Hierarchically Structured Recommender System for Improving NPS of a Company

Jieyan Kuang[1], Albert Daniel[1], Jill Johnston[1], and Zbigniew W. Raś[1,2]

[1] Univ. of North Carolina, College of Computing and Informatics, KDD Laboratory, Charlotte, NC, 28223, USA
[2] Warsaw Univ. of Technology, Inst. of Computer Science, 00-665 Warsaw, Poland
{jkuang1,ras}@uncc.edu

Abstract. The paper presents a description of a hierarchically structured recommender system for improving the efficiency of a company's growth engine. Our dataset (NPS dataset) contains answers to a set of queries (called questionnaire) sent to a randomly chosen groups of customers. It covers 34 companies called clients. The purpose of the questionnaire is to check customer satisfaction in using services of these companies which have repair shops all involved in a similar type of business (fixing heavy equipment). These shops are located in 29 states in the US and Canada. Some of the companies have their shops located in more than one state. They can compete with each other only if they target the same group of customers. The performance of a company is evaluated using the Net Promoter System (NPS). For that purpose, the data from the completed questionnaires are stored in NPS datasets. We have 34 such datasets, one for each company. Knowledge extracted from them, especially action rules and their triggers, can be used to build recommender systems giving hints to companies how to improve their NPS ratings. Larger the datasets, our believe in the knowledge extracted from them is higher. We introduce the concept of semantic similarity between companies. More semantically similar the companies are, the knowledge extracted from their joined NPS datasets has higher accuracy and coverage. Our hierarchically structured recommender system is a collection of recommender systems organized as a tree. Lower the nodes in the tree, more specialized the recommender systems are and the same the classifiers and action rules used to build their recommendation engines have higher precision and accuracy.

1 Introduction

Net Promoter Score (NPS) is used to measure a customer's loyalty to a product or service provider [11], [12]. It is based on the response to a 1 to 10 scale with 0 being very unlikely to recommend the provider and 10 being very likely to recommend. Net Promoter System is based on the fundamental assumption that customers can be divided into three categories: *promoters*, *passives*, and *detractors*. Promoters are loyal enthusiasts who are buying from a company and urge their friends to do the same. Passives are satisfied but unenthusiastic customers who can be easily taken by the competition. Detractors are less than loyal customers who may urge their friends to avoid that company [10]. Customers are categorized based on their answers to the likelihood to recommend question. The figure below explains how these three categories are computed.

C. Cornelis et al. (eds.): RSCTC 2014, LNAI 8536, pp. 347–357, 2014.

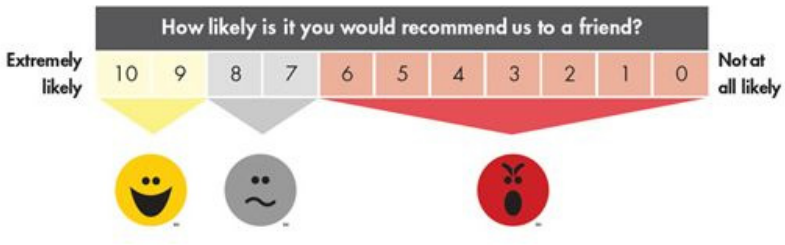

Fig. 1. Net Promoter Score (NPS)

Customers falling into interval 9-10 are seen as promoters, into 7-8 as passives, and into 0-6 as detractors. The partition into these three categories is widely accepted by business organizations but still other discretizations of NPS can be taken into consideration especially when classifiers extracted from NPS datasets do not have acceptable precission/recall for one of these three categories. The classical way to evaluate the efficiency of a company's growth engine is to compute NPS efficiency rating which is defined as the percentage of customers who are promoters minus the percentage who are detractors. Companies with the most efficient growth engines such as Amazon, Costco, Vanguard, or Dell have NPS efficiency ratings between 50 to 80 percent. But even these companies still have room for improvement.

