WHO IS WHO WITH BEHAVIORAL DATA

AN ALGORITHM TO ATTRIBUTE THE DEVICE'S NAVIGATION TO USERS SHARING THE SAME DEVICE

Carlos Ochoa – Netquest
Chief Client Officer
cochoa@netquest.com
www.netquest.com

Carlos Bort – Netquest
Head of Data Science
carlosebort@gmail.com
www.netquest.com

Miquel Porcar – Netquest
Data Scientist
mporcar@netquest.com
www.netquest.com

Abstract - Passive data is powerful but still faces many challenges to gain trust as a way to understand people's online behaviors. One major challenge is separating the data from several individuals sharing one single browsing device. Existing solutions to overcome this difficulty are clearly unsatisfactory. A new method to separate navigation data without asking users, preserving the passive nature of the data, is explored in this paper.

INTRODUCTION

Data fuels market research; insight generation is impossible without relevant and appropriate data to support it. Fortunately, we are seeing tremendous growth in the amount of new sources and the diversity of data available for research, at a low cost that used to be unimaginable.

Any technological disruption is as much an opportunity as it is a challenge, no matter the field we are talking about. The market research industry is experiencing a major disruption, and it is no exception to this rule.

For many years, the number of ways of accessing consumer data was limited and stable. With the advent of the internet, researchers started to adopt this new channel to access potential respondents. It was at the beginning of this century and, since then, things have rapidly evolved. In developed markets, most of the data is collected by means of online access panels (ESOMAR, 2016; Baker et al., 2013; Lozar-Manfreda & Vehovar, 2008).

However, despite this considerable progress in the way consumers are accessed, the nature of the collected data has not evolved at the same pace. Offline surveys are increasingly replaced by online surveys; traditional focus groups sometimes are replaced by online focus groups; and the same may be said of many other traditional methods. In other words, the internet has made data collection more cost efficient and fast, but not radically different. At least, up to now.

After nearly 15 years, it is only now that we are witnessing a real revolution in the way researchers use the internet, motivated by several factors: widespread adoption of the social media, irruption of the mobile internet and consolidation of e-commerce, among others. And, of course, the learning curve of the internet adoption has been got ahead. As a result, new data types are now available and new methodologies are being developed on top of them.

Passive online data collection has emerged as one of the most promising groundbreaking methodologies. One of its most powerful variants is the installation of an online meter on the browsing devices of members of an online access panel to record information on their online behaviors (visited websites, apps usage, search terms), as well as their opinions (via survey).

Passive data has proved to have an edge over survey data when researching online behaviors, overcoming (1) human memory limitations and (2) lack of sincerity (Revilla, Ochoa, Loewe and Voorend, 2015). However, passive online data collection still faces several challenges that prevent broader adoption. First, the large amount of data generated per individual makes the analysis complex; additionally, some of the uses being given to such data requires new analytical methods, as the traditional are facing significant constraints. Second, individual navigation may be spread across different devices (smartphone, tablet, personal PC, professional PC…) which would require installing a meter in all the individual's devices to get the full picture. Finally, some browsing devices are shared among several users, preventing to know for certain which browsing information comes from each one.

This latter issue produces serious discomfort to researchers, as it is an objective distortion of the data. Existing solutions do not enjoy widespread support due to several drawbacks: reduced representativeness, increased measurement error or lack of transparency on how they work. In fact, some solutions may be worse than the problem they try to solve.

This paper presents a completely new approach to overcome the user identification problem: separating individual's navigation by means of an algorithm that just looks at the data. As it will be shown, succeeding in doing so is only possible if browsing information is a personal trait, something unique that unequivocally identifies each individual the same way a fingerprint does (PII).

This paper is organized in several sections:

In **section I,** information on the data used to carry out this research is shared.

In **section II**, existing solutions and their limitations are reviewed.

In **section III**, the key hypothesis that must be valid to make our purpose possible is detailed (i.e. the way each individual browses the internet is unique), as well as some data supporting it.

Section IV provides a detailed description of the proposed algorithm, while **section V** shares its results.

In **section VI** we will explore how some limitations of this solution could be overcome, suggesting further research to improve results.

SECTION I. THE DATA

Data source

We use data from the Netquest's Behavioral Panel in US, UK and Spain. The final algorithm was trained on data from the Spanish panel, as it has been recording behavioral data for a longer period of time.

These behavioral panels are built on top of existing online access panels, so online behavioral and survey data can be collected from the same sample of individuals. To do so, a subsample of the access panel is invited to install tracking software (from now on called the "meter") on their browsing devices (PCs, tablets and smartphones). The meter collects data on the individuals' online activity, such as URLs of the visited webpages, time of the visits, and app use in the case of mobile devices.

As all the metered panelists are also members of the survey panel, their basic sociodemographic information is known as well as some profiling data on different topics (e.g. automotive, healthcare, FMCG, etc.). When regular panelists are invited to install the meter, they are asked to complete an installation survey that asks how many devices they use to browse the internet and, for each device, (1) type of device, (2) main use of the device (personal/professional) and (3) whether it is shared or not. Panelists can install the meter in all their devices and they are rewarded for each one (up to three different devices). However, they are not obliged to track all their devices.

We are interested in panelists that have installed the meter in a shared PC or tablet. Although mobile devices can be shared occasionally, they are mainly single-person devices.

Definitions

Behavioral data produced by a meter is a record of visited webpages, like the one shown in figure I.

Start data and time	**Webpage URL**
2016-03-04 T19:04:48	http://www.google.com
2016-03-04 T19:04:56	https://www.google.com/search? q=bestsellers+2017
2016-03-04 T19:05:25	https://www.amazon.com/
2016-03-04 T19:05:42	https://www.amazon.com/gp/site-directory
2016-03-04 T19:05:58	https://www.amazon.com/books-used-books-textbooks/b/
…	

Figure I. An example of the behavioral data collected by a meter, also known as clickstream. Data has been simplified: additional metadata is also recorded, such as device type, user id, etc.

For the sake of clarity, we define here two key words that will be used throughout this paper.

Webpage/URL: A webpage is a particular file on the Internet that can be accessed by an individual through a browser. A webpage is described by a URL, an address that univocally identifies a webpage (www.amazon.com/help/display.html). The terms webpage and URL will be used interchangeably from now on.

Website/Domain: A website is a connected group of webpages regarded as a single entity, under the same domain name. So, https://www.amazon.com/gp/site-directory and https://www.amazon.com/books-used-books-textbooks/b/, are two webpages under the same website, described by the domain name amazon.com. The terms website and domain will be used interchangeably from now on.

The dataset

We have data available from our target group (shared PCs and tablets). However, we cannot produce a validation dataset to train and test an algorithm. For our purpose, a validation dataset would be a collection of visited webpages from a shared device, each webpage properly labelled as belonging to the right user. Without a validation dataset, it is not possible to measure how accurately an algorithm separates navigations.

This is precisely one of the main obstacles of this work: the lack of a validation dataset. In order to get one, we should rely on some of the already existing methods to separate navigations; however, those methods' accuracy is under suspicion.

To overcome this difficulty, an **artificial validation dataset** was used. It was built by mixing two individual's navigations from non-shared devices, as if both individuals were sharing the same device. Knowing who is the real author of each webpage visit, allows us to use this dataset to measure how well a classification algorithm performs in identifying the right user.

In particular, the artificial dataset used to test the algorithm was built by a two-stage sampling process:

- Stage 1: a random sample of N=200 individuals was drawn from the Spanish Behavioral Panel.
- Stage 2: 1,000 couples of navigations were mixed by randomly selecting couples of panelists from the initial sample.

