IMPROVING CLUSTERING OF WEB BOT AND HUMAN SESSIONS BY APPLYING PRINCIPAL COMPONENT ANALYSIS

Grażyna Suchacka
Institute of Informatics
University of Opole
ul. Oleska 48
45-052 Opole, Poland
E-mail: gsuchacka@uni.opole.pl

KEYWORDS
Principal Component Analysis, PCA, Dimensionality reduction, Feature selection, *k*-means, Clustering, Classification, Internet robot, Web bot, Bot detection, Web server, Log analysis

ABSTRACT

The paper addresses the problem of modeling Web sessions of bots and legitimate users (humans) as feature vectors for their use at the input of classification models. So far many different features to discriminate bots' and humans' navigational patterns have been considered in session models but very few studies were devoted to feature selection and dimensionality reduction in the context of bot detection. We propose applying Principal Component Analysis (PCA) to develop improved session models based on predictor variables being efficient discriminants of Web bots. The proposed models are used in session clustering, whose performance is evaluated in terms of the purity of generated clusters. The efficiency of the proposed approach is experimentally verified using real server log data. Results show that PCA may be very efficient in dimensionality reduction and feature selection for session classification aiming at distinguishing Web robots.

INTRODUCTION

In the prevalent era of Big Data and cloud computing, there is an increasing demand for providing distributed computer systems with high level of security and privacy (Jakóbik 2016). This problem is especially valid on the Web, where a significant part of traffic is due to autonomous intelligent agents – Web robots (bots). Robots are programs designed to traverse the Web using hyperlinks and perform different activities – indexing Web contents for search engines, harvesting e-mail addresses, collecting business intelligence knowledge, etc. In reality, many bots disguise themselves by taking on user agents of legitimate Internet browsers, which makes their detection on the Web a challenging task.

A problem of discriminating bots from humans on the Web has gained a huge attention from the business and the scientific community in recent years. In particular, server access log data has been extensively studied to reconstruct and explore Web sessions based on historical HTTP requests. Many studies addressed characterization of bot and human sessions in terms of some statistical features and they exposed evident differences in Web traffic patterns depending on a client type (Dikaiakos et al. 2005; Suchacka 2014). This in turn has motivated research on Web bot detection, especially by applying machine learning techniques to session classification (Alam et al. 2014; Bomhardt et al. 2005; Hamidzadeh et al. 2018; Stevanovic et al. 2012; Suchacka and Sobków 2015; Zabihimayvan et al. 2017). Different session features were considered for identifying robots.

The goal of our study is to investigate possibilities of improving Web session classification in terms of differentiation of bots from humans by applying Principal Component Analysis (PCA) and to experimentally verify the proposed approach using the real e-commerce traces. We develop a basic and PCA-based session models, which are then used in session clustering with the *k*-means algorithm. The clustering performance is evaluated in terms of an ability to partition bots and humans into distinct clusters.

The remainder of this paper is structured as follows. The next Section outlines preliminaries and states the problem. Then the research methodology is discussed, including reconstruction of sessions from logs, feature extraction, session labeling, and developing PCA-based session representations for the use in classification models. The data used in experiments is briefly characterized and results of PCA and *k*-means are presented and discussed. The last Section concludes the paper and indicates possible directions of future work.

BACKGROUND AND PROBLEM STATEMENT

Web traffic coming to a server is recorded in standard access logs. Log entries include such data as IP addresses, user agents, time stamps, URIs of requested resources, HTTP methods, and response status codes. Based on log data one can reconstruct *sessions*, i.e. sequences of HTTP requests corresponding to Web clients' visits on a server. Different session features may be computed from HTTP data and used to build feature vectors as the input for classification models.

Furthermore, depending on a classification goal, various session classes may be defined. In this paper we are interested in unsupervised classification of sessions of two classes: Internet robots and legitimate users (humans). Thus, each session is represented as a pair of a feature vector (predictor variables) and a class label (1 for bots, 0 for humans). A research question is how to improve a session representation, i.e., what feature set to

Communications of the ECMS, Volume 33, Issue 1,
Proceedings, ©ECMS Mauro Iacono, Francesco Palmieri,
Marco Gribaudo, Massimo Ficco (Editors)
ISBN: 978-3-937436-65-4/978-3-937436-66-1(CD) ISSN 2522-2414

select in order to achieve efficient discrimination of robots from humans with the use of unsupervised learning (clustering) techniques.

