Automatic device detection in web interaction

I. Perona, A. Yera, O. Arbelaitz, J. Muguerza, N. Ragkousis, M. Arrue, J.E. Perez, X. Valencia
{inigo.perona,ainhoa.yera,olatz.arbelaitz,j.muguerza}@ehu.eus,ragkousis_nikos@hotmail.com,{myriam.arrue,juaneduardo.perez,xabier.valencia}@ehu.eus

Dept. of Computer architecture and Technology
University of the Basque Country
M. Lardizabal, 1, 20018 Donostia, Spain

Abstract. Digital competences are considered basic nowadays what makes important to familiarize and capacitate people with disabilities in the use of digital devices and applications and to adapt the site interaction to their necessities. Most of the current adaptable systems are linked to predefined user profiles. However, the automatic detection of user characteristics will introduce the possibility of automatic adaptations to the user characteristics. This work focuses on identifying one of the important user characteristics; the device being used to interact with the computer. Based on web user interaction data collected by Remotest platform, the complete data mining process has been carried out to build a system able to identify the used device. This preliminary system is able to efficiently determine the used device with an accuracy of 93.07%.

Keywords: Data mining, machine learning, accessibility, adaptive web

1 Introduction

In the last decades, the trends have led to a dramatic increase in the information stored on the web and its use. Website accesses have become an important tool for information seeking, communication and participation processes in our society, and consequently, digital competences are considered basic nowadays. This makes important to familiarize and capacitate people with disabilities in the use of digital devices and applications and to adapt the site interaction to their necessities.

Unfortunately, a theoretically accessible design might not be enough to ensure that people with disabilities access the website fluently. In this context adaptation of the site to the users becomes crucial. Adaptations could be determined according to the results of specific questionnaires but this would be limited to the users participating in the questionnaire. Moreover, in general web applications it is very easy to fail to recognize the full range of types of users who might be interested in using or who might need to navigate in them [1].

Another option to propose adaptations is the analysis of the interaction of the users with the website. This option will be more general and applicable to new users accessing the site.

The specific adaptations required will depend on the user characteristics, the problems the user is having while navigating, etc. In this context, the detection in use time of the navigation problems or the type of device being used, will be a compulsory initial stage to later be able to automatically adapt the site to the user and thus, to improve the user experience.

The use of web mining [2] for these objectives has many advantages. It is not disruptive, it is based on statistical data obtained by real navigation data (decreasing the possibility of false assumptions) and is itself adaptive (when the characteristics of the user change, the collected data allows the automatic change of the interaction schema). When the user is a person with physical, sensory or cognitive restrictions, data mining is the easiest (and frequently almost the only) way to obtain information about the use habits or characteristics of the person.

We present in this work a preliminary approach to automatically detect the device being used to interact with the computer while navigating in the Web. The system was built based on data collected by Remotest [3], a tool to collect the complete user interaction data. The process includes the complete data mining process where after prior meetings with the accessibility experts, decisions were made on the features to extract from the collected data. Then the feature extraction and calculation system was built, and finally, different machine learning algorithms combined with some feature selection options where applied to build a system able to differentiate the device being used in the navigation.

The results showed that the application of a complete data mining process to the data collected by Remotest is a promising strategy to automatically detect user characteristics and then propose the specific adaptations.

The paper proceeds describing in the next section the platform used to collect the data. Section 3 describes the experiments carried out to build the system and Section 4 describes the database generation process. In Section 5 we describe the built classifiers, analyse results and propose improvements. Finally, we draw some conclusions and mention some future works that are still to be done in Section 6.

2 Remotest

The Remotest platform [3] provides the necessary functionalities to assist researchers to define web-based user experiments, manage experimental remote/in situ sessions and analyse the gathered interaction data. This platform admits a wide range of experiments. The architecture of the platform consists on a hybrid architecture model that includes some functions in a client-side module and the other ones in some server-side modules. The platform is split into four modules: Experimenter Module (EXm), Participant Module (PAm), Coordinator Module (COm) and Results Viewer Module (RVm).