This approach limits the validity of the results that have been obtained:

- Two real users sharing the same device might have navigations more similar than two random users selected from the panel. In other words, this work aims to contribute to solving the shared devices issue by proving that two independent user's navigations can be separated; the next step will be to test this solution on two navigations coming from the same device.
- We have focused our research on separating two navigations, while a browsing device might be shared by three or more individuals.

In the final section, some considerations will be shared on how these limitations could be overcome.

Sessions

The algorithm described in section IV uses the concept of browsing session. A session is defined as a group of successive webpage visits, in a way that the time lapse between consecutive visits is shorter than a timeout parameter.

As will be detailed later, the classification algorithm assumes that all the visits within a session belong to the same user. This information is key for the accuracy of the output. So, the timeout (time difference to consider a new session) is a tuning parameter of the model and not an intrinsic feature of the data; that is, we need to define the timeout in the most convenient way to maximize the accuracy of the algorithm.

Tuning parameters such as the timeout can be adjusted using different solutions: “A general approach that can be applied to almost any model is to define a set of candidate values, generate reliable estimates of model utility across the candidates’ values, then choose the optimal settings.” (Kuhn and Johnson, 2010).

The parameter tuning process should be part of the algorithm creation. The optimal timeout value: (1) places as many webpage visits in the same session as possible, but (2) limiting the risk of grouping together visits from different users. This could be called the information-precision trade-off.

However, we cannot find the optimal timeout through our artificial data. Our dataset is made up from pairs of independent navigations mixed together, so it does not provide relevant information to find the right balance in the information-precision trade-off.

In view of that fact, we decided to use a timeout of 30 minutes, following a standard used by popular analytical tools such as Google Analytics (https://support.google.com/analytics/answer/2731565). The resulting sessions were manually inspected, ensuring that the division of the webpage visits among sessions made sense.

SECTION II. EXISTING SOLUTIONS

To our knowledge, three main solutions have been proposed to overcome the shared device issue.

The **first solution is to limit the data collection to non-shared devices**. This way, misclassification is avoided but at the expense of reducing the data availability and introducing sample coverage error (i.e. people sharing devices may be different from people not sharing devices).

The **second option is to add a “login dialog” to the meter**; so, each time the user starts a browsing session (or the browser has been inactive for some time), a pop-up message asks about his/her identity. Theoretically, this solution ensures that each webpage visit recorded by the meter is attached to the right user. In practice, serious doubts arise on the reliability of this identification method. Ultimately, asking people about their identity while they use the internet may violate the passive nature of the data, producing: increased churn rate of the participants and misreported identities due to both lack of attention when using the login dialog and social desirability. One of the goals of collecting data passively is to observe people’s activity without affecting their behaviors; by adding a login dialog this benefit might vanish.

Finally, some companies claim that they identify the user behind the device by **analyzing his/her keyboard keystroke pattern**. These companies do not disclose details on how this technique works and how well it performs, a fact that may cause distrust among researchers. Even if we accept this technique is truly effective, the rapid evolution of the browsing devices may challenge its further development: new touch keyboards, autocomplete features in the address bar of the browsers, voice typing, etc. On top of that, browsers are evolving towards greater control of which data is shared with third party applications; so, in the future, using metadata (such as keystroke data) might be problematic.

Table I summarizes the pros and cons of each solution.

Technique	**Pros**	**Cons**
Avoid non-shared devices	• Simplicity. • No misclassification errors.	• It skips the problem, does not solve it. • Representativeness issue: lack of data from shared devices. • Valuable data is not used.
Login dialog	• It would be perfect... if users were perfect. • Easily applicable to more than two users.	• It violates the passive nature of the data: people are constantly aware of being tracked. People may hide part of their navigation. • Users' misuse (lack of attention when selectin the identity in the login dialog) may produce more harm than good.
Analysis of the keyboard keystroke patterns	• Non-intrusive.	• Lack of transparency on how it works. • Challenged by devices' interface evolution.

Table I. Pros and cons of existing solutions to separate navigations.

A new approach, like the one proposed in this paper, would enjoy clear advantages over the existing solutions:

- Simplicity: separation is achieved by just inspecting the data.
- Pure passive: panelists are not asked to provide information while navigating.
- Robustness against future technological limitations that may restrict the possibility of collecting metadata.

Lastly, a final thought about how these solutions might evolve in the future: new technical capabilities can make separation techniques not necessary. Passive facial recognition is a good candidate; Apple is the most recent technology company who is utilizing facial recognition to identify device users to unlock phones. Such system could be used to separate navigations. However, it may be perfectly possible as well that such information is not available for third party applications running on the device, as it is currently happening with other sensitive information.

SECTION III. MAIN HYPOTHESIS

To what extent is the way you browse the internet different from the way others do? This is the key question whose answer determines whether our approach makes sense or not. And, of course, our initial hypothesis is that the answer to this question is yes.

A simple exploration of the data at hand (sample from the Behavioral Netquest Panel described in section I) helps to support this hypothesis (table 2).

Sample size N	200
Country	Spain
Type of device	Non-shared PC or Tablet
Time period	1 month (June 2016)
For the whole sample…	
Sessions	77,842
Visited webpages	1,259,076
Visited domains	363,929
Unique visited domains	17,309
For each panelist average [minimum – maximum]	
Sessions	389.2 [5 ⟷ 1,345]
Webpages per session	16.2 [1 ⟷ 1,708]
Domains per session	4.7 [1 ⟷ 67]
Unique visited domains in one month	175.2 [4 ⟷ 915]
For each couple of panelists…	
% shared domains	4.0% [0% ⟷ 25%]

Table 2. Dataset description. Data for panelists and couples of panelists is shown in the format "average [minimum ⟷ maximum]".

People in the sample visited 175.2 different domains on average in a month, ranging from 4 to 915. But the relevant fact to our purpose is that a pair of randomly chosen panelists from the sample only share 4% of their unique visited domains.

This fact, that supports our hypothesis, could seem somehow surprising. It is well known that websites such as Google and Facebook capture most of the Internet traffic in the Western world. In fact, 98.5% of the panelists in the sample have visited Google while 86.0% have visited Facebook in the time period under analysis. However, that does not mean that people only browse such popular websites. Figure 2 shows which percentage of the panelists have visited (at least once) each of the different domains that are present in the data.

