DE SIMONI Pietro
SESSI Michela

Stockholm University

Report of Laboratory Exercise 2 :

Web Usage Mining "Light"

Group 15

DE SIMONI Pietro
SESSI Michela

Introduction

Nowadays, Internet provides a large amount of data, not always structured, but potentially very useful. This data could be transformed in truthful information and become tips to solve problems in the real life.

The aim of this exercise is to show how easy it is to collect user information from the web using freely available resources even although there are no particular programming skills required. We take advantages of Amazon Wishlists and a geocoding service. The idea of the lab is to find potential terrorist or subversives monitoring books preferences.

The report is divided in three sections following the exercise:

1. **Keywords and matches**
2. **Sorting, linking and analyzing**
3. **Finding addresses and mapping**

Content & Discussion

1. Subversive keywords and matches

The first task we did in the lab was downloading a dataset of 6000 amazon wishlists, and some scripts that we later used to analyze them.
A list of keywords was provided, it was supposed to refer to potentially suspicious books like as religious books, or radical movements books. We added to this list *Guerrilla Warfare* and *Carl Marx* as other examples of suspicious readings.
Then we made a search on all the 6000 wishlists, to see how many users had in their wishlist one (or more) of the mentioned books. We got 377 matches, that means around 6 percent of the users in our dataset. This number is probably too high, because if we apply this technique to all the US population (325 million people) we expect proportionally to have about 20 million matches. This number is clearly too big to be of any help. It might be used as a starting point, and the results could be matched with other sources to reduce the number of suspects.

2. Sorting, linking and analyzing

It is useful to sort the matches: we proceeded to get one directory with each keyword and the files that contain these keywords. In other words, there was a folder named as each keyword and inside it all the html matched.

In this way, it’s easy to count file matches for each keywords. The keyword *Bible* is the most popular with 324 matches: we noticed that a lot of books contain the word “bible” in the title even if it is not about the religious Bible. For example we had some matching generated by books like “The programmer’s bible” which was clearly not what we were looking for.

The other keyword reported a number of matches definitely lower, not more than 8. In example we report the keyword *Stuart Mill* (1), *Ralph Nader* (1) and *Slaughterhouse-five* (8). The results of our new keywords are different: *Guerrilla Warfare* obtained 5 matches (so, we found a keyword that can be considered relevant), *Carl Marx* is probably a not useful keyword, in fact it didn’t obtain any matches. Lastly, it is relevant showing that the keyword *GreenPeace* does not match with any wishlist in our database (that we want to remember being composed only by 6000 files).
Anyway, in order to make a good search, it could be useful study the combination of that matches from different keywords. Adding in the wishlist just one keyword book can’t be considered a threat, on the other hand having multiple titles is more suspicious.

Analyzing the html matches, we noticed that the page refers only to the first page of each user’s wishlist. We can imagine that just the first page was downloaded. In Amazon wishlist system books are always added to the front (so to the first page) and older titles are pushed in background in second, third or more pages. Therefore, it is evident that there is a bias in favor of newer books and older books are not investigated.
For a better analysis, we suggest to improve the study downloading all the pages of the wishlists. It is also possible to think about assign weights based on the number of the page: we can assume that users are more interested in the books that they added recently.

3. Finding shipping addresses and mapping users

Amazon wishlist gave users the possibility to share their home address, in order to let other users to ship them their presents. We were provided a script that let us download the state and city of users sharing this information, and we ran it on all users that had a specific keyword in their wishlist. The results were saved in a txt file.
Then using an online free service (available at http://www.batchgeo.com/), we were able to locate the users on a US map.

The only keyword that gave us a number of result sufficient to be worth mapping was bible so we report and comment that one (*Image 1*).
Data are not enough to outline something real: we want to remember another time that we were analyzing only 6000 wishlists and this exercise is for educational purposes only. However, we can see a correlation between our results and the density of the American population. It would be more useful to represent data in relative terms instead of absolute terms: representing a map with the percentage information based on population density would lead to a more interesting analysis.

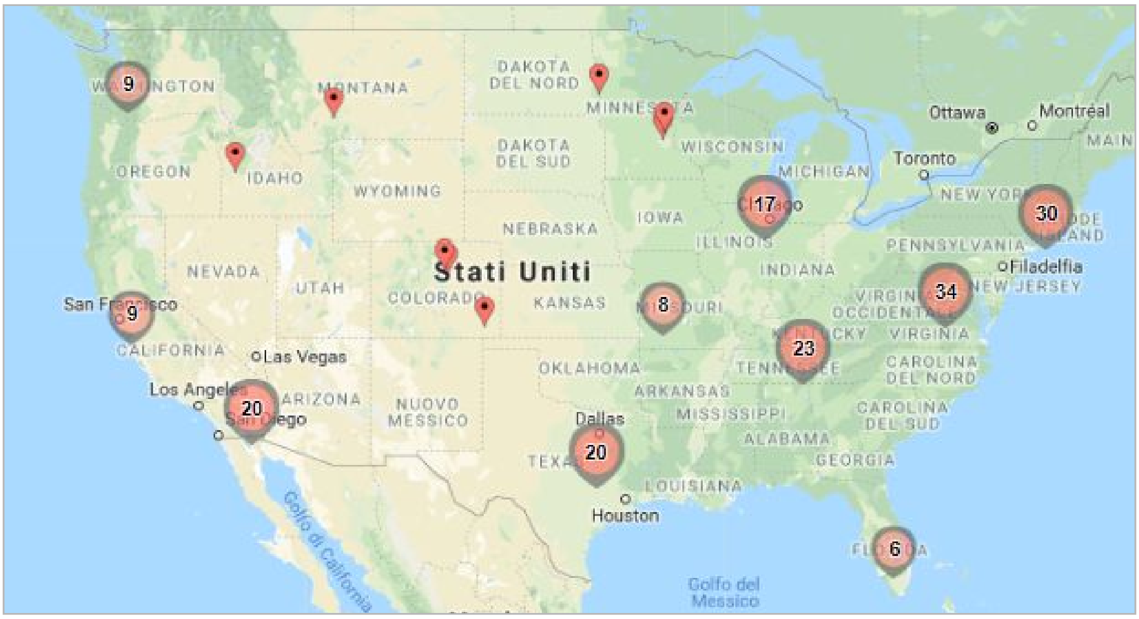

Image 1 - Bible matches in US map

It is possible to extract more detailed information about the users in order to qualcosa: for example some users provides personal description, date of birth, sex. This information alone seems to be useless but after and in addition of a behaviour study could reveal important indications. In order to manage all this information about as many users as possible it would be a good idea to extract only the useful information from the html page, so that the non-relevant parts are not stored uselessly.

Conclusion

Analysis of this kind can be done in different areas of interest, in fact many other sources of data are available, free or paid, and they can be explored to acquire information about many topics. For example a product company could analyze Amazon wishlists in order to collect information about what of their products (and of their competitor’s) are mostly requested by consumers.