Although the RVm processes and visualizes some of the data, to build the system proposed in this work we directly worked with the interaction units or events gathered and interaction information stored in the PAm, such as cursor movements, key presses, scrolls, clicks, etc.

3 Experiments with users

Fifteen subjects took part in the study. Five groups were defined based on the input device used for pointing and clicking actions: (a) two keyboard users, (b) two keyboard users using a headpointer to interact with the keyboard (keyboard+headpointer users), (c) one trackball user, (d) four joystick users, (e) six mouse users.

All subjects from the first four groups (a, b, c and d) were participants with motor-impairments most of them with over seven years of experience and using daily the computer. The subjects in the last group only included users without disabilities who had more than seven years of daily use of the mouse as an input device.

The same Dell Precision M6700 laptop running a 64 bit version of Windows 7 was utilized in all sessions. An additional widescreen LCD monitor (aspect ratio 16:10) with a diagonal size of 24 inches and display resolution set to 1920 x 1200 pixels was used to present stimuli to participants. Firefox add-ons implementing the virtual aids for the cursor were installed in this computer.

Before starting the study, participants were encouraged to adjust the pointer motion behaviour to meet their preferences. Subjects from the a, b, c and d groups used their own personal input devices to complete the study. All non disabled participants (e group) used the same optical USB mouse (Dell M-UVDEL1).

Two different websites were selected as stimuli for the experiment: the Discapnet website `http://www.discapnet.com` which provides information to people with disabilities, and the institutional website from the Council of Gipuzkoa `http://www.gipuzkoa.eus/`. A third informational website about the Bidasoa local area was used `http://www.bidasoaturismo.com` for training purposes, so participants could learn how to use the new cursor virtual enhancements. All three websites claim, within their accessibility sections, to conform to certain level of the WCAG 1.0 guidelines (Discapnet to Level AA, Gipuzkoa and Bidasoa to the Level A).

4 Database generation and feature selection

The design of the device detection system consisted on a complete data mining process. After prior meetings with the accessibility experts, decisions were made on the features to extract from the collected data. Then the feature extraction and calculation system was built to extract the desired values from the information gathered with Remotest.

4.1 Database generation

The interaction of the users with the website was collected with Remotest and converted to a labelled database to be used in a supervised classification environment. We supposed that each of the users interacted similarly in every page visited during the experiments carried out, and that the interaction somehow depended on the type of device the user was using. So, the device being used was used to label the database examples, generating a database with 5 classes: keyboard, keyboard + headpointer, trackball, joystick and mouse.

Each of the entries of the generated database contains the summary of the interaction of a user with a visited page. To summarize this interaction, we tried to extract as many features as possible considered meaningful by the accessibility experts.

We concretely extracted the 19 features summarized in Table 1.

From the experiments carried out, a 5-class unbalanced database with 20 (19 + class) features was generated (see Table 2). All the features were standardized (standard score was calculated) so that their differences in ranges did not affect to the performance of the built classifiers.

4.2 Feature Selection

Although the aim of this preliminary work was to automatically detect the used device, we also aim in a short future, to be able to detect navigation problems found by the users. Consequently, features were extracted thinking in both objectives. For instance, features such as RatioCursorDistOptimal or pStrongDirectionChanges (Table 1) are not intuitively expected to be significant to differentiate between different used devices but we still used them and built classifiers with the complete set of features to confirm our suspicions.

On the other hand, we took into account the expertise of the accessibility experts, who considered only 8 out of the 19 features extracted to be significant to detect the used device. Concretely those marked as priority 1 in Table 1. Consequently we also built classifiers using only these features.

Furthermore, two automatic feature selection algorithms [4] were used to determine the best features to use from the machine learning point of view; we selected two of the most used feature selection algorithms: the Correlation-based Feature Subset Selection [5] and a Wrapper [6] feature selection option which optimizes the features for a given classifier (J48 in our case).

