E-COMMERCE

Dmitriev V.A.

RECOMMENDATION SYSTEMS FOR ONLINE SHOP

Dmitriev V. A., Russian Federation, Moscow, Higher school of Economics, student

Abstract

Recommendation systems represent a diverse class of Web-based applications aimed at predicting the reaction of users to various requests. Inside of any recommendation system installed some algorithm of software implementation. This algorithm, on the one hand, must have a sufficiently high level of prediction and, on the other hand, should be simple enough, does not require much computer time. Therefore it was proposed an algorithm based on a consideration of the formation of a Euclidean space of states with the relevant metric. At ranging of assessment system must use a scale of Saati, which helps to identify all the coordinates in the interval [1,9], where 9 is the maximum deviation from the original item, 1 is the minimal deviation, where all the coordinates of the item viewed by the user are equal to 0, which allows to reduce the calculation significantly.

Keywords: e-commerce system, content-based filtering, collaborative filtering, recommendation system, recommendation service.

I. Introduction

One of the five Internet trends mentioned the magazine Forbes is the individual approach to the client that exactly provided by recommendation services [1]. The theory of recommendation systems appeared at the beginning 2000s, when Amazon implements its first Internet service [2]. Currently, recommender systems are actively used by the most of large online retailers who want to increase the average check and revenues. For example, 70% of the Amazon interface is filled by recommendations. Modern

recommendation services increase fullness of online baskets at 12-60%.

Depending on the available data about the user, recommended items and subject area, it can be selected different approaches to service implementation of the recommendations. Four of them are the most common:

- recommendations which are created manually by employees of online shop,
- content-based filtering approach,
- collaborative filtering approach,
- hybrid recommenred systems.

This article is devoted to content-based filtering recommendation systems. Such systems are based on the comparison of the characteristics of the items between them. Indicators of the effectiveness of this type of system reach a maximum in cases where the item viewed by the user "Not available" or "Out of stock". Online store should offer to user the similar items (goods of the same category, about the same price) in order to avoid loss of profit.

II. Existing algorithms

A. General concepts

Perfect service must combine data on a specific client, the general data on the behavior of customers, information on the properties of items and on the context of the current attention to the client (i.e. at the same time be the integration of first three approaches previously reviewed). The structure of such recommender systems is shown in the Fig. 1.

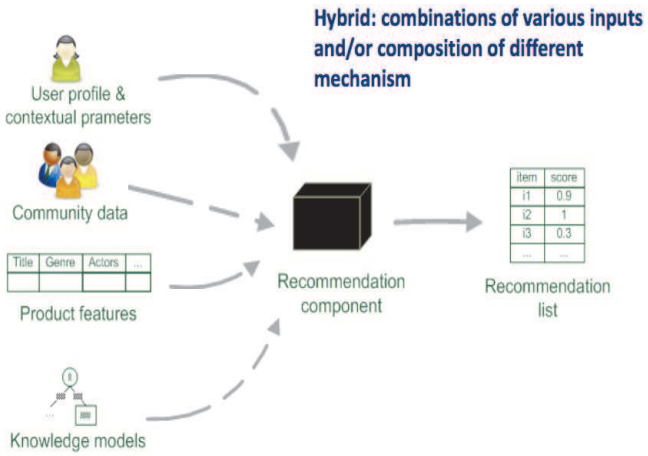

Figure 1. Paradigms of recommendation systems

In this case, the user will see items in accordance with decreasing relevance (fixed number of items from i_1 to i_k). There is one disadvantage: for the treatment of a huge amount of information required huge volumes of data. Before the advent of technologies of Big Data, operate with such huge volumes was impossible. Therefore many online retailers still use a subset of the entire set of data. This reduces processing time.

B. Algorithms of content-based filtering approach

The work of any recommendation system begins with the addition of items that are added to the website indicating certain characteristics (name, company, year, price, stock, etc.) in a form which is suitable for the description of the programmer and the further work. After that, the items need to be compared in order to determine the functional relations between the items. As a metric of comparison can be used the following special measures:

- the Hamming distance,
- the Levenshtein distance,
- the weighted Levenshtein distance,
- the Jaro-Winkler distance (a measure of Jaro-Winkler).

