

Hierarchical Agglomerative Method for Improving NPS

Jieyan Kuang¹, Zbigniew W. Raś^{1,2(✉)}, and Albert Daniel¹

¹ KDD Laboratory, College of Computing and Informatics,
University of North Carolina, Charlotte, NC 28223, USA
{jkuang1, ras}@uncc.edu

² Institute of Computer Science, Warsaw University of Technology,
00-665 Warsaw, Poland

Abstract. The paper proposes a new strategy called HAMIS to improve NPS (Net Promoter Score) of certain companies involved in heavy equipment repair in the US and Canada - we call them clients. HAMIS is based on the semantic dendrogram built by using agglomerative clustering strategy and semantic distance between clients. More similar is the knowledge extracted from two clients, more close these clients semantically are to each other. Each company is represented by a dataset which is built from answers to the questionnaire sent to a number of randomly chosen customers using services offered by this company. Before knowledge is extracted from these datasets, each one is extended by merging it with datasets which are close to it in the semantic dendrogram, have higher NPS, and if classifiers extracted from them have higher FS-score. Action rules are extracted from these extended datasets and used for providing recommendations to clients of how to improve their businesses.

1 Introduction

Improving companies' performance is an important issue nowadays and Net Promoter Score (NPS) is one of the most popular measure for such purpose [7–9]. Net Promoter Score assumes that customers are categorized into three categories: promoter, passive and detractor, which represent customers' satisfaction, loyalty and the likelihood of recommending this client in a descending order.

Our dataset involves 34 clients who are located in different areas crossing the whole United States as well as some parts of Canada. These clients provide similar services to over 25,000 customers. The dataset consists of three categories of values which are collected from the questionnaire answered by randomly selected customers during 2011 and 2012. The first and second category in the questionnaire provide information about customers and services they received and the third category (the key part of the questionnaire) relates to the customers feelings about the services. Here are some examples of questions in these three categories:

- Information about the customer (name, contact phone number).
- Information about the service (name of the client, invoice amount, type of equipment to be repaired).
- Feeling about the service (how many days were needed to finish the job, was the job completed correctly, are you satisfied with the job, likelihood to refer to friends).

Customers are asked to share their feelings about the service by scoring 0 to 10 for all the asked questions in third category. Higher the score is, more pleased the customer is with the service. Based on the average score of all the collected answers for each customer, over 99% of customers are divided into three groups: customers falling into interval 9–10 are seen as promoter, into 7–8 as passive, and into 0–6 as detractor. With the determined NPS status in our dataset, the NPS efficiency rating (defined as the percentage of customers labeled promoter minus the percentage of customers labeled detractor) can be computed for each client.

Our ultimate goal is to improve the service of every client, in another word, improve its NPS. The semantic distance (similarity) between clients, which indicates the similarity of clients' knowledge concerning *Promoter*, *Passive* and *Detractor* hidden in datasets, can be computed. Smaller the semantic distance is, more similar the clients are. Using the notion of semantic distance, we build semantic similarity based dendrogram by following agglomerative clustering algorithm in the domain of datasets representing 34 clients. Next, we propose a method called Hierarchical Agglomerative Method for Improving NPS (HAMIS). Besides semantic similarity in HAMIS, NPS efficiency rating is another primary measure we consider before we merge two semantically similar clients. As a matter of fact, the NPS rating of the newly merged dataset will be higher than or at least equal to the dataset which is used for merging and it is with lower NPS rating than the other dataset. So we can expect that by analyzing the merged dataset of two most semantically similar clients we should be able to offer recommendations to the client with lower NPS rating. However, the consistency of data may decrease in the merged dataset so we also evaluate its representing classifier and if the results are satisfactory we merge two datasets in HAMIS.

Action rules mining is a known strategy in the area of data mining and it was firstly proposed by Ras and Wieczorkowska in [6] and investigated further in [2–5, 10]. 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 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 attributes need to be changed under the situation required by stable part of the rule so the customers classified as detractors will become promoters.

