Effective Web Personalization Using Clustering

D.Vasumathi
Assoc. Professor, Dept.of Computer Science
JNTU College of Engineering
Hyderabad-500085, Andhra Pradesh, India

A.Govardhan
Professor & Head, Dept.of Computer Science
JNTU College of Engineering
Hyderabad-500085, Andhra Pradesh, India

K.Suresh
M.Tech. (IT), Dept.of Computer Science
JNTU College of Engineering
Hyderabad-500085,Andhra Pradesh, India
line 4: e-mail address if desired

***Abstract*— This paper describes web personalization, the essence of personalization is the adaptability of information systems to the needs of the adaptability of information system to the needs of their users. This issue is becoming increasing important on the web, as non-expert users are overwhelmed by the quantity of information available online. The proposed approach aims to mine a reduced set of effective association pattern rules for enhancing the online performance of web recommendations and proposed approach is evaluated based on the efficiency and quality. Adaptive user clustering and profiling is essential to be able to accurately predict user actions. In this paper we present results of our clustering and personalization project.We compare several distance measures used in clustering.We introduce a new measure to assess the quality of clustering independent of the distance measure used in the clustering process. We also compare different strategies(such as clustering users vs. clustering urls).**

I. INTRODUCTION

Recently, many researchers have investigated the generation of user profiles from raw web logs by using various data mining techniques. The technique of clustering individual user sessions is seen as a method to aid in collaborative filtering in [3], [4], [5], [6]. Incremental web-log mining to create adaptive

Web-servers have also been investigated [9]. Most of these approaches concentrate on either user or url clustering, but not on employing both. Likewise, they concentrate more on the clustering/profile creation methods themselves. In this paper we look at employing both user as well as url clustering information for better completeness and relevancy (these attributes are defined below) of the recommended set. We investigate several distance measures commonly employed in the literature and evaluate some others. We have also proposed a measure (the induced entropy) to assess the quality of clusters generated. This quantity does not depend on the distance metrics invoked by the clustering algorithm, it can be computed directly from the original data. Experimental data is presented and discussed.

II. PROPOSED APPROACH

Clustering users seems to be an indirect approach. In a sense, it emphasizes the question "what have like minded users done". The primary emphasis should be on the question "which urls is this particular user likely to visit ?". Then as a secondary option, the user might be told what like minded people have done (Amazon.com does this: after a book purchase, they inform the user which other books were deemed interesting by people who bought the same book). Intuitively it appears that urls should be clustered first, and a list of the (top few) url clusters closest to a user's usage pattern should be retrieved when the user signs on. Accordingly a comprehensive strategy would consist of the following steps:

(1) Cluster the URLs as well as the users (note that clustering of URLs helps answer the question "how similar are url's i and *j*).

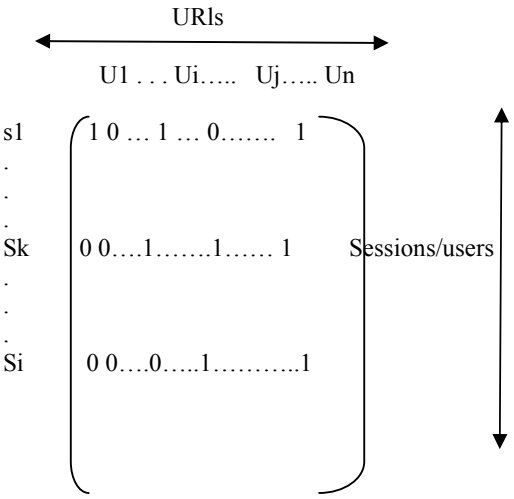
(2) For each user, maintain a list of the (closest few) url clusters. When the user logs-in, first retrieve links from the url clusters he/she is closet to.

(3) The last (auxiliary) step: search the user clusters to find what other urls "like minded" users have visited (that might be missed by the url clusters this particular user likes).

