A Collaborative Filtering Method to Improve the Incident Handling in IT Service Management

Research-in-Progress

Zhe Shan
Lindner College of Business,
University of Cincinnati,
Cincinnati, OH, USA
zhe.shan@uc.edu

Rong Liu
IBM T.J. Watson Research Center,
1101 Kitchawan Rd,
Yorktown Heights, NY, USA
rliu@us.ibm.com

1 Introduction

IT incident management aims to restore normal service operation as quickly as possible and minimize the adverse effect on business operations (The ITIL Open Guide 2010). An incident is any event which is not part of the standard operation of a service and which causes, or may cause, an interruption to or a reduction in the quality of that service. Incident management becomes more and more important as the role of IT in business is critical than ever before. Due to the complexity of IT environment, such as network, hardware, software, etc., usually the incident diagnosis at application level is very challenging (Liu and Lee 2012). The traditional approach passes the unsolved incident from one subject matter expert (SME) to another in a sequential way, which prolongs the processing time, and leads to costly delay in customer service. In this paper, we propose to use the history information of SME's incident processing to identify their preferences or expertise on different types of IT incidents. Based on the generated preference matrix, we can identify an appropriate expert or expert group for different kinds of incidents, which potentially improves the efficiency of incident handling.

2 Background

In the area of IT Service Management (ITSM), Information Technology Infrastructure Library (ITIL) provides a set of practices that focuses on aligning IT services with the needs of business (Rozemeijer and Van Bon 2007; The ITIL Open Guide 2010). Recently, there is growing interest in using statistical analytics to analyzing and managing incidents. In (Zhang et al. 2005), an ensemble of Tree-Augmented Bayesian Network models is provided to correlate workload metrics with service level objectives. A service Delivery Portal introduced in (Lenchner et al. 2009) provides a set of technologies to help system administrators (SA) to diagnose and manage incidents. This platform aggregates various incident data sources and allows SA to search for relevant events based on keywords or incident attributes.

In this paper, we view the SME-incident paring problem in IT incident management as a recommender system problem. Basically, we compare a SME profile to some reference characteristics, and seek to predict the *rating* (sometimes called utility) that a SME would give to an incident which they had not yet considered or decided to accept. Suppose each SME has a unique utility threshold. He only accepts those incidents that have a utility beyond this threshold. Given a list of incidents with highest predicted utility (N best events), we can decide to incident assignments for SMEs.

Recommender systems are usually classified into content-based recommendations, collaborative filtering recommendations, and hybrid approaches (Adomavicius and Tuzhilin 2005). In content-based recommendation methods, the utility of incident *a* for a SME *x* is estimated based on the utilities assigned by this SME to other incidents, which are "similar" to incident *a*. However, the content-based techniques have a number of limitations. For instance, SMEs are limited to being recommended incidents that are similar to those already processed. And, SMEs have to process a sufficient number of incidents before the system can really understand the SMEs' preferences and present the SME with reliable recommendations (Adomavicius and Tuzhilin 2005).

Unlike content-based recommendation methods, collaborative recommender systems try to predict the utility of incidents for a particular SME based on the similar incidents previously processes by other SMEs. For example, in order to recommend incidents to SME *x*, a collaborative recommender system would try to find the similar peers of SME *x*, i.e., other SMEs who have successfully processed similar incidents. Then, only those incidents that are most likely processed successfully by the peers of SME *x* would be recommended. The algorithms for collaborative recommendations can be grouped in two general classes: memory-based and model-based. Memory-based algorithms essentially are heuristics that make incident predictions based on the entire collection of previously processed incidents by the SMEs, while model-based algorithms use the collection of past incidents processing to learn a model, which is then used to make predictions for future incident engagement. In this paper, we start with a memory-based algorithm. Note that, although collaborative systems do not have some of the shortcomings that content-based systems have, they have their own limitations. For instance, they cannot recommend new types of incidents, or recommend incidents to new SMEs. And, they do not take into account contextual knowledge.

3 Proposed Model

In this section, we customize the item-based neighborhood-based CF algorithm (Sarwar et al. 2001) for the IT incident assignment problem.

The algorithm uses the entire SME-incident database to generate a prediction. Every incident is part of a group of incidents with similar features. By identifying the so-called *neighbors* of a new incident, a prediction of preferred SMEs to handle this incident can be produced. It uses the following steps:

a) Rate each occurred engagement between SMEs and incidents.
b) Calculate the similarities between each two incidents (Liu and Lee 2012).
c) Produce a prediction for the active SME by taking the simple average of all the ratings of the SME on related incidents.
d) Find *k* most similar incidents after computing the similarities, then aggregate the neighbors to get the top-N most frequent SMEs as the recommendation.