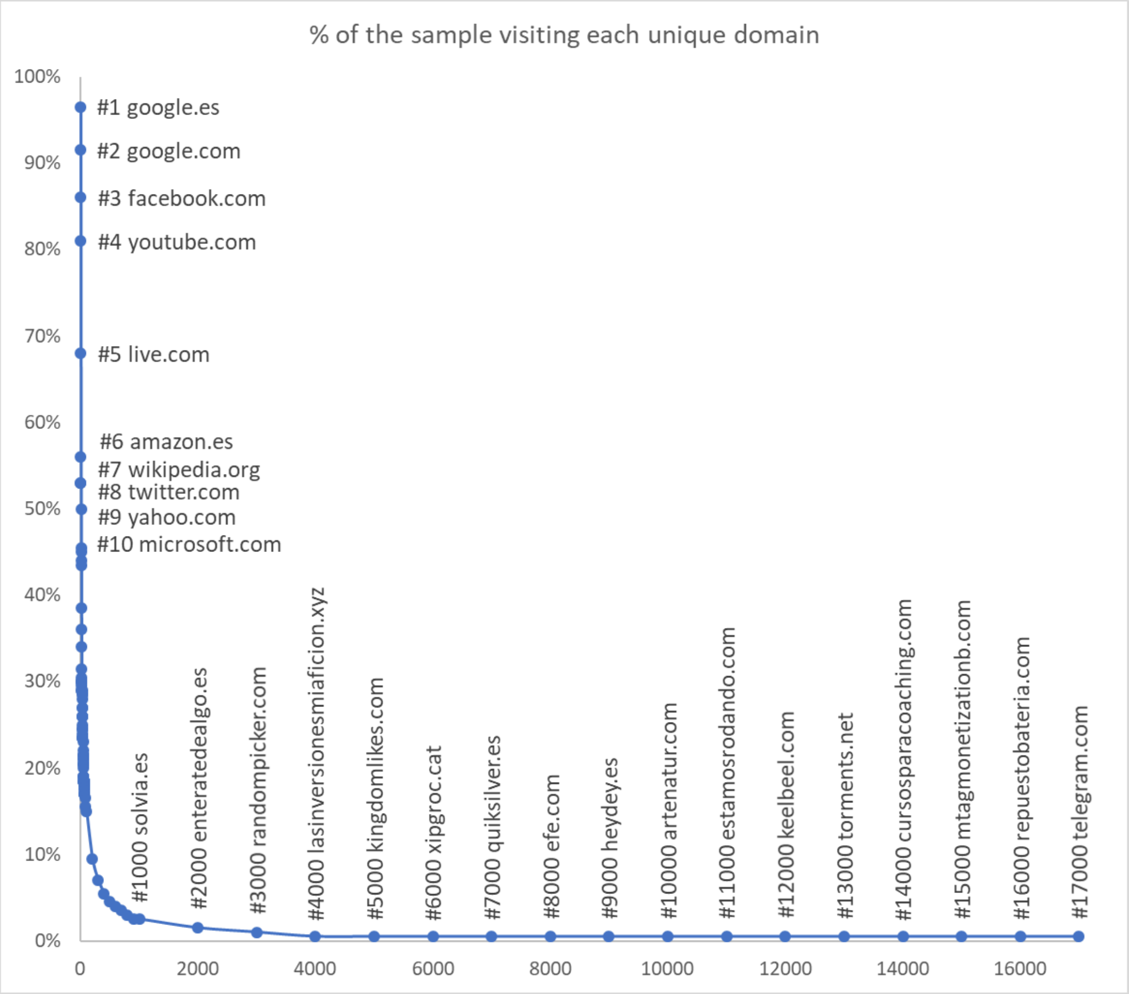


Figure 2. Popularity of each website among the sample. Just a few websites are visited by most of the panelists while a long tail of domains are visited by very few panelists.

Figure 2 follows a typical Pareto distribution: just a few domains are visited by most of the panelists, while there is a long tail of less popular domains visited by a small part of the sample. Popular domains play a minor role to distinguish users; rare domains are the ones that can be crucial to achieve our goal. Our algorithm aims to exploit this opportunity.

SECTION IV. THE ALGORITHM

Choosing learning type

Our purpose is to design a classification algorithm that gets as input the mixed navigation from two users (A and B) and returns each visited webpage properly labelled as belonging to user A or B.

A first decision to be made in pursuing this goal is deciding what type of algorithm should be implemented. Machine Learning literature usually classify algorithms in three broad categories, depending on the way the algorithms learn: supervised, unsupervised and reinforcement learning algorithms.

Reinforcement learning algorithms were rapidly discarded. Reinforcement is a powerful learning method, but such algorithms are “not given examples of optimal outputs (…) but must instead discover them by a process of trial and error” (Bishop, 2006). This learning process requires a reward system: the algorithm needs to know if each step made contributes or not to a success metric. But unfortunately, our problem does not provide such rewards.

On the other hand, **supervised learning** requires training data that “comprises examples of the input vectors along with their corresponding target vectors” (Bishop, 2006). This is precisely what we lack; and what we have tried to replace with the artificial data described in section I.

Supervised learning assumes that a causality relationship exists between some input factors and the classification criteria. Such learning offers some advantages compared to unsupervised learning. There are many and powerful supervised algorithms at hand that have proved to perform extremely well in a wide variety of problems (Kuhn and Johnson, 2013): Classifications Trees, Random Forest, Boosting, Support Vector Machines, etc.

However, after a few attempts to train one of these algorithms, it soon became evident that it was not the right approach. It was pretty easy to train a supervised algorithm to separate the navigation from two particular individuals by using a specific training dataset from these individuals. But the result cannot be generalized to other pairs of individuals, a problem known as overfitting (Kuhn and Johnson, 2013). In other words, if we had data from two specific individuals correctly classified, we could train a specific model for them to classify their future navigation. But this model cannot be used for other individuals.

One key learning from this unsuccessful attempt is the following: while each individual uses the internet in a unique way, it is not easy at all to predict which is this unique way based on personal characteristics.

So, in light of the evidence, an **unsupervised algorithm** was the only option available. As explained by Bishop (2006), “In other pattern recognition problems, the training data consists of a set of input vectors x without any corresponding target values. The goal in such unsupervised learning problems may be to discover groups of similar examples within the data, where it is called clustering (…)”. That description fits perfectly with our problem: regardless of the factors that explain why people browse some websites rather than others, we just want to identify groups of “similar websites” in the hope that this will reveal the identity of the individuals behind.

The solution: an ensemble of different algorithms

A description of the unsupervised algorithm developed to separate navigations is provided below, step by step. All the data manipulations and algorithms have been programmed in R (www.r-project.org), using public libraries.

To facilitate the reader's comprehension, the description is backed with real examples.

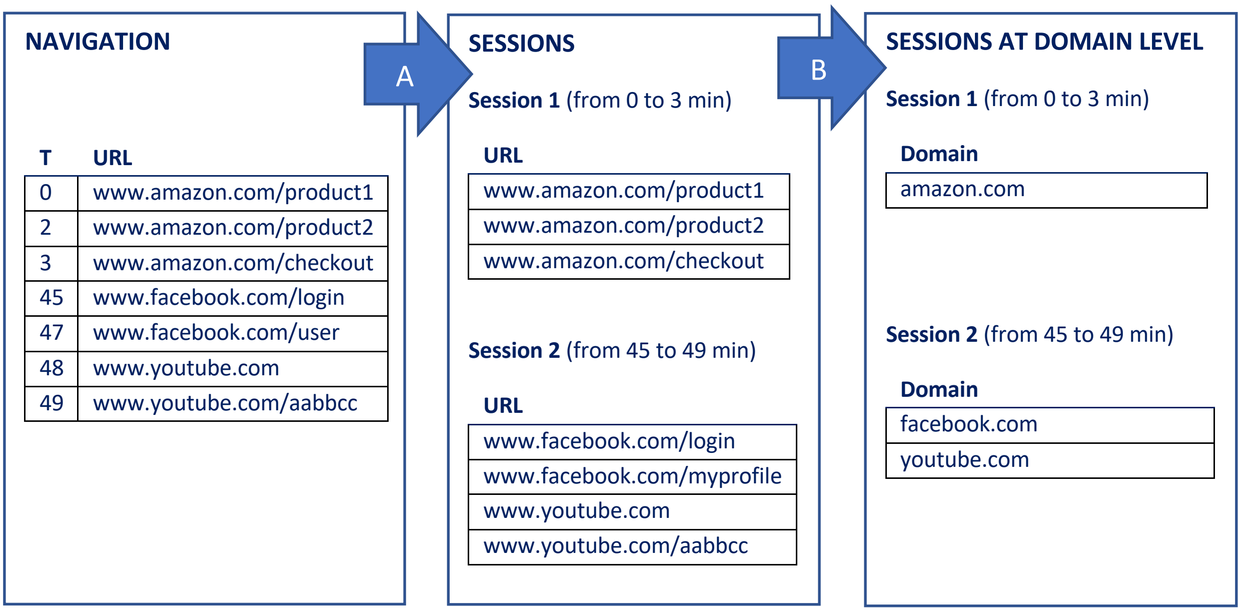
Step 1: Dimension reduction

Navigation data consists of a list of complete URLs, each one formed by a domain name (e.g. amazon.com), sometimes a subdomain (e.g. aws.amazon.com, www.amazon.com) and a page descriptor (e.g. www.amazon.com/cell-phones-service-plans-accessories/b/ref=nav_shopall_wi).

