Analyzing the Evolution of Web Usage Data

Alzennyr da Silva

INRIA-Rocquencourt, France

Outline

- Introduction
- Motivations for this work
- Our proposition
- Clustering approach based on time sub-periods
- The benchmark website analyzed
- Results analysis
- Final conclusion
- Future works
Introduction

- **The WWW:**
  - one of the most relevant examples of voluminous and dynamic data sources

- **Web access patterns have a dynamic nature, due to:**
  - the dynamism of the website’s content and structure
  - or
  - the change of user’s interest

- **Access patterns may depend on:**
  - time of day, day of the week
  - recurrent factors (summer/winter vacations, national holidays, seasonality)
  - non-recurrent global events (epidemics, wars, the World Cup)
  - etc.

Motivations for this work

- The majority of methods in the Web Usage Mining (WUM) domain take into account the **whole period** of usage traces.

  - **Consequence:**
    - the results are those predominant in the entire period of analysis

  - **Negative side effects:**
    - behaviour patterns occurring in short periods of time are not detected by traditional methods
Our proposition

To carry out an analysis on significant time sub-periods, in order to:
- identify the change of user's interest
- follow the evolution of user's profiles over time

using

Summaries to represent user profiles
The website analyzed

- Recife’s (Brazil) Information Technology Centre website (http://www.cin.ufpe.br):
  - static pages (personal web pages, lessons pages, etc.)
  - 91 dynamic pages (maintained by Java servlets in a semantic structure)

- We retrieved the traces of usage:
  - 1 July 2002 – 31 May 2003 (roughly 2Go of raw data)

Common Log Format (CLF)

```
[remotehost] [name] [login] [date] [url] [status] [size] [referrer] [agent]
```

- remotehost: remote identification (hostname or IP address)
- name/login: the remote login name of the user
- date: date and time of the request
- URL: requested page in the site (www.<...>)
- status: returned code (Indicates whether or not the file was successfully retrieved)
- size: the number of bytes transferred
- referrer: the url the client was on before requesting the current url
- agent: the software the client is using
We selected navigations with two shared constraints:
- long
  - number of requests $\geq 10$
  - total duration $\geq 60$ seconds
- those of human origin
  - total duration / number of requests $\geq 4$ (15 requests/min)

After filtering and eliminating the outliers:
- 138,536 navigations
Statistical attributes for navigations’ description

<table>
<thead>
<tr>
<th>Nº</th>
<th>Field</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IDNavigation</td>
<td>Navigation code</td>
</tr>
<tr>
<td>2</td>
<td>NbRequests_OK</td>
<td>Number of successful requests (status = 200) in the navigation</td>
</tr>
<tr>
<td>3</td>
<td>NbRequests_bad</td>
<td>Number of failed requests (status &lt;&gt; 200) in the navigation</td>
</tr>
<tr>
<td>4</td>
<td>MRequests_OK</td>
<td>Percentage of successful requests (~ NbRequests_OK / NbRequests)</td>
</tr>
<tr>
<td>5</td>
<td>NbRepetitions</td>
<td>Number of repeated requests in the navigation</td>
</tr>
<tr>
<td>6</td>
<td>MRepetitions</td>
<td>Percentage of repeated requests (~ NbRepetitions / NbRequests)</td>
</tr>
<tr>
<td>7</td>
<td>TotalDuration</td>
<td>Total duration of the navigation (in seconds)</td>
</tr>
<tr>
<td>8</td>
<td>ADuration</td>
<td>Average of request duration (~ TotalDuration / NbRequests)</td>
</tr>
<tr>
<td>9</td>
<td>ADuration_OK</td>
<td>Average of duration among successful requests (~ TotalDuration_OK / NbRequests_OK)</td>
</tr>
<tr>
<td>10</td>
<td>NbRequests_Sem</td>
<td>Number of requests for the (91) dynamic pages concerning the site’s semantic structure</td>
</tr>
<tr>
<td>11</td>
<td>MRequests_Sem</td>
<td>Percentage of semantic requests (~ NbRequests_Sem / NbRequests)</td>
</tr>
<tr>
<td>12</td>
<td>TotalSize</td>
<td>Total bytes transferred in a navigation</td>
</tr>
<tr>
<td>13</td>
<td>ASize</td>
<td>Average of transferred bytes among requests (~ TotalSize / NbRequests_OK)</td>
</tr>
<tr>
<td>14</td>
<td>MaxDuration_OK</td>
<td>Maximum request duration among successful requests</td>
</tr>
</tbody>
</table>

