INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT

ISSN 2278 – 0211 (Online)
ISSN 2278 – 7631 (Print)

Multi-Objective Recommender System For IT Governance Requirements

Dr. Carsten Mueller
Department Of Information Technologies, University Of Economics, Prague, Czech Republic

Abstract:
IT Governance is a complex system. Information overload and an abundance of choices create situations where selecting one option becomes extremely difficult or even worse, a guessing game. Recommender systems are widely used to alleviate this problem by creating intelligent rankings of items based on an aggregation of user opinions. This research paper presents the multi-criteria ranking algorithm TOPSIS adapted to the problem of dynamic service selection provided based on IT Governance requirements.

Key words: *IT Governance, multicriteria decision making, recommender system, intelligent ranking*

1. Introduction

Information Technology (IT) is among the main capital investments and operating expenditures being made by organizations today. IT Governance (ITG) is specifying the decision rights and accountability framework to encourage desirable behavior in using IT [1]. ITG in a global context has to cater for intensive competition, cultural diversity, and various fluctuating economic conditions. ITG is a complex system. Each ITG implementation takes place in different conditions and circumstances determined by a large set of factors. Interoperability is one of the major challenges to be addressed in achieving an efficient ITG architecture [2].
The amount of information in the dynamic ITG environment is increasing far more quickly than our ability to process it. The recommendation is a way to help users in ITG to find information or services that are most likely to be interested or be relevant to their requirements.

2. Foundations

2.1.Recommender Systems

There are generally two fundamental methods to formulate recommendations both depending on the type of items to be recommended, as well as, on the way that user models [3] are constructed. The two different approaches are content-based [4], [5] and collaborative filtering [6], while additional hybrid techniques have been proposed as well [4].
The challenges for recommendation algorithms expand to three key dimensions, identified as sparsity, scalability and cold-start [7].
Sparsity: Even users that are very active, result in rating just a few of the total number of items available in a database. As the majority of the recommendation algorithms are based on similarity measures computed over the co-rated set of items, large levels of sparsity are detrimental to recommendation systems.
Scalability: Recommendation algorithms are efficient in filtering in items that are interesting to users. They require computations that are expensive and grow non-linearly with the number of users and items in a database. Sophisticated data structures and advanced, scalable architectures are required.
Cold-start: An item cannot be recommended unless it has been rated by a substantial number of users. This problem applies to new and obscure items and is particularly detrimental to users with eclectic taste. Likewise, a new user has to rate a sufficient number of items before the recommendation algorithm be able to provide reliable and accurate recommendations.

2.2. Technique For Order Of Preference By Similarity To Ideal Solution (TOPSIS)

A Multi-Criteria Decision Making (MCDM) problem is concisely expressed in a matrix format, in which columns indicate criteria (attributes) considered in a given problem; and in which rows list the competing alternatives.

Specifically in this context, a MCDM problem with m alternatives ($A_1, A_2, ..., A_m$) that are evaluated by n criteria ($C_1, C_2, ..., C_n$) is viewed as a geometric system with m points in n-dimensional space. An element x_{ij} of the matrix indicates the performance rating of the i^{th} alternative A_i, with respect to the j^{th} criterion C_j.
Hwang and Yoon [11] introduced the TOPSIS method based on the idea that the best alternative should have the shortest distance from the positive ideal solution and farthest distance from the negative ideal solution. TOPSIS method is a multi-attribute decision making approach and stands for technique for ordering preference by similarity to ideal solution [8], [9]. They assumed that if each criterion is monotonously increasing or decreasing, then it is easy to define an ideal solution. Such a solution comprises all the best achievable values of the criteria, while the worst solution is composed of all the worst criteria values achievable, the TOPSIS solution method consists of the following steps [10]:

1) Normalizing the decision matrix
The normalization of the decision matrix is done using the following transformation, for each x_{ij} .

$$n_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} x_{ij}^2}}, i = 1, \ldots, m; j = 1, \ldots n. \qquad (1)$$

2) Constructing the normalized weighted decision
The columns of the normalized decision matrix are multiplied by the associated weights as follows

$$v_{ij} = w_j \cdot n_{ij}, i = 1,2, \ldots, m, j = 1,2, \ldots, n \qquad (2)$$

, where w_j represents the weight of j^{th} criterion,

And $\sum_{j=1}^{n} w_j = 1$.