2 Introduction of Semantic Similarity

The concept of semantic similarity between clients was introduced in [1]. Each client was represented by a tree classifier extracted from its extended dataset. More similar are the tree classifiers representing clients, more close they are semantically. Figure 1 shows the hierarchical clustering of 34 clients with respect to their semantic similarity. In the dendrogram we can easily identify groups of clients which are semantically close to each other. If we use tree structure based terminology, every leaf node represents a client as the numbers show. The depth of a node is the length of the path from it to the root. So larger is the depth of the earliest common ancestor of two clients, more semantically similar they are to each other.

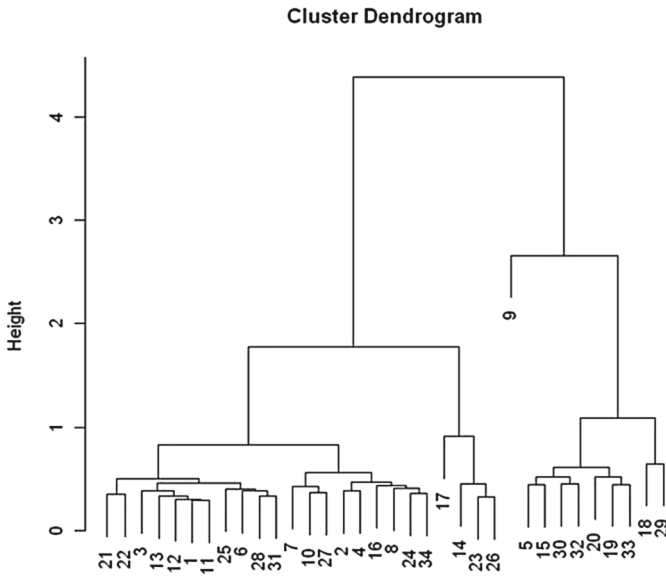


Fig. 1. Hierarchical clustering of 34 clients

3 Hierarchical Agglomerative Method for Improving NPS

HAMIS is enlarging the dataset of any specified client by following a bottom-up path in the hierarchically structured dendrogram based on semantic similarity. In the dendrogram, every leaf node represents a dataset of a corresponding client and every parent node represents the merged dataset of its mergeable children. Therefore, higher the bottom-up path ends in the dendrogram, larger the resulting merged dataset is, namely, more generalized dataset is returned by HAMIS.

The bottom-up path formed during the process links all the successfully merged nodes. Mergeable node means the node that can be used for merging and it is identified by the following criteria:

- It is the most semantically similar node in current situation.
- Its NPS rating is not less than the targeted client.

Given the definition of semantic similarity in the second section, we are capable to quantify the concept of how similar customers from different clients feel about the provided service. Therefore, if the semantic distance between two clients is relatively small, in other words, these clients are semantically close, then we could infer that the customers from these clients think of *Promoter*, *Passive* and *Detractor* in a more similar way, comparing to customers from clients that are further away regarding semantic distance. So it is possible that action rules extracted from the dataset covering all these semantically similar clients are also useful for improving the NPS rating of individual client. Based on the semantic distance retrieved, we clustered all the 34 clients using the agglomerative clustering algorithm and generated a dendrogram as shown in Fig. 1 which provides us with very efficient way to identify the most similar clients. As mentioned previously, each leaf node of the dendrogram stands for each client correspondingly, so the nodes that are semantically closest should be all the leaf nodes on the sibling side. For instance, if the sibling node is a leaf node, then there is only one node available for being the closest. If the sibling node is a parent node, it complicates the situation since the union set of all the leaf nodes under this sibling node should be the most semantically similar, then certainly, all the leaf nodes on the sibling side should be counted in and be checked one by one in a top down sequence following the depth of these nodes.