A. Distance metrics As shown in Figure 1, we create an "incidence matrix"(denoted by **A**), whose rows represent sessions and columns represent urls. If a url was visited in a session, the entry in that column is 1, otherwise it is 0 (as illustrated in figure 1). Note that the "1" could be replaced by the count of number of times a url was visited. This gives more detailed information of visitation patterns. In case users have explicitly ranked urls giving them scores of some sort, those scores can be entered instead of "1"s. We approximate a "session" to be a user. This is not necessarily true, but with cookies this approximation can be close to reality: a cookie

IAMA 2009

can identify whether the session originated from the same browser (which is likely to be used by the same end-user). The incidence matrix is likely to



be sparse and all well known techniques to handle sparse matrices can be applied here. Each url is an *m* dimensional vector and could be treated as a point in an *m* dimensional Euclidean space and any lustering algorithm can be applied. Likewise each session could be treated as a point in an *n* dimensional Euclidean space for the purpose of clustering. However, clustering methods that work directly with normed spaces are typically very slow when the number of dimensions (*m* or *n*) is large which is the case here. Using relational clustering methods that only need the relative distances could be a better approach. The important point is that the distance calculation (based on the huge incidence matrices) is not part of the clustering algorithm itself, (these are typically pre-computed), so the algorithm could be potentially faster. Note that metric space algorithms that update centers of clusters as they iterate will have to work with original spaces which have huge dimensionality. On the other hand, in relational clustering methods the entire information about a pair of points is summarized by a single number irrespective of the dimensionality of the underlying space. Hence relational methods can be expected to be faster and more compact. Moreover, relative distances are more amenable to incremental updating (as more data is collected over time). This coupled with the fact that a relational clustering algorithm could be potentially smaller and faster suggests that reclustering or incrementally updating the clusters would be easier with relational clustering methods. A matrix of pair-wise distances (or similarities among users or urls) which is required for relational clustering can be easily generated from the incidence matrix. For instance $\mathbf{A}^T\mathbf{A}$ is an *n* x *n* matrix which can be thought to indicate the similarities between urls (the *i j*th entry of $\mathbf{A}^T\mathbf{A}$ measures the number of sessions in which both the url's (*i* and *j*) were visited, which can be thought to measure the similarity between the two urls).

Likewise $\mathbf{A}\mathbf{A}^T$ is a measure of similarities between users. Note that each user could be considered a point in an *m* dimensional space with distance among users given by an *m* x *m* matrix (such as $\mathbf{A}\mathbf{A}^T$. However this space is not a normed space (triangle inequality does not hold). Hence relational clustering must be used in this space. In the following, Urls are dented by $\bar{u}$ s (i.e., $\bar{u}_i$ corresponds to the *i*th column of **A**), whereas sessions are denoted by *s* (i.e., s_j corresponds to the *j*th row of **A**). The distance between a pair of corresponding entities (represented by the corresponding vectors) is denoted by *D*. The normal (Euclidean or 2) norm of a vector is denoted by ||.||, whereas the 1-norm of a vector (sum of absolute values of its elements) is denoted by |.|

We have tested the following distance measures.

(i) Distance $D_{ij} = Max - \bar{u}_i^T . \bar{u}_j$ (1)

"Max" is the maximum value of dot product for any pair of vectors. (the dot product itself measures similarity, subtraction from the max is one way to render it to be a distance or dissimilarity)

(ii) Distance $D_{ij} = (1- \cos(\bar{u}_i,\bar{u}_j))^2$ (2)

where $\cos(\bar{u}i,\bar{u}j) = \bar{u}^T_{i.}\, \bar{u}_j / ||\bar{u}_i||.||\bar{u}_j||$ (3)

The main difference wrt. measure (i) above is that the raw popularities of the urls are factored out by dividing by the norm. For instance if incidence values for urls *i*, *j*, *k*, *l* in a row of incidence matrix **A** are 10, 10, 25, 25 then both the url pairs in question are correlated. Pair (*i*, *j*) is correlated, so is pair (*k*,*l*) but in one case (pair (*k*,*l*)), the raw popularity of the urls is higher and contributes more to the similarity.