3) Determining the positive and negative ideal solutions
The positive and negative ideal value sets are determined, respectively, as follows

$$A^+ = (v_1^+, v_2^+, \ldots, v_n^+) = \left\{ \left(\max_j v_{ij} \middle| j \in \Omega_b \right), \left(\min_j v_{ij} \middle| j \in \Omega_c \right) \right\}, \qquad (3)$$

$$A^- = (v_1^-, v_2^-, \ldots, v_n^-) = \left\{ \left(\min_j v_{ij} \middle| j \in \Omega_b \right), \left(\max_j v_{ij} \middle| j \in \Omega_c \right) \right\} \qquad (4)$$

4) Determining the distance from ideal solutions
Two Euclidean distances for each alternative are calculated as follows:

$$S_i^+ = \sqrt{\sum_{j=1}^{n} \left(v_{ij} - v_j^+ \right)^2}, i = 1,2, \ldots, m, \qquad (5)$$

$$S_i^- = \sqrt{\sum_{j=1}^{n} \left(v_{ij} - v_j^- \right)^2}, i = 1,2, \ldots, m \qquad (6)$$

Where S_i^+ and S_i^- represents the distance of alternative A_i from the positive and negative ideal solutions, respectively.

5) Calculating the relative closeness to the ideal solution
The relative closeness to the ideal solution is defined as follows

$$RC_i = \frac{S_i^-}{S_i^+ + S_i^-}, i = 1,2,\ldots,m, 0 \le RC_i \le 1. \quad (7)$$

Where RC_i represents the relative closeness.

6) Ranking the alternatives
Alternatives must be ranked based on RC_i in which the highest score is the best alternative.

2.3. Information Entropy Weight Method

Entropy [11] is the measure of inter user similarity that exists during recommendation generation process. It is expressed in terms of discrete set of probabilities as given in Eq 8.

$$H(D(U_t, U_x)) = -\sum_{i=1}^{n} p(d_i)\log_2 p(d_i) \quad (8)$$

where, $D(U_t, U_x)$ is the difference score rating between the target user U_t and user U_x for n items and $p(d_i)$ is the probability density function of different score rating.

These probabilities depict the degree to which the target user U_t is similar to user U_x. Lower the entropy, higher the degree of inter user similarity.

The weight of the criterion reflects its importance in MCDM. In this paper, an objective weight is applied; named Information Entropy Weight (IEW) based on the information entropy of raw data [13]. Range standardization was done to transform different scales and units among various criteria into common measurable units in order to compare their weights.

$$x'_{ij} = \frac{x_{ij} - \min_{1 \le j \le n} x_{ij}}{\max_{1 \le j \le n} x_{ij} - \min_{1 \le j \le n} x_{ij}} \quad (9)$$

$D'(x')_{m\times n}$ is the matrix after range standardization; $\max x_{ij}$, $\min x_{ij}$ are the maximum and the minimum values of the criterion (j) respectively, all values in D' are ($0 \le x'_{ij} \le 1$). According to the normalized matrix $D' = (x')_{m\times n}$ the information entropy is calculated as shown in the following steps, first in order to avoid the insignificance of $\ln f_{ij}$ in Eq. (11) f_{ij} is stipulated as shown in Eq. (10):

$$f_{ij} = \frac{1 + x'_{ij}}{\sum_{i=1}^{m}(1 + x'_{ij})} \quad (10)$$

$$H_j = -\left(\sum_{i=1}^{m} f_{ij} \ln f_{ij}\right) i = 1,2,\ldots,m;\ j = 1,2,\ldots,n \quad (11)$$

After calculating the various degrees (H_j), the deviation degree of the criterion (j) noted by (G_j) is computed as in Eq. (12):

$$G_j = 1 - H_j,\ j = 1,2,\ldots,n \quad (12)$$

(G_j) is greater if the value of (H_j) is smaller consequently, if the (G_j) is higher, the information entropy (H_j) is lower, which indicates that the more the information criterion (j) provides the greater weight given to the criterion (j).

