ISSN:0975-9646

Effective Classifier for User's Behavioral Profile Classification

Umesh K. Gaikwad , Shirish S. Sane

K. K. Wagh Institute of Engineering Education & Research, University of Pune, Maharashtra, India

Abstract: **Classification of computer users is very useful for assisting them, anticipating their future actions. In addition, it is very useful for making recommendations to a user based on the histories of other users with similar preferences, detecting changes in the behavior of a user, and so on. Several approaches for classifying users are available, however many of them do not consider the changes in user's behavior, as it is essential in some of the categories of users. For example, a computer user behavior is represented as the sequence of commands issued during various sessions. In such cases, the user behavior is not necessarily fixed but rather it changes, it is necessary to consider his evolving nature. Proposed work deals with Prototype Based approach to correctly classify the created profiles. Although there are different strategies are available for generating prototype, it is necessary to investigate effectiveness of statistical distance metrics for prototype creation. The work presented in this paper deals with selection of the best statistical distance metrics for prototype generation. It can be applicable to any environment where user behavior is represented as sequence of actions or events.**

Keywords: **Behavior Recognition, Sample density, Sequence Learning, Prototype Based Classifier, Statistical Distance Metrics**

I. INTRODUCTION

User classification is the process of learning about users by observing the way they use the systems or applications. This process needs the creation of a user profile that contains information that characterizes the usage behavior of a computer user. Observation has shown that users themselves do not know how to articulate what they do, especially if they are very familiar with the tasks they perform. Computer users, like all of us, leave out activities that they do not even notice they are doing. Thus, only by observing users, one can model his/her behavior correctly .

Self learning Prototype based classifiers can play significant role in the area of user classification. A prototype is a data sample that groups several samples which represent a certain class. Prototype generation process can start from scratch. The development of the Prototype Library is gradual. It declares any new sample as a Prototype when that sample cannot be described by the existing Prototypes and when they are descriptive enough.

This paper is organized as follows: Section 2 provides an overview of User profile and existing classification systems. Section 3 describes the structure of Prototype based Classifier for user profile classification. Section 4 contains Experimentation Results. Finally, Section 5 contains concluding remarks.

II. BACKGROUND AND RELATED WORKS

User Profile creation

There exist several definitions for user profile [1]. It can be defined as the description of the user interests, characteristics, behaviors, and preferences. One can prepare user's behavioral profile only by observing activities perform by him during his work[2].Such behavioral profile is very useful in many areas like user recommendation, Intrusion detection etc.

Another very important aspect for creating profile is temporal dependencies among different activities. It is considered that a current event depends on the events that have happened before and it is related to the events that will happen after[3]. Taking this aspect into account, we need sequence learning strategies to create more accurate profile.

User Profile Classification

There is great deal of work in the area of User Classification. For the Web environment, Macedo[3] proposed a system that provides recommended based on the history of use of specific users. Pepyne [4] has modeled users behavior by using queuing theory and logistic regression. For intrusion detection, Coull[5] propose an classification algorithm that uses pair wise sequence alignment to characterize similarity between sequences of users actions. Angelov and Zhou proposed fuzzy classifier for User classification purpose[6]. Although there is a lot of work focusing on user classification in specific environment, it is not clear that they can be transferred to other environments.

Along with this many traditional algorithms are also available which can be used for user classification purpose. In [7], Panda compared different traditional algorithms for classification of user profile-Naive Bayesian (NB), C4.5 and Iterative Dichotomizer 3 (ID3)—for network intrusion detection. According to the work Naive Bayesian performs better to overall classification accuracy. Cufoglu[8] evaluated the classification accuracy of NB[9], IB1[10], Simple CART[11], NBTree[12], ID3[13], J48[14] algorithms with large user profile data. According to the simulation results, NBTree classifier performs the best classification on user-related information.

Several studies[15][16] shows that Prototype based Classification schemes works better in the area of User classification. Better classification needs Quality prototype which can be generated by using different strategies. Statistical metrics plays important role in the process of Prototype generation. However, it is important to note that all of the above approaches ignore the fact that of better statistical distance matrices for classification can affect

overall classification accuracy. There are few attempts of research in this area. J. Iglesias [17] used cosdist for classification, but the work has not focused much more on the use of other statistical measurement techniques as well as their effect on classification accuracy.

