International Journal of Computational Intelligence Systems, Vol. 4, No. 5 (September, 2011), 1062-1069

Identifying Web Users on the Base of their Browsing Patterns

Wojciech Jaworski

Institute of Informatics, University of Warsaw,
Banacha 2, 02-097 Warsaw, Poland
wjaworski@mimuw.edu.pl

Abstract

Our aim is to develop methodology for recognition of Internet portal users on the base of their browsing patterns. This is a classification task in which we have thousands values of decision attribute and objects are described by means of sequences of symbols. We develop feature selectors which make our task tractable. Since the behaviour usually does not distinguish users, we introduce user profiles which are clusters of indiscernible users. Then, we construct classifiers which assign descriptions of user behaviour to user profiles. We also derive quality measures specific for our task.

Keywords: Web traffic analysis, Web usage mining, Browsing behaviour, sequential data classification, quality measures, indiscernibility.

1. Introduction

Web traffic analysis is an area of still growing importance. The ability of determining which pages were displayed by the same user is crucial for this domain. Users are usually identified by the cookies technology. However, this method fails in many cases: user may delete cookie, use more than one computer etc. The only trace that allow us to identify user in such case is his behaviour - the similarity between sequences of Internet pages visited by user which we try to recognize with behaviour of known users.

The problem of user browsing pattern analysis and related to it problem of web log analysis have been broadly studied during last two decades. Early tools for World Wide Web usage mining and analysis were developed in late nineties (for ex. [1]). They applied data mining algorithm in order to discover the most common traversal paths and groups of pages frequently visited together.

Subsequent systems introduced more sophisticated methodology to learn user's browsing patterns. For example [2] uses two types of learning classifier systems: a content-based classifier system for contents change patterns and an action-based classifier system for user's action patterns, while [3] identify and categorize users behavioural patterns to show how certain types of patterns could indicate navigational problems in a website and introduce a graphical representation of browsing patterns.

The task of web users clustering, which is related to the task of web users identification, was dealt with in various contexts. User profiles built using an unsupervised Web page clustering algorithms were used to characterize user ongoing activities and behaviour patterns[4] and to discover knowledge from users webpage navigation[5]. The other worth mentioning applications are dynamic text hyperlinking[6] and context-aware recommendation[7]. The most widespread methodologies are ontology-based profiling[8] or conceptual clustering[9] On the

other hand[10] introduced a generalization-based clustering method was employed which first generalized the sessions and then applied a hierarchical clustering algorithm to find clusters in the generalized sessions.

Recently a large-scale study of online user behaviour was undertook[11], which examined the extent to which pages of certain types are revisited by the same user over time, and the mechanisms by which users move from page to page, within and across hosts, and within and across page types.

In [13] the concept of support-interest was introduced, which is based on the assumption that visitor will backtrack if he do not find the information where they expect.

User behaviour patterns can be used in E-commerce for the purposes of marketing strategies and product offerings, mass customization and personalization, and Web site adaptation[14]. Browsing patterns were used also for derivation of web page semantics: [15] derived semantic breakpoints of long browsing paths, using an iterative process.

2. Problem statement

In this paper we analyze browsing patterns in the context of the following case study: we are given an Internet portal that provides web pages. Each of them belongs to one of 20 thematic categories denoted by $A, B, \ldots, T$.* User activity is reported as a sequence of category identifiers assigned to pages displayed by user. In order to compress the data, subsequent occurrences of the same category are represented as a pair which is composed out of category id and number of occurrences. Formally we denote such sequence as $\{c_i, m_i\}_{i=1}^{n}$, where each $c_i \in \mathscr{C}$ is a category id and m_i is a number of its occurrences. We denote the set of all sequences by $\mathscr{S}$.

We are given two tables. Each of them consists of two fields: user id and user activity. First table describe users' activity during first three weeks and the second one describe the activity during the fourth week. Our goal is to recognize user id on the basis of his activity in the fourth week, having given activities of all users during the first three weeks labeled by user ids.

This is a classification task in which we have 4882 values of decision attribute (there are 4882 users) and objects are described by means of sequences.

However, the behaviour does not distinguish users, rather split them into clusters of indiscernible objects. As a consequence, the classifier should indicate a set of users indiscernible with the one whose description is given as a classifier input, rather than point a single user.