Clustering approach based on time sub-periods

- To split the analyzed period into more significant time sub-periods: *months of the year*

- The clustering is carried out by an adapted version of the dynamic clustering algorithm (Celeux et al. (1989)):
  1. Assignment of new individuals to a previous clustering
  2. Initialization of the algorithm with the results of another clustering carried out by itself
Clustering approach based on time sub-periods

- Algorithm parameters:
  - Number of clusters = 10
  - Number of repetitions = 100

- To carry out four types of clustering:
  1. Global clustering
  2. Independent local clustering
  3. Previous local clustering
  4. Dependent local clustering

(1/4) Global clustering
(2/4) Independent local clustering

Previous and dependent local clustering
(3/4) Previous local clustering

Data set

1st sub-period

clustering (k=10)

2nd sub-period

Allocation step

1st sub-period (p=10)

P1

P2

Pk

n-1th sub-period

nth sub-period

2nd sub-period

n-1th sub-period

nth sub-period

Processed data

3th sub-period

n-1th sub-period

nth sub-period

Allocation step

3th sub-period (p=10)

P1

P2

Pk

n-1th sub-period

nth sub-period

(4/4) Dependent local clustering
Results analysis

- Evaluation criteria:
  - For a cluster-by-cluster analysis
    - F-measure (van Rijsbergen (1979))
  - For a global analysis between two partitions
    - Corrected Rand index (Hubert et Arabie (1985))

Follow-up of cluster prototypes

- To better understand the cluster evolution over time sub-periods, we planned to:
  - Follow the evolution of cluster prototypes (month by month) for the local clustering: independent and dependent
  - Project these prototypes on the factorial plan computed over the total population
Follow-up of cluster prototypes

Fig. 1 Projection and follow-up of cluster prototypes for local clustering.

Intra-cluster variance

Fig. 3 Intra-cluster variance for clustering: independent (black line), dependent (red line) and global (blue line).
Corrected Rand index results

Fig. 4 Cluster-by-cluster corrected Rand index.

F-measure results

Fig. 5 Boxplots corresponding to cluster-by-cluster F-measures.
Conclusion

- The methods of *global* and *dependent local* clustering show that the obtained partition do not change over time or change only a bit.

- The method of *independent local clustering* is more sensitive to changes occurring between two sub-periods.

- The analysis of dynamic data by means of time sub-periods offers advantages:
  - makes the method more effective in terms of cluster discovery
  - allows to overcome difficulties related to physical limitations (memory size, processor speed, etc.)

Future works

- Implementation of other clustering methods
- Application of techniques allowing the automatic discovery of the cluster number
- Identification of merge and split between clusters over time
Thanks for your attention!

Questions

Intra-cluster variation

\[ V(Q) = \sum_{j=1}^{k} \sum_{x \in C_j} d(x, P_j) \]
The F-measure combines the concepts of precision and recall between two $U_i$ and $C_k$ of two partitions.  

The recall is defined as $R(i,k) = \frac{n_{ki}}{n_k}$. It computes the percentage of elements from class a priori $k$ founded in class $i$ obtained by the classification method. The recall also decreases when the number of classes in the partition obtained by the classification decreases.

The precision is defined as $P(i,k) = \frac{n_{ki}}{n_i}$. It computes the percentage of elements from class $i$ founded in the a priori class $k$. The precision increases when the number of classes in the partition obtained by the classification decreases.