The program contains the comparison algorithm (metric comparisons), called a comparator. The comparator output should be 0 if the items are completely different, and 1 if they are identical, otherwise it outputs a value between 0 and 1. It sets a threshold value and, if the comparator derived values above it, then these values are acceptable. Assume that the threshold is 0.7. If the comparator establish a functional relations between the items A and B is equal to 0.6 and between the items D and C is 0.8, on the one hand, the D will be recommended the C, on the other hand, items A and B will not passed this test.

The most common comparator among small online shops is ExactComparator. Its operation is simple: firslly, it compares two strings element by element, then, the output returns 1, if elements are absolutly identity, otherwise it returns 0. Every characteristic has its weight, therefore it can be The advantages of this comparator are performance and productivity. However, there is one major disadvantage. This is best illustrated by the following example.

There is some online shop that sells mobile phones and smartphones; and there is a necessity to find recommendations to the selected IPhone 6. Assume that all smartphones have the following characteristics (properties): a brand, a price, a performance, a weight, a screen size, and a color. As part of a mathematical model, represent any smartphone which is stored in the directory as a vector X_i with

coordinates (a,b,c,d,e,f)(a, b, c, d, e, f), where a is a brand, b is a value, c is a performance, d is a weight, e is a diagonal of the screen and a color is f. Assume that in addition to IPhone 6 there are two items in the shop, one of which should be recommended for IPhone6: IPhone 4 and Samsung Galaxy S5. Finally, denote the number 1 brand Apple and the number 2 Samsung brand. In this case, consider the following values. This model is shown in the Fig. 2.

Figure 2. Example of operation of ExactComparator

In this model we consider the weighted average evaluation of all properties. In other words, the estimate obtained at the output of the comparator is $k_1 = \frac{1}{6} \cdot (a + b + c + d + e + f)$, где $a, b, c, d, e, f = 0 \vee 1$.

Therefore, comparator will return $k_1 = \frac{1}{6} \cdot (1+0+0+0+0+0) = \frac{1}{6}$, $k_1 = \frac{1}{6} \cdot (1 + 0 + 0 + 0 + 0 + 0) = \frac{1}{6}$, if IPhone 6 and IPhone 4 be loaded. At the same time it will return $k_2 = 6 \cdot \frac{1}{6} \cdot 0 = 0$, if IPhone 6 and Samsung Galaxy S5 be loaded. As a result, recommended item for IPhone 6 will be shown exactly IPhone 4 and not Samsung Galaxy S5, which it would be logical to recommend, based on other characteristics. Thus, its main problem is the lack of an intermediate state when comparing the two characteristics (elements).

Consider easyrec, which is the recommendation engine of Web-based application with an open source software enabling to

receive personalized recommendations. This engine is used by more than 10 large shops in Russia and Singapore, but the exact names of the retailers were not disclosed. As an algorithm for determining the recommended item acts profile similarity calculator. The algorithm produces comparing the numbers by using the following principle: "The comparator returns 1 if the numbers are the same and 0 if the different". When it is necessary to compare properties that require a verbal description (a brand, a color, as well as certain technical characteristics), this comparator uses the Levenshtein distance. Consider a simple example: need to calculate the Levenstein distance between the words "cat" and "code". In this case, distance is equal to 3 (necessary to carry out two replacements and one deletion of letters).