Our preliminary analysis of server log data showed that some session features are correlated with each other. This suggests that for a given feature set Principal Component Analysis (PCA) may be successful in generating a more representative set of high-dimensional features, based on generated principal components (PCs). It has to be noted, however, that since class labels are not included in PCA, an impact of PCs on discriminating robots from humans is not obvious.

Thus, the task is to apply PCA to a relatively wide session feature set and then exploit (1) some of PCs and (2) some of features contributing most to these PCs, to create new reduced feature sets to be employed in session clustering. Our hypothesis assumes that reducing the feature set dimensionality may lead to better clustering results in terms of cluster purity, i.e., to provide a higher degree of separation of Web bots from humans.

Principal Component Analysis

Principal Component Analysis is a dimensionality reduction technique (Abbott 2014; Kassambara 2017). Its input is a data matrix with rows corresponding to samples (i.e., sessions in our case) and columns representing variables (i.e., session features), usually scaled to have the standard deviation 1 and the mean 0.

PCA identifies main directions (called *principal components*, PCs) in which the data varies; the directions with the largest variances are considered the most important. The PCA output are two other matrices with eigenvalues and eigenvectors. The *eigenvalues* represent the proportion of variance retained by each PC whereas the *eigenvectors* represent the contributions of each original variable onto each PC (loadings of the variables on the PCs). A PCA model is the set of eigenvectors, which can be saved and applied to new samples.

The PCA method is worth applying especially when variables are highly correlated, which indicates a redundancy in the data. PCA can be used to reduce the number of variables at the input of classification models by combining original variables correlated with each other into a smaller number of new variables, explaining most of the variance in the original variables. Alternatively, the method may be useful in reducing the number of input variables for classification models by leaving only these with the highest loadings on top PCs.

K-means Clustering

The *k*-means algorithm is one of the most popular clustering techniques (Abbott 2014). It aims at partitioning data in groups (clusters) providing relatively small intra-cluster distances and relatively large inter-cluster distances. For a given number of groups the algorithm starts from picking random data points as initial centroids (cluster centers). Then the algorithm iteratively performs the following steps: first it computes the distance between each centroid and each data point, thus assigning the points to the nearest clusters, and then it determines the actual centroids for newly created clusters. The procedure iterates until the cluster membership is stable. The algorithm is usually run multiple times with different random initial centroids.

The most common distance measure in *k*-means is Euclidean distance. The clustering quality may be evaluated with different measures, e.g., the sum of squared error (SSE) or supervised-oriented metrics, like clustering entropy or purity.

METHODOLOGY

Our approach involves the following stages (Fig. 1):

1) Transforming row log data into Web sessions; describing sessions with vectors of features (predictor variables) and class labels (1 for robots, 0 for humans).
2) Application of PCA to generate representative vectors of PCs from original features; selecting subsets of original features to build reduced vectors of features (the ones with the largest contribution to the top PCs).
3) Application of *k*-means for three independent experimental scenarios: (I) to cluster original feature vectors, (II) to cluster PC vectors, and (III) to cluster reduced feature vectors.
4) Measuring clustering results and their comparison between the three scenarios.

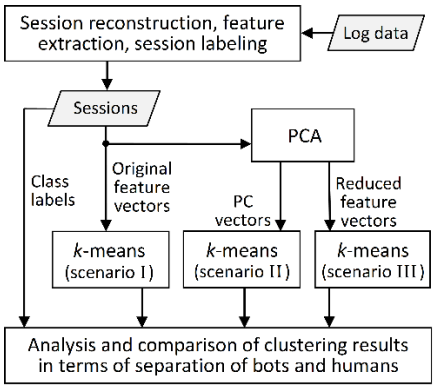


Figure 1: Flowchart of the Proposed Approach

Session Preparation

The first stage is realized by using our dedicated C++ log analyzer. The program reads, cleans, and pre-processes log entries, assemblies HTTP request records and uses them to reconstruct Web sessions. This process is based on a common assumption that an individual Web client is represented by an IP address and a user agent field combined whereas successive sessions of a client are separated by a minimum 30-minute time interval. The analyzer also extracts features and assigns class labels to the reconstructed sessions.

Session Features

Each session is described with 40 features, selected based on the analysis of previous bot detection studies (Alam et

al. 2014; Bomhardt et al. 2005; Hamidzadeh et al. 2018; Stevanovic et al. 2012; Stassopoulou and Dikaiakos 2009; Tan and Kumar 2002; Zabihimayvan et al. 2017).
The first group of features refers to temporal access patterns in session and includes:

- session duration in seconds [s]: feature *duration*;
- the average and standard deviation of time between subsequent requests [s] with regard to all HTTP requests, page requests, and embedded requests: *intervReqAvg*, *intervReqSd*, *intervPagAvg*, *intervPagSd*, *intervEmbAvg*, *intervEmbSd*;
- maximum sustained click rate, measured as the maximum number of page requests within a sliding window of 12, 6, and 3 seconds: *msClickRate12*, *msClickRate6*, *msClickRate3*;
- percentage of requests made during the night hours (12 a.m. – 7 a.m.): *%night*.