The weight (w_j) of the criterion (j) is defined as:

$$w_j = \frac{G_j}{\sum_{j=1}^{n} G_j} = \frac{1 - H_j}{n - \sum_{j=1}^{n} H_j} \tag{13}$$

Where $j = 1,2,\ldots,n$.

3. Recommender

Recommender systems provide the user with a list of recommended items they might prefer, or supply guesses of how much the user might prefer each item [8], [9].

As the mathematical model, $S = \{s_1, s_2, \ldots, s_m\}$ is defined as the vector of the service information and $F = \{f_1, f_2, \ldots, f_n\}$ is defined as the vector of the requirement's contextual features. To represent the relevance performance of the service s_i in the quantitative feature i, the decision matrix is constructed as the following:

$$D = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1n} \\ d_{21} & d_{22} & \cdots & d_{21} \\ \cdots & \cdots & \cdots & \cdots \\ d_{m1} & d_{m2} & \cdots & d_{mn} \end{bmatrix} \tag{14}$$

The decision matrix is normalized following the formula:

$$b_{ij} = \frac{d_{ij}}{\sum_{j=1}^{n} d_{ij}^2}, i = 1,2,\ldots,m; j = 1,2,\ldots,n \tag{15}$$

The normalized value b_{ij} is limited in [0,1]. The utility value of the service s_j is calculated using the formula:

$$RC_i = \frac{t_i^-}{t_i^- + t_i^+}, i = 1,2,\ldots,m \tag{16}$$

where,

$$t_i^+ = \sqrt{\sum_{j=1}^{n} \left(c_{ij} - c_j^+\right)}, i = 1,2,\ldots,m \tag{17}$$

$$t_i^- = \sqrt{\sum_{j=1}^{n} \left(c_{ij} - c_j^-\right)}, i = 1,2,\ldots,m \tag{18}$$

In the above equations, n is the number of requirement's contextual features, r_{ij} is the weighted normalized decision matrix which is calculated by

$$r_{ij} = w_j b_{ij}, i = 1,2,\ldots,m, j = 1,2,\ldots,n \tag{19}$$

where w_j means the requirement's relative need in this feature and r_j^+ , r_j^- are the positive and negative ideal solutions:

$$R^{+} = \{r_1^{+}, r_2^{+}, \ldots, r_n^{+}\} = \left\{\left(\max_i x_{ij} \middle| j \in I\right), \left(\min_i r_{ij} \middle| j \in J\right)\right\} \quad (20)$$

$$R^{-} = \{r_1^{-}, r_2^{+}, \ldots, r_n^{-}\} = \left\{\left(\min_i r_{ij} \middle| j \in I\right), \left(\max_i x_{ij} \middle| j \in J\right)\right\} \quad (21)$$

The more increase the relative closeness RC_i, the more important the utility value of the service s_i.
Finally, by performing the three stages systematically, the algorithm recommends a ranked list with the highest weighted target instances and the requirement obtains the most suitable services.

4. Service Selection Problem Based On IT Governance Requirements

The minimum requirements for its application in the ITG context are as follows Table I:

ITG Service Attribute A	1.5
ITG Service Attribute B	1900
ITG Service Attribute C	20000
ITG Service Attribute D	5.0
ITG Service Attribute E	3
ITG Service Attribute F	7

Table 1: Minimum Requirements

Attributes for the short-listed candidate services are shown in Table II:

Alt.	A	B	C	D	E	F
A_1	2.0	1500	20000	5.5	5	9
A_2	2.5	2700	18000	6.5	3	5
A_3	1.8	2000	21000	4.5	7	7
A_4	2.2	1800	20000	5.0	5	5

Table 2: Attributes For the Candidate Services

4.1. Calculating The Normalized Decision Matrix

The normalized decision matrix is calculated with Eq. 22

$$r_{ij} = \frac{d_{ij}}{\sqrt{\sum_{i=1}^{m} d_{ij}^{2}}} \quad (22)$$

$$N = \begin{bmatrix} 0.4671 & 0.3662 & 0.5056 & 0.5063 & 0.4811 & 0.6708 \\ 0.5839 & 0.6591 & 0.4550 & 0.5983 & 0.2887 & 0.3727 \\ 0.4204 & 0.4882 & 0.5308 & 0.4143 & 0.6736 & 0.5217 \\ 0.5139 & 0.4392 & 0.5056 & 0.4603 & 0.4811 & 0.3727 \end{bmatrix} \quad (23)$$