Reducing the Levenstein distance between two items increases the likelihood of their recommendations to each other. This comparator has its disadvantages, which should be clarified by an example. Firslly, comparison of numbers this comparator produces as well as ExactComparator, which is absolutely not acceptable. Secondly, metric of Levenstein has some limitations too. Consider online shop of smartphones which directory contains the following items: Huawei Ascend Mate 7, Samsung Galaxy S5, IPhone 6. Assume that there are only 2 characteristics, which descibe every item: a brand and a color. Like the previous example, it is also necessary to find a recommendation for iPhone 6. Moreover, every smartphone is also represented like a vector x with two coordinates (a, b), where a is a brand and b is a color. This model is shown in the Fig. 3. Levenstein distance between apple and huawei is equal to 5 and so on: $d(apple, huawei) = 5$, $d(apple, samsung) = 6$, $d(white, gold) = 5$ and $d(white, black) = 5$. Thus, Levenstein distance between IPhone 6 and Huawei Ascend Mate 7 is less, than distance between IPhone 6 and Samsung Galaxy S5. Such decision allows us to conclude that recommendations filter will provide a relatively low-budget Huawei to luxury Apple, instead of Samsung, which corresponds to the customers with the same average income like customers who prefer the company of Steve Jobs.

C. Discussion of the results

Firstly, it was considered ExactComparator which has significant drawbacks, the main of which is the lack of an intermediate state in the calculation of comparative measures between the properties of two items. Secondly, we reviewed easyrec, which, on the on hand, produces comparison of numbers as well as ExactComparator and, on the other hand, uses inappropriate in this case Levenstein distance when comparing properties that require a

verbal description. All these drawbacks should be corrected in our proposed system.

Figure 3. Example of operation of easyrec comparator

III. Our algorithm

A. Solution

The main creative part of creating a recommendation system is the choice of a metric for comparison of two items. We believe that it is necessary to select Euclidean distance as this metric [3]: $d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + ... + (p_n - q_n)^2}$ This measure calculates the distance between two n-dimensional vectors $(p(p_1,..., p_n); q(q_1,..., q_n))$, the coordinates of which are given in numerical form.

In this case, the calculations should be carried out evaluation of the proximity between the two items in the following sequence:

- each item in the shop has to be represented as a vector with n coordinates (numbers),
- comparator performs calculations using the Euclidean measure as an estimate of proximity.

There are two signifacant problems: firstly, translation of verbal characteristics in numeric, secondly, after translation we will have too large spread of values among the various characteristics (for example, among weight and resolution of a camera). Generally speaking, the solution of these two problems will solve the problems identified at the existing algorithms.

Rank (previous)	Manufacturer
1 (1)	Samsung
2 (2)	Apple
3 (8)	Huawei
4 (7)	Sony
5 (5)	ZTE
6 (4)	HTC
7 (6)	RIM
8 (9)	LG
9 (11)	Lenovo

Figure 4. Example of translation

Consider the solution of the first problem. We are sure that any verbal characteristic may be represented in numerical form, using a variety of information about a given characteristic. For solving the problem recommendations of the items of different price categories with different target audience, we offer the use various ratings. This example is presented in the Fig.4 [4]. Such conversion can be done with any properties of the item, whether it is color, the brand, the country of production or complete set.

Consider the solution of the second problem. There are several properties whose values vary in a geometric progression, for example, if there are 100 brands, but only 11 colors, comparing the first and one hundredth brands coordinate difference is equal to 99, but in comparison with the first and eleventh colors corresponding to the difference is equal to 10. In other words, the brand will be of paramount importance, which will lead to an error in the recommendations. In order to avoid errors it is necessary to normalized all properties without exception, i.e. assign values to all of them, for example, in the range [1, 9]. In this case, it is a range of Thomas L. Saati, which is often used in models of decision-making and determines the degree of significant deviation from 0 [5]. This scale helps to rank all of the values for a particular attribute, assigning 9 (maximum deviation), 8 (a very large deviation), 7 (large deviation), 6 (deviation above the mean), 5 (average deviation), 4 (deviation below the mean) 3 (small deviation), 2 (very small deviation), 1 (minimal deviation). These assessments are exposed as follows: if the similarity of the recommended product (Y) from the

viewed product (X) on a specific characteristic is below, than $\frac{100}{9} \approx 11.1\%$, then on the basis of assessment to grade Y as 9, if between $[\frac{100}{9}, \frac{100}{18}]$, i.e. [11.1%, 22.2%], it is necessary to make an assessment of 8, and so on. Accordingly, if a user views an item X, recommendation service should grade the rest of the items, taking all properties of the X as 0. This approach greatly reduces computation, since the comparison occurs with the vector with coordinates (0, ..., 0).