The second group of features is related to resource request patterns in session:

- total numbers of HTTP requests and page requests: *totalReq*, *totalPag*;
- percentages of requests of specific type (page, image, binary program file, compressed file): *%pag*, *%img*, *%binExe*, *%zip*;
- ratios of image to page requests and embedded to page requests: *imgToPag*, *embToPag*;
- maximum number of embedded requests per page: *maxEmbBar*;
- switching factor of file types: *swiFactTyp*;
- statistics for repeated requests, including average and maximum numbers of requests per file, and percentage of repeated requests: *avgReqPerURI*, *maxReqPerURI*, *%repeatedURI*;
- occurance of a request for robots.txt file in session (a Boolean variable): *robots.txt*;
- statistics for data volumes transferred to the client, [KB], including the average, standard deviation, and total transfer volume: *volAvg*, *volSd*, *volTotal*;
- percentages of requests with specific HTTP method (GET, HEAD, and POST): *%GET*, *%HEAD*, *%POST*.

The third group includes features connected with HTTP response status, such as the successful request (2xx), redirection – request moved permanently (301), moved temporarily (302), or not modified (304), client error (4xx), and server error (5xx): *%2xx*, *%301*, *%302*, *%304*, *%4xx*, *%5xx*.
Remaining features are related to request referrers: percentages of HTTP requests and page requests with empty referrer fields and a switching factor on empty referrer: *%empRefReq*, *%empRefPag*, *swiFactEmpRef*.
Our *basic* session model includes all 40 features. They are used as predictor variables at the input of the PCA algorithm (and the *k*-means algorithm for scenario I).

Session Labels
Each session receives a class label: 0 or 1. Labels are not used in PCA nor *k*-means but they are needed to assess the clustering performance. Our session labeling procedure, proposed in (Suchacka and Motyka 2018), relies on querying well-recognized online databases of IPs and user agents known to correspond to robots or Web browsers. It additionally examines keywords suggesting a bot in user agent strings and implements some heuristic rules identifying session patterns untypical for humans.

Dimensionality Reduction and Feature Selection with PCA

We propose to apply PCA to the set of sessions described by 40 features in order to generate multidimensional features (PCs) and to identify these of original features which contribute most to the top PCs. Our goal is twofold:

- identifying a subset of top principal components and using them as new, information-rich session features;
- selecting a subset of the most significant original session features.

Based on PCA results, we propose two improved session representations and experimentally verify their efficiency in session clustering:

1) In the *PC-based model* a session is represented as a vector of selected top PCs so the PCs themselves are used as new input variables to the clustering algorithm (experimental scenario II);
2) In the *feature subset model* a session is represented by some subset of original variables – the top loading variables for selected top PCs are used as input variables to the clustering algorithm (experimental scenario III).

Session Clustering with *k*-means

Clustering of sessions is performed with *k*-means independently for the three scenarios:

- scenario I: clustering of original feature vectors, 40 variables each;
- scenario II: clustering of PC vectors; number of vector elements vary depending on the number of dimensions (1-5) under consideration;
- scenario III: clustering of reduced original feature vectors; number of variables in vectors vary depending on the number of dimensions (1-5).

Experimental Setup

Experiments are performed using R-project, a free software environment for statistical computing (R Project) with FactoMineR (FactoMineR) and factoextra (Factoextra) R packages used for PCA.
An original data matrix is pre-processed at the PCA input: Boolean values are replaced with 1 and 0, missing values are replaced with an average for the given column, data are centered and scaled.
The *k*-means algorithm employs Euclidean distance and *k* varies from 1 to 10. For each *k* the algorithm is run 200 times with different random seeds and finally mean performance scores are computed and reported.
Clustering performance is evaluated in terms of the ability to partition bots and humans into separate clusters.