5 Classification system

The calculated features were used to build classifiers to classify user interaction data according to the device used for navigation. We built classifiers with the complete set of features extracted, the features considered to be the most important by the accessibility experts and the features selected by some automatic feature selection processes.

Name	Priority	Description
NoEvents	1	Number of events recorded by Remotest in the page
NoSpecialKeyPress	1	Number of special (no letter or digit input) keys pressed within the page
NoWheels	1	Number of times the wheel has been used within the page
crossMoves	1	Number of cursor movements in the horizontal or vertical axes
diagonalsMoves	1	Number of movements done in the diagonals (with angles of 45, 135, 220, 315 degrees) within the page
medianGap	1	Median of the gaps, times without movement, appearing in the cursor movements within the page
medianSpeed	1	Median of the cursor movement speed within the page
medianAcc	1	Median of the cursor movement acceleration within the page
NoKeyPress	2	Number of keys pressed while navigating in the page
CursorDist	2	Distance given in number of pixels travelled by the cursor within the page
RatioCursorDistOptimal	2	Ratio between the distance travelled by the cursor within the page and the optimal distance between its initial and final position
pStrongDirectionChanges	2	Percentage of changes that are considered to be strong (see NoStrongDirectionChanges feature)
NoClicks	3	Number of clicks recorded by Remotest in the page
NoScrolls	3	Number of times the scroll has been used within the page
pStraightDirections	3	Percentage of movements done in cross or diagonal
NoDirectionChanges	3	Number of direction changes performed by the user within the page
NoStrongDirectionChanges	3	Number of strong direction changes (relative position of the cursor changes more than 45 degrees) performed by the user within the page
areasMoves	3	Number of movements in directions which are different to horizontal and vertical axes or diagonals
totalTime	3	Total time spent in the page

Table 1. Description of the features extracted for each visited page.

class	**Number of examples**
Joystic	584
Keyboard	347
Keyboard+headpointer	338
Trackball	171
Mouse	235
Total	1675

Table 2. Class distribution of the generated database.

Experiments were run in Weka [7] with 4 basic classifiers, Naive Bayes [8], IBK [9], SVM [10] and J48 [11] with default parameters and two decision tree (J48) based meta classifiers, bagging and boosting, with 25 iterations. A five fold cross-validation (5 fold-CV) strategy was used for validation (80% for training and 20% for testing). As it can be observed in Table 3, the four databases differing in the contained features were evaluated:

- The most important features according to the experts (P1 features)
- All the extracted features (All features)
- The features selected by the Correlation-based Feature Subset Selection method (CF Subset Eval)
- The features selected by the wrapper selection method with J48 as classifier and Genetic Search as search algorithm (Wrapper J48)

	Classifiers					
Used Features	Naive Bayes	IBK	SVM	J48	Bagging	Boosting
P1 features	66.09	67.46	62.09	71.88	74.75	75.52
All features	57.85	67.82	67.1	74.45	79.34	79.76
CF Subset Eval	59.64	68.96	64.12	74.87	77.97	77.25
Wrapper J48	57.85	67.88	66.57	75.82	79.7	80.78

Table 3. Classification results for different feature sets and classifiers.

The values in in Table 3 show that classification rates were not as high as expected. Generally the best rates were obtained with the most complex classifiers or meta classifiers: bagging and boosting.

Focusing the analysis on how the sets of features affect to the performance of the system, it seems that the set proposed based on the experience of the accessibility experts (P1 features) is only the best option in the case of Naive Bayes classifiers. The rest of the classifiers behave better using the complete set of features or automatically selected sets of features.

These two outcomes lead us to analyse confusion matrices in order to discover the source of the error on the one hand, and, to analyse the set of features considered to be important by the automatic feature selection methods on the other hand.

