Using Data Mining and Recommender Systems to Facilitate Large-Scale, Open, and Inclusive Requirements Elicitation Processes

Carlos Castro-Herrera[+], Chuan Duan[+], Jane Cleland-Huang[+], and Bamshad Mobasher[*]
Systems and Requirements Engineering Center[+]
Center for Web Intelligence[*]
DePaul University
{ccastroh, duanchuan, jhuang, mobasher}@cs.depaul.edu

Abstract

Requirements related problems, especially those originating from inadequacies in the human-intensive task of eliciting stakeholders' needs and desires, have contributed to many failed and challenged software projects. This is especially true for large and complex projects in which requirements knowledge is distributed across thousands of stakeholders. This short paper introduces a new process and related framework that utilizes data mining and recommender technologies to create an open, scalable, and inclusive requirements elicitation process capable of supporting projects with thousands of stakeholders. The approach is illustrated and evaluated using feature requests mined from an open source software product.

1. Problem statement

Requirements elicitation is a human-intensive task in which analysts proactively identify stakeholders' needs, wants, and desires, using a broad array of elicitation techniques such as interviews, surveys, brainstorming sessions, Joint Application Design (JAD), and ethnographic studies [4]. Unfortunately, there are numerous accounts of large projects which have failed, primarily due to problems in scaling up the requirements process. This short paper describes a new requirements framework which utilizes data mining and machine learning techniques to address these problems in large-scale systems.

In our framework, stakeholders' needs are first gathered using a web-enabled elicitation tool. The needs are then processed using unsupervised clustering techniques in order to identify dominant and cross-cutting themes around which a set of discussion forums are created. Stakeholders are assigned to these forums based upon the needs they have contributed. They then work collaboratively in these forums to transform their needs into more formal requirements. To help keep stakeholders informed of relevant forums and requirements, our framework also utilizes a collaborative recommender system which recommends forums based on the interests of similar stakeholders. These additional recommendations increase the likelihood that critical stakeholders will be placed into relevant forums in a timely manner.

The need for this type of recommender system is clearly illustrated through an examination of the requirements features of open source projects. For example, in SugarCRM, a large open-source customer management system, users create new feature requests by browsing through a list of existing threads and determining whether to submit to an existing thread or create a new one. An analysis of the resulting threads showed that many users created either a new thread for each feature request, or placed requests into one or two mega-threads. Neither of these approaches is ideal in an online requirements gathering tool, as the resulting threads are either too isolated or too large to effectively support collaborative requirements activities.

2. Forum recommendations

Recommender technologies, such as those adopted in our framework, have traditionally been used in information systems to dynamically target content to one or more users, and also in e-commerce domains to recommend products to customers [1]. The recommendation problem is typically formulated as a prediction task in which a predictive model is built according to prior training data and then used in conjunction with the dynamic profile of a new user to predict the level of interest of that user in a target item. Recommender systems generally fall into three categories: content-based systems which make recommendations based on semantic content of data