(iii) Distance $D_{ij} = (1-\rho_{ij})^2$ where (4)

ρ_{ij}= correlation = $\bar{u}^T_{i.}\, \bar{u}_j - \langle\bar{u}_j\rangle.\langle\bar{u}_j\rangle / \sigma_{wi}\,\sigma_{wj}$ (5)

where ($\bar{u}i$) is simply the average (mean) defined as the sum of all the element's of the vector *ui* divided by the number of elements and,σ_{wi} = standard deviation = $\sqrt{\langle\bar{u}^2\rangle - \langle\bar{u}i\rangle^2}$

(iv)Distance $D_{ij} = ||\bar{u}_i - \bar{u}_j||$ (Euclidean norm) (6)

(v) Distance $D_{ij} = P(u_i\tilde{u}_j) + P(\tilde{u}_i u_j)$ (7)

where $P(u_i\, \tilde{u}_j)$denotes the probability of the event [visiting url *i* AND not visiting
url *j*]. Since

$P(u_i\tilde{u}_j) + P(\tilde{u}_i u_j) = P(u_i)+P(u_j)-2P(u_i u_j)$ (8)

We need to calculate the probability of the event [u_i is visited AND u_j is visited]. Since the sessions k can be approximated to be mutually independent events which together cover the entire event space, using conditional probabilities we get

$$P(u_iu_j) = \Sigma_k P(u_iu_j|k).P(s_k) \quad (9)$$

where $P(u_iu_j|k)$ is the conditional probability of the event [visiting urls i AND j | session k] It is straightforward to verify that

$$P(s_k) = |s_k| = \Sigma_i a_{ki} / M \quad (10)$$

where, $M = \Sigma_i \Sigma_j a_{ij}$ (sum of all elements of matrix **A**). Invoking the assumption that visiting *ui* and *uj* are independent
events, the conditional probability of visiting links i and j given session k is

$$P(u_iu_j|k) = a_{ki} a_{kj} / |s_k|^2 \quad (11)$$

substituting from (10) and (11) into (9), we get

$$D_{ij} = 1/ M (\bar{u}_i+\bar{u}_j-2\Sigma_k a_{ki} a_{kj} / |s_k|) \quad (12)$$

Clustering and Recommendation Since these two steps is not the main focus of our investigation,they are briefly outlined in this subsection. Each of the above distance measures was used to construct distance matrices for clustering urls as well as users. The next step was to invoke a relational clustering algorithm (which used the distance various matrices). We used the fuzzy-C medoids clustering algorithm [10], [11], [3], [12]. Note that the algorithm itself is immaterial, any of he large number published in the literature could be used in this step. We used the fuzzy-C medoids method because it seems to perform well and a robust implementation of that method was easily available to us. Reduced matrices are generated after clustering (this would be the whole point behind clustering). For example, suppose only urls are clustered. The reduced matrix would have fewer columns since all columns (urls) belonging to the same cluster are combined into a single column. Each entry in the new column is the sum of corresponding entries in the columns that got collapsed into this column. For instance, suppose url's i, j and k get clustered into 1 group. Then, the 3 columns i, j, k in the original incidence matrix **A** are replaced by a single column c where each element of (vector) c is the sum of corresponding elements
of i, j, k, i.e.

$$c_r = a_{ri} + a_{rj} + a_{rk}, \text{ where } r=1,\ldots\ldots,m \quad (13)$$

The same method is used when sessions are clustered: now multiple rows (belonging to the same cluster) are collapsed into one row representing the cluster. This way, no matter how the matrix is reduced, the sum of all elements always remains M. The next step is recommending urls based on the clustering. The way urls are recommended is this: a row from the incidence matrix is read off, corresponding to a session-cluster (or an individual session if no clustering has been done on sessions). The total incidence of that row is summed (this is simply the sum of all elements in that row). Urls from url clusters with the highest incidence in that row are recommended until the total incidence of the recommended urls exceeds the a threshold. The threshold function used was(totalIncidence/numSessionsInCluster), where the numerator is the row-sum mentioned above and the denominator is the number of sessions that got clustered together (i.e., the number of rows that have been collapsed into the current row).

