Modification of SEP/COP, an efficient method to find the best partition in hierarchical clustering, to improve a link prediction system

L. Sestorain, I. Perona, A. Yera, O. Arbelaitz, J. Muguerza
{lsestorain001@ikasle.,inigo.perona@,ainhoa.yera@,olatz.
arbelaitz@,j.muguerza@}@ehu.eus

Dept. of Computer architecture and Technology
University of the Basque Country
M. Lardizabal, 1, 20018 Donostia, Spain

Abstract. In this work, we improved the link prediction part of a web mining system developed for a tourism website (Bidasoa Turismo, BTw). First, we replaced the PAM clustering algorithm used in the profiling part of the system with the adaptation of other system, SEP/COP, which is based on a hierarchical clustering algorithm and is able to automatically adjust to the number of clusters required. Secondly, we modified the implementation of the exploitation part, i.e. the use of these profiles for link prediction, so that it adapts better to the system. Thirdly, we applied both systems, the original and the improved one, to another environment called Discapnet which is a website aimed at people with disabilities. Values of the calculated performance metrics confirm the improvement of the system and its generality for different environments.

Keywords: Data mining, link prediction, clustering

1 Introduction

Nowadays, owing to the increase of Internet usage, many new tasks have become necessary. One example of these tasks is link prediction which is used in a wide range of areas. For instance, investigating relationships between malicious websites, security agencies can find new websites that they have not looked up yet. Moreover, in social networks, this tool helps to propose friends, pages or events the user may like. What is more, link prediction systems could also be useful to apply in other areas such as medicine or biology, for example to discover relations which otherwise would need a long and expensive research.

After all, any natural environment that can be expressed using a network has an applicable prediction method. This method can provide answers for essential questions of its environment.

That is why many researchers have developed link predictions systems for web environments. Arbelaitz et al. designed for example a general web mining system [1] which includes a link prediction system as well as an interest profiling system based on the data provided by the local DMO (Destination Marketing Office) of the Bidasoa Turismo website (BTw). In the link prediction system, users are classified in different groups and their corresponding profiles, i.e. the set of URLs likely to be visited by each group of users, created. These profiles are used to propose links to the new users while they are navigating in the website.

Having this system as a starting point, the aim of the work presented in this paper is to improve the part of the system devoted to link prediction. Therefore, during the next pages, we study different unsupervised classifiers and improvement options so as to know which of them are appropriate for the system. After that, we choose the tests, fix the variables and execute the modified system in order to analyze the results. Finally, taking into account all the results, we prepare the best system and apply it to another website aimed at people with disabilities.

The next section describes the original system as well as its limitations. Section 3 describes the profiling system, Section 4 describes the changes we made in the SEP algorithm so as to adapt it to our system. In Section 5 we explain the changes made in the explotation part. In Section 6 we show the experiments and main results given with both environments and, finally, we draw some conclusions and mention some future works that are still to be done in Section 7.

2 Original system

The system was created based on the data of a tourism website called Bidasoa Turismo (BTw) [1]. The huge change that tourism industry has experimented was the main reason to start the project. Since travellers have started to use the net, intelligent systems have become necessary. These information systems must give the most important information to customers and service providers, help them to decide, help them move and offer them the best travel experience. It is a general automatic and non-invasive system . It combines web usage and content mining techniques to achieve the next three purposes:

1. Create navigation profiles in order to use them for link prediction.
2. Enrich profiles with semantic information so as to diversify and/or improve them. In addition, with this tool, DMOs (Destination Marketing Organization) can propose links combining their own interests with users' tastes and interests.
3. Get language-dependent interest profiles to help DMOs with future web design or marketing campaings.

In this paper, we explain the improvement of the first part, the one which creates navigation profiles.

2.1 Data preparation

First, the data received from BTw needed to be preprocessed. The reference system was built based on nearly ten months of usage data used in the experiments:

from January 2012 to October 2012. The tourism agents provided a database containing a total of 3,636,233 user requests. The information was collected in a *log* file. These documents are standarized and have several fields specified. However, only the IP address of the user who makes the request, the time the request was recorded, the URL that was requested and the status of the request were taken into account. The number of requests was reduced by eliminating erroneous requests and all the indirect requests such as requests to complete the web page with images and videos or URLs used for administration purposes.