We have decided to focus our analysis on domain names. For our purpose, all this information around the domain name (i.e. subdomains and page descriptors) adds much more complexity. As the whole idea behind the separation is to exploit coincidences in the same session, working with precise URLs would require much more data to train an algorithm.

The same applies to multiple visits per domain. We could potentially separate two users visiting the same domain if one user tends to visit many pages in the same session and the other one just a few. But this information is much less relevant than the simple fact of whether a domain has been visited or not, so we decided to not use it.

So finally, data is reduced to domain level: a complete navigation is transformed in a list of sessions (figure 3 – A), and each session is transformed into a list of unique domains visited in that session (figure 3 – B). Once the session is properly assigned to the right user, all the suppressed information around the domain can be recovered in order to assign webpage visits to each user.



NAVIGATION

T	URL
0	www.amazon.com/product1
2	www.amazon.com/product2
3	www.amazon.com/checkout
45	www.facebook.com/login
47	www.facebook.com/user
48	www.youtube.com
49	www.youtube.com/aabbcc

SESSIONS

Session 1 (from 0 to 3 min)

URL
www.amazon.com/product1
www.amazon.com/product2
www.amazon.com/checkout

Session 2 (from 45 to 49 min)

URL
www.facebook.com/login
www.facebook.com/myprofile
www.youtube.com
www.youtube.com/aabbcc

SESSIONS AT DOMAIN LEVEL

Session 1 (from 0 to 3 min)

Domain
amazon.com

Session 2 (from 45 to 49 min)

Domain
facebook.com
youtube.com

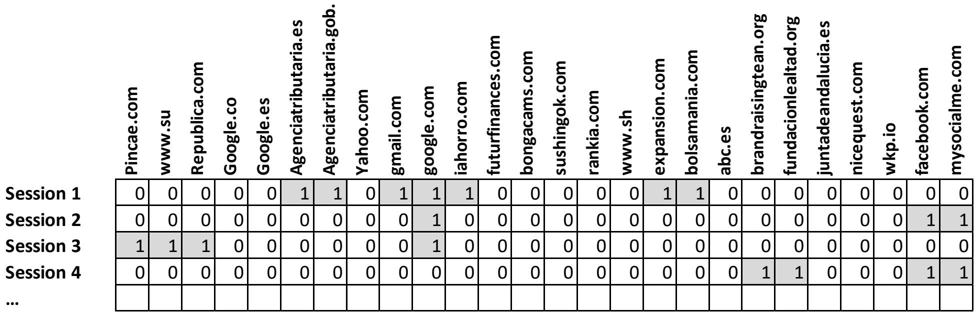
Figure 3. Dimension reduction: navigation data is first grouped into sessions; then, URLs are reduced to unique domains per session.

Step 2. Similarity matrix

A similarity matrix is evaluated for the list of unique domains present in the navigation data. Each cell of this matrix contains a measure of how likely two domains appear together in a session, what is a sort of correlation.

Several ways to create such matrix were tested, without relevant differences in the result. So, the following simple method was finally employed:

(1) First, the browsing data for each couple of panelists is binary coded in a matrix ***M***. This matrix has as many rows as sessions and as many columns as unique domains in the joint navigation. So, each row is a sequence of ones and zeros that represents whether each possible domain is present at each session (Figure 4).



	Pincae.com	www.su	Republica.com	Google.co	Google.es	Agenciatributaria.es	Agenciatributaria.gob.	Yahoo.com	gmail.com	google.com	iahorro.com	futurfinances.com	bongacams.com	sushingok.com	rankia.com	www.sh	expansion.com	bolsamania.com	abc.es	brandraisingtean.org	fundacionlealtad.org	juntadeandalucia.es	nicequest.com	wkp.io	facebook.com	mysocialme.com
Session 1	0	0	0	0	0	1	1	0	1	1	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
Session 2	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
Session 3	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Session 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1
…																										

Figure 4. Matrix M binary codifies sessions and unique domains.

(2) Once ***M*** is built, the similarity matrix ***S*** is calculated by means of a matrix product $\boldsymbol{M^T M}$. The similarity matrix ***S*** has the following properties:

a. Each cell represents the similarity of a pair of domains. So, the matrix is symmetric.

b. The minimum value of each cell is zero, that means that both domains never appear together in a browsing session (minimum similarity).

c. The maximum value of each cell is the number of times both domains appear together in a browsing session.

d. The diagonal of that matrix represents the similarity of each domain with itself. Because of the matrix **M** is computed, each diagonal cell equals the number of times each domain appears in the navigation. It can be removed as it does not provide useful information.

Domains that appear together more frequently in the sessions will score high in the similarity matrix ***S***. As we assume that sessions are owned by a single user, high similarity indicates high likelihood of belonging to the same user. A visualization of such matrix is shown in figure 5 for a couple of individuals.

Theoretically, the above procedure to create the similarity matrix has some drawbacks. For instance, domains that appear more often tend to score higher. In other words, the matrix is not normalized. For instance, say that a pair of domains A and B appear just two times in the navigation but always together, while another couple of domains C and D appear tens of times but only two times together. Both pairs A-B and C-D will get the same similarity score (2), while it seems clear that A-B are more similar than C-D.

Different strategies to overcome this alleged limitation were tested. Even though some of these strategies produced similarity matrices in greater accordance with what we may expect, none of them improved the final performance in the ultimate goal of the algorithm: correct separation of individual’s navigations.