However, merging a targeted client with a semantically similar client whose NPS rating is lower won't fully match our expectation, since our goal is to improve the target's NPS rating, not conversely. But what we can be certain of is that merging a client with other client whose NPS rating is not lower gives us a dataset with higher or at least the same NPS rating. Let's assume that $NPS[i]$ and $NPS[j]$ are the NPS ratings of two clients i and j . Then

$$NPS[i] = \frac{Num[i, Promoter]}{Num[i, *]} - \frac{Num[i, Detractor]}{Num[i, *]}.$$

By $Num[i, Promoter]$ and $Num[i, Detractor]$ we mean number of *Promoter* and *Detractor* records in dataset of client i respectively, and by $Num[i, *]$ we mean the total number of records in i regardless of the class categories.

$$\text{Meanwhile, } NPS[j] = \frac{Num[j, Promoter]}{Num[j, *]} - \frac{Num[j, Detractor]}{Num[j, *]}.$$

By $Num[j, Promoter]$ and $Num[j, Detractor]$ we mean number of *Promoter* and *Detractor* records in dataset of client j respectively, and by $Num[j, *]$ we mean the total number of records in j regardless of the class categories.

Also we assume that $NPS[j] \geq NPS[i]$ and $NPS[i \cup j]$ is the NPS rating of the union set of client i and j , so we can expect $NPS[i \cup j] \geq NPS[i]$, because

$$\begin{aligned}
& \text{if } NPS[j] - NPS[i] = \\
& \left(\frac{Num[j, Promoter]}{Num[j, *]} - \frac{Num[j, Detractor]}{Num[j, *]} \right) - \left(\frac{Num[i, Promoter]}{Num[i, *]} - \frac{Num[i, Detractor]}{Num[i, *]} \right) \geq 0, \\
& \text{then } NPS[i \cup j] - NPS[i] = \\
& \left(\frac{Num[j, Promoter] + Num[i, Promoter]}{Num[j, *] + Num[i, *]} - \frac{Num[j, Detractor] + Num[i, Detractor]}{Num[j, *] + Num[i, *]} \right) - \\
& \left(\frac{Num[i, Promoter]}{Num[i, *]} - \frac{Num[i, Detractor]}{Num[i, *]} \right) \geq 0.
\end{aligned}$$

Thus, we can surely get a joined dataset with non-decreased NPS rating. In addition, continuously keeping track of the quality of classifiers extracted from the merged dataset during the entire procedure is advantaging for achieving the best performance of generalization. Classification results show the quality of datasets for mining action rules and worse classifiers lead to poor confidence of action rules. Accordingly, we must make sure the classifiers are under improvement. To evaluate the classifiers, we use F-score that includes both accuracy and coverage of classification into consideration. As a popular measure of assessing the classification performance, F-score offers us a comprehensive and accurate view on our data.

Therefore, the three criteria mentioned above make up the foundation of algorithm HAMIS which is presented thoroughly in the next section.

3.1 Presentation of HAMIS

Technically speaking, the purpose of the algorithm HAMIS is to keep expanding the targeted client by unionizing it with all the clients satisfying the conditions. Unless the resulting dataset for chosen client can't be expanded any further, the algorithm would be repeatedly executed. And the algorithm returns resulting dataset when it ends. As HAMIS is built on the basis of a dendrogram regarding semantic distance, we describe the procedure using tree structure related terminology. The algorithm is designed as presented in Algorithm 1.