4.2. Assign Weights For Each Attribute

Assign weights for each attribute such that their sum will be equal one.

$$\sum_{i=1}^{n} w_i = 1 \quad (24)$$

$$w_1 + w_2 + w_3 + w_4 + w_5 + w_6 = 1 \quad (25)$$

w_1 = 0.2	**w_2 = 0.1**	**w_3 = 0.1**	**w_4 = 0.1**	**w_5 = 0.2**	**w_6 = 0.3**

Table 3: Weights for Each Attribute

The weights for each attribute are stored in a vector.

$$W = \begin{bmatrix} 0.2 \\ 0.1 \\ 0.1 \\ 0.1 \\ 0.2 \\ 0.3 \end{bmatrix} \tag{26}$$

Calculating the weighted normalized specification matrix.

Relative importance of the attributes with their normalized value is used to create unique parameter for the candidate service.

$$V_{ij} = N_{ij} W_i \tag{27}$$

$$V_{ij} = \begin{bmatrix} 0.0934 & 0.0366 & 0.0506 & 0.0506 & 0.0962 & 0.2012 \\ 0.1168 & 0.0659 & 0.0455 & 0.0598 & 0.0577 & 0.1118 \\ 0.0841 & 0.0488 & 0.0531 & 0.0414 & 0.1347 & 0.1565 \\ 0.1028 & 0.0439 & 0.0506 & 0.0460 & 0.0962 & 0.1118 \end{bmatrix} \tag{28}$$

4.3. Obtain The Ideal (V^*) And The Negative Ideal (V^-) Solutions From The Weighted Decision Matrix V

The weighted normalized attributes for the +ve and –ve benchmark are obtained as

$$V^* = [0.1168\ 0.0659\ 0.0531\ 0.0414\ 0.1347\ 0.201] \tag{29}$$

$$V^- = [0.0841\ 0.0366\ 0.0455\ 0.0598\ 0.0577\ 0.111] \tag{30}$$

4.4. Compute The Separation Measures

Compute the separation measures from the ideal (Si^*) and the negative ideal (Si^-) solutions for all alternatives, $i = 1, \ldots, m$.

$$Si^* = \sqrt{\sum_{j=1}^{n} \left(v_{ij} - v_j^*\right)^2} \tag{31}$$

$$Si^- = \sqrt{\sum_{j=1}^{n} \left(v_{ij} - v_j^-\right)^2} \tag{32}$$

Ideal solution	Ideal solution
$S_1^* = 0.0545$	$S_1^- = 0.0983$
$S_2^* = 0.1197$	$S_2^- = 0.0439$
$S_3^* = 0.0580$	$S_3^- = 0.0920$
$S_4^* = 0.1009$	$S_4^- = 0.0458$

Table 4: Values of Separation Measures

4.5. Determine Relative Closeness To Ideal Solution

For each alternative determine the relative closeness to the ideal solution $\left(C_i^*, i = 1, \ldots, m\right)$ as

$$\boxed{C_i^* = \frac{S_i^-}{\left(S_i^* + S_i^-\right)}} \quad (33)$$

Relative closeness values

- $C_1^* = 0.643$
- $C_2^* = 0.268$
- $C_3^* = 0.613$
- $C_4^* = 0.312$

The closeness rating is a number between 0 and 1, with 0 being the worst possible and 1 the best possible solution.

4.6. Rank The Preference Order

Determine the preference order by arranging the alter-natives in the descending order of $C_i^*, i = 1, \ldots, m$.

The ranks for the service alternatives in the requirement selection problem emerge as A1, A3, A4, A2.

5. Evaluation

5.1. Data Set

The experimental data comes from an in-house IT Governance recommendation system based on ITG components (see http://www.itg-components.com) named

ITG Service Recommendation System (ITGRS).

The ITGRS database currently consists of 2068 ratings provided by 114 requirements to 641 services, which belong to at least 1 of 21 categories in the context of ITG.