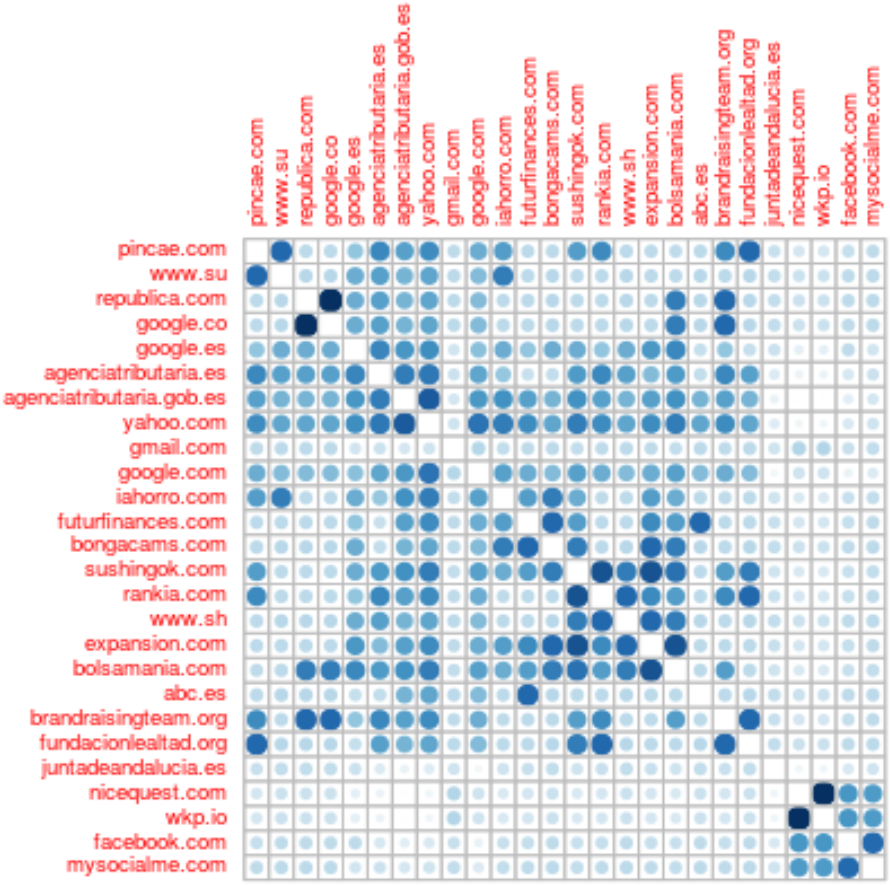


Figure 5. Visual representation of a similarity matrix of unique domains for a couple of individuals. Dark areas mean high similarity between domains; that is, domains that frequently appear together in the same sessions.

Step 3. Multidimensional scaling

The similarity matrix ***S*** tells which domains appear together more often and which ones do not. But we aim to combine this pair wise information to get a global picture. Could we spread the domains onto a plane, so the similar domains are placed together and the dissimilar ones are placed distant? If so, we could separate two groups of domains in the hope that each group belongs to a different individual.

An intuition on how domains can be placed in a plane in such way is the following:

(1) First, the similarity matrix ***S*** can be transformed into a distance matrix ***D*** by inverting each cell one by one (***D***=1/***S***). So instead of a measure of similarity, we get a measure of dissimilarity: the higher the value in a cell, the less likely two domains appear together in a session, the less likely they both belong to the same individual.

(2) Place the first unique domain in the center of a plane.

(3) Take the second domain and place it at a distance from the first one according to the information in the distance matrix ***D***.

(4) Things get more interesting with the next domains; if you try to proceed in a similar way as with the second domain, the distance with the first and the second domain must be considered at the same time. But both distances may be incompatible, so a compromise among different distances must be reached.

(5) This process gets increasingly complex as more domains are placed in the plane because the coherence between pair wise distances gets harder. So, for instance, when placing the 6th domain, its distance with the 1st, 2nd, 3rd, 4th and 5th domains must be balanced.

Fortunately, this placement of domains in a physical space can be done using a well-stablished algorithm, a Multidimensional Scaling (MDS; Young, 1987). An MDS algorithm aims to place several elements in a N-dimensional space such that the between-elements distances are preserved as well as possible. Each element is then assigned coordinates in each of the N dimensions.

How a MDS works can be seen with a simple example. Consider the distances between nine American cities (table 3).

	BOS	**CHI**	**DC**	**DEN**	**LA**	**MIA**	**NY**	**SEA**	**SF**
BOS	0	963	429	1,949	2,979	1,504	206	2,976	3,095
CHI	963	0	671	996	2,054	1,329	802	2,013	2,142
DC	429	671	0	1,616	2,631	1,075	233	2,684	2,799
DEN	1,949	996	1,616	0	1,059	2,037	1,771	1,307	1,235
LA	2,979	2,054	2,631	1,059	0	2,687	2,786	1,131	379
MIA	1,504	1,329	1,075	2,037	2,687	0	1,308	3,273	3,053
NY	206	802	233	1,771	2,786	1,308	0	2,815	2,934
SEA	2,976	2,013	2,684	1,307	1,131	3,273	2,815	0	808
SF	3,095	2,142	2,799	1,235	379	3,053	2,934	808	0

Table 3. Distance between nine US cities.

Running a MDS on this data, a pair of coordinates is assigned to each city. Once represented in a graph, the position assigned to each city approximately reproduces the shape of the US map (figure 4). In other words, the best representation on a 2-dimensional space of the distances between cities is the United States of America.

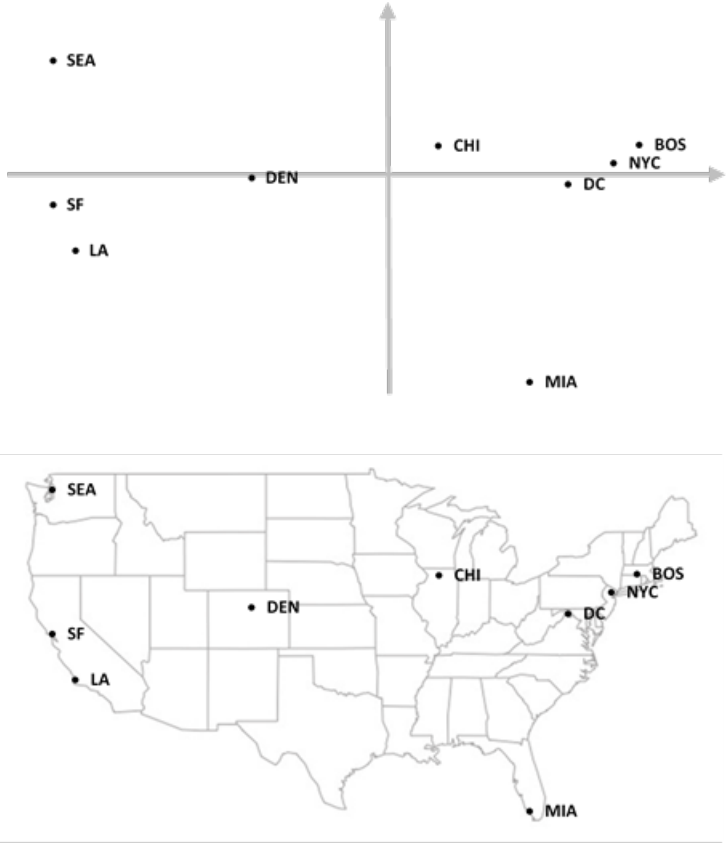


Figure 4. The MDS procedure locates the cities in a plane in a way that between-cities distance is preserved as much as possible. This results in the real relative location of each city.

The same MDS procedure can be applied to the between-domains distance matrix **D**. As we want to identify just two groups of domains, we can use a 1-dimensional plane (N=1), that is, a simple line. If the hypothesis supporting this work is valid, domains that are visited only by user A should be placed at one end of the line, while domains visited only by user B should be placed at the opposite end of the line. Domains that are visited by both users should be placed in the half-way.

Considering the midpoint of the line (coordinate x=0) as the threshold to separate both users, the distance to this midpoint can be considered a propensity score. Domains with large scores (positive or negative) are highly likely to be visited by only one of two users; these are highly **discriminant domains**. On the contrary, domains with scores close to zero are likely to be shared between both users and therefore, **non-discriminant**.

Figure 7 shows an example of the resultant scores for a couple of users.