In the procedure of HAMIS, the resulting node is defined as N and it is initialized with the input targeted node N_{target} . Once N has been given, the nodes that are semantically closest to it are retrieved and stored in a list naming N_c . Accordingly, N_c contains all the leaf nodes on the sibling side of current N in the dendrogram and they are the candidates for being mergeable with N . It is apparent that at least one candidate is required to proceed, otherwise, it means the node N has reached the root and there is no more node available for merging. When proceeding, the following part is the main part in HAMIS and it iterates through all the candidates in N_c on the foundation of other two merging criteria mentioned above: NPS rating and F-score. If a candidate $N_c[i]$ does not have lower NPS rating than the targeted node N_{target} , then the candidate is qualified for merging. And the merged result is temporarily stored as N_m . N_m can't become the new resulting node N yet unless its F-score is greater or at least equal to F-score of current N . Thus, if the resulting node N is replaced by the merged result N_m , it suggests the merging process for current candidate succeeds and the new N will be used for next candidate in N_c if there are still

Algorithm 1. Hierarchical Agglomerative Method for Improving NPS

```

procedure HAMIS( $N_{target}$ )
   $N \leftarrow N_{target}$   $\triangleright N$  is a node including only the targeted node initially
  repeat
     $N_0 \leftarrow N$ 
    retrieve  $N_c$  which is a list of candidates  $N_c[1], N_c[2], \dots, N_c[n]$ ;
    while  $N_c \neq \emptyset$  do
      get next available candidate  $N_c[i] \in N_c$  ( $i \in \{1, 2, \dots, n\}$ );
      if  $NPS[N_c[i]] \geq NPS[N_{target}]$  then
         $N_m \leftarrow \{N_c[i]\} \cup N$ 
        if  $Fs[N_m] \geq F[N]$  then
           $N \leftarrow N_m$   $\triangleright N$  is replaced by  $N_m$ 
        end if
      end if
       $N_c \leftarrow N_c \setminus \{N_c[i]\}$   $\triangleright$  remove it from the list
    end while
    if  $N_0 \neq N$  then
       $N$  climbs to its parent node in upper level;
    end if
  until  $N_0 = N$   $\triangleright$  We have the final dataset return if no more merging happens.
  return  $N$ 
end procedure

```

any. When a candidate fails merging with N , the same resulting node N will be used for another generalization attempt with next available candidate. The main part will not end until all the candidates have been checked. If there are more than one candidate found in N_c , they will be checked in a top down order based on the depth of them in the dendrogram, smaller the depth of a candidate is, earlier the candidate will be checked. So candidates are stored in an ascending order with regards to their depth in dendrogram, saying that for each candidate $N_c[i] (i \in \{1, 2, \dots, n\})$,

$depth[N_c[i+1]] > depth[N_c[i]]$, where $depth[N_c[i]]$ is the depth of node $N_c[i]$ in dendrogram and $i \in \{1, 2, \dots, n-1\}$.

Each candidate is examined in almost same way, while the only difference is a new resulting node iteratively generated by successful merging process. Every time a new node is merged in the resulting node, the newly updated resulting node is replacing the current one. When the main part of the algorithm is finished with the resulting node being updated, it will climb up one level in the dendrogram and become the parent node of previous position. With a new resulting node at a new depth, HAMIS will keep going until the resulting node is not changed after the main part ends or it has reached the root.

4 Experiment

To show the running process of algorithm HAMIS in our domain, we take Client 2 as a target example, and the relevant data used during this procedure are shown