In order to formulate actions for improving the performance of a company, we need to know why a customer is or is not likely to recommend the company to their colleagues/friends. This is why customers are asked to complete a questionnaire which gives us personal information about them and about their satisfaction with the services provided by a company. Examples of questions included in the questionnaire are give below:

- name of the customer
- name of the organization (client)
- invoice amount
- internal contact name (person with whom you deal in a company)
- how many days was needed to repair the equipment
- when the equipment was delivered to repair shop
- any disagreements?
- name/type of the equipment to be repaired
- are you satisfied with the job

More questionnaires are completed by customers, larger dataset for mining is available. Classifiers extracted from this dataset, defining each of the NPS categories, are evaluated using confusion matrix. If their accuracy and coverage is high then the action rules built from them will have high confidence as well. Different types of classifiers have been tested with a goal to identify which one has the highest accuracy/coverage.

Our dataset (NPS dataset) contains answers to a set of queries (called questionnaire) sent to a randomly chosen groups of customers. The purpose of the questionnaire is to check customer satisfaction in using services of 34 repair shops (clients) involved

in a similar type of business (fixing heavy equipment). These shops are located in 34 states in the US and Canada and they can compete with each other only if they are geographically closely located. For each shop, we extracted a number of classifiers from its NPS dataset. We also extended the original NPS dataset by developing and adding new groups of attributes, including temporal attributes. The accuracy/coverage of several classifiers for the category promoter become very high for all 34 shops. However, the accuracy and coverage of classifiers for the categories passive and detractor still remains low. So, action rules can not be built from pairs of classification rules [5] unless we are interested to build recommender system for each shop (or groups of shops) which only will target customers who completed the questionnaire. To apply action rules successfully for other customers, we have to extract these rules either directly from the dataset or built them from action reducts [2],[5].

The concept of an action rule was proposed by Ras and Wieczorkowska in [7] and investigated further in [1], [3], [9], [14]. Action rules describe possible transitions of objects from one state to another with respect to a distinguished attribute called the decision [7]. In our application domain, we are only interested in transitions from detractors and passives to promoters. We assume that attributes used to describe customers are partitioned into stable and flexible. Values of flexible attributes can be changed. In our domain, invoice amount or name/type of equipment to be repaired are examples of stable attributes. "Client name", "how many days are needed by the shop to fix the equipment" are examples of flexible attributes. "Client name" is a flexible attribute because customers may decide to change shops for fixing their equipment. In early papers, action rules have been constructed from two classification rules $[(\omega \wedge \alpha) \rightarrow \phi]$ and $[(\omega \wedge \beta) \rightarrow \psi]$, where ω is a stable part for both rules. Action rule was defined as the term $[(\omega) \wedge (\alpha \rightarrow \beta)] \Rightarrow (\phi \rightarrow \psi)$, where ω is the description of clients for whom the rule can be applied, $(\alpha \rightarrow \beta)$ shows what changes in values of flexible attributes are required, and $(\phi \rightarrow \psi)$ gives the expected effect of the action. Let us assume that ϕ means *detractors* and ψ means *promoters*. Then, the discovered knowledge shows how values of flexible attributes need to be changed so the customers classified as detractors will become promoters.

2 Action Rules

In this section we recall the definition of an information system (also called a dataset), action set, and also recall the classical strategy of constructing action rules from action sets.

By an information system [4] we mean a triple $S = (X, A, V)$, where:

1. X is a nonempty, finite set of objects
2. A is a nonempty, finite set of attributes, i.e.
   $a : U \longrightarrow V_a$ is a function (can be partial function) for any $a \in A$, where V_a is called the domain of a
3. $V = \bigcup\{V_a : a \in A\}$.

For example, Table 1 shows an information system S with a set of objects $X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$, set of attributes $A = \{a, b, c, d\}$, and the set of their values $V = \{a_1, a_2, b_1, b_2, c_1, c_2, d_1, d_2\}$.