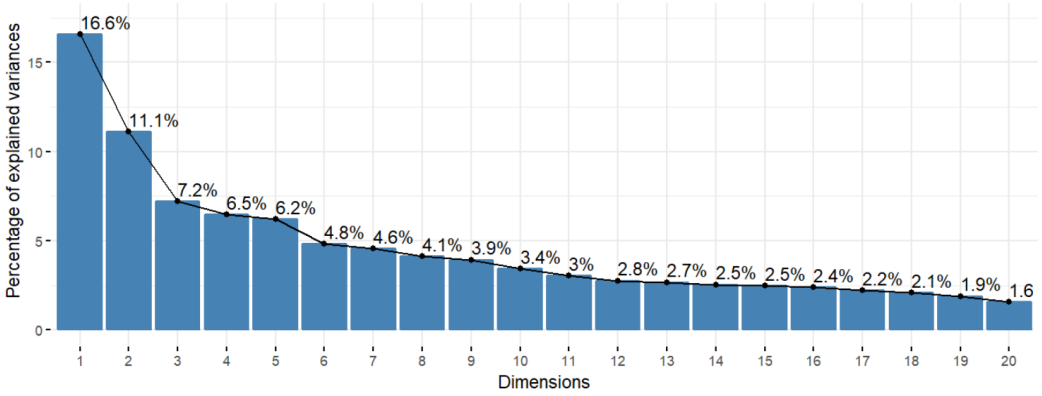


Figure 2: Scree Plot for the First 20 Dimensions

This is measured with the *purity* metric (Tan et al. 2006), expressing an extent to which clusters contain sessions of a single class. The purity of cluster *i* is:

$$p_i = \max_j p_{ij}, \quad (1)$$

where $p_{ij} = m_{ij}/m_i$, where m_{ij} is the number of sessions of class *j* in cluster *i* and m_i is the number of all sessions in cluster *i*. The overall clustering purity is:

$$purity = \sum_{i=1}^{k} \frac{m_i}{m} p_i, \quad (2)$$

where *k* is the number of clusters.

In real Web traffic the classes are imbalanced so we apply data under-sampling by randomly drawing the majority class sessions into the final dataset. A balanced session set allows us to intuitively interpret the purity measure. In the case of two classes, purity of 0.5 means that each cluster contains 50% of sessions of each class while purity of 1 means the ideal separation between classes.

RESULTS AND DISCUSSION

Dataset Description

Log data used in the analysis are for an online store offering car parts and accessories. Recorded traffic included 773 874 HTTP requests received from 13th October to 20th December, 2015. Among 8 408 sessions 71.1% were driven by bots. After data under-sampling the final dataset included 4 856 sessions with equal proportions of robots and humans.

PCA Results

Since in the basic session model each sample is represented with 40 variables, PCA generated 40 principal components. Table 1 shows eigenvalues for 15 top PCs. The eigenvalues express variation of data retained by individual PCs, which is the highest for the first PCs and smaller for the subsequent ones. The eigenvalues are helpful in determining the number of principal components to be kept in PC-based models. A popular rule-of-thumb to do this for standardized data is keeping PCs with eigenvalues > 1. One can notice in Table 1 that only 14 out of 40 PCs have eigenvalues greater than one. The first five principal components are much more significant than others – they together explain nearly half of total information contained in the data.

Another way of determining the number of final PCs is to analyze a scree plot, presenting eigenvalue variances ordered from the largest to the smallest. A scree plot for the first twenty dimensions is shown in Fig. 2. It reveals upcasts after the first PC (from 16.6% to 11.1%), after the second one (from 11.1% to 7.2%) and after the next three PCs (from 6.2% to 4.8%).

Thus, five top principal components (dimensions) were selected to develop new session models based on PCs.

Table 1: First 15 Components from the PCA

Dim.	Eigenvalue	Variance [%]	Cumulative variance [%]
1	6.63	16.58	16.58
2	4.45	11.12	27.70
3	2.89	7.21	34.91
4	2.58	6.46	41.37
5	2.48	6.19	47.56
6	1.93	4.82	52.38
7	1.83	4.57	56.94
8	1.64	4.11	61.05
9	1.57	3.93	64.98
10	1.37	3.42	68.40
11	1.22	3.04	71.44
12	1.10	2.76	74.20
13	1.07	2.67	76.88
14	1.02	2.54	79.42
15	0.99	2.47	81.89

PC-based Session Model

Five PC-based vector representations of sessions were developed and analyzed in the experimental scenario II, for dimensions 1 to 5. In the first model (*PC-based/1 dim*) only one dimension is taken into account so a feature vector representing a session contains only one element – the first principal component, PC1. The second model (*PC-based/2 dim*) is based on two dimensions so a feature vector of each session contains two top principal components, PC1 and PC2. Each subsequent model is augmented with one additional dimension, up to the fifth model (*PC-based/5 dim*) with a five-PC vector.