Formally, our task is described as follows: we are given a set of users U, a function $g : U \to \mathscr{S}$ that describe user behaviour during the first three weeks. We are also given a multiset V of sequences generated by users during the fourth week. Our goal is to generate function $f : V \to P(U)$ that will be a classifier which indicates users indiscernible with the one who generated classified sequence.

For the purposes of classifier validation, we use a function $h : U \to \mathscr{S}$ which describe user behaviour during the fourth week. h is the function approximated by classifier.

3. Data description and preprocessing

We analyze web traffic data from Polish web sites employing *gemiusTraffic* study. Data was acquired by Gemius company through the use of scripts, placed in code of the monitored web page. The scripts reported to the *gemiusTraffic* platform each Page View - event of displaying the monitored web page. Page Views were grouped into Visits - uninterrupted series of Page Views on a given web site executed by the same user.

This dataset was used as a subject of task in ECML/PKDD'2007 Discovery Challenge[16]. However the goal of that task was different that the one we pursue here. The problem objective was to predict user behaviour, while we are recognizing users on the basis of their behaviour.

The original dataset was composed out of two tables. First of them consisted of information about

*In the original dataset categories were denoted by means numbers from 1 to 20. However for the sake of clarity we mapped numeric ids into letters.

users: their country, operating system, browser, etc. This information is irrelevant for our task, so we did not use the first table.

Second table described users' activity in terms of Visits. It consisted of the following fields: user id, timestamp and sequences of categories of pages displayed during the Visit. For each user we have merged subsequent Visits into one sequence of Page Views. This operation significantly simplified the structure of data. According to our tests, the information about how Page Views are grouped into Visits is not helpful in our task.

4. Quality measures

Before we construct a classifier we must define measures of its quality. Our task is different from a typical classification problem, because of the large number of decision attribute values. That is why traditional quality measures[17] developed for classification tasks (such as accuracy) do not suit to our task.

The classifier indicates a set of users indiscernible with the one whose description is given as its input. Hence the quality measures should compare the set of users indicated by the classifier with the set of users indiscernible with a given one (the latter set we denote also as a set of relevant users).

For a single set of relevant users our task is similar to the problem of information retrieval: searching for documents relevant for a given query. In this case *precision* and *recall* are adequate quality measures. They can be seen as a measures of classifier exactness and completeness.

We define *precision* as a number of relevant users indicated by a classifier divided by the total number of users indicated by a classifier, and we define *recall* as the number of relevant users indicated by a classifier divided by the total number of relevant users.

Formally, let $[u]$ denote a set of users indiscernible with $u \in U$ (equivalence class of the indiscernibility relation). For a given classifier f and user $u \in U$ we define precision and recall as follows:

$$\mathbf{Precision}(f,u) = \frac{|[u] \cap f(h(u))|}{|f(h(u))|},$$

$$\mathbf{Recall}(f,u) = \frac{|[u] \cap f(h(u))|}{|[u]|},$$

where $h(u)$ is a sequence generated by user u and $f(h(u))$ is a value calculated by classifier for this sequence.

Now, we extend these measures on the sets of users. Since precision and recall are identical for indiscernible objects it is natural to state that

$$\mathbf{Precision}(f,[u]) = \mathbf{Precision}(f,u),$$

$$\mathbf{Recall}(f,[u]) = \mathbf{Recall}(f,u).$$

Precision and recall of sets that are sums of indiscernibility classes are weighted means of their values calculated separately for each indiscernibility class, i.e.,

$$\mathbf{Recall}(f,X) = \frac{1}{|X|}\sum_{k=1}^{n} |[u_k]|\mathbf{Recall}(f,[u_k]),$$

$$\mathbf{Precision}(f,X) = \frac{1}{|X|}\sum_{k=1}^{n} |[u_k]|\mathbf{Precision}(f,[u_k]),$$

where $X = [u_1] \cup [u_2] \cup \cdots \cup [u_n]$.