Table 1. Information System S

	a	b	c	d
x_1	a_1	b_1	c_1	d_1
x_2	a_2	b_1	c_1	d_1
x_3	a_2	b_2	c_1	d_2
x_4	a_2	b_2	c_2	d_2
x_5	a_2	b_1	c_1	d_1
x_6	a_2	b_2	c_1	d_2
x_7	a_2	b_1	c_2	d_2
x_8	a_1	b_2	c_2	d_1

Additionally, we assume that $A = A_{St} \cup A_{Fl}$, where attributes in A_{St} are called *stable* and attributes in A_{Fl} are called *flexible*. "Customer name" is an example of a stable attribute. "Interest rate" for each customer account is an example of a flexible attribute.

Let $S = (X, A, V)$ is an information system, where $V = \bigcup\{V_a : a \in A\}$.

By an *atomic action set* we mean a singleton set containing an expression $(a, a_1 \rightarrow a_2)$ called atomic action, where a is an attribute and $a_1, a_2 \in V_a$. If $a_1 = a_2$, then a is called stable on a_1. Instead of $(a, a_1 \rightarrow a_1)$, we usually write (a, a_1) for any $a_1 \in V_a$.

By *Action Sets* we mean a smallest collection of sets such that:

1. If t is an atomic action set, then t is an action set.
2. If t_1, t_2 are action sets, then $t_1 \cup t_2$ is a candidate action set.
3. If t is a candidate action set and for any two atomic actions $(a, a_1 \rightarrow a_2)$, $(b, b_1 \rightarrow b_2)$ contained in t we have $a \neq b$, then t is an action set.

By the domain of an action set t, denoted by $Dom(t)$, we mean the set of all attribute names listed in t.

By an *action rule* we mean any expression $r = [t_1 \Rightarrow t_2]$, where t_1 and t_2 are action sets. Additionally, we assume that $Dom(t_2) \cup Dom(t_1) \subseteq A$ and $Dom(t_2) \cap Dom(t_1) = \emptyset$. The domain of action rule r is defined as $Dom(t_1) \cup Dom(t_2)$.

Now, we give an example of an action rule assuming that our information system S is represented by Table 1, a, c are stable and b, d are flexible attributes. Expressions (a, a_2), $(b, b_1 \rightarrow b_2)$, (c, c_2), $(d, d_1 \rightarrow d_2)$ are examples of atomic actions. Expression $(b, b_1 \rightarrow b_2)$ means that the value of attribute b is changed from b_1 to b_2. Expression (c, c_2) means that the value c_2 of attribute c remains unchanged. Expression r = $[\{(a, a_2), (b, b_1 \rightarrow b_2)\} \Rightarrow \{(d, d_1 \rightarrow d_2)\}]$ is an example of an action rule. The rule says that if value a_2 remains unchanged and value b will change from b_1 to b_2, then it is expected that the value d will change its value from d_1 to d_2.

3 Extracting Classifiers and Action Rules from NPS Datasets

In this section, we present a brief discussion on some classifiers built from NPS datasets representing answers to the customer satisfaction questionnaire completed by about 50,000 customers

The first result concerns classifiers built from the union of all NPS datasets which contains the answers collected from the customers for all the clients. We tested the classifiers available in WEKA by using the average confusion matrix for 10 random samplings, extracted from the union of all NPS datasets, each one covering about 1200 customers. J48 gave the best results which are presented in Table 2.

Table 2. Confusion Matrix for J48 and the original NPS dataset covering all clients

	$Promoter$	$Passive$	$Detractor$
$Promoter$	87	407	0
$Passive$	77	422	1
$Detractor$	29	170	0

We can see that Table 2 presents confusion matrix for the dataset with 494 promoters, 500 passives, and 199 detractors. So the average, total number of customers is 1193. NPS score is 494/1193 - 199/1193 = 0.41 - 0.17 = 0.24 (24 percent). The large number of $Passive$ customers (almost 1/2) is the main reason of this very low NPS score. To improve that score, action rules need to be extracted from NPS dataset. The first step to achieve our goal is to improve the classifiers extracted from that dataset. A number of new attributes have been constructed using text mining methods and added to the original NPS dataset. These attributes include: *Overall satisfaction*, *Likelihood to be a repeated customer*, *Technician arrived when promised*, and *Repair completed correctly*. The new average confusion matrix obtained from 10 random samplings using J48 classifier for the extended NPS dataset is shown in Table 3. Obviously the NPS score did not change because the decision column is the same.