Moreover, user sessions were identified by using the IP adress and the time between the requests. So, sessions with no activity for 10 minutes were considered as finished. In addition, requests with a number of clicks out of a certain range are also erased. Lastly, those URLs belonging to the most dynamic or volatile parts of the website were not considered either because these parts were out of the scope of the reference research work.

Finally, we got a database with 10,790 cases. Each case represents a session which is an ordered sequence of URLs. On average, sessions have 10 URLs.

2.2 Profiling

The system is adapted to the sequence representation of the user sessions; it uses PAM (*Partitioning Around Medoids*) [2] to do the clustering and the Edit Distance [3] [4] metric, a well known distance to compare sequences, to compare user sessions. The algorithm needs to have defined the number of clusters to be obtained from the database. This parameter k depends on the structure of the database and the accuracy of the profiles we want to create. The greater k is, the more specific the profiles will be.

The outcome of the clustering process is a set of groups of user sequences that show similar navigation patterns. After getting the groups, SPADE [5], an efficient algorithm for Mining Frequent Sequences, is used to find the most frequent sequences of each cluster in order to define the profiles. SPADE counts how many times each sequence appears in the whole database as a subsequence. Therefore, the same as with PAM happens with SPADE because it needs to define the minimum support. If the support of a sequence is greater than the minimum support, then the sequence is a frequent sequence.

Figure 1 shows the scheme of the architecture of the profiling system. First, we have the information from the server in log files. After preprocessing the data, we get a database of user sessions which are ordered sequences of URLs. Then, we cluster them into groups with similar navigation patterns and finally, we get the profiles and the most frequent sequences of each group in order to use them for link prediction.

As the minimum support of SPADE conditions the user profile assigned to each cluster, based on previous works, we fixed that parameter to 0.2 and the number of URLs of the profile or URLs to be proposed to 4.

2.3 Explotation

In order to find the most similar profile to the new sequence that we are testing, we only take into account the first part of the sequence (25%). This way, we

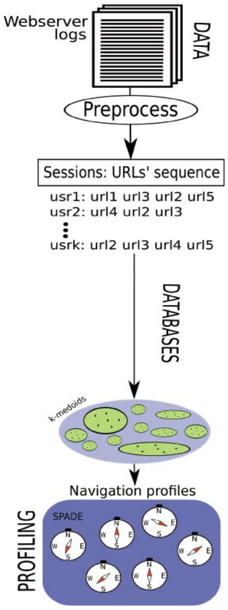

Fig. 1. Scheme of the architecture of the profiling system.

simulate the real situation in a website. These first clicks are considered the navigation of the new user.

Using this part of the sequence and the K-NN classifier [6], a profile is assigned to the new user. The classifier's criteria to choose the most similar cluster is the distance between the testing sequence and the cluster, in other words, between the testing sequence and the medoid of the cluster. After that, the system proposes the 4 URLs of this profile to the user. In case there are not enough acceptable links, it looks in the next closest cluster. In order to evaluate the system, the system compares the links it proposed with the links that follow in the sequence.

2.4 Evaluation

The cross validation evaluation method is used in the system to evaluate the proposals. It divides the data into 10 folds: seven of them are used to learn; two, for validation or selection of the best values for the parameters, and one, for testing. As sequences have 10 URLs on average, we express them like URL1, URL2, URL3, URL4, URL5, URL6, URL7, URL8, URL9, URL10. So, we take the first two clicks (URL1, URL2), 25%, from the testing sequence to determine which is the closest cluster, select the proposed URLs according to the profile of the selected cluster and compare the links that the system proposes with the rest, URL3, URL4, URL5, URL6, URL7, URL8, URL9, URL10, for the link prediction system.

The evaluation metrics used to evaluate the comparisons are precision, recall and f-measure. The first one represents the percentage of the URLs used among those which were proposed. The second measure indicates the percentage of the proposed URLs among those which were used. It is also called sensitivity. Lastly, f-measure is a measure to combine the results of precision and recall.

2.5 Results

Before starting the analysis of the results, in the original system, there are several parameters we need to fix. We fixed all of them unless the k parameter of PAM algorithm in BTw based on the previous works [1] [7].