A. *Evaluation of Clusters*

No matter how sophisticated (or simple for that matter) he distance measure and the clustering algorithm is, in the end what matters is the quality of clusters produced. The assessment of quality should be independent of the distance measure itself as far as possible (the clustering algorithm presumably already minimizes (optimizes) an objective function which includes the distances). Hence we used the following method. Since the end product is a set of recommendations, it is natural to evaluate how useful the recommendations were. Denote by R the set of urls recommended to a user, and by V the set of urls visited. Let $|S|$ denote the cardinality of a set S. Then we define

Relevance $R = |R| \cap |V| / |R|$ (14)

Completeness $C = |R| \cap |V| / |V|$ (15)

Bad recommendations
$B = |R| - (|R| \cap |V|)$ (16)

The score could be some combination f relevance, completeness and the number of bad recommendations,

score/quality $Q = f(R,C,B)$ (17)

simplest f = linear combination
= $\alpha R+\beta C-\gamma B$

which is a linear combination of R,C,B where α, β, γ are constants. See [13] for another method of combining the scores into one value. We look at R,C,B separately without combining hem into one value. These attributes (R,C,B, etc.) can be judged subjectively since they can be assigned values/importance (at least subjectively). For instance, in some cases, completeness might be the main criteria whereas in some other cases, relevancy might be the chief attribute. As an example, consider a mobile user who asks for a list of restaurants within 1 mile of their current location. The businesses paying the recommendation service provider want relevancy: if their web-site is recommended to the user, they

want the user to at least visit that web-site (($V \cap R)/R$ to be maximum). The recommendation service provider on the other hand would like to primarily worry about completeness: they'd not be very happy if the user took none of their suggestions which would signal to them that the user did not like anything they recommended and may not pay for their service next time around (they want $(V \cap R)/V$ to be maximum)). For this reason, these scores are together referred to as "subjective scores" in the remainder of the manuscript. We define the entropy E of the original matrix A to be

$$E = -(\Sigma_i \Sigma_j a_{ij} / M \lg a_{ij} / M) \qquad (18)$$

In a sense, E measures the spread of total incidence M among the sessions and urls. If, and only if, only one entry is non-zero then $E = 0$ (since $x \lg x \approx 0$ as $x \to 0$, x dominates $\lg x$). On the other extreme, if all entries are identical then $E = \lg(mn)$ where mn is the total number of elements in the (incidence) matrix. For any other distribution, the E is in between these two extremes. It is seen that E is biased by the dimensions of the matrix (for example consider two different matrices with all entries identical, one with dimensions $(m1,n1)$ and the other with dimensions $(m2,n2)$, even though they are both equally spread, their E values are different). The reduced matrix (generated by clustering) can be shown to have less E than the original matrix. This is intuitive, clustering is trying to reduce the spread by concentrating many incidence entries into one, and is also reducing the dimensions. Trying to minimize E is therefore not a good way to assess the quality of clustering (trivially group all sessions and all urls into 1 cluster which will yield a 1 x 1 matrix whose $E = 0$, but this clustering is certainly not what is desired). Instead we look at the E of another matrix *induced* by the reduced matrix (the induced entropy is denoted by E). The induced matrix has the exact same dimensions as the original matrix. Suppose that columns (urls) $c1,c2,\ldots..,ck$ got collapsed into 1 column because of url clustering and rows (sessions) $r1,r2,\ldots\ldots,rl$ got clustered together. In the reduced matrix the k x 1 block in the original matrix is replaced by a single element whose value σ is the sum of all the elements in the original k x 1 sub-block as explained above. To generate the induced matrix this sum σ is uniformly spread over a k x l sub-block, i.e., each entry in the sub-block now has the value σ/kl . In other words the single entry in the reduced matrix is replaced by a k x l sub-block.

III. Experimental Results and Conclusion

We tested the different distance metrics using the evaluation criteria described above. Real web access logs from our departmental web-server were parsed to generate an incidence matrix of dimension 2474 (sessions) x 3013 (urls). From this incidence matrix, 10 different distance matrices were generated, one set of 5 matrices for pair-wise distances among urls, corresponding to each of the 5 distance metrics described above. The other set of 5 distance matrices was for pair-wise distances among sessions/ users. The fuzzy C-medoids relational clustering method was then used to cluster the data. This algorithm does not figure out the number of clusters by itself, that. number must be provided as an input parameter to the algorithm.