Table 3. Confusion Matrix for J48 and NPS dataset with new features

	$Promoter$	$Passive$	$Detractor$
$Promoter$	407	80	7
$Passive$	123	327	50
$Detractor$	23	77	99

Much better results we received for the extended NPS datasets covering certain combinations of clients, especially single clients. Table 4 shows confusion matrix for the Tree Classifier using Rough Set Exploration System ($RSES$) for a dataset which represents two clients.

Assume now that we use $RSES$ Tree Classifier to construct action rules by pairing classification rules describing $Detractor$ with classification rules describing $Promoter$. The goal is to reclassify as many $Detractors$ as possible to $Promoters$. The average confidence of action rules will be $0.993 \cdot 0.849 = 0.84$ (see the Accuracy column in Figure 2). Our action rules can target only 4.2 (out of 10.2) detractors. So, we can expect $4.2 \cdot 0.84 = 3.52$ detractors moving to the promoter status. The NPS score for the initial dataset covering two clients was 0.80. After applying our action rules we get:

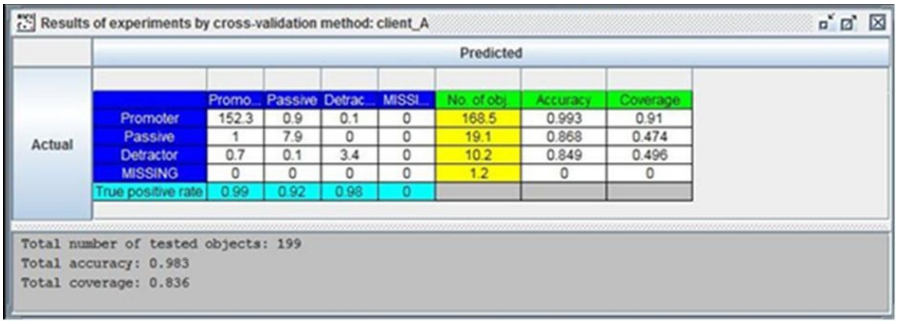

Fig. 2. Two-Clients Confusion Matrix for $RSES$ Tree Classifier

- Number of $Promoters$ = $168.5 + 3.52 = 172.2$
- Number of $Passives$ = 19.1
- Number of $Detractors$ = 10.2 - $3.52 = 6.68$

Total number of customers = 197.8. So, the new NPS = $[\frac{172.2}{197.8}$ - $\frac{6.7}{197.8}] = 0.87 - 0.03 = 0.84$, which means 4 percent of improvement in NPS is expected.

4 Clustering of Clients Based on Semantic Similarity in NPS Datasets

In this section, we introduce the notion of semantic similarity between clients. Informally speaking, we say that two clients are semantically similar if they agree on the knowledge concerning $Promoter$, $Passive$, and $Detractor$ which is hidden in their NPS datasets. Stronger is the agreement, semantically more similar they are. NPS Datasets of two or more clients semantically similar can be joined together giving us larger NPS datasets for mining. Larger the datasets, our own confidence in the results shown in confusion matrices is higher.

Assume now that $RC[1]$, $RC[2]$ are the sets of classification rules extracted from the questionnaire-type datasets (NPS datasets) collected for clients $C1$, $C2$. Also, we assume that

$RC[1] = RC[1, Promoter] \cup RC[1, Passive] \cup RC[1, Detractor]$,
where $RC[1, Promoter] = \{r[1, Promoter, i] : i \in I_{Pr}\}$, $RC[1, Passive] = \{r[1, Passive, i] : i \in I_{Ps}\}$, $RC[1, Detractor] = \{r[1, Detractor, i] : i \in I_{Dr}\}$, where $\{r[1, Promoter, i] : i \in I_{Pr}\}$ is a collection of classification rules defining "Promoter", $\{r[1, Passive, i] : i \in I_{Ps}\}$ is a collection of classification rules defining "Passive", and $\{r[1, Detractor, i] : i \in I_{Dr}\}$ is a collection of classification rules defining "Detractor".

