

DYNAMIC CONTROL OF DISTANCE EDUCATION PROGRESS ON EXAMPLE OF CHESS PROBLEMS

Zakharov V.B.

k.f-m.n.,n.s

SPIN: 7551-4861

Kravtsova M.E

student

Moscow State University, Moscow, Moscow, Leninskie Gory 1, 119991

Abstract. *The paper considers the ways to range progress of students in the distance education system Chess King. The basic idea is in using Elo rating system. The article offers a new formula that uses success percent for every exercise instead of solved/not solved values. Google Analytics cloud service is used for data storing and analysis. For the current moment Chess King range system was successfully tuned using data of hundreds thousands of students and successfully used.*

Key words: *distance education, chess, range system, Google Analytics.*

Introduction.

Problem objective ratings the level reached by the person in charge some areas (game, training, worker process, etc.) constantly occurred, occurs and will occur in human form activities. Depending on the situation, evaluation can occur both in the simplest way principle, type set-off/non-set-off (fit/outfit), and on difficult ones formulas including many those factors.

We will name the system, carrying out specified evaluation, the rating system (RS).

The main goals of the rating systems:

1. The most objective assessment of the level of each student.
2. Ranking students either by location, or by certain numerical methods estimated, or by level.
3. Offer required paths further education depending on the student's rating.
4. Raise profile motivation level students to training using their aspiration how to dial higher level rating, and surpass other students.

The most important resume of working with the RS is splitting students of the program monitoring and production recommendations for further information training. In compliance with this build typical RS it should include these following actions:

1. Production formulas for evaluations.
2. Accounting students' solutions of the tasks and the rating calculation according to the formulas.
3. Selection logical conditions, allowing split students on groups.
4. Analysis of the received results.
5. Modification of formulas and recalculation of ratings in case the RS is in doubt.

Statement of the problem.

The data on solving problems was to create a RS for the Chess King teaching system. The Chess King system allows students to solve chess problems of various

complexity and subjects. Data on solving problems are collected in a single database. Chess King system is available on Web, Android, Windows and iOS platforms. The number of installations of different versions of the training system at the moment has reached about half a million devices. Therefore, in addition to the implementation of rating calculation tasks, a significant part of the work is associated with the storage and processing of the results of hundreds millions solutions.

Elo rating system.

One of the standards for evaluating the power of a chess game is the Elo rating. It was decided not to leave it in the training system, but to adapt the Elo system to the problem-solving process as much as possible. Elo rating calculation is based on mathematical expectation. The basic formula is:

$$E_A = \frac{1}{1 + 10^{\frac{R_B - R_A}{400}}} \quad (1)$$

where:

E_A - expected value of a quantity the number of points that player A will score in a game with B;

R_A – player rating A;

R_B – player rating B.

Also, this formula can be represented as follows:

$$E_A = \frac{Q_A}{Q_A + Q_B}$$

where:

$$Q_A = 10^{R_A/400}, \quad Q_B = 10^{R_B/400}$$

Writing the same formula for player B will have the same denominator in the fraction. That means that the expected result of player A is Q_A/Q_B times more than the expected result of player B. Also, for every 400 points of the advantage rating over the opponent, the expected result is increased 10 times compared to the expected score of the opponent.

A sum of expected values of both players is equal to 1. In practice, since the real strength of each player is not known, the expected ratings are calculated using the current ratings of the players.

New player rating A is calculated by the formula:

$$R'_A = R_A + K(S_A - E_A)$$

where:

K – coefficient, which value equal to 10 for the strongest players (rating 2400 and higher), 20 – for players with rating less than 2400, and 40 - for new players, and for players up to 18 years old, whose rating are below 2300;

S_A – player points A actually scored (1 point for a win, 0.5 for a draw and 0 for a loss);

R'_A – new rating player A.

When a player's actual tournament scores exceed their expected scores, the Elo system takes this as evidence that player's rating is too low, and needs to be adjusted upward. Similarly, when a player's actual tournament scores fall short of their

expected scores, that player's rating is adjusted downward. Elo's original suggestion, which is still widely used, was a simple linear adjustment proportional to the amount by which a player over performed or underperformed their expected score [1].

Collection statistics about the course decision.

It was decided to use the free analytic service Google Analytics as a means of implementation. Its capabilities include the collection of arbitrary numerical data, which are described by string characteristics indicated during sending. These characteristics are the category, action and label of the data to be sent. When displaying information, you can filter data by these characteristics and get the average value for all elements satisfying the filter.

Google Analytics was chosen because it meets all the necessary requirements. Obviously, you could write your own server for searching all this data, but it requires too much effort and does not make sense until absolutely necessary.

To achieve the main goal of this work, it is necessary to know every move that the user makes, whatever it is (true or false). In Chess King chess courses, statistics come to Google Analytics as follows:

1. Event Label - a textual representation of a chess board position (FEN) and a perfect move.
2. Event action - task number.
3. Event category - solving problems.
4. The average value of the event - is the move right, doubtful or wrong.

Google Analytics inside itself is able to process this data in the necessary way, and also apply various filters when collecting statistics. For example, you can collect statistics for a specific version of an application or for a specific date range.

One of the features used by Google Analytics is the summation of sent events. Accordingly, another parameter appears for data processing - the total of events, which shows how many people made this move in solving this problem.

The collected data can be used not only to determine the rating of tasks. It is also possible, for example, to find moves in tasks that cause the greatest difficulties for users.

Calculation issue rating.