Statistical Metrics & User Classification

From the scientific and mathematical point of view, similarity/distance is defined as a quantitative degree that enumerates the logical separation of two objects represented by a set of measurable attributes/characteristics[18]. Measuring similarity or distance between two data points is a core requirement for several data mining and knowledge discovery tasks that involve distance computation. Examples include clustering (k-means), distance-based outlier detection etc. [19]. There are a wide variety of distance metrics are available which give significant results of similarity/distance calculation between two items. Some of them are cosine distance, Euclidean distance, squared Euclidean, Chebyshev distance metrics and Manhattan metrics[20][21][22].

It is very interesting to investigate the effect of statistical metrics on the overall accuracy of classification. Work presented in this report proposes prototype based classifier in which prototypes are generated by using five major distance matrices. The performance of the system is analyzed considering total no. of prototype generated and classification accuracy with each metrics.

III. EFFECTIVE PROTOTYPE BASED CLASSIFIER

Proposed prototype based classification approach for user profile consists of different stages.

1] User behavior profiles Construction: Generate a Users Profile by considering a sequence of activities and their support value which can be calculated using no. of occurrences.

2] Prototype based Classifier Building:

- Calculate Statistical distance between profile using suitable statistical distance metrics. To select better metrics proposed approach has used various distance metrics such as Cosine, Euclidean, Chebyshev, Manhattan etc.
- Calculate density of the User profile using distance among profiles.
- Generate prototype for classification by considering density factor of different profile.

3] User Profile classification: Classify the sample using prototypes generated by classifier.

Figure shows Architecture diagram of the proposed approach

3.1 User Profile Creation

As temporal dependencies among activities are useful, proposed approach creates a user profile as a distribution of relevant subsequences. A activity sequence is an ordered list of elements (events, commands,...) that represents a behavior (pattern) of the user. it can be interpreted as {e1 → e2 →... → en} where n is the length of the sequence. First step is to extract the significant pieces of the sequence of commands that can represent a pattern of behavior. The construction of a user profile from a single sequence of commands is done by a two steps process: 1. Segmentation of the sequence of commands 2. Creation of the user profile. These steps are detailed in the following section.

3.1.1 Subsequence Generation:

The sequence of activities is segmented in subsequence of equal length from the first to the last element and store it into suitable storage along with its frequency count. Let's consider following sequence of activities perform by the user,

{w5 → w1 → w5 → w1 → w5 → w3}

Firstly, this sequence must be split into different segments of equal length. Let's consider sub-sequence length is three, then the sequence get split in two sub-sequences:{w5 → w1 → w5} and {w1 → w5 → w3}. Because of repeating and significant sub-sequences are important to determine the sequence pattern, the suffixes of the sub-sequences are also studied. In the illustration along with first subsequence it's suffixes {w1 → w5} and {w5} are also considered for the profile.

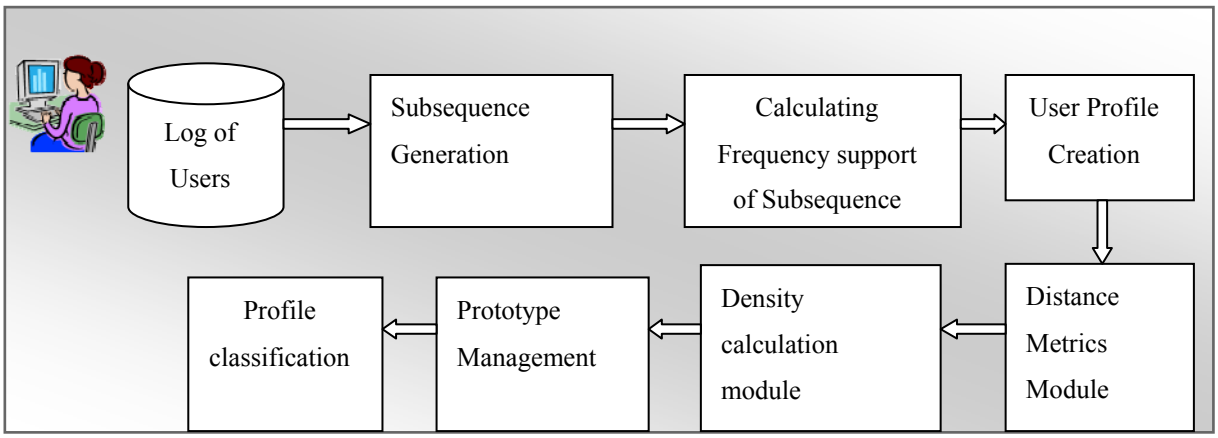

Figure 3.1 Architecture diagram

3.1.2 Profile creation:

After completion of segmentation phase, User profile is created using subsequences and it's supports value . Following formula is used for calculating support value of subsequences

$$\text{support of subsequence} = \frac{\text{No. of occurances of the subsequence}}{\text{Total No. of subsequences of the equal length}} \quad [3.1]$$