In a similar way, we define
$RC[2] = RC[2, Promoter] \cup RC[2, Passive] \cup RC[2, Detractor]$,
where $RC[2, Promoter] = \{r[2, Promoter, i] : i \in J_{Pr}\}$, $RC[2, Passive] = \{r[2, Passive, i] : i \in J_{Ps}\}$, $RC[2, Detractor] = \{r[2, Detractor, i] : i \in J_{Dr}\}$.

By $C1[1, Promoter, i], C1[1, Passive, i], C1[1, Detractor, i]$ we mean confidence of $r[1, Promoter, i]$, $r[1, Passive, i]$, and $r[1, Detractor, i]$ in a dataset for client $C1$, respectively.

By $C2[1, Promoter, i], C2[1, Passive, i], C2[1, Detractor, i]$ we mean confidence of $r[1, Promoter, i]$, $r[1, Passive, i]$, and $r[1, Detractor, i]$ in a dataset for client $C2$, respectively.

By $C2[2, Promoter, i], C2[2, Passive, i], C2[2, Detractor, i]$ we mean confidence of $r[2, Promoter, i]$, $r[2, Passive, i]$, and $r[2, Detractor, i]$ in a dataset for client $C2$, respectively.

By $C1[2, Promoter, i], C1[2, Passive, i], C1[2, Detractor, i]$ we mean confidence of $r[2, Promoter, i]$, $r[2, Passive, i]$, and $r[2, Detractor, i]$ in a dataset for client $C1$, respectively.

Now, we can introduce the concept of semantic similarity between clients $C1$, $C2$ denoted by $SemSim(C1, C2)$.

$SemSim(C1, C2)$ =

$$\frac{\Sigma\{|C1[1,Promoter,k]-C2[1,Promoter,k]|:k\in I_{Pr}\}}{card(I_{Pr})} + \frac{\Sigma\{|C1[1,Passive,k]-C2[1,Passive,k]|:k\in I_{Ps}\}}{card(I_{Ps})} + \frac{\Sigma\{|C1[1,Detractor,k]-C2[1,Detractor,k]|:k\in I_{Dr}\}}{card(I_{Dr})} + \frac{\Sigma\{|C2[2,Promoter,k]-C1[2,Promoter,k]|:k\in J_{Pr}\}}{card(J_{Pr})} + \frac{\Sigma\{|C2[2,Passive,k]-C1[2,Passive,k]|:k\in J_{Ps}\}}{card(J_{Ps})} + \frac{\Sigma\{|C2[2,Detractor,k]-C1[2,Detractor,k]|:k\in J_{Dr}\}}{card(J_{Dr})}.$$

Figure 3 shows the hierarchical clustering of 34 clients with respect to their semantic similarity. Clients which are semantically and geographically close to each other can have their datasets merged and the same considered as a single client from the business perspective (customers have similar opinion about them). For each state (its abbreviation is given), we list clients operating in that state with their respective NPS values:

AB- $\{(clint - 9, NPS = 0.503)\}$, AZ- $\{(client - 8, NPS = 0.802)\}$, CA- $\{(client - 13, NPS = 0.777), (client - 16, NPS = 0.767), (client - 17, NPS = 0.848), (client - 24, NPS = 0.724)\}$, GA- $\{(client - 34, NPS = 0.779)\}$, ID- $\{(client - 30, NPS = 725)\}$, IL- $\{(client - 1, NPS = 0.836)\}$, KS- $\{(client - 7, NPS = 0.771)\}$, KY- $\{(client - 31, NPS = 0.804)\}$, LA- $\{(client - 19, NPS = 0.705)\}$, MN- $\{(client - 3, NPS = 0.823)\}$, MO- $\{(client - 7, NPS = 0.771), (client - 10, NPS = 0.788)\}$, MS- $\{(client - 23, NPS = 0.860), (client - 26, NPS = 0.828)\}$, NC- $\{(client - 4, NPS = 0.803), (client - 11, NPS = 0.797), (client - 12, NPS = 0.722), (client - 27, NPS = 0.760), (client - 32, NPS = 0.740)\}$, ND- $\{(client - 3, NPS = 0.823)\}$, NE- $\{(client - 21, NPS = 0.732)\}$, NV- $\{(client - 5, NPS = 0.771)\}$, OH- $\{(client - 22, NPS = 0.820), (client - 28, NPS = 0.779)\}$, OK- $\{(client - 29, NPS = 0.710)\}$, PA- $\{(client - 6, NPS = 0.788), (client - 25, NPS = 0.800)\}$, QC- $\{(client - 14, NPS = 0.721)\}$, SC- $\{(client - 2, NPS = 765), (client - 11, NPS = 0.797)\}$, SD- $\{(client - 2, NPS = 0.765), (client - 3, NPS = 0.823)\}$, SK- $\{(client - 18, NPS = 0.636)\}$, TN- $\{(client - 26, NPS = 0.828)\}$, TX- $\{(client - 15, NPS =$

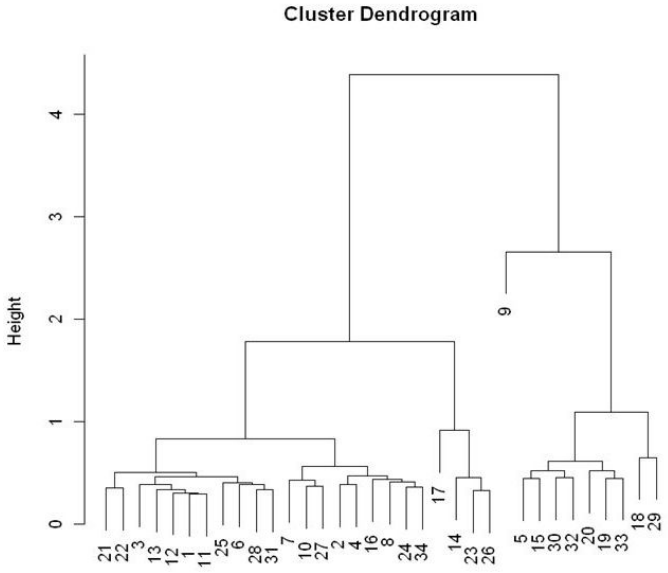

Fig. 3. Hierarchical clustering of 34 clients

$0.762), (client - 20, NPS = 0.675), (client - 29, NPS = 0.710)\}$, UT- $\{client - 5, NPS = 0.771)\}$, WV- $\{(client - 28, NPS = 0.779)\}$; WY- $\{(client - 30, NPS = 0.725), (client - 33, NPS = 0.772)\}$, VA- $\{(client - 11, NPS = 0.800)\}$.

From the Figure 2, we can observe that $\{23, 26\}$, $\{24, 34\}$, $\{1, 11\}$, $\{21, 22\}$, $\{28, 31\}$, $\{19, 33\}$ are examples of six clusters of semantically similar clients. Clients in the cluster $\{23, 26\}$ are both located in Mississippi so they are geographically close as well. Their NPS ratings are 0.860 and 0.828, respectively. Since they both target the same group of customers, Client 26 can improve its ratings by using recommendations based on action rules extracted from the NPS dataset covering both clients. Clients in the cluster $\{24, 34\}$ are far away from each other. One is in California and the other in Georgia. NPS rating for Client 24 is 0.724 and for Client 34 is equal to 0.80. They do not compete for the same group of customers, so the strategy for improvement of the NPS rating for Client 24 is more challenging. In this particular case, Client 24 has the worst NPS ratings among clients in California (Client 13, 16, 17). Client 16 is semantically the closest one to Client 24 and also its NPS rating is the closest to the NPS rating of Client 24. So, we have two options. We can merge NPS datasets of clients 16 and 24, and next extract action rules from the joined dataset to get improvement of the NPS score for Client 24 or we can merge the NPS datasets of Clients 24 and 34, and next extract action rules from the joined dataset. Following the first option we are targeting the same group of customers whereas in the second, the customers are from two different states and geographically far away from each other. It is quite possible that customers from two different states evaluating their local clients may follow different