On the other hand, values of precision and recall for sets of users should be an arithmetic mean of their values calculated for each user:

$$\mathbf{Precision}(f,X) = \frac{1}{|X|}\sum_{u \in X} \mathbf{Precision}(f,u).$$

$$\mathbf{Recall}(f,X) = \frac{1}{|X|}\sum_{u \in X} \mathbf{Recall}(f,u).$$

Now, we show that the above definitions are equivalent. Let $X \subset U$ be such that $X = [u_1] \cup [u_2] \cup \cdots \cup [u_n]$.

$$\begin{aligned}
\mathbf{Recall}(f,X) &= \frac{1}{|X|}\sum_{u \in X} \mathbf{Recall}(f,u) \\
&= \frac{\sum_{k=1}^{n} |[u_k]|\mathbf{Recall}(f,u_k)}{\sum_{k=1}^{n} |[u_k]|} \\
&= \frac{\sum_{k=1}^{n} |[u_k]|\mathbf{Recall}(f,[u_k])}{\sum_{k=1}^{n} |[u_k]|} \\
&= \frac{\sum_{k=1}^{n} |[u_k] \cap f(h(u_k))|}{\sum_{k=1}^{n} |[u_k]|}.
\end{aligned}$$

Analogically

$$\begin{aligned}\textbf{Precision}(f,X) &= \frac{\sum_{k=1}^{n}|[u_k]|\textbf{Precision}(f,[u_k])}{\sum_{k=1}^{n}|[u_k]|} \\ &= \frac{\sum_{k=1}^{n}|[u_k]|\frac{|[u_k]\cap f(h(u_k))|}{|f(h(u_k))|}}{\sum_{k=1}^{n}|[u_k]|}.\end{aligned}$$

Precision and recall may be combined into a single measure called *F-score* which is their harmonic mean:

$$F(f,X) = 2\cdot\frac{\textbf{Precision}(f,X)\cdot\textbf{Recall}(f,X)}{\textbf{Precision}(f,X)+\textbf{Recall}(f,X)}.$$

Apart from the above measures we will consider also *average cluster size* defined as

$$\begin{aligned}\textbf{ClusterSize}(f,X) &= \frac{1}{|X|}\sum_{u\in X}|f(h(u))| \\ &= \frac{\sum_{k=1}^{n}|[u_k]|\cdot|f(h(u_k))|}{\sum_{k=1}^{n}|[u_k]|},\end{aligned}$$

where $X\subset U$ is such that $X=[u_1]\cup[u_2]\cup\cdots\cup[u_n]$.

We assume that recall is the most important quality measure: high recall guarantee that the relevant user will be among users indicated by classifier. The second important measure is the cluster size: it tells us how many users will be on average indicated by classifier.

5. Feature selection

Users cannot be classified by classifier that simply receive sequences of Page Views as input data and tries to find dependence between user id and the sequence. The reason is that such classifier would have an infinite Vapnik–Chervonenkis dimension[18], so it would not be learnable[19].

In order to construct learnable classifier, we had to select relevant features from the original data. We accomplished this goal adaptively guessing how relevant information is encoded in data: we prepared and tested several features and, on the basis of tests results, we developed feature selectors described in this and the following sections.

We assume that user preferences did not change during four weeks covered by data. The reason for this assumption lies in the fact that four weeks is a very short period, so there is a small probability that user will change his preferences. On the other hand users behave pretty randomly so it is very hard to distinguish preference change from random noise.

As a consequence, categories that appear in sequence generated by a given user during first three weeks should be identical to those which he generated during fourth week. However, some rare categories may appear only in one sequence as a statistical noise. In order to eliminate such noise we introduce thresholds: we eliminate from sequence all categories that were viewed less then a given number of times.

For a given threshold t function $a_t:\mathscr{S}\to P(\mathscr{C})$ selects features as follows:

$$a_t\left(\{c_i,m_i\}_{i=1}^{n}\right)=\{c:\sum_{i:c_i=c}m_i>t\}\cup\{\arg\max_{c}\sum_{i:c_i=c}m_i\}.$$

a_t generates a set of categories, such that user displayed more than t pages belonging to each of them. In order to avoid users described by empty feature set, the most frequent category is always added to the set.

Table 1. The most commons user profiles generated with threshold $t=30$.