in Table 1. As the semantic similarity based clustering dendrogram is given in Fig. 1 and the part of it related to our example is shown in Fig. 2, we observe that node $\{2\}$ representing Client 2 is labeled in green at the bottom and it is the initial node. As a sibling node to node $\{2\}$, node $\{4\}$ is the only candidate in N_c which is most semantically similar to node $\{2\}$, and NPS rating of node $\{4\}$ shown in Table 1 is higher. In addition, F-score calculated by J48 in WEKA for merged node $\{2, 4\}$ is also higher than current resulting node $\{2\}$, which are 0.788 comparing to 0.783, hence the merged node $\{2, 4\}$ successfully replaces $\{2\}$ and become the new resulting node. Meanwhile, there are no more unchecked candidate in N_c , so HAMIS is done with current depth and will continue with the new resulting node by climbing up to the parent node which is labeled in blue. At a new position, because the sibling node of current resulting node is not a leaf node, and leaf nodes $\{16\}$, $\{8\}$, $\{24\}$ and $\{34\}$ on the sibling side should be included in candidate set as we defined, and they are labeled in blue as well. According to the depth of each candidate in dendrogram, they will be checked following the top down sequence which is node $\{16\}$ first, then node $\{8\}$ and $\{24\}$, and $\{34\}$ at the end. Then HAMIS attempts to merge these candidates to resulting node individually, but it turns out none of them can successfully merge with $\{2, 4\}$. When it comes to node $\{16\}$, although its NPS rating is just a little bit higher than for the targeted node $\{2\}$, the F-score of $\{2, 4, 16\}$ is much lower than for $\{2, 4\}$, so the merging of $\{16\}$ and $\{2, 4\}$ fails and the main part goes to the next one, which is node $\{8\}$. The case for node 8 is exactly the same as for node $\{16\}$, so $\{2, 4\}$ is still the resulting node without being changed and it keeps going to node $\{24\}$ and $\{34\}$. But neither of them can be merged with $\{2, 4\}$ due to either low NPS ratings or lower F-score of joined nodes. Consequently, node $\{2, 4\}$ has not been replaced with any new merged node after all the candidates have been checked, which suggests $\{2, 4\}$ is the most generalized in our program for Client 2. Thus, HAMIS ends here and returns $\{2, 4\}$.

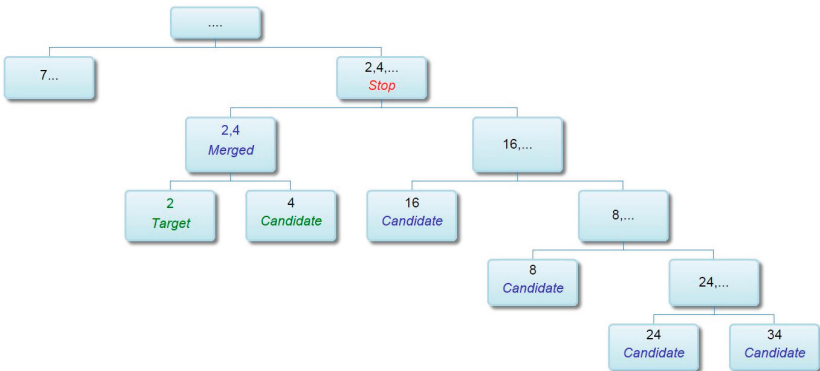


Fig. 2. Example of running HAMIS with Client 2 selected (Color figure online)

Table 1. NPS rating and Fscore of relevant nodes

	$N\{2\}$	$N\{4\}$	$N\{16\}$	$N\{8\}$	$N\{24\}$	$N\{34\}$
NPS rating	0.765	0.803	0.767	0.802	0.724	0.779
	$N\{2\}$	$N\{2, 4\}$	$N\{2, 4, 16\}$	$N\{2, 4, 8\}$	$N\{2, 4, 24\}$	$N\{2, 4, 34\}$
F-score	0.783	0.788	0.776	0.786	NA	0.778

In the next step, we are going to generate action rules for both generalized dataset and original dataset of Client 2. Before the program starts, we need to specify the necessary attributes. Certainly the promoter status should be the decision attributes and the transitions we are interested in are from *Detractor* to *Promoter*. The customers' personal information related attributes should be seen as stable attributes, in our experiment, attributes like customers' name, location and contact number are set as stable attributes. Then the attributes about customers' feeling and comment are selected as attributes which can change (they are flexible), and these are the keys for improving NPS ratings since they will tell us about what actions we should adopt. For example, attributes evaluating if the job is done correctly and the timeframe of technician's arrival are flexible attributes. Based on our personal knowledge about the dataset, we expect that a huge number of action rules will be generated and we only pay attention to the ones with sufficiently high confidence, so we intend to get the action rules with at least 80% confidence.