criteria when they answer the questionnaire. The optimal solution to solve this problem is to test both options and chose the one which is giving us better improvement in NPS score.

5 NPS-Based Recommender System

In this section, we present the methodology which can be followed to build a hierarchically structured recommender system with nodes being recommender systems of different generalization levels. Leaves of the tree represent personalized recommender systems built from classifiers and action rules extracted from NPS dataset of a single client (single company). The root of the tree represents the most generalized recommender system which is built from classifiers extracted from the union of all NPS datasets covering all 34 clients. The internal nodes of the tree are built following the dendrogram presented in Figure 3. Lower the nodes in the tree, more specialized the recommender systems are. Also, lower the nodes in the tree, the accuracy and precision of the classifiers assigned to them is higher.

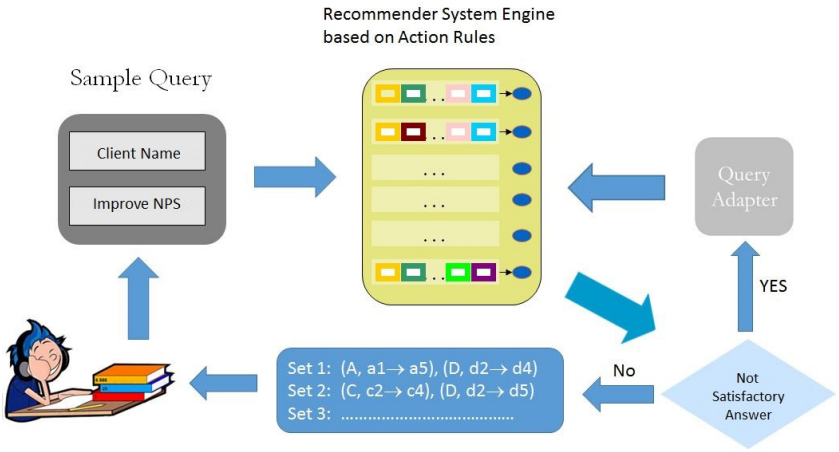

Fig. 4. Recommender System

Figure 4 presents Flexible Query Answering System (FQAS) built on the top of a hierarchically structured recommender system. Client (company) which is not satisfied with its current NPS ratings can submit a query to FQAS asking what can be done to improve its NPS. The easiest answer can be obtained from the recommender system constructed from the NPS dataset of that client but to get better recommendation we have to use recommender systems which are not personalized. Namely, we should consider getting help from recommender systems assigned to the nodes forming the path which starts from the leaf (our client) in the dendrogram and leads to the root. So, the process is bottom-up. For instance, if the Client 24 wants to improve its ratings, action rules from its NPS dataset are extracted. If the hints from the recommender system

based on these rules are not satisfactory, Client 24 is clustered with Client 34 (see the dendrogram) since these two clients are semantically the most similar. If the NPS rating of Client 34 is higher than the rating of Client 24, action rules are extracted from the union of the corresponding two NPS datasets and hints based on them are given to the client. Otherwise, we should check all leaves in the dendrogram leading us to the parent of Client 24 and 34. We should chose that leaf which represents a client with NPS rating higher than the rating of Client 24 and also which is semantically the closest one to Client 24.