The F-measure between the a priori partition $U$ in $K$ classes and the partition $P$ obtained by the classification method is defined as:

$$F = \sum_{k=1}^{K} \left( n_k, n \right) \max_j \left( \frac{2.R(k, j).P(k, j)}{R(k, j) + P(k, j)} \right)$$

F-measure for the a priori class $k$:

$$F(k) = \max_j \left( \frac{2.R(k, j).P(k, j)}{R(k, j) + P(k, j)} \right)$$
### Corrected Rand index

\[
CR = \frac{\sum_{i=1}^{R} \sum_{j=1}^{C} \binom{n_{ij}}{2} - \binom{n}{2}^{-1} \sum_{i=1}^{R} \binom{n_{i}}{2} \sum_{j=1}^{C} \binom{n_{j}}{2}}{\frac{1}{2} \left[ \sum_{j=1}^{C} \binom{n_{j}}{2} + \sum_{i=1}^{R} \binom{n_{i}}{2} \right] + \left( \binom{n}{2}^{-1} \sum_{i=1}^{R} \binom{n_{i}}{2} \sum_{j=1}^{C} \binom{n_{j}}{2} \right) - \binom{n}{2}^{-1} \sum_{i=1}^{R} \binom{n_{i}}{2} \sum_{j=1}^{C} \binom{n_{j}}{2}}
\]

### Key statistics

- **After the pre-processing and data selection:**
  - 138,536 navigations
  - 184,275 pages (where 91 dynamics)
  - 56,314 users
  - Average duration of page visualization:
    - 1.19 minutes
The authors propose a method of recursive division for discovering sequential patterns of weak support (until 0.006%):

- hacking activities
- minority users’ behaviours

The split is based on a classification over the whole log and on time.

The dynamic clustering method

Let \( E \) be a set of \( n \) objects \( \{s_1, \ldots, s_n\} \) described by \( p \) variables, \( \Lambda \) be a set of prototypes and \( \psi \) be a distance function on \( D_x \times \Lambda \).

Each object \( s \) of \( E \) is described by a vector \( x_s \) of \( D_x \) (the representation space of elements in \( E \)).

The problem is to find simultaneously:

- one partition \( P = (C_1, \ldots, C_K) \) of \( E \) in not empty \( K \) classes
- the prototypes \( L = (L_1, \ldots, L_K) \) of \( \Lambda \) which optimise the criteria \( \Delta(P, L) \):

\[
\Delta(P, L) = \sum_{k=1}^{K} \sum_{x_s \in C_k} \psi(x_s, L_k) \quad C_k \in P, L_k \in \Lambda
\]
The dynamic clustering algorithm
Diday (1971)

(a) Initialization
Choose $K$ distinct prototypes $L_1,\ldots,L_K$ in $\Lambda$

(b) Allocation
For each objet $s_i$ of $E$ compute the index $l$ of the affectation class which verifies $l = \arg \min_{k=1,\ldots,K} \psi(x_i, L_k)$

(c) Representation
For each class $k$ find the prototype $L_k$ in $\Lambda$ which minimizes $w(C_k, L) = \sum_{s \in C_k} \psi(x_s, L)$

Repeat (b) and (c) until the convergence

The original $k$-means algorithm

Suppose we have a sample of infinite size.
With the $x_t$ implementation, we only have information regarding the sample of size $t$.

Initialization
Choose $K$ points in $\mathbb{R}^p$ $L_0 = (L_0^1, \ldots, L_0^K)$

At the $t$ step We associate the $x_i$ implementation to the class $k$ which has the nearest prototype $k = \arg \min_{l=1,\ldots,K} \psi(L_l^i, x_i)$

We modify the prototype of the class $k$ by $L_{t+1}^k = \frac{n_k L_k^i + x_i}{n_k + 1}$ where $n_k$ is the number of implementation already put into the class $k$.

Stopping criterion we must have $\psi(L_{t+1}^i, L_i) \leq \varepsilon$
Fig. 2. Cluster prototypes projection for clustering: global (G1, G2, ..., G10), dependent local (o) and independent local (+).