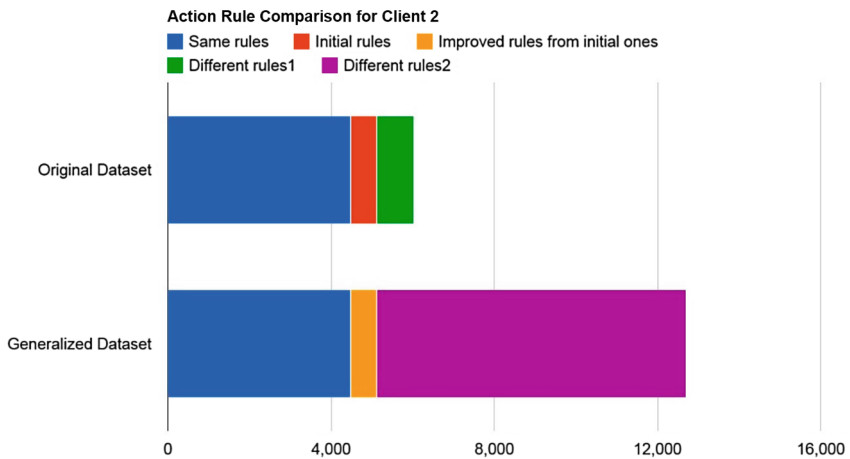

Fig. 3. Action rule comparison for Client 2 (Color figure online)

Figure 3 shows the results of comparing action rules extracted from dataset $\{2, 4\}$ to dataset $\{2\}$ alone. In the figure, blue bars display the number of exact same rules with same support and confidence found in both datasets, red bar

represents the rules extracted from dataset $\{2\}$ which are not found the same in dataset $\{2, 4\}$ but the action sets associated with these rules are contained in the action sets associated with rules extracted from $\{2, 4\}$ with higher confidence or support, which is marked using orange bar on the bottom. Last but not the least, green bar and pink bar show the unique rules in both action rule sets respectively that don't exist in the other action rule set. Firstly, we can easily see that there are twice as many as rules generated from the expanded dataset. More specifically, we found 12,715 action rules from the larger dataset while 6,026 from the original dataset. At the same time, nearly 75% of action rules from dataset of Client 2 can be found in the set of action rules from the more generalized dataset $\{2, 4\}$ with same support and confidence. And over 10% of action rules found in original dataset can be found in the set of action rules from generalized dataset with higher support or confidence. Furthermore, a lot of new action rules have been discovered and over 70% of the new action rules are with remarkably high confidence.