6 Conclusion

The paper presents preliminary results which finally will lead us to the construction of a flexible hierarchically structured recommender system for improving NPS of a company in a global competitive market. Thirty four companies (clients) form the domain for the agglomerative clustering algorithm based on their semantic distance. Clients are compared in terms of the similarity of their knowledge concerning the meaning of three concepts: promoter, passive, and detractor. The resulting dendrogram is a skeleton for the collection of hierarchically structured recommender systems. Lower the nodes in the dendrogram, more specialized the recommender systems are. The recommendations are based on action rules which are extracted from the datasets assigned to all nodes of the dendrogram. Higher a node in the dendrogram, the dataset assigned to it is larger - it is built by taking the union of all datasets assigned to the ancestors of that node. The questionnaire sent to the customers allows them to enter statements in the text format explaining their ratings. Information included in these statements will help us to find triggers for action rules. The triggers are also called meta-actions [13], [6].

References

1. He, Z., Xu, X., Deng, S., Ma, R.: Mining action rules from scratch. Expert Systems with Applications 29(3), 691–699 (2005)
2. Im, S., Ras, Z., Tsay, L.-S.: Action reducts. In: Kryszkiewicz, M., Rybinski, H., Skowron, A., Raś, Z.W. (eds.) ISMIS 2011. LNCS (LNAI), vol. 6804, pp. 62–69. Springer, Heidelberg (2011)
3. Paul, R., Hoque, A.S.: Mining irregular association rules based on action and non-action type data. In: Proceedings of the Fifth International Conference on Digital Information Management (ICDIM), pp. 63–68 (2010)
4. Pawlak, Z.: Information systems - theoretical foundations. Information Systems Journal 6, 205–218 (1981)
5. Ras, Z., Dardzinska, A.: From Data to Classification Rules and Actions. In: Proceedings of the Joint Rough Sets Symposium (JRS 2007). LNCS (LNAI), vol. 4482, pp. 322–329. Springer (2011)
6. Raś, Z.W., Dardzińska, A.: Action Rules Discovery based on Tree Classifiers and Meta-Actions. In: Rauch, J., Raś, Z.W., Berka, P., Elomaa, T. (eds.) ISMIS 2009. LNCS (LNAI), vol. 5722, pp. 66–75. Springer, Heidelberg (2009)
7. Raś, Z.W., Wieczorkowska, A.A.: Action-rules: how to increase profit of a company. In: Zighed, D.A., Komorowski, J., Żytkow, J.M. (eds.) PKDD 2000. LNCS (LNAI), vol. 1910, pp. 587–592. Springer, Heidelberg (2000)

8. Raś, Z.W., Wyrzykowska, E.z., Wasyluk, H.: ARAS: Action rules discovery based on agglomerative strategy. In: Raś, Z.W., Tsumoto, S., Zighed, D.A. (eds.) MCD 2007. LNCS (LNAI), vol. 4944, pp. 196–208. Springer, Heidelberg (2008)
9. Rauch, J., Šimůnek, M.: Action rules and the GUHA method: Preliminary considerations and results. In: Rauch, J., Raś, Z.W., Berka, P., Elomaa, T. (eds.) ISMIS 2009. LNCS (LNAI), vol. 5722, pp. 76–87. Springer, Heidelberg (2009)
10. Reichheld, F.F.: The one number you need to grow. Harvard Business Review, 1–8 (December 2003)
11. SATMETRIX, NET Promoter (2014), http://www.satmetrix.com/net-promoter/
12. SATMETRIX, Improving your net promoter scores through strategic account management, white paper (2012)
13. Tzacheva, A., Ras, Z.W.: Association action rules and action paths triggered by meta-actions. In: Proceedings of 2010 IEEE Conference on Granular Computing, Silicon Valley, CA, pp. 772–776. IEEE Computer Society (2010)
14. Qiao, Y., Zhong, K., Wang, H.-A., Li, X.: Developing event-condition-action rules in real time active database. In: Proceedings of the 2007 ACM Symposium on Applied Computing, Seoul, pp. 511–516 (2007)