5.1 Analysis of the source of the error

To analyse the source of the error, we selected one of the best classifiers, the outcome of a J48 based boosting process applied to a dataset generated using the features selected with the Wrapper Feature selection process, and, studied its confusion matrix (see Table 4).

classified as	a	b	c	d	e	F- Measure
joystic = a	114	0	0	1	2	0.95
keyboard = b	0	49	20	1	0	0.726
keyboard+headpointer = c	0	16	51	0	0	0.739
mouse = d	8	0	0	35	4	0.805
trackball = e	1	0	0	3	30	0.857

Table 4. Confusion matrix + F-measure. Boosting with a Wrapper feature selection

The values clearly show that the main source of error comes from not being able to differentiate classes keyboard and keyboard+headpointer. This was to be expected somehow, since although managing it differently, in both cases the finally used device is the keyboard. On the other edge, the joystic users are very accurately classified obtaining a F-measure value of 0.95 and the mistakes done with mouse and trackball users are also few.

In this sense, in a first approach we simplified the problem to four classes, that is, we joined into the same class keyboard users and keyboard+headpointer users. The new database had still 1675 examples but distributed now in the following way: Joystic (584), Keyboard (685), Trackball (171) and Mouse (235). However, in a future approach we will design a hierarchical classifier able first to discriminate between the two main groups (joystic/trackball/mouse or keyboard/keyboard+headpointer) and then the specific device within each of them; the features might also require to be specifically selected for each second level classifier.

5.2 Solving the 4-class problem

The same experiments described in the previous sections were repeated in Weka for the new 4-class database; Naive Bayes, IBK, SVM and J48, bagging and boosting models were built and evaluated based on a 5 fold-CV strategy.

As it could be expected, the values in Table 5 show that classification rates increased for all classifiers. What means that the systems built combining the features extracted from the experiments carried out with Remotest with machine learning algorithms are able to differentiate the used device accurately.

Comparing classifiers' performance, the same trends observed in the 5-class database were repeated; the best rates were obtained with the most complex classifiers or meta classifiers bagging and boosting.

	Classifiers					
Used Features	Naive Bayes	IBK	SVM	J48	Bagging	Boosting
P1 features	82.99	83.52	80.48	87.22	89.19	90.392
All features	77.31	83.16	84.54	88.9	92.42	93.07
CF Subset Eval	81.19	82.75	80.54	87.34	90.69	91.22
Wrapper J48	77.49	84.12	83.52	89.67	92.48	92.66

Table 5. Classification results for different feature sets and classifiers in the 4 class database.

With regard to the sets of features seeming to work better, they are again the automatically selected ones or the complete set of features.

If we further analyse the confusion matrices (see Table 6 for an example), we realize that the classifier was able to nearly perfectly differentiate the keyboard users from the rest, maintaining the general ability to differentiate devices for the three remaining options. This classifier could be used as a first classifier in a hierarchical system where the classification of the classes joystic/trackball/mouse can be refined in a second stage by a specifically designed classifier.

classified as	a	b	c	d	F- Measure
joystic = a	108	2	5	2	0.919
keyboard = b	1	136	0	0	0.989
mouse = c	6	0	40	1	0.842
trackball = d	3	0	3	28	0.862

Table 6. Confusion matrix. Boosting with a Wrapper feature selection for the 4-class problem.

5.3 Analysis of the importance of the features

As in any data mining process, the features used in the classification process affected the efficiency of the classifiers. As stated before, it seems that the features considered to be the most important by the accessibility experts were not the best to use for classification. Therefore, we considered that the analysis of the features selected by the two feature selection processes applied to the 2 databases (5-class and 4-class) could give us and the experts some clues about the importance of the extracted features. Table 7 contrasts the selection done by the experts and the one done by automatic algorithms. Each of the features could have been selected at most 4 times.