Table 2: Eigenvectors of Top Five PCs

Variable	PC1	PC2	PC3	PC4	PC5
duration	-0.15	0.13	0.08	0.15	0.08
intervReqAvg	**-0.27**	-0.13	0.10	-0.16	0.02
intervPagAvg	-0.22	-0.15	0.13	-0.18	0.04
intervEmbAvg	-0.13	0.04	0.00	0.14	-0.02
intervReqSd	**-0.30**	-0.09	0.07	-0.08	-0.03
intervPagSd	-0.19	-0.15	0.10	-0.13	0.05
intervEmbSd	-0.08	0.05	-0.03	0.08	-0.04
msClickRate12	-0.01	**0.35**	-0.08	0.15	-0.26
msClickRate6	-0.02	**0.34**	-0.09	0.16	**-0.29**
msClickRate3	-0.01	**0.31**	-0.10	0.14	**-0.30**
%night	-0.04	0.02	-0.02	0.11	0.01
totalReq	-0.02	**0.36**	0.22	-0.08	**0.29**
totalPag	-0.05	**0.35**	0.17	-0.08	**0.33**
%pag	**-0.36**	0.03	0.00	0.01	-0.07
%img	0.26	0.03	0.10	-0.03	-0.15
%binExe	**0.29**	-0.06	-0.03	0.00	0.19
%zip	0.21	-0.05	-0.12	0.03	0.25
imgToPag	0.11	0.00	**0.47**	-0.05	-0.25
embToPag	0.14	-0.01	**0.46**	-0.05	-0.22
maxEmbBar	0.15	0.01	**0.43**	-0.04	-0.22
swiFactTyp	0.15	0.00	-0.15	0.10	0.14
avgReqPerURI	-0.01	**0.29**	0.05	-0.24	0.15
maxReqPerURI	-0.01	**0.29**	0.05	-0.24	0.15
%repeatedURI	0.05	0.15	-0.17	-0.19	-0.18
robots.txt	-0.10	0.10	-0.05	**0.27**	0.04
volTotal	-0.04	0.24	0.17	0.05	**0.30**
volAvg	-0.16	0.02	0.03	0.13	-0.04
volSd	-0.02	0.03	0.06	0.08	-0.04
%GET	-0.01	-0.14	0.16	**0.35**	0.07
%HEAD	0.00	0.02	-0.03	-0.04	-0.04
%POST	0.01	0.14	-0.15	**-0.35**	-0.07
%2xx	0.05	0.03	0.13	**0.37**	0.15
%301	-0.08	-0.01	-0.02	-0.02	-0.06
%302	-0.03	0.05	-0.13	-0.10	-0.13
%304	0.04	0.01	-0.07	-0.11	-0.07
%4xx	0.14	0.05	-0.15	-0.17	-0.10
%5xx	-0.11	-0.08	0.01	**-0.27**	-0.02
%empRefReq	**-0.34**	0.03	-0.01	0.04	-0.05
%empRefPag	**-0.33**	0.03	0.01	0.06	-0.04
swiFactEmpRef	0.02	0.02	-0.05	0.01	-0.02

Original variables contribute to individual PCs to different degrees, expressed by eigenvectors. Table 2 displays the eigenvector values (loadings) for top five PCs, with the highest contributions for each PC stressed in boldface.

One can see that each PC is dominated by different variables, e.g., almost all of the variables contribute to the first PC but to very different degrees, with the following top loaders: *intervReqAvg*, *intervReqSd*, *%pag*, *%binExe*, *%empRefReq*, and *%empRefPag*. Similar levels of their loadings onto PC1 indicates that these variables are correlated with each other.

Plots in Fig. 3 visualize contributions of all variables to top five PCs. For a given PC the most significant variables have contribution values exceeding the expected average contribution (marked with a red dashed line). These variables were selected as candidates for the reduced feature vectors representing sessions in experimental scenario III.

Feature Subset Session Model

We developed and analyzed five session representations based on reduced feature vectors, starting from taking only one dimension into account and augmenting the subsequent models by one additional dimension, up to the fifth model, which is based on all five dimensions.

In the first model (*Feature subset/1 dim*) each session is represented by a 11-element vector, including values for the features with the largest contributions to PC1 (separated from other features with a vertical line in Fig. 3a): *%pag*, *%empRefReq*, *%empRefPag*, *intervReqSd*, *%binExe*, *intervReqAvg*, *%img*, *intervPagAvg*, *%zip*, *intervPagSd*, and *volAvg*. Similarly, vector representations in models *Feature subset/2 dim*, *Feature subset/3 dim*, *Feature subset/4 dim*, and *Feature subset/5 dim* contain 18, 20, 22, and 23 elements, respectively – variables most contributing to dimensions 1-2, 1-3, 1-4, and 1-5, respectively (cf. Fig. 3).