For each value of number of clusters, the algorithm was run 30 times. The best clustering

("best" decided based on the value of the objective function minimized by the algorithm) from each batch was used to evaluate the scores mentioned in section 2.3 above. The whole thing was run 42 times for each distance measure and the average scores, with standard deviation in parentheses, are tabulated in Tables I–II . The first column lists the number of sessions (out of the original 2474) for which the intersection between the set of recommended urls R and the set of visited urls V was nonzero, i.e., at leastone of the recommended urls was visited. This is the primary of aspect of the subjective scores we are concerned with. For those sessions for which $R \cap V \neq 0$, the average relevance and completeness is listed in columns 2 and 3 (the average was computed over the total number of sessions with non-zero intersection). In the tables, the 4th column (titled $|B|$) shows the average number of bad recommendations averaged over those sessions for which $R \cap V = 0$. The fact that the average is 1 or 2 indicates that a small number of useless urls were recommended when there was no intersection (between recommended and visited) which is good. The first 4 columns in the tables list the subjective scores. strategies. First strategy is to cluster on urls alone (reduced matrix dimensions 2474x900). In this case all sessions had non-zero $R \cap V$.

The last column lists the entropy of the induced matrix. In each case (row of the table), the induced matrix has the same dimensions as the original incidence matrix (i.e., 2474 x 3013). The entropy of the original incidence matrix was $E = 13.3582$. Table 1 (looking at the "number intersected" column)demonstrates that when the number of clusters is large (i.e,the distances between elements within a cluster are small), the Euclidean norm distance metric leads to (relatively) better subjective scores.Table 3 compares different

Hence there are no sessions to average the bad ecommendations over, which is why the corresponding entry indicates "not applicable". Second strategy is to cluster on sessions (reduced matrix dimensions 35x3013). The third row indicates the following strategy: cluster on urls first to obtain a reduced matrix 2474x900, and then cluster this matrix on sessions (so that overall reduced matrix is 35x900). This clustering of sessions is in general different from the one obtained in in the second strategy. The session-cluster-membership mapping generated by this two step process is then applied to the original
matrix to generate an alternate session-clustered matrix of size 35x3013 and the subjective scores and entropy for this reduced matrix are indicated in the table. The last two strategies yield overall reduced matrix of size 35x900 and test if the ordering of reduction steps matters. The table demonstrates that clustering urls leads to good completeness, whereas clustering sessions leads to better relevance. For the most part the induced entropy is better (lower) when the

subjective scores are better. We are further investigating the few cases where the scores go one way and the entropy goes the other way.

REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. (references)

[2] S. Schechter M. Krishnan and M. D. Smith, "Using path profiles to predict http requests," in *Proceedings of the 7th International World Wide Web Conference, Brisbane, Australia*, April 1998.

[3] A. Joshi, C. Punyapu, and P. Karnam, "Personalization and asynchronicity to support mobile web access," in *Proc. Workshop on Web Information and Data Management, 7th Intl. Conf. on Information and Knowledge Management*, November 1998.

[4] O. Nasraoui, H. Frigui, A. Joshi, and R. Krishnapuram, "Mining web access logs using relational competitive fuzzy clustering," *Eight International Fuzzy Systems Association World Congress*, August 1999.

[5] B. Mobasher, R. Cooley, and J. Srivastava, "Creating Adaptive Web Sites Through Usage-Based Clustering of URLs," in *Proceedings of the IEEE Knowledge and Data Engineering Exchange Workshop (KDEX)*, 1999.

[6] Mobasher, . Cooley, and J. Srivastava, "Automatic Personalization Based On Web Usage Mining," *Communications of the ACM*, vol. 43, no. 8, pp. 142–151, Aug. 2000.

[7] B. Mobasher, H. Dai T. Luo, M. Nakagawa, Y. Sun, and J. Wiltshire,"Discovery of aggregate usage profiles for Web personalization," in *Proceedings of the WebKDD Workshop at the ACM SIGKKD, Boston*, Aug. 2000.