3.1 Engagement Rating

Unlike the Netflix case in which user rates a movie in a scale of 5 (0-4), the engagement between SME and incident does not conclude with a score. So how should we measure the service quality in each engagement? Figure 1 shows the general information contained in the incident ticket.

Incident ID: INC1
Open Time: 8/3/2010 8:31:47 AM Close Time: 7/16/2010 6:55:20 AM
Description: The USER xxx has a successful *login* into the hub after *registration*, but he is *unable to access* SAP. Every time when he clicks on Sap work place, the screen goes blank!

Incident ID: INC2
Open time: 8/23/2010 2:02:16 PM Close Time: 7/28/2010 6:34:41 PM
Had system reimaged a few months ago, has not been able to perform *goods movement*, had previous incident opened (INCx) to report *authorization* issue, but is still unable to complete work, referring back to appropriate parties at higher severity. Please help to check.

Figure 1. Incident Examples (Liu et al. 2013)

For those measures listed in (The ITIL Open Guide 2010; UCISA 2011), most of them are designed for the summary level, not individual engagement. Based those measures, we developed our own metrics as follows,

- The engagement rating ranges from 0 to 1, in which *0* is the best, and *1* is the worst.
- If an incident is rejected by the assigned SME, this engagement is rated as *1*.
- Otherwise, the engagement is rated as $\frac{1}{1+\frac{t_s}{t_a}}$, where t_s is the actual service time, and t_a is the average service time. Smaller t_s is, closer the rate is to 1. When t_s becomes larger, the rate is close 0.
- The average time to achieve incident resolution t_a can be broken down by the following unit levels: Type, Category, Priority-impact-urgency, and Service (UCISA 2011).

3.2 Similarity Computation

For item-based CF algorithms, the basic idea of the similarity computation between item *i* and item *j* is first to work on the users who have rated both of these items and then to apply a similarity computation to determine the similarity, $w_{i,j}$, between the two co-rated items of the users. However, in our context most incidents only occurred once. So each incident has only one assigned SME, and correspondingly one rate. Therefore, it is not feasible to apply the traditional item-based algorithm. Instead, we adopt a vector-space model (Salton et al. 1975) with slight modifications to calculate the similarity score for returned incidents, as we did in (Liu and Lee 2012). In other words, the similarity between two documents can be measured by treating each document as a vector of word frequencies and computing the cosine of the angle formed by the frequency vectors, as shown below,

$$W_{i,j} = \cos(\vec{i}, \vec{j}) = \frac{\vec{i} \bullet \vec{j}}{\|\vec{i}\| * \|\vec{j}\|} \tag{1}$$

Figure 2 shows an example in a table form.

Incident		Keyword Vector							$v_1 \cdot v_2$ (or v_3)	$\lvert v_1\rvert * \lvert v_2\rvert$ (or $\lvert v_3\rvert$)	*Similarity*
		SomeApp	*ServerXYZ*	*ServerXYZ2*	*mq*	*mq conn.*	*dependent app*				
							DepApp1	*DepApp2*			
v_1	IN1	1	1	1	1	1		2			
v_2	IN2	1	1		1				3	$\sqrt{5} * \sqrt{3}$	0.77
v_3	IN3		1					2	5	$\sqrt{9} * \sqrt{5}$	0.75

Figure 2.Calcualting Similarity Score (Liu and Lee 2012)

3.3 Prediction and Recommendation Computation

In the neighborhood-based CF algorithm, a subset of nearest neighbors of the active incident are chosen based on their similarity with it, and a weighted aggregate of their ratings is used to generate predictions for the active incident.

For the incident-based prediction, we adopt the simple weighted average to predict the rating, $P_{e,i}$, for SME e on incident *i*.

$$P_{e,i} = \frac{\sum_{n \in N} r_{\mathrm{e,n}} w_{\mathrm{i,n}}}{\sum_{n \in N} | w_{\mathrm{i,n}} |} \tag{2}$$

where the summations are over all other rated incidents $n \in N$ for SME *e*, $w_{i,n}$ is the weight between incidents *i* and *n*, $r_{e,n}$ is the rating for SME *e* on incident *n*.

3.4 Incident-based Top-N Recommendations Algorithm

Top-N recommendation is to recommend a set of N top-ranked SMEs that will be appropriate to a certain incident. Top-N recommendation techniques analyze the SME-incident matrix to discover relations between different incidents and use them to compute the recommendations.

Incident-based top-N recommendation algorithms firstly identify the *k* most similar incident (nearest neighbors) to the incident using the technique introduced in (Liu and Lee 2012), in which each incident is treated as a vector in the *m*-dimensional keyword space and the similarities between the active incident and other incidents are computed between the vectors.