1176	*Q*	27	*F*	11	*ILQ*	5	*KLQ*
834	*GQ*	27	*CGQ*	11	*HPQ*	5	*A*
676	*G*	25	*FQ*	11	*GLNQ*	4	*S*
186	*LQ*	23	*CQ*	11	*FG*	4	*QR*
157	*GPQ*	22	*GLPQ*	10	*LPQ*	4	*LNQ*
116	*GN*	19	*GNPQ*	10	*FGQ*	4	*IL*
108	*L*	17	*GOQ*	10	*EHQ*	4	*GM*
99	*PQ*	17	*EQ*	10	*EGQ*	4	*GHNQ*
89	*GLQ*	15	*MQ*	9	*KQ*	4	*EGIQ*
87	*HQ*	15	*GKQ*	9	*GK*	4	*EGHQ*
80	*GNQ*	15	*CG*	9	*E*	4	*EG*
73	*IQ*	14	*NQ*	8	*K*	4	*DGQ*
69	*I*	14	*N*	7	*HLQ*	4	*CPQ*
61	*GP*	14	*GH*	7	*GO*	4	*CGLQ*
58	*P*	14	*C*	7	*GMQ*	3	*R*
52	*GIQ*	13	*GNP*	7	*GILQ*	3	*M*
45	*GHQ*	13	*CGPQ*	6	*GIPQ*	3	*KPQ*
40	*H*	12	*OQ*	6	*GHIQ*	3	*JQ*
36	*GL*	12	*GHPQ*	6	*FGPQ*	3	*IP*
29	*GI*	11	*IPQ*	5	*O*	3	*HNQ*

Function a_t generates indiscernibility classes, which may be interpreted as user profiles. The most common user profiles, generated with threshold $t=30$, are presented in Table 1. Size of profiles

decreases exponentially, we have a few large clusters and many small ones: there are 245 profiles total. The number of categories in profiles is small, typical user visits pages belonging to 1, 2 or 3 categories.

We tested also alternative feature selection function $b_t : \mathscr{S} \to P(\mathscr{C})$

$$b_t\big(\{c_i, m_i\}_{i=1}^n\big) = \{c : \sum_{i:c_i=c} m_i > t \cdot \sum_{i=1}^{n} m_i\}.$$

It selects categories which are present in at least t percent of pages displayed by user.

Our feature selectors seek for information concerning the thematic categories in which user is interested. That is why, the order of visited pages is irrelevant and the number of pages belonging to a given category is important only in terms of being satisfying required threshold.

6. Classifier $f_{\text{simple}=}$

Now, we construct a simple classifier which will test our features selectors.

Classifier $f_{\text{simple}=}$ is parametrized by thresholds t_1 and t_2. It selects a set $b_{t_1}(s)$ of categories which are present in at least t_1 percent of pages in input sequence and for each user u it selects a set $b_{t_2}(g(u))$ of categories which are present in at least t_2 percent of pages displayed by u during first three weeks. Then it returns a set of users for whom sets $b_{t_1}(s)$ and $b_{t_2}(g(u))$ are identical. Formally, we define $f_{\text{simple}=} : V \to P(U)$ as follows

$$f_{\text{simple}=}(s) = \{u \in U : b_{t_1}(s) = b_{t_2}(g(u))\}.$$

Table 2 presents test results for this classifier for selected recall values. We tested every possible combination of t_1 and t_2 with step 0.01 and we selected best classifiers with respect to a given recall. As we see the classifier is good for applications where need fairly high recall, small cluster size and large precision is needed. However it is impossible to obtain recall greater then 0.79. Note, that in order to obtain recall greater then 0.75 t_1 and t_2 were set above 0.5. It means that in this case classifier compares most common categories only.

Performance of an analogical classifier that selected features according to a_t was inferior to $f_{\text{simple}=}$

Table 2. Tests of $f_{\text{simple}=}$ classifier with feature selector b_t, $t_1, t_2 \in \{0.00, 0.01, \ldots, 0.80\}$.

t_1	t_2	**Recall**	**Precision**	F	**ClusterSize**
0.07	0.08	0.5008	0.4914	0.4961	435.1
0.10	0.11	0.5518	0.5458	0.5488	527.2
0.15	0.16	0.6016	0.5934	0.5975	672.8
0.17	0.21	0.6501	0.6200	0.6347	778.4
0.26	0.26	0.7014	0.7012	0.7013	1027.3
0.54	0.60	0.7503	0.8010	0.7748	1297.6
0.57	0.60	0.7614	0.7882	0.7746	1306.1
0.53	0.52	0.7704	0.7725	0.7714	1362.4
0.50	0.48	0.7800	0.7777	0.7788	1417.2
0.41	0.43	0.7900	0.7926	0.7913	1459.5

