www.ijemr.net ISSN (ONLINE): 2250-0758, ISSN (PRINT): 2394-6962

Volume-5, Issue-2, April-2015

International Journal of Engineering and Management Research

Page Number: 383-385

# Guru-Mantra: Recommendation System for Career Enhancement of Employee

Kaustubh Ner[1], Sudarshan Ekambe[2], Shubham Dosi[3]
[1,2,3]Computer Department, Sinhgad College of Engineering, INDIA

**ABSTRACT**

**Provide career growth recommendations to an employee to achieve a target career milestone in terms of a skill/role/domain expertise leveraging enterprise-wide data repository capturing employee work history, growth, experience, etc.To estimate 'utility' or 'value' added by an event to an employee's utility or value (or a dimension of utility/value) and to understand possible career paths available for a profile.**

***Keywords*** – Recommendation System, Career Enhancement, Collaborative Filtering.

## I. INTRODUCTION

Recommender systems are increasingly being used to predict user preferences and suggest products and services based on his/her previous choices. Recommender systems have made their way into career recommendation and job recommendation systems. However, their use for intra-organization decision-making remains negligible.

Guru-Mantra is a recommender system that builds on an employee's interests and also his performance (based on employer provided performance ratings and also ratings taken from other sources) to recommend to him activities (projects, trainings, certifications) which are likely to improve his expertise in the recommended field in turn resulting in achieving career milestones such as appraisals.

The system will learn from employee profile and their work history, several most effective chain of events e.g. allocation-skill-project-training-certification-role change-promotion, which have certainly improved the 'utility' and performance of the employee.

To make improvement recommendations for a given time period in future by matching an employee's profile and work history with employees having best 'utility' and performance, possibly using sequence of events.

## II. BACKGROUND

The use of recommendation systems for intra industry purposes remains minimum. The general corporate firm focuses on the requirements of the industry at any time. However, it rarely takes into account the likes and dislikes of the employees. Consideration of which might result into higher productivity levels and a higher retention ratio.

## III. THE RECOMMENDATION SYSTEM

A recommendation system can be designed using many standard models. Some of which are [1]:

1. Collaborative Filtering: This method measures how similar one user is to the active user. This similarity is used to generate recommendations for the active user. The most commonly implemented collaborative filtering algorithm is the k-Nearest Neighbour i.e. kNN.
2. Content-Based Filtering: This method derives recommendations only based on the past choices of the user.
3. Demographic Filtering: This method uses the demographic factors such as age, sex, locality, etc for generation recommendations.
4. Hybrid: This method as the name suggests is a combination of two or more of the above mentioned methods. Most of the modern commercial recommendation systems are hybrid systems.

## IV. GURU-MANTRA

We have strived to implement a system which looks into the profile of the employee for whom the

recommendations need to be generated (this employee has been referred to as the active employee henceforth) and the profile of other employees. The system matches the employees with each other to derive a measure of similarity.

1. We have constructed a role hierarchy tree so that the active employee's recommendations can be limited to the employees who are closer to his role. This is done so that the generated recommendations remain relevant.
2. A technology tree and role hierarchy has been constructed so as to enable us to find similarities between different technologies. For this purpose we have implemented the taxonomy matcher.
3. Once the similarity between employees is found, we find the best times in the other employee's career i.e. promotions, role increments, etc.
4. Assuming that the employee has done some good work during this period, we extract his trainings and certifications.
5. These trainings and certifications are then ranked on the basis of user similarity and the active employee's preferences and dislikes.
6. The final ranked recommendations are displayed.

## V. BLOCK REPRESNTATION

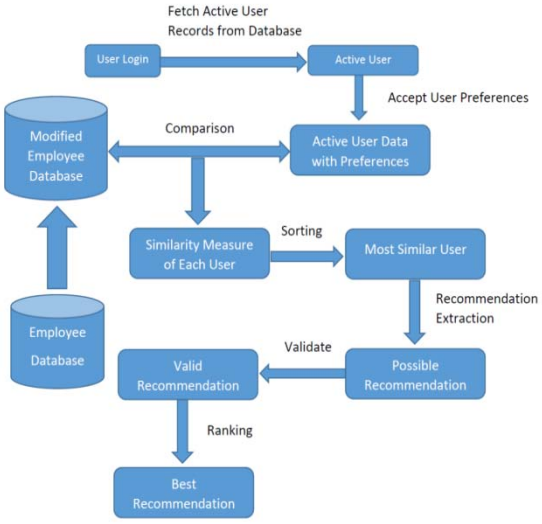


**Fig. 1. Block Diagram**

## VI. RESULT DISCUSSION

To analyze the results generated by the recommender system, we have written a test program to automatically run tests on the system. This program passes different inputs to the program and tests the results against the logically expected results. We have written 7 tests. Each of these tests were run on a set of 100 employees. The tests are as follows:

**Table 1 . Test Description**

| TEST | CRITERIA | RESULT |
|---|---|---|
| 1 | No Preference | Same As Major Tech |
| 2 | No Like, Positive Dislike | Any Other Than Major |
| 3 | No Like, Negative Dislike | Same As Major Tech |
| 4 | Positive Like, No Dislike | Same As Major Tech |
| 5 | Negative Like, No Dislike | Tech Given As Like |
| 6 | Positive Like, Negative Dislike | Same As Major Tech, Not Dislike Tech |
| 7 | Negative Like, Positive Dislike | Tech Given As Like, Not Dislike Tech |

On running the tests, we get the results as pass or fail. The criteria for deciding pass or fail is the occurrence or non-occurrence of a recommendation in the top 5 generated recommendations.
The test statistics are as follows:

**Table 2. Test Statistics**

| | Pass | Fail |
|---|---|---|
| Test 1 | 39 | 61 |
| Test 2 | 65 | 35 |
| Test 3 | 98 | 2 |
| Test 4 | 55 | 45 |
| Test 5 | 99 | 1 |
| Test 6 | 82 | 18 |
| Test 7 | 92 | 8 |

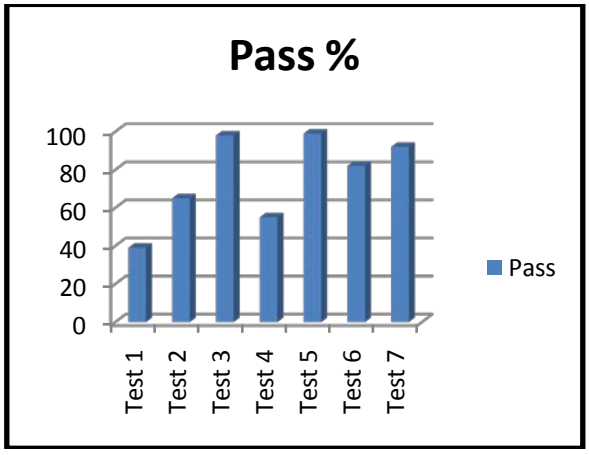


**Fig. 2. Accuracy Testing Graph**

## VII. CONCLUSION

The implemented system is capable of generating good recommendations provided the data is available for it to learn. The system is also capable of improving its recommendations over time as more and more data becomes available for it to learn from

## VIII. ACKNOWLEDGEMENTS

The Guru-Mantra system is sponsored by Tata Research Development and Design Center, Pune and ably guided by Mr. Rajiv Srivastava. Also thanks to mentor and guide Prof. P. M. Kamde from Sinhgad College of Engineering, Pune.

## REFERENCES

[1] Bobadilla, Jesús, et al. "Recommender systems survey." Knowledge-Based Systems 46 (2013): 109-132.
[2] Sarwar, Badrul, et al. "Item-based collaborative filtering recommendation algorithms." Proceedings of the 10th international conference on World Wide Web. ACM, 2001.
[3] Goldberg, Ken, et al. "Eigentaste: A constant time collaborative filtering algorithm." Information Retrieval 4.2 (2001): 133-151.
[4] Linden, Greg, Brent Smith, and Jeremy York. "Amazon.com recommendations: Item-to-item collaborative filtering." Internet Computing, IEEE 7.1 (2003): 76-80.
[5] Zhang, Jing, et al. "Collaborative filtering recommendation algorithm based on user preference derived from item domain features." Physics A: Statistical Mechanics and its Applications 396 (2014): 66-76.
[6] Sammut, Claude, and Geoffrey I. Webb, eds. Encyclopedia of machine learning. Springer Science & Business Media, 2011.
[7] Deng-Neng, C. H. E. N., and Yao-Chun Chiang. "Combining personal Ontology and collaborative filtering to design a document recommendation system."Journal of Service Science and Management 2.04 (2009): 322.
[8] Yu, Hwanjo, and Sungchul Kim. "SVM Tutorial—Classification, Regression and Ranking." Handbook of Natural Computing. Springer Berlin Heidelberg, 2012. 479-506.
[9] Radlinski, Filip, and Thorsten Joachims. "Query chains: learning to rank from implicit feedback." Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining. ACM, 2005.
[10] Joachims, Thorsten. "Optimizing search engines using clickthroughdata."Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2002.