K-means Results

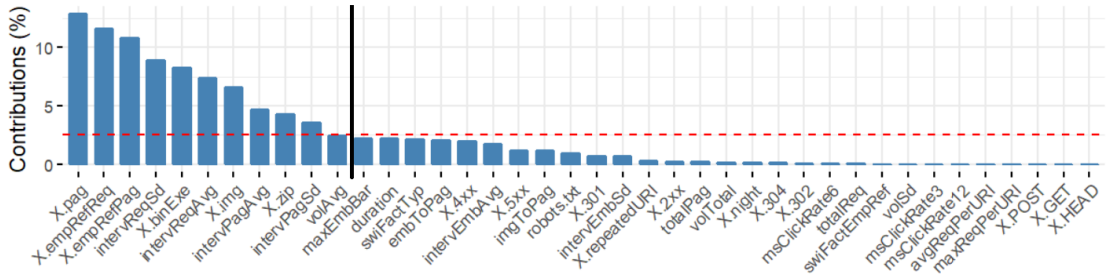
The analysis of clustering results reveals that the basic session model, consisting of 40 features, allows the *k*-means algorithm to efficiently separate bots from humans to a large extent (see a black line in Fig. 4a, 4b). The resulting purity ranges from 0.86 for $k = 2$ to 0.94 for $k = 10$. So even a very coarse division of sessions into two groups gave the purity of 0.86, which is a promising result, confirming a good selection of predictor variables. Increasing the number of clusters to three results in purity exceeding 0.93 and the further increase in k affects the increase in purity only slightly.

Fig. 4a shows that for two clusters the PC-based session models for all five dimensions are better than the basic model; however, for $k > 4$ the advantage is kept only by the PC models based on the first and two first dimensions. In particular, representing a session with one top principal component leads to the highest clustering purity, approaching the level of 0.95 even for only two groups.

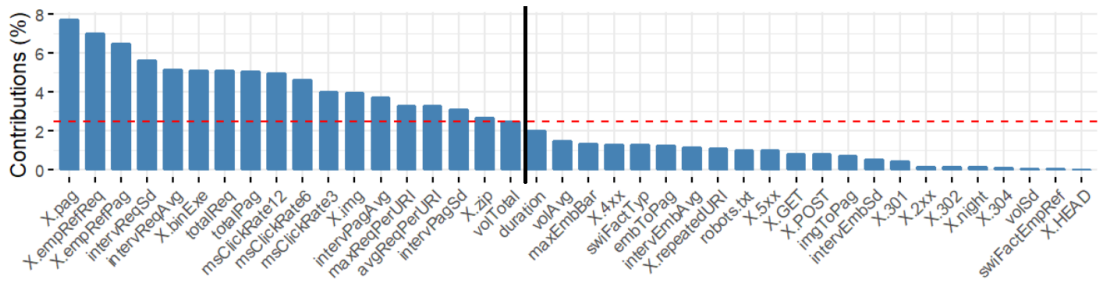
Fig. 4b shows the similar superiority of session models based on subsets of original features. In this case clustering results for models based on dimensions 3 – 4 are even better than for the corresponding PC models. The feature subset model based on one dimension, representing a session with 11 most significant features (*Feature subset/1 dim*) leads to the highest purity, ranging from 0.93 for $k = 2$ to 0.96 for $k = 10$.

Fig. 4 confirms that taking into consideration only the first principal component and top session features contributing to it gives the highest degree of separation of Web robots and legitimate users, regardless of the number of clusters.

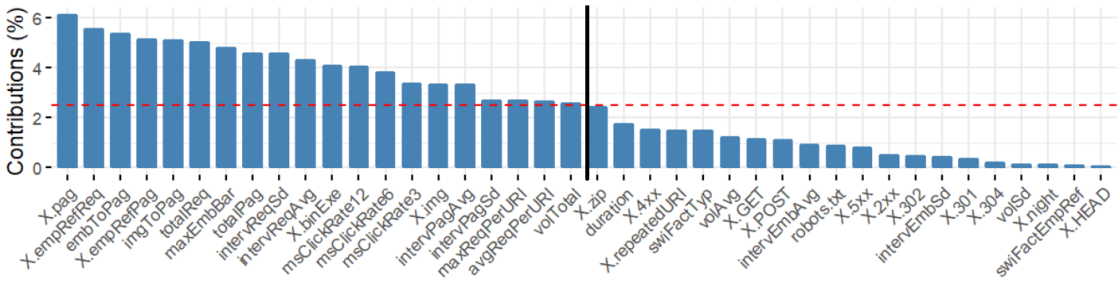
Regarding a possible implementation of the approach on a real Web server, it has to be noted that although the dimensionality of improved session models is reduced, still all the original features are needed to generate the PCs. However, since principal components are linear projections of original variables, the PCA algorithm is computationally fast even for a large number of input variables.