- Minimum support for SPADE: 0.2
- Number of URLs to propose for SPADE: 4
- β to calculate f-measure: 0.5 (in order to give more importance to precision)
- k for PAM in BTw: 50, 100, 150, 200, 300, 400, 500[1]
- k for PAM in Discapnet: the values proposed as best ks in a previous work carried out with Discapnet.

After analysing the results obtained from the execution, we realised there are two main problems:

1. The results differ depending on the execution, in other words, on the data that is used in the folds of that certain execution. As we divide the database into 10 folds to do the cross validation, we use different data to learn, validate or test in each execution. Therefore, this points to the possibility of being under a dataset shift problem.
2. The parameter k affects to the results. In our experiments, we try many values for k. Anyway, we cannot be sure that the optimal value is among them so we can just use them as an approach. Moreover, if we changed the amount of information, the database or the environment, we would need to do all the examination again. The best solution is, for sure, to find a way to look for this value automatically, but how can we do that?

3 Modifications in clustering

Hierarchical algorithms show the whole hierarchy of clusters instead of showing just one partition so they give us more information. At the same time, they are much more complex. That is why they represent the hierarchy in a structure similar to trees called dendrogram. The nodes of the dendrogram indicate the clusters and the branches, the unions.

Nevertheless, we need to get the best partition, not the whole hierarchy. For this reason, it is necessary to select a partition from the hierarchy, i.e. to cut the dendrogram. Sometimes, the optimal partition is explicitly shown in the hierarchy. This means that we can get it by cutting the dendrogram horizontally.

[1] We chose these values in order to search the best k in a wide range of values.

But in some other cases, even though the optimal partition is in the dendrogram, it requires to cut it in a different way.

Owing to that fact, we used SEP (Search over the Extended Partition set) [8] which looks for the best partition among all the possible partitions of the dendrogram, including those that need a non-horizontal cut. This search space is called *the extended partition set*. The results obtained from the extended partition set must be at least as good as the results obtained from the hierarchy.

The use of SEP combined with COP would have the advantage of not having to explore a wide range of values for k looking for the best partition as it is required with most of the clustering algorithms.

The algorithm for SEP can be seen in the algorithm 1.

Algorithm 1 SEP(V, $Node$)

```
$C \leftarrow$ cluster in $Node$
if $Node$ is a leaf node then
    return $\{C\}$
else
    $Union \leftarrow \emptyset$
    for all $Child \in Node$ do
        $Union \leftarrow Union \cup \mathrm{SEP}(\mathrm{V}, Child)$
    end for
    if $\mathrm{V}(\{C\}) < \mathrm{V}(Union)$ then
        return $\{C\}$
    else
        return $Union$
    end if
end if
```

Where V is the cluster validation index (CVI) used to measure the optimality of a partition. CVIs evaluate partitions according to the cohesion of the objects inside the cluster and the difference or distance of the objects among clusters. Yet, SEP needs a special CVI so that the index can also evaluate partial partitions (P^Y). Because of that, we use COP (Context-independent Optimality and Partiality) [8].

The value of COP is estimated based on the intra-cluster variance or cohesion and the inter-cluster variance or distance, measured by complete linkage. It is defined like as

$$\mathrm{COP}(P^Y, X) = \frac{1}{|Y|} \sum_{C \in P^Y} |C| \frac{\mathrm{intra}_{COP}(C)}{\mathrm{inter}_{COP}(C)},$$

where

$$\mathrm{intra}_{COP}(C) = 1/|C| \sum_{x \in C} \mathrm{d}(x, \bar{C})$$

$$\mathrm{inter}_{COP}(C) = \min_{x_i \notin C} \max_{x_j \in C} \mathrm{d}(x_i, x_j)$$

and $\mathrm{d}(x_i, x_j)$ is the Euclidean distance between points x_i and x_j. Since COP cannot evaluate the root and leaf nodes, we set its value to 1 in these cases. Thus, it is able to analyse a partial partition.

As we are working with sequences, we cannot use the Euclidean distance, so we use the Edit distance instead. Moreover, we use medoids instead of centroids ($\bar{C}$). These changes and the structure of the dendrogram lead to the malfunctioning of the system with SEP –it always returned two clusters– so we made some modifications on the algorithm. Due to the analysis we made, we realised that there were leafs involved near the root, and therefore, the COP value of the union was always greater than the node's COP value in these cases. SEP understands a lower value as a better partition so that is why we lost the union and it returned always two clusters –it did not return the root because its COP value is 1, the worst possible value .