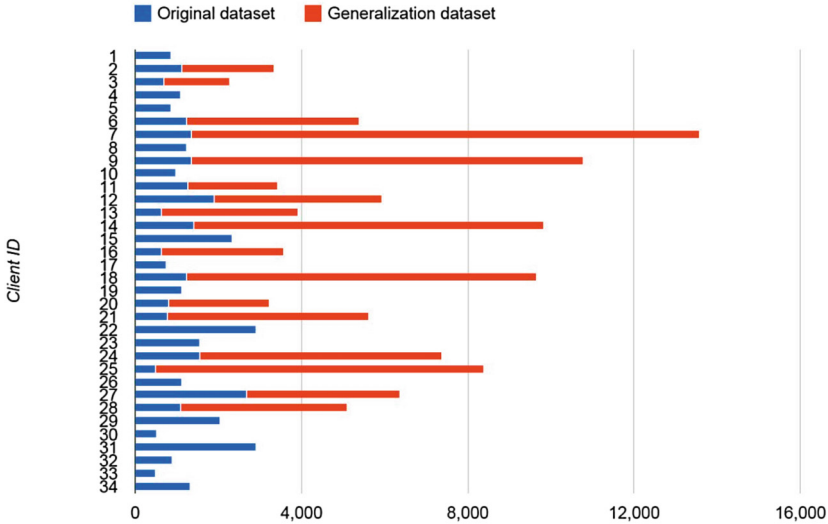


Fig. 4. Performance of HAMIS on 34 clients

In order to get more convincing results, we apply HAMIS to all 34 clients individually and retrieve the generalized datasets. From the results in Fig. 4, we get 18 out of 34 clients who are generalized by HAMIS, and averagely, the generalized dataset for each client is three times as large as the original dataset. The largest generalized dataset is from Client 7 which is far more than other expanded datasets. For comparing action rules, the results vary with the different generalized datasets associated with clients, and the number of action rules generated from those generalized dataset of clients is at least two times bigger than that from a client alone, which is still within our expectation.

5 Conclusion and Further Work

The paper presents HAMIS which is one of the main modules of a hierarchically structured recommender system for improving NPS. We have shown that by expanding datasets assigned to nodes of the dendrogram, recommender systems can give clients more promising suggestion for improving their NPS score. With the hierarchical dendrogram, HAMIS continually enlarges the dataset by following a bottom up path starting from the chosen node, higher the path ends, more generalized the dataset becomes. After applying HAMIS to all 34 clients in our domain, we notice that over half of the datasets can be expanded as shown in Fig. 4. As what we expected, action rules mined from extended datasets are far more promising than the ones from original datasets, no matter in quantity or in quality. Thus, the more generalized (more extended are the datasets) recommender system is built using HAMIS, better recommendations for improving the NPS score of clients can be given.

However, there are still some clients who can't benefit from HAMIS due to the failure of generalization based on semantic similarity. And 90% of them are caused by the lower NPS score of their most semantically similar clients. For example, Client 5 (in Nevada) is one of the clients which failed in HAMIS, but its geographical neighbor Client 17 (in California) has been expanded with Client 23 (in Mississippi) which can be seen in Fig. 4. Meanwhile, we speculate that knowledge hidden in dataset of Client 5 could be similar to the knowledge hidden in Client 17 to some extent, since customers locating nearby possibly have common thoughts about how they felt about the services and how they would like to be served. Therefore, it is interesting to think that Client 5 can benefit from the advices of Client 17 and even from Client 23, but the problem is how the geographic distance influences the generalization. What's more, the fact that the clients are geographically located close to each other does not necessarily imply they are semantically similar. The most fitting example is Client 24 which is in California and Client 34 which stays at Georgia. They are physically far away, but they are treated as most semantically similar as it is shown in Fig. 1, and Client 34 can't be generalized with Client 24 because of lower NPS rating. So this makes us wonder, maybe we can find another client near Client 34 which can offer some help. When looking into the location of Client 34, we happen to discover another interesting case, for Client 2 (in South Carolina), although it has been generalized using semantic similarity, it is surrounded by several clients with higher NPS ratings. Living such competitive environment, customers here could have stricter requirement for clients and harder to satisfy. With all the concerns above, we still have some space to make progress.

References

1. Kuang, J., Daniel, A., Johnston, J., Raś, Z.W.: Hierarchically Structured Recommender System for Improving NPS of a Company. In: Cornelis, C., Kryszkiewicz, M., Ślęzak, D., Ruiz, E.M., Bello, R., Shang, L. (eds.) RSCCTC 2014. LNCS (LNAI), vol. 8536, pp. 347–357. Springer, Heidelberg (2014)

2. 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)
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. 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)
5. He, Z., Xu, X., Deng, S., Ma, R.: Mining action rules from scratch. *Expert Syst. Appl.* **29**(3), 691–699 (2005). Elsevier
6. Raś, Z.W., Wiczorkowska, 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)
7. Reichheld, F.F.: The one number you need to grow. *Harvard Bus. Rev.* **18**, 46–54 (2003)
8. SATMETRIX, NET Promoter (2014). <http://www.satmetrix.com/net-promoter/>
9. SATMETRIX: Improving your net promoter scores through strategic account management, White paper (2012)
10. 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)