3.2 Classifier Building

3.2.1 User Behavior Representation in common data space

Observation of sequence of activity is needed For user behavior representation. They are converted into the corresponding distribution of subsequences using segmentation strategy .In order to classify user behavior, these distributions must be represented in a data space. For this reason, each distribution will be considered as a data vector that defines a point that can be represented in the data space. By considering this approach The data space of n dimension will get created , where n is the number of the different subsequences that could be observed. Let's consider following subsequence
with their support value,
User 1: (ls-0.5, date-0.3, pix-0.2, cat-0.75, vi-0.1)
User 2: (ls-0.6, date-0.1, vi-0.2, rm-0.8, emac-0.3)
User 3: (ls-0.3, vi-0.5, mail-0.9)

In this example, the distribution of the first user consists of five subsequences of commands therefore we need a 5 dimensional data space to represent this distribution (each different subsequence is represented by one dimension). If we consider the second user, we can see that 2 of the 5 previous subsequences have not been typed by this user (pix, cat). Also, there are 2 new subsequences (emacs and rm) so the representation of this value in the same data space needs to increase the dimensionality of the data space from 5 to 7. To sum up, the dimensions of the data space represent the different subsequences typed by the users and they will increase according to the different new subsequences obtained. for above example data space will be as shown in table 3.1.

Table 3.1 :Common Dataspace

User	ls	date	pix	cat	vi	rm	Emac	mail
1	0.5	0.3	0.2	0.75	0.1	------	------	-----
2	0.6	0.1	-----	-----	0,2	0.8	0.3	-----
3	0.3	-----	------	------	0.5	------	------	0.9

3.2.2 Calculating Distance

On Entry of new sample its future vector get extracted. Then in next stage distance between future vector of current sample and future vectors of all previous samples of data space get calculated. For calculating distance various metrics are available.

3.2.3 Selection of Metrics for distance calculation

In order to analyze effectiveness of various distance metrics for classifying user profile, proposed approach encode 5 different metrics for calculating distance. Their details are as follows Let's consider $\mathbf{p} = (p_1, p_2,..., p_n)$ and $\mathbf{q} = (q_1, q_2,..., q_n)$ are two vectors in the space where n is no. of attributes, then the different distances from **p** to **q** or from **q** to **p** is given by:

1] Cosine distance:

$$cosine(p_n, q_n) = 1 - \frac{\sum_{i=1}^{n} p_i q_i}{\sqrt{\sum_{i=1}^{n} {p_i}^2 \sum_{i=1}^{n} {q_i}^2}} \quad [3.2]$$

2]Euclidean:

$$Euclidean(p_n, q_n) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \quad [3.3]$$

3]Squared Euclidean:

$$SquaredEuclidean(p_n, q_n) = (p_1 - q_1)^2 + (p_2 - q_2)^2 + (p_3 - q_3)^2 \dots\dots (p_n - q_n)^2 = \sum_{i=1}^{n} (p_i - q_i)^2 \quad [3.4]$$

4]Manhattan distance:

$$Manhatten\ (p_n, q_n) = |(p_1 - q_1)| + |(p_2 - q_2)| + |(p_3 - q_3)| + \cdots + |(p_n - q_n)| = \sum_{i=1}^{n} |(p_i - q_i)| \quad [3.5]$$

5]Chebyshev Distance:

$$Chebyshev(p_n, q_n) = MAX|(p_1 - q_1)|,\ |(p_2 - q_2), \dots\dots, |(p_n - q_n) = Max|(p_i - q_i)| \quad [3.6]$$

3.2.3 Density Calculation

After distance calculation next step is to calculate density of a sample. Statistical distance of a sample from all other samples of data space is used for density calculation. The density (D) of the Kth data sample x_k is calculated by equation(2) , which represents a part of the accumulated distance between a sample and all the other k - 1 samples in the data space. The result of this mapping represents the density of the data that surrounds a certain data sample k.

$$D\ (X_i) = \frac{1}{1 + \sum_{i=1}^{k-1} \frac{dist^2(x_k, x_i)}{k-1}} \quad [3.7]$$

Where dist represents the distance between two samples in the data space. In proposed application, the data are represented by a set of positive support values. So it is possible to simplify the calculation of the above-mentioned expression. For this reason, one can use the following equation 3.8 instead of equation no. 3.7. It uses simply the distance instead of square of the distance.