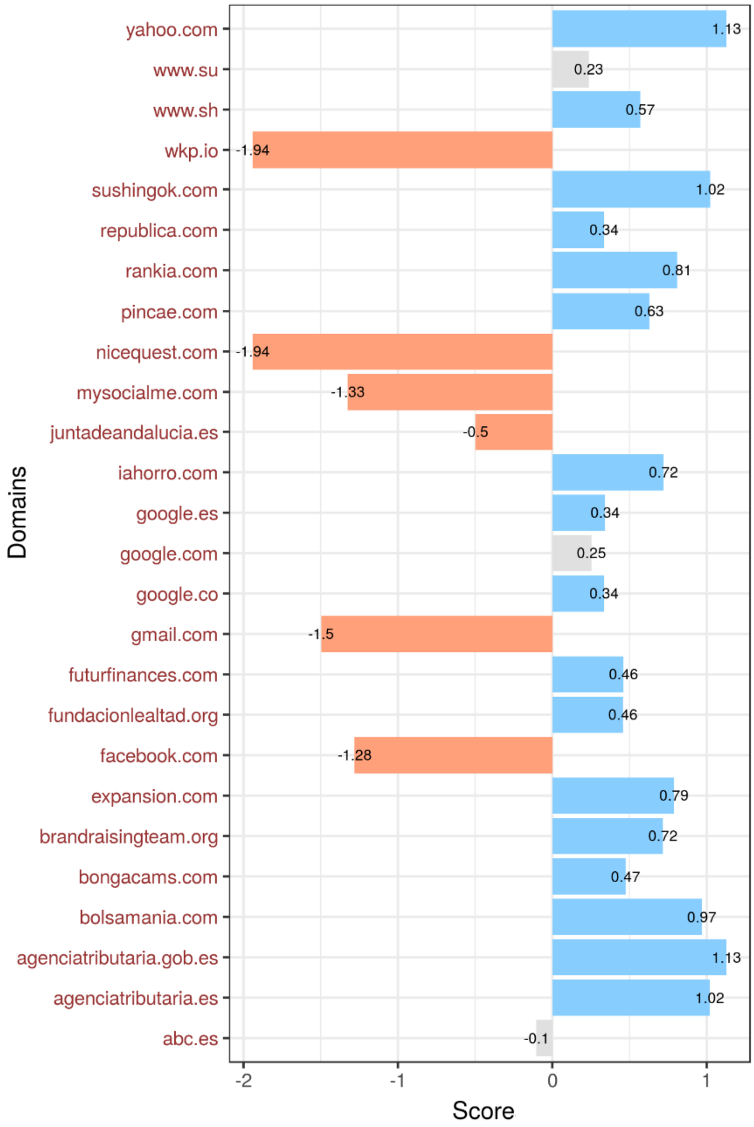


Figure 7. Propensity scores of a pair of users. Large scores are associated to discriminant domains (blue and orange), while small scores (grey) to non-discriminant ones.

Step 4. Classification of sessions

Up to step 3 we have found a first classification criterion at domain level: positive scores are assigned to user A and negative ones to user B. However, this criterion is too extreme: even close-to-zero score domains (non-discriminant) are assigned the same way than large score domains are. For instance, according to the data shown in figure 7, the domain google.com (score +0.25) should be assigned always to user A, while it is likely that both users visit google.com.

The accuracy can be improved by computing average scores per session. For instance, consider a session with four domains (facebook.com, gmail.com, mysocialme.com and republica.com) with respective scores -1.28, -1.5, -1.33 and 0.34. Consider also that positive scores are assigned to user A and negative ones to user B. When assigned at domain level, facebook.com, gmail.com and mysocialme.com are assigned to B and republica.com to A. But if we evaluate the average score of the session (-0.9), the four domains are assigned to B (see Figure 8).

This procedure allows to assign non-discriminant domains more accurately, taking advantage of the fact that they are part of a session that may include discriminant domains. The larger a session is, the more effective is this method.

Of course, short sessions with non-discriminant domains are more likely to be misclassified. Fortunately, short sessions impact much less in the global accuracy.

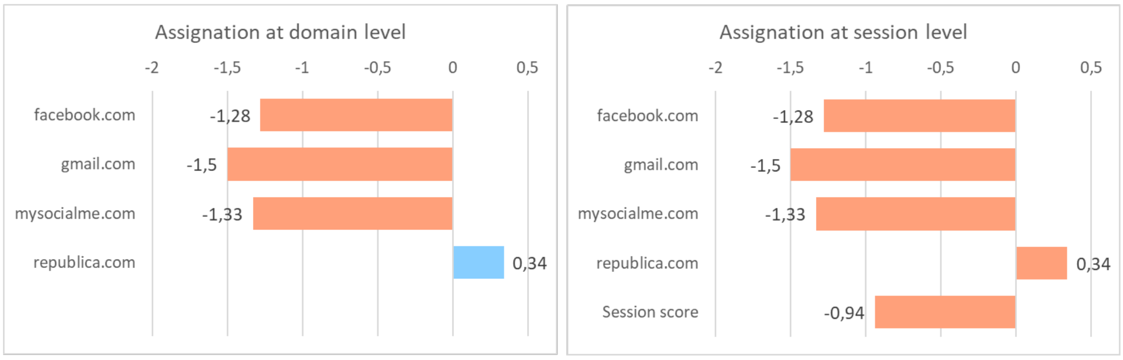


Figure 8. When considered at domain level, each domain is assigned based on its score, even those with small scores. When considered at session level, all four domains are assigned together. In this case, republica.com is assigned to user B (orange) despite its score is positive.

Step 5. Who is the panelist?

One final step is missing. Up to step number 4 we have separated domains in two groups, A (positive scores) and B (negative scores). But, who is the user we are interested in?

If we certainly know that the user we are interested in (target user) visits a particular discriminant domain, this is enough to decide. We call this domain, the one that is visited exclusively by the target user, the **hook**. If the hook is positive, the target user is A, otherwise is B.

But, how can we get such domain from the target user? Fortunately, when installing the meter in an online panel such as Netquest, a hook is always available: the domain of the panel website, the one accessed when the panelist participates in surveys. As such domain is only visited by the target user, is the perfect hook.

And what about if both users sharing a browsing device are members of the panel? Theoretically, both users would visit the panel domain and it would no longer be a hook. However, we can benefit from the fact that panelists need to log in the panel website to participate in surveys. When users log in into a website, the URL changes specifically for each user, so we can still use the panel domain as a hook; or more precisely, two slightly different versions of the panel domain. For instance, the domain nicequest.com becomes “nicequest.com/userA” for user A, and “nicequest.com/userB” for user B. So, in fact, when both users are panelists, we have two different hooks at our disposal; that makes the user identification even more reliable.

