1B | 2B | — | 2 | 0.092 | 0.151 | 0.238 | 0.358 | 0.513 | 0.665 | 0.78 | 0.862 | 0.917 |
| 1B | — | 3B | 0 | 0.051 | 0.085 | 0.139 | 0.218 | 0.327 | 0.463 | 0.602 | 0.72 | 0.813 |
| 1B | — | 3B | 1 | 0.067 | 0.111 | 0.178 | 0.274 | 0.402 | 0.548 | 0.682 | 0.786 | 0.864 |
| 1B | — | 3B | 2 | 0.089 | 0.146 | 0.231 | 0.349 | 0.5 | 0.652 | 0.77 | 0.855 | 0.912 |
| — | 2B | 3B | 0 | 0.046 | 0.078 | 0.127 | 0.201 | 0.303 | 0.431 | 0.569 | 0.695 | 0.795 |
| — | 2B | 3B | 1 | 0.062 | 0.102 | 0.165 | 0.255 | 0.377 | 0.517 | 0.652 | 0.764 | 0.848 |
| — | 2B | 3B | 2 | 0.087 | 0.143 | 0.226 | 0.341 | 0.49 | 0.639 | 0.757 | 0.845 | 0.906 |
| 1B | 2B | 3B | 0 | 0.042 | 0.071 | 0.116 | 0.183 | 0.277 | 0.395 | 0.523 | 0.644 | 0.748 |
| 1B | 2B | 3B | 1 | 0.06 | 0.099 | 0.159 | 0.245 | 0.362 | 0.495 | 0.622 | 0.731 | 0.818 |
| 1B | 2B | 3B | 2 | 0.084 | 0.137 | 0.217 | 0.328 | 0.471 | 0.617 | 0.733 | 0.823 | 0.888 |

The blue-highlighted 0.128 means that when the team is leaded by 3 points with no out and runner on the second base,

its WPA is 0.114.

Database Requirements?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| Inning | Outcount | Base | Score | Win(%) |
| 1 | 0 | 0 | -4 | 50 |
| 1 | 0 | 0 | -3 | 50 |
| 1 | 0 | 0 | -2 | 50 |
| 1 | 0 | 0 | -1 | 50 |
| 1 | 0 | 0 | 0 | 50 |
| 1 | 0 | 0 | 1 | 50 |
| 1 | 0 | 0 | 2 | 50 |
| 1 | 0 | 0 | 3 | 50 |
| 1 | 0 | 0 | 4 | 50 |
| 1 | 0 | 1 | -4 | 50 |
| 1 | 0 | 1 | -3 | 50 |
| 1 | 0 | 1 | -2 | 50 |
| 1 | 0 | 1 | -1 | 50 |
| 1 | 0 | 1 | 0 | 50 |
| 1 | 0 | 1 | 1 | 50 |
| 1 | 0 | 1 | 2 | 50 |
| 1 | 0 | 1 | 3 | 50 |
| 1 | 0 | 1 | 4 | 50 |
| 1 | 0 | 2 | -4 | 50 |
| 1 | 0 | 2 | -3 | 50 |
| 1 | 0 | 2 | -2 | 50 |
| 1 | 0 | 2 | -1 | 50 |
| 1 | 0 | 2 | 0 | 50 |
| 1 | 0 | 2 | 1 | 50 |
| 1 | 0 | 2 | 2 | 50 |
| 1 | 0 | 2 | 3 | 50 |
| 1 | 0 | 2 | 4 | 50 |
| 1 | 0 | 3 | -4 | 50 |
| 1 | 0 | 3 | -3 | 50 |
| 1 | 0 | 3 | -2 | 50 |
| 1 | 0 | 3 | -1 | 50 |
| 1 | 0 | 3 | 0 | 50 |
| 1 | 0 | 3 | 1 | 50 |
| 1 | 0 | 3 | 2 | 50 |
| 1 | 0 | 3 | 3 | 50 |
| 1 | 0 | 3 | 4 | 50 |
| 1 | 0 | 12 | -4 | 50 |
| 1 | 0 | 12 | -3 | 50 |
| 1 | 0 | 12 | -2 | 50 |
| 1 | 0 | 12 | -1 | 50 |
| 1 | 0 | 12 | 0 | 50 |
| 1 | 0 | 12 | 1 | 50 |
| 1 | 0 | 12 | 2 | 50 |
| 1 | 0 | 12 | 3 | 50 |
| 1 | 0 | 12 | 4 | 50 |
| 1 | 0 | 13 | -4 | 50 |
| 1 | 0 | 13 | -3 | 50 |
| 1 | 0 | 13 | -2 | 50 |
| 1 | 0 | 13 | -1 | 50 |
| 1 | 0 | 13 | 0 | 50 |
| 1 | 0 | 13 | 1 | 50 |
| 1 | 0 | 13 | 2 | 50 |
| 1 | 0 | 13 | 3 | 50 |
| 1 | 0 | 13 | 4 | 50 |