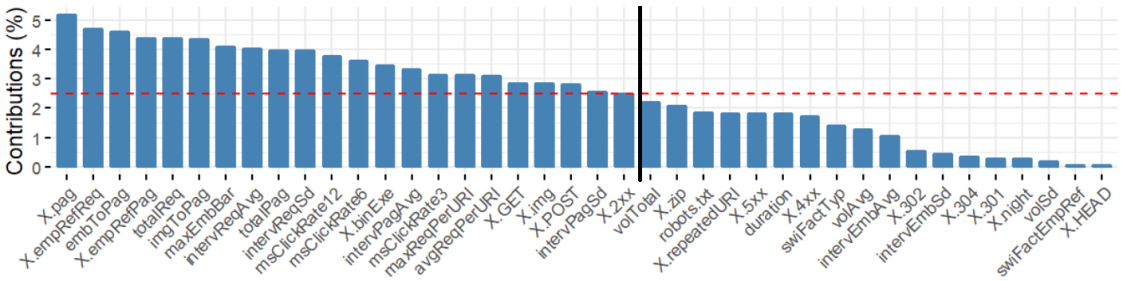
(a) Contribution of Variables to Dim 1



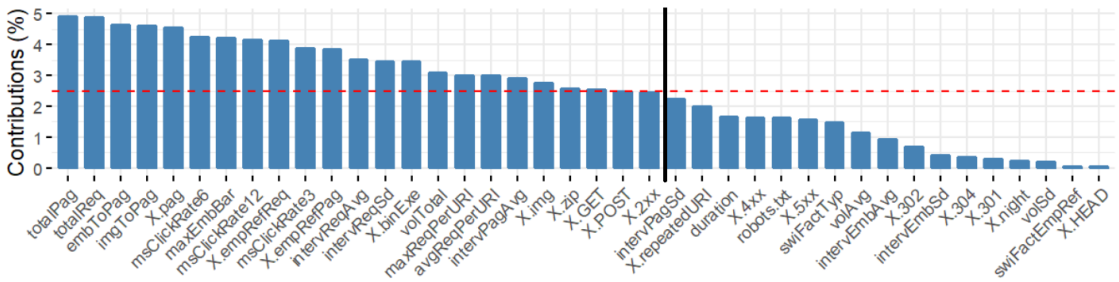
(b) Contribution of Variables to Dim 1-2



(c) Contribution of Variables to Dim 1-2-3

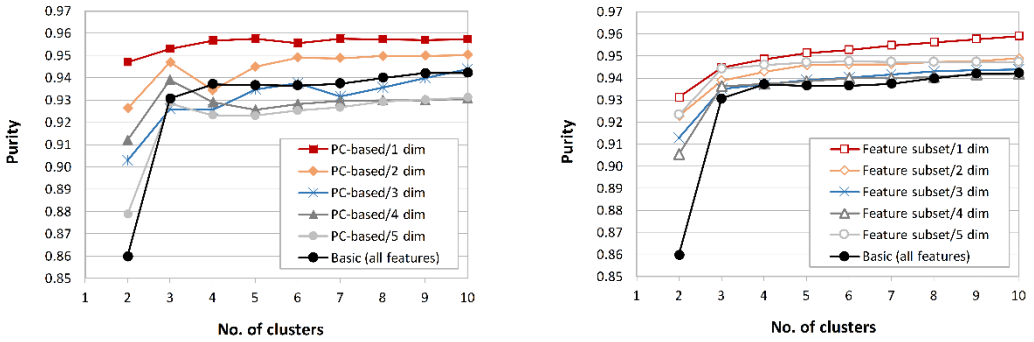


(d) Contribution of Variables to Dim 1-2-3-4



(e) Contribution of Variables to Dim 1-2-3-4-5

Figure 3: Contributions of Original Session Features to Top Five Dimensions



(a) (b)

Figure 4: Total Purity of Clusters Generated with *k*-means for Sessions Described with Original Feature Vectors (*Basic - all features*), as well as with (**a**) PC Vectors and (**b**) Reduced Feature Vectors

CONCLUSIONS

In this paper we proposed a new PCA-based approach to dimensionality reduction and feature selection for representing Web sessions of bots and humans for the use in classification tasks. Three types of session models were proposed: a basic, 40-feature model, a principal component-based model, and a reduced feature set model. The efficiency of the approach was experimentally evaluated by clustering sessions with the *k*-means algorithm.