7. Classifier $f_{\text{simple}\subseteq}$

The classifier $f_{\text{simple}\subseteq} : V \to P(U)$ selects a set $a_t(s)$ of categories which are present in at least t pages in input sequence. Then it returns set of users for whom set $a_t(s)$ is a subset of a set of categories of pages visited by this user during first three weeks. It is defined as:

$$f_{\text{simple}\subseteq}(s) = \{u \in U : a_t(s) \subseteq a_0(g(u))\}.$$

Table 3 presents performance of $f_{\text{simple}\subseteq}$ used with features selected by a_t. It offers arbitrarily high recall, however it has large cluster size and small precision.

Performance of an analogical classifier that selected features according to b_t was inferior to $f_{\text{simple}\subseteq}$.

Table 3. $f_{\text{simple}\subseteq}$ classifier quality.

t	**Recall**	**Precision**	F	**ClusterSize**
0	0.4318	0.0119	0.0232	1211.6
1	0.6059	0.0226	0.0436	1647.3
2	0.6956	0.0312	0.0597	1910.3
3	0.7522	0.0385	0.0733	2099.7
5	0.8191	0.0504	0.0949	2356.7
7	0.8558	0.0597	0.1116	2543.1
10	0.8947	0.0721	0.1335	2747.9
15	0.9295	0.0892	0.1629	3000.6
20	0.9480	0.1044	0.1881	3177.1
25	0.9592	0.1156	0.2064	3301.4
30	0.9664	0.1268	0.2242	3405.6
35	0.9746	0.1397	0.2443	3497.0
40	0.9801	0.1512	0.2620	3567.9
50	0.9846	0.1721	0.2931	3674.5

Performance of classifier $f_{\text{simple}\subseteq}$ depends significantly on the number of categories present in cluster. Table 4 shows that large cluster size is a consequence of poor distinctive ability of classifier for users with small number of categories. On the other hand, for clusters specified by more than three categories, we observe decrease of recall. That is why, now, we will develop classifiers dedicated to clusters with a certain number of categories. Then we combine them into one classifier which will work with recall 0.9.

Table 4. Dependence $f_{\text{simple}\subseteq}$ classifier performance from number of categories present in cluster, with threshold $t = 30$.

No cats	No users	Recall	Precision	F	Cluster Size
1	2224	0.9977	0.1797	0.3046	4276.1
2	1752	0.9657	0.1126	0.2018	3194.1
3	692	0.9031	0.0291	0.0565	1909.7
4	185	0.8648	0.0083	0.0165	1011.3
5	25	0.7600	0.0025	0.0050	420.9
6	3	0.6666	0.0059	0.0118	214.0
7	1	1.0000	0.0045	0.0091	218.0

8. Classification of sequences specified by one category

In this case we know that, during the fourth week, examined user displayed pages belonging to only one category. The only information that may help us to indicate such user is the percent of Page Views that belong to that category during the first three weeks of user activity.

Let s be a sequence such that $a_{30}(s)$ is a singleton. Let $\mathbf{t} = (t_A, t_B, \ldots, t_T)$. The classifier $f_{\mathbf{t}} : V \to P(U)$ is defined as:

$$f_{\mathbf{t}}(s) = \{u \in U : a_{30}(s) = \{c\} \wedge c \in b_{t_c}(g(u))\}.$$

In the definition above $a_{30}(s) = \{c\}$ means that c is the only category that appears in sequence s more than 30 times and $c \in b_{t_c}(g(u))$ means that more than t_c percent of pages displayed by user u is labeled with category c.

Table 5 presents optimal $\mathbf{t}$ and classifier quality for all one categorial clusters. As we can see each cluster has its own specific character: clusters differs in size, threshold value and precision. Some clusters are simple to classify while others (see P-cluster for example) are pretty hard. In general $f_{\mathbf{t}}$ reduced average cluster size from 4276.1 to 1624.1 maintaining recall at the level 0.9.