So as to solve this problem, we modified slightly SEP algorithm; we multiplied the union's COP value with a variable. This variable takes into account the number of leafs we have in the union. We calculate this proportion by dividing the number of leafs in the union with the total number of leafs in the dendrogram. Anyway, with this proportion, we would get a bigger number as we approach the root. So we define that variable as $1 - proportion$.

The results of the experiments can be seen in section 5. In the next section, we explain the changes made in the exploitation part of the system.

4 Modifications in explotation

As we mentioned before, K-NN is used to find the most similar cluster to the sequence we are testing. After the clustering, the medoid for each group is calculated which is, somehow, the center. In other words, it is the sequence that minimizes the distance to all the other sequences in the cluster. So when we have a new sequence, we compare it with the medoid of each cluster in order to decide which one is the closest to it. We found three possibilities to improve this process:

1. The original system compared directly 25% of the new sequence to the medoid in order to select the most similar profile. Being Edit Distance a global distance, the difference in length of the compared sequences might be a source of error in the selection of the closest profile.
   In order to solve this problem, we adjust the size of the medoid to the length of the part of the new sequence we are testing, considering, at most, two more URLs. Sometimes, it may happen that the medoid is shorter than the tested sequence. In these cases, the whole medoid has been taken into account.
2. While we were working on the task aforementioned, we realised that the distances used to choose the most similar cluster are absolute. The same example shows the inappropriateness of this fact. However, the importance of this issue minimizes after shortening the medoids in the comparison.
   We calculated the relative distance by dividing the absolute distance with the length of the longest sequence –it can be either the medoid or the testing

sequence. This way, the distance value is always between 0 and 1. After these two changes, the closest profile to the new sequence can be fairly chosen.

3. At the moment of proposing links, the original system only takes into account the closest cluster. Anyway, the links that this profile proposes may not be the most suitable links. There may be cases where the proposals of the closest profile have a very low support and therefore, proposals from the next closest cluster are more appropriate.

   The last modification affects the criteria to choose the proposals so as to take into account not only the distance to the cluster, but also the support of each link proposed. Firstly, we take the two closest profiles of the testing sequence. After that, we choose the proposals depending on the value we get from multiplying the support of each proposal with $1 - distance$. The bigger the value is, the more suitable the proposal will be.

   In case we do not get enough proposals from the two closest profiles, we keep looking for more links from the third cluster ahead.

5 Experiments

After making all the changes, we have executed the system with PAM and SEP algorithms and the data of two different environments: BTw (a tourism website), which is the website the original system was created for, and Discapnet (a website for disabled people) which is a new environment for trials.

In Table 1, we show for both environments, BTw (left) and Discapnet (right), average results obtained with the original system (based on PAM) and the system built replacing PAM by SEP (column Original). The values represent the f-measure values obtained with the link prediction system, using the first 25% of the test sequence to assign a profile. The same way, the table includes the results of the last version of the system (column Improved).

	BTw		**Discapnet**	
	Original	**Improved**	**Original**	**Improved**
PAM	0.220	0.293	0.173	0.248
SEP	0.207	0.280	0.160	0.235

Table 1. Average results for the original and the improved systems.

We can see in Table 1 that although, regardless of the environment, the best results are the ones obtained with PAM, SEP's outcome is really close and under no circumstances should we forget all the advantages it offers us.

As for the improved results, the same trend detected for the original systems applies. PAM's results are a little bit better than those obtained with SEP but both improve considerably with the modifications in the exploitation phase being this improvement proportionally higher in the case of SEP algorithm. Improvements for average results and BTw database are 28,45% for the PAM based system and 37,43% for the SEP based system, and, these improvements are in Discapnet database, 43,55% for PAM and 46,50% for SEP.

Both systems, the one using PAM and the one using SEP obtain improvements with each of the modifications made to the exploitation phase. As an example, we show in Figure 2 how the values evolve in the case of BTw database. The lower line represents the results of the system implemented with SEP while the upper line represents the outcome we got with the system implemented with PAM. On the other hand, the points on the left hand side show the result of the original systems whereas the points on the right end of the graphic display the result for the improved version of the system.