$$D_k(z_k) = \frac{1}{1 + \sum_{i=1}^{k-1} \frac{dist(x_k, x_i)}{k-1}} \quad [3.8]$$

3.2.4 Creating New Prototypes,

The density of the new sample (z_k) is compared with the density of the existing prototypes. A new prototype is

created if its value is higher than any other existing prototype. If the new data sample is not relevant, the overall structure of the classifier is not modified. Otherwise, if the new data sample has high descriptive power and generalization density, the classifier evolves by adding a new prototype, which represents a portion of the observed data sample. Condition for creating new prototype is as shown in equation 3.9.

$$\text{Density}(z_k) > Density(\text{Prototype}_i) \quad [3.9]$$

Where,

z_k =current sample, D(z_k)=density of current data sample, i=1 to No of prototypes

3.2.5 User Profile Classification

In order to classify a new data sample, compare it with all the prototypes stored in the Prototype Library. This comparison is done using Statistical distance among sample and all other prototypes. The smallest statistical distance determines the closest similarity. Sample is classified to class label of a prototype with closet similarity. The time consumed for classifying a new sample depends on the number of prototypes and its number of attributes.

IV. RESULTS

The performance of the proposed system is measured over the real world dataset called Greenberg's 168 Users Unix Dataset. As discussed in previous chapters first phase of the proposed approach is generating user profile using user activity log. For this purpose proposed approach has used segmentation strategies in which sequence of activities are converted into various subsequences. Each user profile is represented using future vector. It consists of support value of all the different subsequences of commands obtained for all the users. These subsequences act as attributes of a profile. There are subsequences which do not have a value because the corresponding user has not used those commands. In such a case, in order to be able to use this data for training the classifiers, the proposed approach has considered this value as 0.

Tables 4.1 shows number of different attributes (subsequences) obtained using different number of commands for training (10, 20, 30, 40 and 50 commands per user)To classify User profile, proposed system calculates its distance from all other profiles of data space and then by using this distance, density of that profile is calculated. The density is used for prototype generation and classification.

Table 4.1: Total No. of Attributes obtained

No. of Commands Per User	Different No. of Attributes Generated
10	3,166
20	6, 569
30	9,656
40	12,633

For the best selection of distance metrics the system considers various distance metrics, namely, Cosine, Euclidean, Squared Euclidean, Manhattan and Chebyshev. A table 4.2 shows the classification accuracy using various distance metrics.

Table 4.2: Classification Accuracy with Different Metrics

Statistical Distance Metrics	Average Classification Accuracy(10,20,30,40,50 Commands/user)
Cosine	36.54
Euclidean	44.75
Squared Euclidean	44.99
Manhattan	58.3
Chebyshev	49.63

It also shows that Manhattan and Chebyshev has better average classification accuracy for classifying the profile. As shown in figure 4.3 one can observe comparison of Classification accuracy with various distance metrics.

One can consider Classification accuracy as an evaluation criteria for measuring the performance of the system. Depending on this **Manhattans** distance metric provides the best average classification accuracy for user profile. But several studies shows that Classification Accuracy itself not a complete criteria. So, for evaluating system performance, it was decided to use new criteria for measurement. This criterion is **Good classification accuracy with less no. of prototypes**.

Prototype Generated by All schemes for classification is also varies. Though Manhattan has better accuracy still it generated more number of prototypes as compare to other approaches. So, by considering both criteria experimentation shows that Chebyshev provides Good classification accuracy with less no. of prototypes.

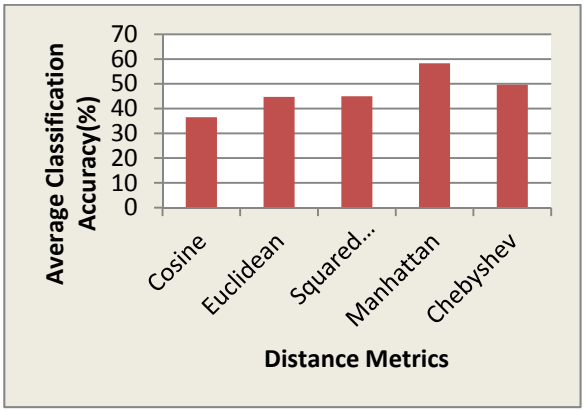

Figure 4.1: Classification Accuracy Graph

V. CONCLUSION

A novel approach, Prototype Based Classifier for User Profile classification using various statistical distance metrics has been presented in this paper. Sequence learning plays an important role in User Profile characterization. The proposed approach follows segmentation based sequence learning strategy to create a better user profile. On entry of new, sample for