Table 5. Optimal parameters of $f_{\mathbf{t}}$ classifier and its performance on single categorial clusters. Last verse describes classifier on all single categorial clusters together.

Cluster	No users	t	Recall	Precision	F	Cluster Size
A	5	0.02	1.0000	0.0352	0.0680	142.0
B	2	0.48	1.0000	1.0000	1.0000	2.0
C	14	0.01	1.0000	0.0204	0.0400	686.0
D	2	0.08	1.0000	0.0909	0.1667	22.0
E	9	0.07	0.8889	0.0661	0.1231	121.0
F	27	0.25	0.9259	0.3676	0.5263	68.0
G	676	0.50	0.9038	0.3790	0.5341	1612.0
H	40	0.19	0.9000	0.2000	0.3273	180.0
I	69	0.31	0.9275	0.3879	0.5470	165.0
J	1	0.62	1.0000	1.0000	1.0000	1.0
K	8	0.43	1.0000	0.2857	0.4444	28.0
L	108	0.33	0.9074	0.2700	0.4161	363.0
M	3	0.76	1.0000	0.7500	0.8571	4.0
N	14	0.58	0.9286	0.4062	0.5652	32.0
O	5	0.53	1.0000	0.4167	0.5882	12.0
P	58	0.03	0.9310	0.0432	0.0826	1250.0
Q	1176	0.46	0.9022	0.5247	0.6635	2022.0
R	3	0.23	1.0000	0.3750	0.5455	8.0
S	4	0.30	1.0000	0.3636	0.5333	11.0
joint	2224		0.9069	0.4356	0.5885	1624.1
$f_{\text{simple}\subseteq}$	2224		0.9977	0.1797	0.3046	4276.1

9. Classification of sequences specified by two categories

Consider the following sequences:

$$CQCGQGQGQGQGQGQGQGQC,$$

$$FGFGFGFGFGFGFGFGQFQF,$$

$$FGFIFIFGFGFGFIF.$$

They consists of long periods of alternating occurrences of two categories. We observed that mosts of sequences have such structure.

We recognize such periods when they have at least four occurrences and we replace recognized periods with new categories. Each occurrence of such new category is assigned with the number of Page Views equal to the sum of Page Views that appear in recognized period.

For example the above sequences are processed into:

$$CQC\underline{GQ}C,\ \underline{FG}\underline{FQ},\ FG\underline{FI}\underline{FG}IF.$$

While classifying sequence specified by two categories (we will denote them as c_1 and c_2) we consider following features:

- the quantity of occurrence of c_1 category in sequence,
- the quantity of occurrence of c_2 category in sequence,
- the quantity of occurrence of c_1c_2 category created from alternating occurrences of c_1 and c_2.

We analyze performance of four classifiers:

$$\begin{aligned} f_{\mathbf{t},\mathbf{t}'}(s) &= \{u \in U : a_{30}(s) = \{c_1, c_2\} \\ &\quad \wedge c_1 \in b_{t_{c_1c_2}}(g(u)) \wedge c_2 \in b_{t'_{c_1c_2}}(g(u))\}, \\ f_{\mathbf{t}''}(s) &= \{u \in U : a_{30}(s) = \{c_1, c_2\} \\ &\quad \wedge c_1c_2 \in d_{t''_{c_1c_2}}(g(u))\}, \\ f_{\mathbf{t},\mathbf{t}'\wedge\mathbf{t}''}(s) &= f_{\mathbf{t},\mathbf{t}'}(s) \cap f_{\mathbf{t}''}(s), \\ f_{\mathbf{t},\mathbf{t}'\vee\mathbf{t}''}(s) &= f_{\mathbf{t},\mathbf{t}'}(s) \cup f_{\mathbf{t}''}(s). \end{aligned}$$

Performance of $f_{\mathbf{t},\mathbf{t}'}$ and $f_{\mathbf{t}''}$ indicate that both quantity of single categories occurrences and quantity of alternating occurrences of categories carry important information. $f_{\mathbf{t},\mathbf{t}'\wedge\mathbf{t}''}$ and $f_{\mathbf{t},\mathbf{t}'\vee\mathbf{t}''}$ tested possible combinations of this information. Tests show that $f_{\mathbf{t},\mathbf{t}'\vee\mathbf{t}''}$ classifier gives best results.