1090-705x/08 $25.00 © 2008 IEEE
DOI 10.1109/RE.2008.47

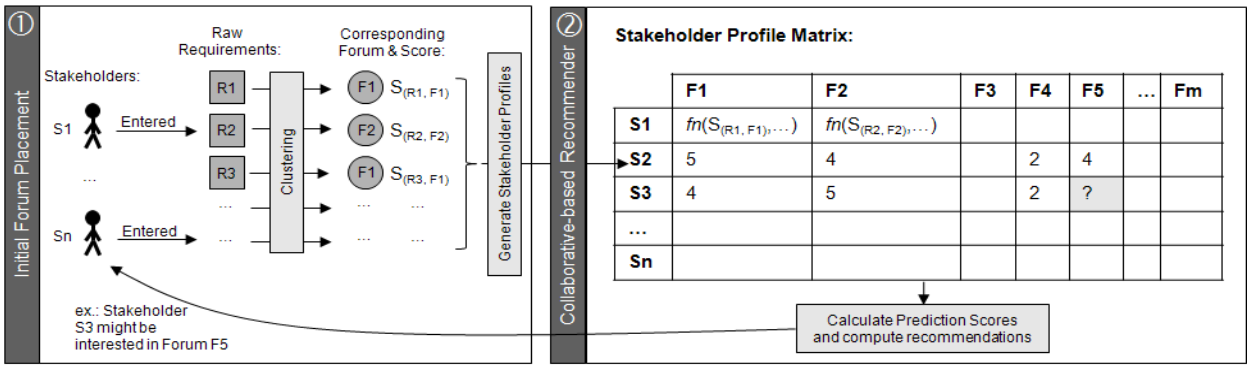


Figure 1. Content-based and Collaborative recommenders

[7], collaborative-filtering systems which make recommendations by examining past interactions of a user with the system and then identifying other stakeholders with similar interests [8], and knowledge-based systems which make recommendations based on knowledge of the user and pre-established heuristics. Our proposed model is a hybrid one that recommends forums to stakeholders and infers knowledge of the user by examining the distribution of topics across the stakeholders' needs.

2.1 Forum creation

Following standard information retrieval techniques, stakeholders' needs were initially stemmed to their root forms, common (stop) terms were removed, and the tf-idf (term frequency, inverse document frequency) values for all remaining terms were computed[6]. Intuitively, tf-idf weights terms more highly if they occur less frequently and are therefore expected to be more useful in expressing unique concepts in the domain. Each need was represented as a weighted vector of terms, and the complete set of vectors χ were then used to determine the optimal number of clusters K heuristically using a technique known as cover coefficient presented by Can [2].

Based on extensive experimentation [5], we adopted the bisecting clustering algorithm to dynamically build the discussion forums. This algorithm, which relies on K-means clustering to consecutively bisect a larger cluster into two smaller ones, was chosen because of its fast running time and relatively high quality results. The clustering algorithm first assigns all needs to a single cluster, and then each cluster c_i in the current clustering C, is bisected using 2-means clustering. The objective function E is computed over the resulting clusters as the sum of cohesion where $E = \sum_{i=1}^{K} \sum_{x \in c_i} s(x, m_i)$, for which m_i is the centroid of cluster c_i, and $s(x, m_i)$ represents the similarity score between artifact x and m_i, computed using a standard cosine similarity. The cluster that returns the highest gain of E score after bisecting is removed from the clustering, and replaced by its two derived clusters.

2.2 User profiles

Typically, a recommender system constructs a user's profile by inferring customers' interests through examining a history of purchases or web-pages viewed, or through explicit ratings that the customer may have made for various items. However, in our framework, a stakeholder's membership in a given forum is predicted by considering both the quantity of the stakeholder's raw needs placed into the associated cluster, and the distance d of each of these needs to the centroid. More formally membership level M of stakeholder s in forum f is computed as follows $M(s,f) = \sum_{x \in R_s}(p_x) / \sum_{x' \in R}(p_{x'})$ where R represents all the requirements in the forum, R_s represents those requirements in the forum belonging to stakeholder s, and p_x represents the similarity, or proximity, of requirement x to the centroid of the cluster. Intuitively this computes the fraction of forum f 'owned' by stakeholder s. In our initial study, stakeholders scoring membership values greater than zero for any forum are considered to be members of that forum.

2.3 Collaborative Filtering recommenders

Placing stakeholders into forums based only on their prior contributions and stated interests, misses the opportunity for more proactive recommendations. Collaborative recommendations can therefore be made by identifying neighborhoods of users with similar interests, and then using these neighborhoods to predict the interest that a particular user might have in

a forum for which he or she has no current known interests. Such collaborative recommenders could be particularly useful in the requirements process, as they facilitate the cross-pollination of ideas.

The following algorithm, which is often used in collaborative filtering recommenders, was used in our experiments to predict the level of interest that a user *u* might have in forum *f* given a set of ratings (ranged over *r*, with $\bar{r}$ indicating an average rating):

$$pred(u, f) = \bar{r}_u + \frac{\sum_{n \in nbr(u)} userSim(u, n) \cdot (r_{ni} - \bar{r}_n)}{\sum_{n \in nbr(u)} userSim(u, n)}$$

and where $n \in nbr(u)$ represents that *n* is a neighbor of *u* [8]. Note that in our context, the membership score of a stakeholder in a forum is used as the rating.