The lowest level of sparsity for the tests is defined as

$$\boxed{114 \times 641 - \frac{2068}{114} \times 641 \cong 0.9717} \quad (34)$$

5.2. Coverage Metric

Coverage is a measure of the percentage of items for which the recommendation system provides predictions. A basic coverage metric is the percentage of items to which predictions are available. Coverage is reduced by defining small neighborhood sizes or by sampling users to compute predictions.

5.3. Accuracy Metrics

The performance of recommender systems is often evaluated by the predictive accuracy and classification accuracy [12]. They are divided into two main categories: statistical accuracy and decision-support accuracy metrics.

Statistical accuracy metrics evaluate the accuracy of a pre-diction algorithm by comparing the numerical deviation of the predicted ratings from the respective actual user ratings. Frequently used are mean absolute error (MAE), root mean squared error (RMSE) and correlation between ratings and predictions [6].

MAE is used to evaluate the effectiveness of the approach. It measures the average absolute deviation between the predicted a predicted rating and the true rating is given by:

$$MAE = \frac{\sum_{\{u,i\}} \left| p_{u,i} - r_{u,i} \right|}{n} \tag{35}$$

Where $p_{u,i}$ the predicted rating for user u on item i . $r_{u,i}$ is the actual rating and n is the total number of rating. The lower the MAE, the more accurate the predictions would be, allowing for better recommendations to be formulated.

The MAE has two advantages: (1) Easy to calculate, easy to understand; (2) The evaluation standard is explicit and easy to evaluate the performance of different algorithms.

In Table V the comparison for a random recommender and the approach developed in this research article is presented. R is the random selection of recommendations and A is the approach developed in this research paper.

	0.972	**0.975**	**0.98**	**0.985**	**0.99**	**0.995**	**0.999**
R	3.166	3.515	3.414	3.024	3.256	3.174	3.398
A	0.838	0.915	1.065	1.142	1.284	1.626	1.662

Table 5: Statistical Accuracy Of Different Prediction Algorithms In Terms Of Mean Absolute Error (MAE) With Respect To Different Sparsity Levels

6. Conclusion

The typical MCDM approaches focus on a set of feasible alternatives and considers more than one criterion to deter mine a priority ranking for alternative information. The main purpose of this paper is to develop a TOPSIS method to select services appropriately bases on requirements for an ITG environment from available alternatives. The problem has been described as a multi-decision making method with the focus on (dynamic) service selection. A practical experiment was presented to valid its applicability.

7.References

1. P. Weill and J. Ross, IT Governance: How Top Performers Manage IT Decision Rights for Superior Results. Harvard Business School Press, 2004.
2. M. Turner, D. Budgen, and P. Brereton, “Turning software into a service,” Computer, vol. 36(10), pp. 38–44, 2003.
3. R. B. Allen, “User models: theory, method and practice,” International Journal of Man & Machine Studies, 1990.
4. M. Balabanovic and Y. Sholam, “Combining content-based and collaborative recommendation,” Communications of the ACM, vol. 40(3), 1997.
5. D. Kalles, A. Papagelis, and C. Zaroliagis, Algorithmic aspects of web intelligent systems. Springer, 2003, ch. Web Intelligence, pp. 323–345.
6. J. L. Herlocker, J. A. Konstan, and J. Riedl, “Explaining collaborative filtering recommendations,” in Proceedings of the ACM Conference on Computer Supported Cooperative Work, 2000.
7. Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” Computer, vol. 42(8), pp. 30–37, 2009.
8. J. Wei, “Topsis method for multiple attribute decision making with incomplete weight information in linguistic,” Journal of Convergence
Information Technology, vol. 5:10, 2010.
9. Z. Zeng, “A personalized product recommender system based on semantic similarity and TOPSIS method,” Journal of Convergence Information Technology, vol. 6:7, 2011.
10. Ahi, M. Aryanezhad, B. Ashtiani, and A. Makui, “A novel approach to determine cell formation, intracellular machine layout and cell layout in the CMS problem based on TOPSIS method,” Computers & Operations Research, vol. 36, pp. 1478–1496, 2009.
11. E. Shannon and E. Claude, “Prediction and entropy of printed english,” The Bell System Technical Journal, vol. 30, pp. 50–64, 1951.
12. L. Si and R. Jin, “Flexible mixture model for collaborative filtering,” Proceedings of the Twentieth International Conference on Machine Learning, 2003.