Table 6. Parameters of classifier $f_{\mathbf{t},\mathbf{t}'\vee\mathbf{t}''}$ for sequences with two categories.

Cluster	No user	t	t'	t''
CG	15	0.01	0.01	0.88
CQ	23	0.22	0.02	0.33
EQ	17	0.03	0.00	0.83
FG	11	0.32	0.01	0.08
FQ	25	0.08	0.00	0.38
GH	14	0.00	0.00	0.00
GI	29	0.20	0.01	0.05
GL	36	0.06	0.17	0.01
GN	116	0.95	0.48	0.05
GP	61	0.01	0.66	0.00
GQ	834	0.09	0.11	0.37
HQ	87	0.45	0.01	0.15
IQ	73	0.09	0.07	0.04
LQ	186	0.28	0.21	0.22
MQ	15	0.37	0.23	0.11
NQ	14	0.43	0.75	0.02
OQ	12	0.02	0.00	0.43
PQ	99	0.79	0.00	0.04

A possible reason for such behaviour is that quantity of single categories occurrences is relevant feature for short sequences and quantity of alternating category occurrences is relevant for long sequences. So, these features operate on disjoint sets of sequences and should be connected by disjunction.

Tables 6 and 7 presents optimal parameters and performance of $f_{\mathbf{t},\mathbf{t}'\vee\mathbf{t}''}$ classifier for clusters that have more then 10 users. Clusters have their own specific character similarly as single categorial clusters. For some of them, it was impossible to find recall greater then 0.9. Classifier reduced average cluster size from 3194.1 to 1714.0 maintaining recall at the level 0.9.

In order to avoid overfitting we used $f_{\text{simple}\subseteq}$ only for clusters that have less then 11 users.

Table 7. Performance of classifier $f_{\mathbf{t},\mathbf{t}'\vee\mathbf{t}''}$ for sequences with two categories. Last verse describes performance of classifier on all two categorial clusters together.

Cluster	No user	Recall	Precision	*F*	Cluster Size
CG	15	0.9333	0.0648	0.1212	216.0
CQ	23	0.8261	0.0662	0.1226	287.0
EQ	17	0.8824	0.0498	0.0943	301.0
FG	11	0.8182	0.0383	0.0732	235.0
FQ	25	0.9200	0.0271	0.0526	849.0
GH	14	0.8571	0.0037	0.0073	3265.0
GI	29	0.9310	0.0685	0.1277	394.0
GL	36	0.9167	0.0815	0.1497	405.0
GN	116	0.9052	0.4217	0.5753	249.0
GP	61	0.9016	0.0469	0.0891	1173.0
GQ	834	0.9005	0.2950	0.4444	2546.0
HQ	87	0.9080	0.0607	0.1138	1302.0
IQ	73	0.9041	0.1486	0.2553	444.0
LQ	186	0.9032	0.3597	0.5145	467.0
MQ	15	0.8000	0.4286	0.5581	28.0
NQ	14	0.9286	0.3714	0.5306	35.0
OQ	12	0.9167	0.0161	0.0317	682.0
PQ	99	0.9091	0.0317	0.0612	2841.0
joint	1667	0.9010	0.2431	0.3829	1714.0
$f_{\text{simple}\subseteq}$	1752	0.9657	0.1126	0.2018	3194.1

10. Results

We define our ultimate classifier f_{combined} as follows: for a given sequence, we select relevant categories using feature selector a_{30}, then we check the number of categories in obtained set:

- if there is only one category, we classify sequence using $f_{\mathbf{t}}$ classifier with parameters specified in Table 5;

- if there are two categories and this pair of categories is present in Table 6, we classify sequence using $f_{\mathbf{t},\mathbf{t}'\vee\mathbf{t}''}$ classifier with parameters specified in Table 6;
- otherwise we classify sequence by means of $f_{\text{simple}\subseteq}$ classifier.

Performance of f_{combined} is presented in the Table 8 in comparison with other classifiers is presented.

Table 8. Classification results.