classification, the proposed system calculates its statistical distance from all other samples of data space and then by using the statistical distance, density of that sample is calculated which is further used for prototype generation. Prototype is generated only if density of new sample is more than all other previous prototypes. One major aspect during this process is distance metrics for calculating the distance. It is important to note that selection of statistical distance metrics for generating prototype may affect the overall classification accuracy as well as performance of the system. For better selection of metrics proposed approach have considered 5 different metrics such as Cosine, Euclidean, Squared Euclidean Chebyshev and Manhattan. The result shows that Chebyshev metrics perform significantly well in terms of number of prototypes generated as well as classification accuracy.

REFERENCES:

1. Jose Antonio Iglesias, Agapito Ledezma, "Creating User Profiles from a Command-Line Interface: A Statistical Approach", UMAP, vol. 5535 of LNCS, pp. 90–101, Springer, 2009.
2. Hackos, J.T., Redish, J.C.: User and Task Analysis for Interface Design. Wiley, Chichester (1998).
3. Macado
4. D.L. Pepyne, J. Hu, and W. Gong, "User Profiling for Computer Security", Proc. American Control Conference, pp. 982-987, 2004.
5. S.E. Coull, J.W. Branch, B.K. Szymanski, and E. Breimer, "Intrusion Detection: A Bioinformatics Approach", Proc. Ann. Computer Security Applications Conf. (ACSAC), pp. 24-33, 2003.
6. P. Angelov and X. Zhou, "Evolving Fuzzy Rule-Based Classifiers from Data Streams", IEEE Transactions on Fuzzy Systems: Special Issageue on Evolving Fuzzy Systems, vol. 16, pp.1462-1475, 2008.
7. M. Panda and M.R. Patra, "A Comparative Study of Data Mining Algorithms for Network Intrusion Detection", Proc. Int'l Conf. Emerging Trends in Eng. and Technology(ICETET '08), pp. 504-507, 2008.
8. A. Cufoglu, M. Lohi, and K. Madani, "A Comparative Study of Selected Classifiers with Classification Accuracy in User Profiling", Proc. WRI, World Congress on Computer Science and Information Eng. (CSIE), pp. 708-712, 2009
9. Harry Zhang "The Optimality of Naive Bayes", *FLAIRS Conference, pp. 562-567,2004.*
10. Cover TM, Hart PE "Nearest neighbor pattern classification", IEEE Transactions on Information Theory , vol. 13(1), pp. 21-27,1967.
11. Wei-Yin Loh "Classification and regression trees", Inc. *WIREs Data Mining Knowledge Discovery,* Volume 1, Issue 1, pp. 14-23 , 2011.
12. Ron Kohavi, "Scaling Up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid", Second International Conference on Knowledge Discovery and Data Mining, pp.202-207, 1996.
13. P.E. Utgoff, "Incremental Induction of Decision Trees", Machine Learning, vol. 4(2), pp. 161-186, 1989.
14. J. Quinlan, "Data Mining Tools See5 and c5.0", http://www.rulequest.com/see5-info.html
15. Isaac Triguero, Joaquin Derrac, "A Taxonomy and Experimental Study on Prototype Generation for Nearest Neighbor Classification", IEEE Transactions On Systems, Man, And Cybernetics, vol 42(1), pp.86-100, 2012
16. Salvador García ,"A memetic algorithm for evolutionary prototype selection: A scaling up approach", Pattern Recognition, vol.41 , pp. 2693 – 2709,2008
17. Jose Antonio Iglesias " Creating Evolving User Behavior Profiles Automatically", IEEE Transactions On Knowledge And Data Engineering, Vol. 24(5), pp.854-867, 2012.
18. Sung-Hyuk Cha, "Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions", International Journal of Mathematical Models And Methods In Applied Sciences, Volume 1(4) , pp.300-307, 2007
19. Manohar M G ,"A study on similarity measure functions on engineering materials selection", *CSCP*, pp. 157–168 , 2011.
20. Deza E. and Deza M.M., "Dictionary of Distances", Elsevier, 2006
21. Krause E.F., "Taxicab Geometry An Adventure in Non-Euclidean Geometry", Machine Learning, vol.6, pp.37–66, 1995
22. Seung-Seok Choi, Sung-Hyuk Cha, and Charles C. Tappert, "A Survey of Binary Similarity and Distance Measures", Journal on Systemics, Cybernetics and Informatics, Vol. 8(1), pp. 43-48, 2010