We could conclude from this analysis that only five out of the eight features considered very important by the experts where considered effective for the classification process by most of the 4 automatic feature selection processes carried out. However, there were other three features, pStraightDirections, areasMoves

		Automatically selected	
		Very popular	**Less popular**
Experts	**Priority 1**	NoEvents(4) NoWheels(4) medianGap(4) medianAcc(4) medianSpeed(3)	crossMoves(2) NoSpecialKeyPress(2) diagonalsMoves(1)
	No priority 1	pStraightDirections(4) areasMoves(4) NoKeyPress(3)	RatioCursorDistOptimal(2) NoDirectionChanges(2) totalTime(2) NoClicks(2) CursorDist(1) nStrongDirectionChanges(1) NoScrolls(2) pStrongDirectionChanges(0)

Table 7. Features selected by the experts versus automatically selected features.

and NoKeyPress, not considered by the experts that are important for classification and will probably need to be taken into account by the experts in the future.

Although as stated before some features seemed not to be determinant intuitively for device detection, there was a single one, pStrongDirectionChanges, not selected by any of the feature selection processes executed. However, this feature will probably be informative for problem detection.

6 Conclusions and further work

The results show that the application of a complete data mining process to the data collected by Remotest is a promising strategy to automatically detect user characteristics and then propose the specific adaptations. We concretely tried initially to differentiate 5 types of devices, Joystic, Keyboard, Keyboard+headpointer, Trackball and Mouse but came out to the conclusion that the characteristics of classes keyboard and keyboard+headpointer were very similar and finally built a system able to differentiate 4 devices with an average accuracy value of 93.07 for the best classifier.

On the other hand the performed analysis showed that not all the features considered of highest priority by the accessibility experts were important from the classification point of view, whereas some of the features considered no so important were.

As further work we propose the design of a hierarchical classifier able first to discriminate between the two main classes (joystic/trackball/mouse or keyboard/keyboard+headpointer) and then the specific device within each of them; the features required for each second level classifier might also be different. Furthermore, we intend to implement the online linkage to the detected device with the adaptations defined for each type of device. Furthermore, we claim that the same methodology can be used to automatically detect the types of user characteristics or problems the users are having and we plan to complete the system so that it is able to do it.

Acknowledgements. This work was funded by the University of the Basque Country (UPV/EHU), general funding for research unit Basque Advanced Informatics Laboratory (BAILab) (grant UFI11/45); by the Ministry of Economy and Competitiveness of the Spanish Government and the European Regional Development fund- ERFD (eGovernAbility project, TIN2014-52665-C2-1-R). The authors are grateful to the Elkartu association of people with physical disabilities and, specially, to the study participants.

References

1. Dillon, A.: Beyond usability: process, outcome and affect in human computer interactions. Canadian Journal of Information Science **26** (2001) 57–69
2. Liu, B.: Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data (Data-Centric Systems and Applications). Springer-Verlag New York, Inc., Secaucus, NJ, USA (2006)
3. Valencia, X., Pérez, J.E., Muñoz, U., Arrue, M., Abascal, J.: Assisted Interaction Data Analysis of Web-Based User Studies. In: INTERACT 2015: 15th IFIP TC 13 International Conference, Bamberg, Germany, 2015,. Springer International Publishing, Cham (2015) 1–19
4. S. Garca, J. Luengo, F.H.: Data Preprocessing in Data Mining. Springer International Publishing Switzerland (2015)
5. Hall, M.A.: Correlation-based Feature Subset Selection for Machine Learning. PhD thesis, University of Waikato, Hamilton, New Zealand (1998)
6. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artificial Intelligence **97** (1997) 273 – 324 Relevance.
7. Witten, I.H., Frank, E., Hall, M.A.: Data Mining: Practical Machine Learning Tools and Techniques. 3rd edn. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2011)
8. John, G.H., Langley, P.: Estimating continuous distributions in bayesian classifiers. In: Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence. UAI'95, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (1995) 338–345
9. Aha, D.W., Kibler, D., Albert, M.K.: Instance-based learning algorithms. Machine Learning **6** (1991) 37–66
10. Platt, J.C.: Advances in kernel methods. MIT Press, Cambridge, MA, USA (1999) 185–208
11. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1993)