Classifier	Recall	Precision	F	ClusterSize
f_{combined}	0.9015	0.2860	0.4342	1651.9
$f_{\text{simple}\subseteq}$	0.6059	0.0226	0.0436	1647.3
$f_{\text{simple}\subseteq}$	0.9033	0.0762	0.1405	2808.3
$f_{\text{simple}=}$	0.7900	0.7926	0.7913	1459.5

11. Future work

In future we plan to investigate the information carried in number of Page Views. Tests revealed that there exists a loose correlation between a total number of pages viewed by user during first three weeks and a number of pages that he displayed during fourth week.

Randomness in sequences is an obstacle in finding arranged structures longer than two alternating symbols. However, low entropy of sequences suggests that it could be possible to recognize more complex patterns of user behaviour.

The phenomenon of alternating categories itself is also interesting and worth of further analysis.

Acknowledgment

The research has been partially supported by grants N N516 368334 and N N516 077837 from Ministry of Science and Higher Education of the Republic of Poland.

1. Wu, K.-L., Yu, P. S., Ballman, A. SpeedTracer: A Web usage mining and analysis tool. *IBM Systems Journal*, **37**(1) (1998)
2. Nagino, N., Yamada, S., Future View: Web Navigation Based on Learning User's Browsing Patterns. In: *IEEE/WIC International Conference on Web Intelligence (WI 2003)* (2003)
3. Ting, I-H., Clark, L., Kimble, C., Kudenko, D., Wright, P. APD - A Tool for Identifying Behavioural Patterns Automatically from Clickstream Data. *Knowledge-Based Intelligent Information and Engineering Systems*, LNCS 4693, 66-73 (2007)
4. Godoy, D., Amandi, A. Exploiting User Interests to Characterize Navigational Patterns in Web Browsing Assistance. *New Generation Computing*, **26**(3), 259–275 (2008)
5. Shahabi, C., Zarkesh, A.M., Adibi, J., Shah, V. Knowledge discovery from users Web-page navigation. In proc. *Seventh International Workshop on Research Issues in Data Engineering*, 20–29 (1997)
6. Yan, T., Jacobsen, M., Garcia-Molina, H., Dayal, U. From User Access Patterns to Dynamic Hypertext Linking. In *Fifth International World Wide Web Conference*, Paris, France (1996)
7. Godoy, D., Amandi, A. Learning Browsing Patterns for Context-Aware Recommendation. In *IFIP AI*, 61–70 (2006)
8. Middleton, S., Shadbolt, N., Roure, D. Ontological user profiling in recommender systems. *ACM Transactions on Information Systems*, **22**(1), 54–88 (2004)
9. Godoy, D., Amandi, A. Modeling user interests by conceptual clustering. *Information Systems*, **31**(4-5), 247265 (2006)
10. Fu, Y., Sandhu, K. and Shih, M. Y. Clustering of Web Users Based on Access Patterns. In: *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Springer, San Diego (1999)
11. Kumar, R., Tomkins, A. A characterization of online browsing behavior. In *Proceedings of the 19th international conference on World wide web (WWW 2010)*, 561–570 (2010)
12. Srivastava, J., Cooley, R., Deshpande, M., Tan, P.-N. Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data. *SIGKDD Explorations* **1**(2) (2000)
13. Zhou, H.-f., Hei, X.-H., Lu, L.-T. Mining interesting knowledge from Web-log. *Wuhan University Journal of Natural Sciences*, **9**(5), 569–574 (2004)
14. Song, Q., Shepperd, M. Mining web browsing patterns for E-commerce. *Computers in Industry*, **57**(7), 622-630 (2006)
15. Sreenath, D. V., Grosky, W. I., Fotouhi, F. Emergent Semantics from Users' Browsing Path. In proc. *1st NSF/NIJ conference on Intelligence and security informatics*, Springer-Verlag Berlin, Heidelberg (2003)
16. Jaworska, J., Nguyen, H. S. Users Behaviour Prediction Challenge, in: Nguyen, H. S. (Eds.) *Proc. ECML/PKDD2007 Discovery Challenge* (2007)
17. Guillet, F., Hamilton, H.J. (Eds.): *Quality Measures in Data Mining*, Studies in Computational Intelligence 43, Springer (2007)
18. Vapnik, V. N., *Statistical Learning Theory*, Wiley, New York (1998)
19. Angluin, D. Inductive inference of formal languages from positive data. *Information and Control*, **45**, 117–135 (1980)