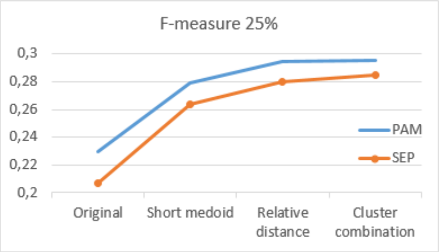

Fig. 2. Graphic of the improvement obtained in both environments.

It can be clearly seen that the improvement has been more or less the same for the two systems implemented with different algorithms, regardless of which is the application environment. However, SEP has got a greater change with the BTw than PAM.

Before finishing this section, it should be pointed out that PAM has required a lot of research for the adjustment of the parameter k in order to get these results. Nevertheless, SEP has adapted really well with any of the environments and has achieved almost as good results as PAM without the necessity of searching for the best k. In that context, the system based on SEP could be directly applied to any other new database, whereas the system based on PAM would require the exploration of a wide range of values of k.

6 Conclusions and further work

Thanks to the evaluation of the link prediction system developed for a tourism website (Bidasoa Turismo, BTw), we confirmed that clustering algorithms such as PAM require a lot of time and effort to adjust the k parameter to obtain the best possible results for a system. What is more, if the amount of data, the database or the environment changes, all that research becomes useless and the work has to be done again.

Therefore, the modification of the SEP algorithm so that the SEP/COP methodology can be used in this context and its use has been a great progress since we could directly use the system with the data of a new environment, without the need of adjusting any parameter. Furthermore, it has been able to give nearly as good results as PAM.

Lastly, regarding to the changes made in the exploitation part, they all contributed to substantially improve the performance of the system. The most significant change was obtained with the first change –adjusting the medoid to compare it with the part of the test sequence. Anyway, using the relative distance and combining two clusters also made a little difference.

On the whole, the system based on SEP obtains an improvement of 23.75% over the original system based on PAM in BTw and of 35.92% in Discapnet.

Although substantial improvements were obtained for the link prediction system this work showed that the analysis of new options could be worth it to further rise the performance of the link prediction system.

Although the original system uses Edit distance, which is a global distance, it could be interesting to try a local distance since they face the difficulty of the fields with low similarity level or using a sliding window otherwise.

On the other hand, we have seen that the original SEP algorithm did not work in this system. We made some modifications on it to make it work but that adjustment is not final. More experiments could lead the algorithm to get as good results as PAM, or even better.

Acknowledgements. This work was funded by the University of the Basque Country UPV/EHU (BAILab, grant UFI11/45); by the Department of Education, Universities and Research (grant IT-395-10); and by the Ministry of Economy and Competitiveness of the Spanish Government and by the European Regional Development Fund - ERDF (eGovernAbility, grant TIN2014-52665-C2-1-R).

References

1. Arbelaitz O., Gurrutxaga I., Lojo A., Muguerza J., Perez J.M., Perona I. : Web usage and content mining to extract knowledge for modelling the users of the bidasoa turismo website and to adapt it. Expert Systems with Applications **40** (2013) 7478–7491
2. Kaufman L., Rousseeuw P.J.: Finding Groups in Data: An Introduction to Cluster Analysis. Wiley-Interscience (1990)
3. Gusfield, D.: Algorithms on Strings, Trees, and Sequences - Computer Science and Computational Biology. Cambridge University Press, New York, NY, USA (1997)
4. Chordia B., Adhiya K.: Grouping web access sequences using sequence alignment method. Indian Journal of Computer Science and Engineering (IJCSE) **2** (2011) 308–314
5. Zaki M.J.: Spade: An efficient algorithm for mining frequent sequences. Machine Learning **42** (2001) 31–60
6. Dasarathy S.: Nearest neighbor (NN) norms: NN pattern classification techniques. IEEE Computer Society Press, New York, NY, USA (1991)
7. Arbelaitz O., Lojo A., Muguerza, J., Perona, I.: Global versus link prediction approach for discapnet: website focused to visually impaired people. Preprints of the Federated Conference on Computer Science and Information Systems (2014) 51–58
8. Gurrutxaga I., Albisua I., Arbelaitz O., Martin J.I., Muguerza J., Perez J.M., Perona I.: SEP/COP: An efficient method to find the best partition in hierarchical clustering based on a new cluster validity index. Pattern Recognition **43** (2010) 3364–3373