[8] B. Mobasher, H. Dai, T. Luo, and M. Nakagawa, "Improving the Effectiveness of Collaborative Filtering on Anonymous Web

[9] Data," in *Proceedings of IJCAI Workshop on Intelligent Techniques for Web Personalization (ITWP01), Seattle*, 2001.

[10] P. Domingos, C. Anderson, and D. Weld, "Adaptive Web Navigation for Wireless Devices," in *Proceedings of the International*

[11] *Joint Conference on Artificial Intelligence (IJCAI), Seattle*, 2001,pp. 879–884.

[12] T. Kamdar and A. Joshi, "On creating adaptive web servers using weblog mining," Tech. Rep., CS-TR-00-05,CS Department,

[13] UMBC, 2000.[10] A. Joshi and R. Krishnapuram, "Robust fuzzy clustering methods to support web mining," in *Workshop in Data Mining and Knowledge Discovery SIGMOD*, 1998, vol. 15, pp. 1–8.

[14] A. Joshi, K. Joshi, and R. Krishnapuram, "On mining web access logs," Tech. Rep., CS Department, UMBC, 1999.

[15] R.Krishnapuram, A.Joshi, O.Nasraoui, and Liyu Yi, "Low complexiy fuzzy relational clustering algorithms for web mining," in *IEEE Transcations on Fuzzy Systems*, 2001.

[16] David D. Lewis andWilliam A. Gale, "A sequential algorithm for training text classifiers," in *Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval*, W. Bruce Croft and Cornelis J. van Rijsbergen, Eds., Dublin, IE, 1994, pp. 3–12, Springer Verlag, Heidelberg, DE

Table 1: scores for the different distance metrics when number of URL clusters 900 and number of user clusters = 35

Distance Measure ↓	Average Scores over 42 runs (Std. dev. in parentheses)				
	Number intersected	Relevance	Completeness	|B |	Induced Entropy E
(i) (1-cos)2	506(42)	.96(.017)	.34(.056)	1.094(.051)	18.65(.17)
(ii) (1-p)2	494(91)	.96(.017)	.34(.068)	1.114(.060)	18.66(.17)
(iii)Euclidean Norm	750(114)	.98(.009)	.34(.039)	1.047(.040)	19.06(.19)
(iv)Dot product	272(82)	.92(.034)	.33(.079)	1.130(.040)	19.56(.20)
(v) Probability	167(93)	.94(.084)	.67(.109)	1.014(.023)	21.32(.20)

Table 2:Scores for the different distance metrics when number of url clusters = 50 and number of user clusters = 35

Distance Measure ↓	Average scores over 42 runs (Std. dev. in parentheses)				
	Number intersected	Relevance	Completeness	|B|	Induced Entropy E
(i) $(1\text{-cos})^2$	178(66)	.81(.107)	.29(.080)	1.81(.34)	20.36(.39)
(ii) $(1\text{-p})^2$	137(58)	.77(.095)	.28(.055)	1.75(.34)	20.50(.33)
(iii)Euclidean Norm	225(96)	.78(.109)	.25(.059)	2.03(.62)	20.91(.32)
(iv)Dot product	71(36)	.84(.113)	.30(.073)	1.30(.25)	21.04(.41)
(v) Probability	99(64)	.92(.102)	.55(.221)	1.14(.13)	22.33(.14)

Table 3: Scores for different strategies for the $(1\text{-cos})^2$ distance metric. number of url clusters=900 and number of session clusters=35.

Strategy ●	Average Scores over 42 runs (Std. dev. in parentheses)				
	Number intersected	Relevance	Completeness	|B |	Induced Entropy E
Cluster Urls only	2474(0)	.38(.018)	1.0(0)	n/a	14.71(.06)
Cluster sessions only	1260(41)	.99(.017)	.41(.015)	1.004(.060)	18.33(.17)
Cluster sessions method2	1212(48)	.99(.005)	.40(.017)	1.013(.012)	18.02(.17)
Cluster Urls then sessions	272(83)	.96(.017)	.34(.056)	1.094(.051)	18.65(.17)
Cluster sessions then Urls	521(113)	.96(.023)	.37(.062)	1.153(.185)	18.98(.19)