Intuitively, this computes the average ratings that neighbors have given a forum, while taking into consideration the similarity of the neighbors and the fact that some users are more optimistic than others. User similarity, *userSim(u,n)*, between user *u* and the neighbor *n* was computed as follows, using a version of the Pearson correlation in which $CR_{u,n}$ denotes the set of corated items between *u* and *n*:

$$userSim(u, n) = \frac{\Sigma f \in CR_{u,n} (r_{uf} - \bar{r}_u)(r_{nf} - \bar{r}_n)}{\sqrt{\Sigma f \in CR_{u,n} (r_{uf} - \bar{r}_u)^2} \sqrt{\Sigma f \in CR_{u,n} (r_{nf} - \bar{r}_n)^2}}$$

This correlation metric generates numbers between 1 and -1, where users in perfect agreement score 1, and users in perfect disagreement score -1. Such recommendations can only meaningfully be made to those stakeholders who have sufficient interests registered in their user profile to support neighbor identification.

3. Evaluating the framework

The effectiveness of this recommender system in the requirements domain was evaluated using a set of 1000 feature requests mined from SugarCRM. These feature requests were contributed by 523 different stakeholders over a two year period, and distributed across 293 threads.

3.1 Unsupervised clustering of Sugar data

The SugarCRM feature requests were clustered using the bisecting clustering algorithm and the quality of the resulting clusters was evaluated using two standard cohesion (CH1 and CH2) metrics and one standard coupling (CP) metric described by Zhao et al [10]. The results, depicted in Table 1, show that the automated clustering methods adopted in this paper returned significantly higher quality clusters than a random approach, however not quite as good as the human created clusters. It should be noted that these standard metrics are skewed against our experiment,

Table 1. Coupling and cohesion of Sugar data

Clustering Type	Cohesion		Coupling
	CH1	CH2	CP
Native threads	44.40	565.29	326.97
Dynamic clusters	39.38	467.73	380.47
Random clusters	10.66	247.67	444.88

because they provably favor the characteristics of the native threads which contained numerous small clusters. Nevertheless, the challenge of building large-scale systems makes it imperative to improve requirements clustering techniques in order to provide automated support for online requirements processes.

In addition to the quality metrics, Normalized Mutual Information (NMI) was used to evaluate the similarity between the generated clusters and the native user threads. NMI [9] is a well-known information theoretical clustering comparison metric which measures the extent that the knowledge of one clustering reduces uncertainty of the other. Informally, the metric calculates the amount of mutual information between each pair of clusters across two clusterings, then computes the normalized sum of the pair-wise mutual information scores. An evaluation of our automatically generated forums returned an NMI score of 0.670 which means that there are significant similarities between the two clusterings, and implies that the automated technique produces a significant number of clusters that have cohesive themes and would be acceptable to human stakeholders.

3.2 Collaborative Recommendations

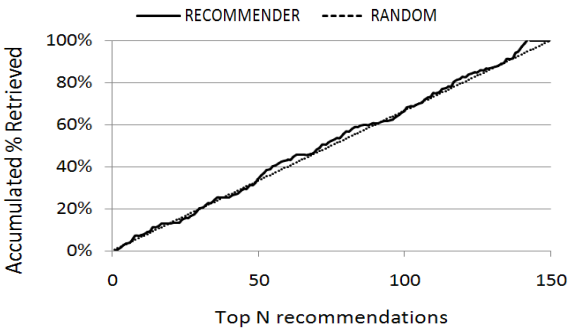
Because it was not feasible to ask the original SugarCRM stakeholders to evaluate the usefulness of collaborative filtering recommendations, we adopted a standard leave-one-out cross validation technique that iteratively removed one known interest at a time, and then measured the ability of the collaborative filtering recommender to recommend it back to the user. In our context, an interest is represented by the membership score of a stakeholder in a forum. The following steps were taken for each known forum of interest *i* for each stakeholder *s* who had three or more registered interests: (1) interest *i* was removed from the user profile of stakeholder *s*, (2) the collaborative recommender was run and recommendations were generated for stakeholder *s*, (3) the recommendations were ranked according to score, (4) the results were analyzed to determine if the desired interest was recommended in the top *n* recommendations.

The experiment was first run using the SugarCRM native threads and then repeated using the automatically generated clusters. In each case, the recommendations generated by the collaborative recommender were compared against the random case.