After the *k* most similar incidents have been discovered, their corresponding rows in the SME-incident matrix *R* are aggregated to identify a set of SMEs, *C*, who engaged with the group together with their ratings. With the set *C*, incident-based CF techniques then recommend the top-N most helpful SMEs in *C* that the active incident has not been assigned (in case it has been assigned to some SME, and then be rejected).

4 Discussion and Conclusion

IT incident management, which ensures high levels of service quality and availability, is a significant challenge primarily because enterprises often maintain many applications in shared dynamic IT environments. In this paper, we present a collaborative filtering algorithm for incident recommendation. By mining the past experiences of incident processing of SMEs, we expect to find most capable SMEs for each incident, and therefore significantly increase the efficiency of incident handling. Currently, we are finalizing the data cleaning, and will report the preliminary results at the conference presentation.

Although the algorithm introduced in this paper is incident-based, we also plan to use the social network information mined from (Liu et al. 2013) to calculate the similarity between SMEs. Based on this similarity, we can develop a SME-based algorithm, similar to the user-based CF algorithm. Moreover, by collecting more background information of SMEs, we plan to use a hybrid and probabilistic approach by combining content-based and collaborative methods, which helps to increase the accuracy of SME recommendation and therefore improve the service quality of incidents. In the hybrid approach, the unknown utilities are calculated as

$$u_{x,a} = E(u_{x,a}) = \sum_{i=0}^{n} i * \Pr(u_{x,a} = i | u_{x,a'}, a' \in A_c) \quad (3)$$

We assume that rate values are integers between *0* and *n* (we define rate as [0,1] in section 3.1), and the probability expression is the probability that SME *x* will give a particular rate to incident *a* given that SME's ratings of the previously engaged incidents. To estimate this probability, we plan to use two models: cluster models and Bayesian networks. In the first model, like-minded SMEs are clustered into classes. Given the SME's class membership, the SME ratings are assumed to be independent, i.e., the model structure is that of a naïve Bayesian model. The number of classes and the parameters of the model are learned from the data. The second model represents each incident in the domain as a node in a Bayesian network, where the states of each node correspond to the possible rating values for each incident. Both the structure of the network and the conditional probabilities are learned from data. Then, we can combine the outputs obtained from both models into one final recommendation using either a linear combination or a voting scheme. By this way, we incorporate both the profile information of SMEs and incidents into a single statistical model.

5 References

Adomavicius, G., and Tuzhilin, A. 2005. "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," *IEEE Trans. on Knowl. and Data Eng.* (17:6), pp. 734-749.

Lenchner, J., Rosu, D., Velasquez, N. F., Guo, S., Christiance, K., DeFelice, D., Deshpande, P. M., Kummamuru, K., Kraus, N., and Luan, L. Z. 2009. "A Service Delivery Platform for Server Management Services," *IBM Journal of Research and Development* (53:6), pp. 2: 1-2: 17.

Liu, R., Agarwal, S., Sindhgatta, R. R., and Lee, J. 2013. "Accelerating Collaboration in Task Assignment Using a Socially Enhanced Resource Model," in Proceedings of *11th International Conference on Business Process Management*: Springer, pp. 251-258.

Liu, R., and Lee, J. 2012. "It Incident Management by Analyzing Incident Relations," in *10th International Conference on Service-Oriented Computing*. Springer, pp. 631-638.

Rozemeijer, E., and Van Bon, J. 2007. *Frameworks for It Management*. Van Haren.

Salton, G., Wong, A., and Yang, C.-S. 1975. "A Vector Space Model for Automatic Indexing," *Communications of the ACM* (18:11), pp. 613-620.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. 2001. "Item-Based Collaborative Filtering Recommendation Algorithms," in: *Proceedings of the 10th international conference on World Wide Web*. Hong Kong, Hong Kong: ACM, pp. 285-295.

The ITIL Open Guide. 2010. "Itil Incident Management." Retrieved on Sep 26, 2014, from http://www.itlibrary.org/index.php?page=Incident_Management

UCISA. 2011. "Itil – Incident Management: Key Performance Indicators (Kpis) and Reports." Available at https://www.ucisa.ac.uk/~/media/Files/members/activities/ITIL/service_operation/incident_management/ITIL_IM%20KPIs%20and%20reports%20pdf.ashx

Zhang, S., Cohen, I., Goldszmidt, M., Symons, J., and Fox, A. 2005. "Ensembles of Models for Automated Diagnosis of System Performance Problems," in Proceedings of *Dependable Systems and Networks, 2005. DSN 2005. Proceedings. International Conference on*: IEEE, pp. 644-653.