To calculate the task rating, a new formula was derived based on the Elo rating. The probability of winning player A playing with player B is (1) by definition.

The collected data give the following values. Let T – the total number of solutions to the problem, S – the total number of correct solutions to the problem (with these solutions, the problem is the “loser”). Let player B – the task itself. Then we get that the probability of winning a player is also equal to:

$$E_A = \frac{T - S}{T}$$

Then it is possible to equate two of the resulting value:

$$\frac{T - S}{T} = \frac{1}{1 + 10^{\frac{R_B - R_A}{400}}}$$

By logarithm, the formula for the task rating is derived from the obtained

$$R_B = R_A - 400 * \lg \frac{S}{T - S}$$

identity:

But this formula is not suitable for the final calculation of the task rating, since some conditions are not taken into account.

Firstly, since players of different ratings and different strengths have different “sensitivity” to losses and victories in the Elo rating, it was customary to make a coefficient of 400 in the Elo rating. In the case of tasks, one cannot talk about the “strength” of the opponent, as it’s just a task with a fixed rating. Therefore, in the formula, you must enter the parameter - "sensitivity" of the tasks of the current course. This parameter changes from zero to one.

Secondly, when a task is considered as a rival of a student, it is not correct to talk about a draw here. Generally, it exists if a person scores half the points of the task. However, it is not very well defined; since it is not clear which of the moves, such a result was achieved. Therefore, it was decided not to consider the student’s probability of winning over a task in terms of “victory - defeat - draw”. For probability, the following formula is used:

$$\frac{b(g - 1)}{g(b - 1) + b(g - 1)}$$

where:

b - the ratio of the number of incorrect decisions to the total number of solutions to the problem

g - the ratio of the number of correct solutions to the total number of solutions to the problem

And then the expression under the logarithm takes the form:

$$\frac{g(b - 1)}{b(g - 1)}$$

Thirdly, there can be several moves in a task. Each of them has a different complexity, so it was necessary that the influence of simple and obvious moves be minimized. The rating of the task itself is largely determined by complex moves. Below, when describing the algorithm of the program, it will be described how complex and simple moves were taken into account.

So, final formula looks like:

$$R_B = R_A - 400 * flex * \lg \frac{g(b - 1)}{b(g - 1)}$$

For the full processing of all data, four modules were written. The following stages of the program can be distinguished:

1. Uploading data from the Google server to text file and its analysis.
2. Uploading and parsing data from a file to a dictionary in the program.
3. Calculation tasks rating and the entire course in general.

At the first stage, all data is downloaded to the computer. A script written in Python is run directly from the program. As parameters, it receives the application identifier and the date to which it wants to collect statistics. The start date in the script

by default is the release date of the first course.

At the second stage, all obtained statistics are loaded into memory in a format convenient for further processing. To store information on a specific task, a class was created with fields in which data about number of solutions is stored. When deriving the formula for calculating the rating of a task, the question of the maximum reduction in the influence of simple moves on the rating was already raised. For this, the difference between the relations of the right decisions to the total number and the wrong decisions to the total number is first found. And, accordingly, the smaller this difference, the more simple this move is considered. To ensure that it is not taken into account as much as possible, a certain number multiplies the number of all decisions. During testing, it was concluded that the most successful is multiplication by coefficients that are squares of numbers from 1 to 10.

At the third stage, the ratings of all tasks and the overall rating of the course are calculated. Accordingly, first the rating for each move is calculated, then the rating for the task, as the arithmetic average of the ratings received, and the last step is the course rating, as well as the arithmetic average of the ratings of all the tasks that make up this course.

Conclusion.

When writing this paper, a large amount of data was analyzed. These data were obtained over several years and now they have found practical application. The data show the solution to the learning problems of more than a million users. The program created as part of this work is able to automatically download all this data from the Google server and then process it to obtain the desired result.

To process the collected statistics of solutions, a mathematical analysis of the data was initially carried out. The main formula for calculating the task complexity was derived. This formula is applicable only for chess problems, but it can also be derived for any other educational field in which the use of the rating system is available. An increase in the number of different parameters is also available, if necessary.

The resulting system is self-correcting. That is, depending on the data collected and the intermediate results obtained on their basis within the program, the calculation of the final result may vary.

The work of the resulting program was tested in practice. For each training course, an approximate rating was initially known. The expected results were almost completely identical to those obtained. Of course, there was no absolute coincidence, since it happens that a person with a low level can try to solve problems for professionals and thereby give not very objective data for processing.

One of the reasons of mismatches was in order the students solve their problems. Similar tasks of the same level are evaluated harder if they go in the beginning and easier if they go in the end. As a student solves more problems in some theme, better he knows this theme, more easily he solves the tasks of the same level. So only random mode of problems offering provides exact ratings.

Future research will allow to use multi-dimensional rating system based on basic chess skills such as opening play, middlegame, endgame, tactics, strategy, mating, development.

In future, it is planned to finalize the program so that it can dynamically process new data with a certain frequency. Accordingly, the task complexity can change in real time, and each time it will become more accurate.

References:

1. Elo rating system [Electronic resource] // Wikipedia [Official website]. URL: https://en.wikipedia.org/wiki/Elo_rating_system . (accessed 03.05.2018)
2. Analytics Reporting API version 4 [Electronic resource] // Google Developers [Official website]. URL: <https://developers.google.com/analytics/devguides/reporting/core/v4/> (accessed 17.05.2018)
3. Мостяев А. И. Методика создания переносимых приложений на мобильных платформах. Московский государственный университет им. М. В. Ломоносова, Москва, 2015