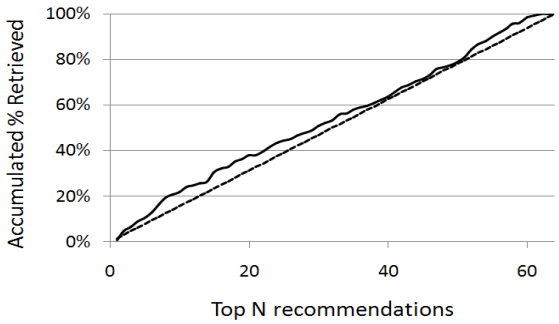
For the native SugarCRM feature requests, there were a small number of relatively large clusters, and a significant number of smaller clusters. In contrast, the automated clustering produced fewer and more uniform sized clusters. For this reason there were 156 recommendable items when native threads were used and only 25 for the automated clusters.

The results, showed that the collaborative filtering recommender outperformed the random predictor when applied to the automated clusters, shown in Figure 2b, but did not show any significant improvement when applied to the native SugarCRM threads, shown in Figure 2a. Our initial observations suggest that this difference in performance may be explained by the fact that popular themes such as calendar features or lead management were integrated into undesirable mega-threads in the native SugarCRM data but were clustered into more evenly grained topics in the automated clusters. The granularity of the automated clusters increased the opportunities for meaningful recommendations.

These results suggest that recommender technologies can be used in a meaningful way to help place stakeholders into forums during the requirements elicitation process, if forums are built around cohesive clusters of needs.



a. User defined threads



b. Automatically generated clusters

Figure 2. Collaborative filtering results for leave-one-out cross validation analysis.

4. Conclusions and future work

In conclusion, this paper has proposed a new framework for managing distributed and large scale requirements processes. Although stakeholders are initially placed into forums based on the needs they have contributed, this paper explored the use of collaborative recommendations for placing stakeholders into additional forums that might be of relevance to them. The results indicated that the collaborative recommender performed better than the random case; however additional work is needed to compare these results to those obtainable using different types of recommender systems.

Acknowledgments

The work described in this paper was partially funded by NSF grants CCR-0306303, CCR-0447594, and IIS-0430303.

References

[1] Basu, C., Hirsh, H., & Cohen, W. "Recommendation as Classification: Using Social and Content-Based Information in Recommendation" *National Conference on Artificial Intelligence*, Madison, WI, pp. 714-720.

[2] Can, F. and Ozkarahan, E. A. 1990. Concepts and effectiveness of the cover-coefficient-based clustering methodology for text databases. *ACM Trans. Database Syst.* Vol. 15, No. 4, Dec. 1990, pp. 483-517.

[3] Cleland-Huang, J., and Mobasher, B., "Using Data Mining and Recommender Systems to Scale up the Requirements Process", *Ultra-Large-Scale Software-Intensive Systems*, ICSE, Leipzig, Germany, May 2008.

[4] Davis, A., Dieste, O., Hickey, A., Juristo, N., & Moreno, A. "Effectiveness of Requirements Elicitation Techniques", *IEEE Intn'l Requirements Engineering Conf.*, Minneapolis, MN, Sept. 2006, pp. 179-188.

[5] Duan, C., *Clustering and its Application in Requirements Engineering*, Technical Report #08-001 , School of Comp., DePaul University, February, 2008.

[6] Frakes W.B, and Baeza-Yates, R, *Info retrieval: Data structures and Algorithms*, Englewood Cliffs, NJ: Prentice-Hall , 1992

[7] Pazzani, M., & Billsus, D. "Content-Based Recommendation Systems". *In P. Brusilovsky, A. Kobsa, & W. Nejdl, The Adaptive Web: Methods and Strategies of Web Personalization*. 2007, Berlin Heidelberg NewYork: Springer-Verlag.

[8] Schafer, J. B., Frankowski, D., & Shilad, S., "Collaborative Filtering Recommender Systems" *In P. Brusilovsky, A. Kobsa, & W. Nejdl, The Adaptive Web: Methods and Strategies of Web Personalization*. New York: Springer-Verlag. 2007.

[9] Strehl, A. and Ghosh, J. 2003. Cluster ensembles – a knowledge reuse framework for multiple partitions. *J. Mach. Learn.* Res. 3 (Mar. 2003), 583-617.

[10] Zhao, Y. and Karypis, G. 2001. Criterion functions for document clustering: Experiments and analysis. Tech.l Report TR #01--40, Dept. of Comp. Science, Univ. of Minnesota.