Finally, if the employee of online shop believes that the one property is more important than another one (for example, market trends), we can offer the following options:

Consider weighted Euclidean distance between the items, i.e. $d(p,q) = \sqrt{\sum_{i=1}^{n}(w_i \cdot (p_i - q_i)^2)}$ (1). This measure calculates the distance between two n-dimensional vectors $(p(p_1,..., p_n); q(q_1,..., q_n))$.

Reduction the scale range Saati. For example, if the brand twice as important, than the color, the color should be considered to the range [1, 9], and brand with the range [1, 4.5].

For clarity, it is an example: consider an online shop with directory where contains 4 different smartphones (smartphone appears as a vector X_i with coordinates (a, b, c, d, e, f, g), where a is a brand, price is a b, c is a performance, weight is a d, e is a diagonal, f is a color, g is a number of pixels of the front camera). This model is shown in the Fig. 5. Now we need to choose the recommendations in order of relevance for IPhone 6, then $X_{IPhone6}$(0, 0, 0, 0, 0, 0, 0). To calculate the weighted Euclidean distance between the vectors we introduce weights: $w_a = 0.3, w_b = 0.2, w_c = w_d = w_e = w_f = w_g = 0.1$. As a result, using (1) we obtain the following values: $d(X_{IPhone6}, X_{Samsung}) \approx 2$, $d(X_{IPhone6}, X_{IPhone4}) \approx 3.97$, $d(X_{IPhone6}, X_{Huawei}) \approx 3.59$. As a result, items inside recommendation system will be displayed in order of increasing distance of Euclid, i.e., Samsung will be the first, the second will be IPhone 4 and the last will be Huawei.

IV. Conclusion

Firstly, it should be noted that all the drawbacks of the existing recommendation systems, which we specified above have been corrected. We mean that we relieved the system from using Levenstein distance (translation of verbal characteristics to

numerical allows to do it) and entered the intermediate values in comparing the two characteristics.

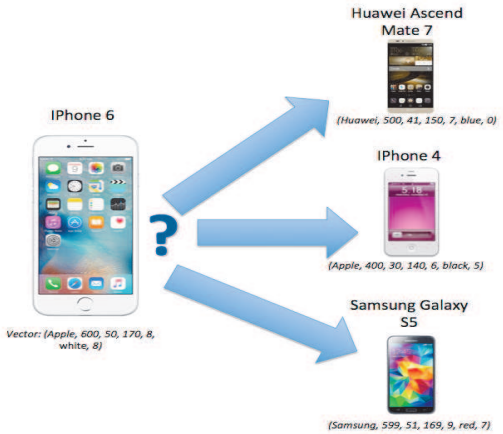

Figure 5. Example of opertion of our algorithm

The development of article is planned in the framework of the program implementation of the algorithm, a detailed study of its efficacy (formal and experimental methods) and to determine the scope of its application in the practice of online stores.

It is also planned to conduct studies to improve existing collaborative recommendation services by the use of metrics biophysics and genetics, such as a measure of the Simpson or Jaccard index.

References:

[1] Five Trends Shaping The Future Of Customer Service In 2015 / M. Blake. – Forbes magazine, 2014.

[2] Item-to-Item Collaborative Filtering / G. Linden, B. Smith, J. York // IEEE INTERNET COMPUTING, Los Alamitos. – CA USA, 2003.

[3] Encyclopedia of Distances / Deza, M. Marie. – Springer, 2013.

[4] Top 10 smartphone makers: Nokia last, Motorola out as Chinese companies take over / H. Viktor. – 2012.

[5] Decision Making with Dependence and Feedback: The Analytic Network Process / T. Saati. – RWS Publications, 1996