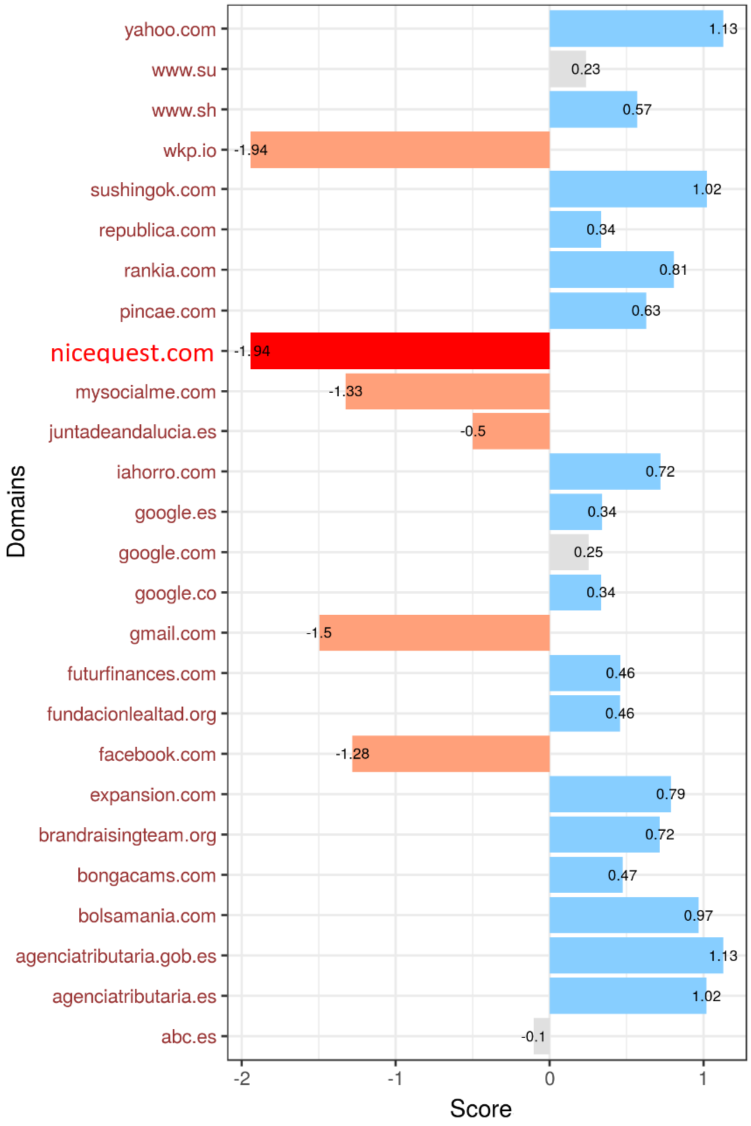


Figure 9. The sign of the hook’s score determines whether the target user is A or B. In this example, the hook (nicequest.com) determines that the target user is B (the one with negative scores).

SECTION V. PERFORMANCE

In order to assess how well this algorithm performs, a success metric must be defined. Several options are available:

- **Session accuracy**: the percentage of sessions (as defined in section I) assigned to the right user.

- **Domain accuracy**: the percentage of unique domains per session assigned to the right user. This metric can be applied directly to the data once the dimension reduction is applied (see section IV, step 2).
- **URL accuracy**: Each domain in the dataset corresponds to several URLs (different webpages from the same domain that the user visits in the same session). Once the domains are classified, the underlying URLs are implicitly classified, resulting in the URL accuracy metric. URL accuracy may differ from domain accuracy only if (1) some domains receive more page visits than others and (2) these domains have a significant different accuracy.

We have computed all the above metrics, but we have used the domain accuracy as the key success metric, the one used to compare algorithms' performance.

Table 4 shows the different resultant metrics for our dataset.

Success metric	**Average**
Session accuracy	84.0%
Domain accuracy	87.3%
URL accuracy	87.5%

Table 4. Performance of the algorithm (different metrics)

A naïve classifier (that is, assigning domains randomly to each user) may reach a 50% domain accuracy, so this is the base accuracy we aim to improve. Considering this fact, 87.3% domain accuracy should be considered a promising result.

Note that the domain accuracy is greater than the session accuracy (87.3% vs 84.0%). This is due to the fact (explained in section IV, step 4) that larger sessions tend to be better classified. URL accuracy, on the contrary, is pretty similar to domain accuracy, meaning that domains that receive more webpage visits are not better classified.

Figure 10 shows the accuracy for each pair of users in the dataset, ordered by accuracy. It is interesting to note that the average accuracy (87.3%) is not the result of a homogenous performance among all the cases; on the contrary, the algorithm seems to work very well (accuracy > 85%) for a near 75% of the cases, while dramatically fails in some cases. In these cases, the algorithm performs even worse than a naïve random classification algorithm.

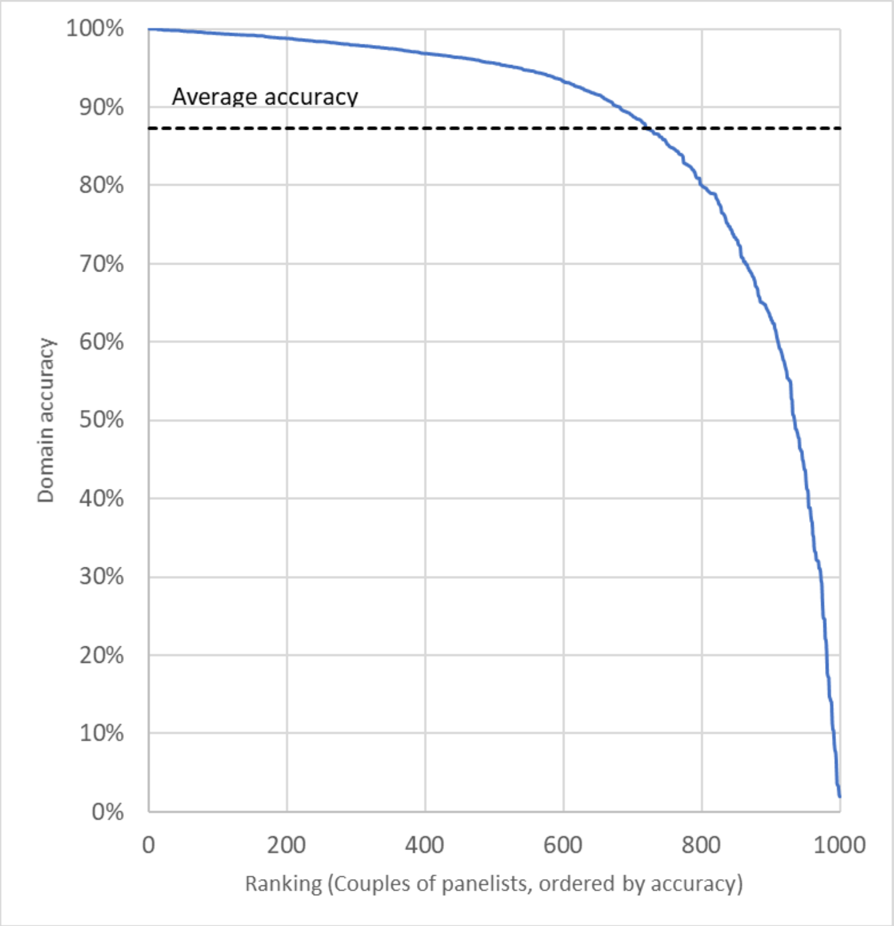


Figure 10. The overall accuracy is a mix of many well classified cases and a few very bad classified ones. If we order the 1,000 cases, most of them perform between 90% and 100% but some perform lower than 50%.

Looking at these problematic cases some insights are revealed:

- In some cases, the hook has been mistakenly classified. The result of this misclassification is harmful: the accuracy goes towards zero (in fact, towards 1 minus the accuracy that should be achieved if the hook were properly classified). A hook misclassification may occur when (1) the panelist has visited the panel website very few times (e.g. he/she has participated in few surveys) and (2) those visits have occurred in very short sessions.
- Some particular cases were found in which the pair of mixed navigations seemed to be produced by three different users instead of two. This fact can be easily visualized by executing the MDS algorithm using two dimensions (a plane) instead of one (a line). Figure 11 shows one of these cases. When doing so, domains are displayed in three different areas that can be easily separated. A possible explanation is that although we are producing the artificial dataset by combining navigations from devices that are allegedly not shared, some of them are actually shared. Some users may have reported misleading information in the installation survey regarding the shared condition of their

devices, or this condition has changed after some time. In fact, this finding reinforces the approach we have taken: relying more in the data that in declared information.