The clustering results show the superiority of the proposed session models, especially the ones based on the first principal component (one PCA dimension), over the basic model in terms of the overall clustering purity. This suggests that using outcomes of Principal Component Analysis to build multidimensional and reduced session representations may be beneficial for discriminating Web bots from legitimate users.

Our methodology may be used to develop more effective bot detection methods. In future work we are going to verify the proposed approach for other server log data in order to generalize the results. We are also planning to integrate it with session classifiers, both based on supervised and unsupervised learning, and to verify the classification performance on a test dataset.

REFERENCES

Abbott, D. 2014. *Applied predictive analytics*. Wiley, Indianapolis, IN, USA.

Alam, S.; G. Dobbie; Y.S. Koh; and P. Riddle. 2014. "Web bots detection using particle swarm optimization based clustering," in *Proc. of IEEE CEC'14*. IEEE, 2955-2962.

Bomhardt, C.; W. Gaul; and L. Schmidt-Thieme. 2005. "Web robot detection – preprocessing Web logfiles for robot detection," in New Developments in Classification and Data Analysis. Berlin, Heidelberg: Springer, 113-124.

Dikaiakos, M.D.; A. Stassopoulou; and L. Papageorgiou. 2005. "An investigation of Web crawler behavior: Characterization and metrics," *Comput. Commun.* 28(8), 880-897.

Factoextra: Extract and visualize the results of multivariate data analyses, https://cran.r-project.org/web/packages/factoextra /index.html

FactoMineR: Multivariate exploratory data analysis and data mining, https://cran.r-project.org/web/packages/FactoMineR/ index.html.

Hamidzadeh, J.; M. Zabihimayvan; and R. Sadeghi. 2018. "Detection of Web site visitors based on fuzzy rough sets," *Soft Comput.* 22(7), 2175-2188.

Jakóbik, A.: Big data security. 2016. *Resource Management for Big Data Platforms*, Springer, Chapter 8, 241-261.

Kassambara, A. 2017. *Practical guide to principal component methods in R*. STHDA.

R Project, *The R project for statistical computing*, http://www.r-project.org.

Stassopoulou, A. and M.D. Dikaiakos. 2009. "Web robot detection: A probabilistic reasoning approach." *Comput. Netw.* 53(3), 265-278.

Stevanovic, D.; A. An; and N. Vlajic. 2012. "Feature evaluation for Web crawler detection with data mining techniques," *Expert Syst. Appl.* 39(10), 8707-8717.

Suchacka, G. 2014. "Analysis of aggregated bot and human traffic on e-commerce site". *FedCSIS'14*, 1123-1130.

Suchacka, G. and I. Motyka. 2018. "Efficiency analysis of resource request patterns in classification of Web robots and humans". In *Proc. of ECMS'18*, 475-481.

Suchacka, G. and M. Sobków. 2015. "Detection of Internet robots using a Bayesian approach," In *Proc. of IEEE 2nd Int. Conf. on Cybernetics*, 365-370.

Tan, P.-N. and V. Kumar. 2002. "Discovery of Web robot sessions based on their navigational patterns," *Data Min. Knowl. Discov.* 6(1), 9-35.

Tan, P.-N.; M. Steinbach; and V. Kumar. 2006. *Introduction to data mining*. Pearson Addison-Wesley, Boston, MA.

Zabihimayvan, M.; R. Sadeghi; H.N. Rude; and D. Doran. 2017. "A soft computing approach for benign and malicious Web robot detection," *Expert Syst. Appl.* 87, 129-140.

AUTHOR BIOGRAPHIES

GRAŻYNA SUCHACKA received the M.Sc. degrees in Computer Science and in Management, as well as the Ph.D. degree in Computer Science (with distinction) from Wroclaw University of Science and Technology, Poland. Now she is an assistant professor in the Institute of Informatics at the University of Opole, Poland. Her research interests include data analysis and modeling, data mining, and Quality of Web Service with special regard to bot detection and electronic commerce support. Her e-mail address is: `gsuchacka@uni.opole.pl`.