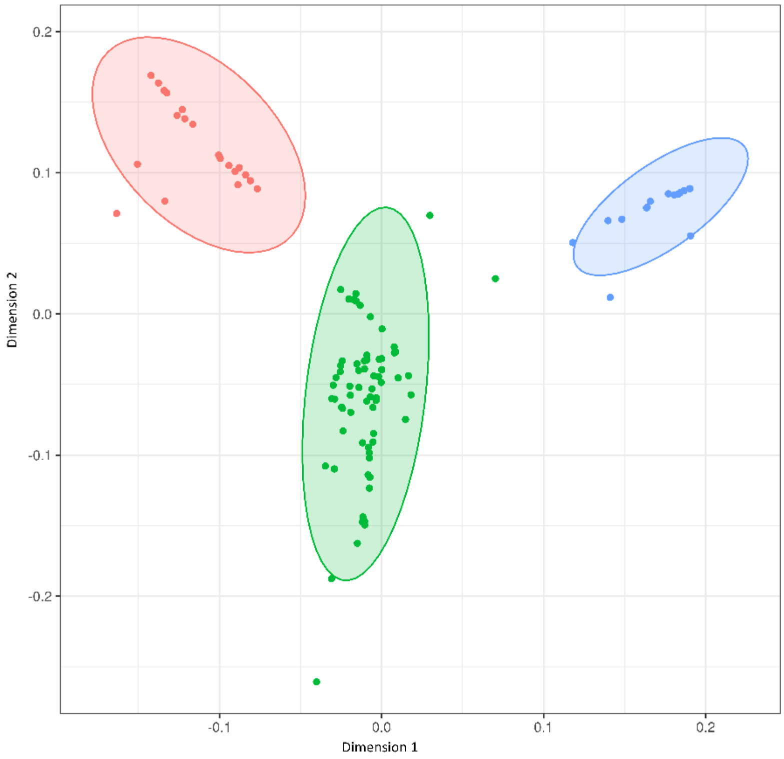


Figure 11. An example of a particular case in the dataset that may be produced by three users instead of two. A 2-dimensional MDS allows to visualize three different regions in the domain space.

SECTION VI. FURTHER RESEARCH AND APPLICABILITY

Our research has been tested on artificial data, as detailed in section I. Despite results are promising, an objection could be made on the fact that we have not tested the algorithm on navigations coming from a real shared device. If people sharing a device navigate in a much more similar way, the main hypothesis that supports this work would be seriously compromised.

A real validation dataset from shared devices would be needed. But we should not underestimate how difficult to obtain one is: whenever we ask people to identify themselves when browsing, we will risk suffering a serious bias as mentioned in section II.

However, even if the algorithm performs worse than expected in real shared devices, it can still be improved to boost the accuracy. The following two techniques deserve further research:

- Instead of working at domain level in general, some non-discriminant domains could be split in subdomains. For instance, “facebook.com” would be a non-discriminant domain if both users visit Facebook. But considering ”facebook.com/userA” and ”facebook.com/userB” as different domains, a non-discriminant domain is transformed into two highly discriminant ones. It is the same strategy suggested to split hooks when two panelists share a domain (section IV, step 5). By applying similar preprocessing techniques to some popular domains, accuracy can be improved.
- Panelists could be asked to provide some additional information on how they browse the internet to improve the algorithm. Questions such as “Could you provide us some websites you usually visit that nobody else at your home does?” would improve accuracy and prevent a wrong identification of the hook. An alternative approach is to ask the user to identify herself in some initial browsing sessions. Although the user may alter the way he/she navigates in that session (e.g. avoiding sensible websites), the collected data could be enough to better train the algorithm. If people are required to identify their browsing sessions occasionally, for calibrating the algorithm, the passive nature of the data would not be compromised and churn rate would be limited.

Beyond its performance, the algorithm offers several advantages that facilitate its usage in production environments, compared to existing solutions.

- It is simple. It can be easily programmed in almost any programming language.
- It does not need to be executed in real time. In fact, it can be executed whenever it is needed, as it only requires (as input) the same data that is going to be processed. The only requirement is to have enough data stored to properly assign each domain to each user.
- The performance of the algorithm improves over time. The longer the period of time under analysis, the better the detection of similarities among domains and the better the accuracy. However, a limit should be set up to account for the possibility that users change their navigation habits. This topic requires further research.

CONCLUSIONS

We have shown that the way people use the internet is a personal trait that, in fact, should be considered as Personal Identifiable Information (PII). We can benefit from this fact to separate navigations of two or more users that are sharing the same metered browsing device.

To prove this concept, we have developed an algorithm that exploits the correlation among domains in browsing sessions. The algorithm was tested on a dataset created for this purpose, by mixing navigations of members of a Behavioral Panel (Netquest).

We have reached 87.3% of accuracy in identifying website visits and 87.5% in identifying URLs visits.

This result opens a new line of research. There is plenty of room for improvement, as was presented in this paper.

Finally, we hope this work launches a debate about how we, as researchers, should approach new methodologies and data sources. The method we have explored to separate navigations is not perfect: some web visits, particularly those that are part of short browsing sessions, may be misclassified. This fact may cause discomfort in some researchers that prefer to work with the apparent certainty that self-reported data provides.

Despite self-reported data on online behaviors may be severely distorted, it is a safe place for researchers: if data makes no sense, just blame the online panel for not being able to recruit honest participants, without bearing in mind that is impossible to be honest when asked to report short and repetitive interactions that are repeated many times a day.

Behavioral data follows what may be called the "Heisenberg's uncertainty principle on behavioral data". Just like the original Heisenberg's principle stated that some pairs of physical properties cannot be known with unlimited precision, a perfect knowledge on online behaviors cannot be reached, while knowing all the surrounding circumstances without uncertainty. To know the latter, we need to ask. When asking, we alter behaviors and data is not passive anymore. New methods as the one presented in this paper offers a way to reduce the uncertainty around the behavioral data, without seeking to eliminate it completely.

Working with new types of data means taking risks; it means working with imperfect datasets, far away from the simplicity of survey data.

REFERENCES

[1] Baker, R., Brick, J.M., Bates, N.A., Battaglia. M., Couper, M.P., Dever, J.A., Gile, K.J., Tourangeau, R., (2013), "Summary report of the AAPOR Task Force on non-probability sampling", *Journal of Survey Statistics and Methodology,* 2013;1:90–143.

[2] Bishop, C. M (2006), "Pattern Recognition and Machine Learning", ISBN-10: 0-387-31073-8.

[3] Kuhn, M., Johnson K., (2010), "Applied Predictive Modeling", ISBN 978-1-4614-6848-6.

[4] Lozar-Manfreda, K. & Vehovar, V. (2008), “Internet surveys”, in E.D. de Leeuw, J.J. Hox & D.A. Dillman (Red.), International handbook of survey methodology. New York: Erlbaum.

[5] ESOMAR (2016), “Global Market Research 2016”, ISBN: 92-831-0282-7.

[6] Revilla, M., Ochoa, C., Loewe G., Voorend, R. (2015), “When should we ask, when should we measure? Comparing information from passive and active data collection”, ESOMAR CONGRESS 2015, ISBN: 92-831-0283-5.

[7] Young, Forrest W. (1987). Multidimensional scaling: History, theory, and applications. Lawrence Erlbaum Associates. ISBN 978-0898596632.