



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Lecture with Computer Exercises:
Modelling and Simulating Social Systems with
MATLAB

Project Report

Facebook, for Fun and Profit

Simona CONSTANTINESCU & David TORTEL

Zurich
May 2011

Agreement for free-download

We hereby agree to make our source code for this project freely available for download from the web pages of the SOMS chair. Furthermore, we assure that all source code is written by ourselves and is not violating any copyright restrictions.

The program and source code used for data collection was reused and modified with the explicit consent of the original authors but may not be published due to copyright restrictions.

Simona Constantinescu

David Tortel

Acknowledgements

We would like to thank William Messenger for all his wisdom, guidance, availability, confidence and willingness to assist us and without which this paper would not have reached this quality. We are extremely grateful to him.

We would also like to thank Arthur Schmitt for his continuous advice, suggestions and inspiration on many different problems so critical to the success of the project.

Abstract

This paper is the final report for the course `Modeling and Simulating Social Systems with MATLAB`[26]. The course aimed to offer insight on how to use tools such as MATLAB in order to model and analyze social systems. In this paper, we try to develop a new way of approaching and analyzing social networks that is based on bilateral relationships; such networks are often represented as graphs showing the concatenation of ties between users, with ties standing for the relationship that exists between them. We will try to demonstrate why this view of the social network is somehow old fashioned, and then come up with a new view that we believe is more accurate. Indeed, we will focus on the social network as a meeting point for people sharing interests and ideologies, arguing that the social network is becoming a channel for mass communication. To do so, we will motivate our work, based on events that occurred recently in the world. Once motivated, and having formulated the questions we will ask about the existence of this new vision of social networks, we will implement a tool that enables us to gather pertinent data from a social network. The analysis of this data, based on mathematical algorithms, aims to show whether or not our vision was realistic. In the last part, we will explain some applications we see in such types of approaches. To our knowledge, the idea we are developing is new and has never been studied before.

Contents

1	Introduction	8
1.1	Internet evolution	8
1.1.1	Web	8
1.1.2	Web2.0	8
1.2	New mass communication channel	9
1.2.1	Social network evolution	9
1.2.2	Emergence of groups	9
1.2.3	A new framework for mass communication	9
1.3	New approach to the Social Network	10
1.4	Consequences of a new approach	10
1.4.1	Information diffusion	10
1.4.2	Asymmetry	11
1.5	Problematic	11
1.6	Reasoning	11
2	Description of the model and Implementation	12
2.1	Choosing a Social Network	12
2.1.1	Two-way relationship criteria	12
2.1.2	Legitimacy	12
2.1.3	Public API	14
2.1.4	Local application	14
2.2	Data to get	15
2.2.1	Facebook information about users	15
2.2.2	The user table	15
2.2.3	Groups	16
2.3	Implementing the Facebook app'	16
2.3.1	Registration process	16
2.3.2	Giving permissions	17
2.3.3	FQL requests	17
2.3.4	Translation	18
2.4	Getting the data	18
2.4.1	First data	18
2.4.2	Anonymity	19
2.4.3	Axe of study	20
2.4.4	Facebook group	20
2.4.5	Group similarity	20
2.4.6	Modeling the data	21
2.4.7	Creating the matrix	22

2.5	Reducing the data	22
2.5.1	The problem of huge datasets	22
2.5.2	Dividing the datasets into sub datasets	23
2.5.3	Sampling	23
3	Introduction and Research Question	24
3.1	Clustering	24
3.2	Similarity	25
3.3	Good clustering	25
4	Data statistics	25
4.1	The Data matrix	25
4.2	Data Analysis	26
4.2.1	Users Analysis	26
4.2.2	Groups Analysis	26
5	Types of Clustering	27
5.1	Hierarchical Clustering	27
5.2	Single Level Hard Clustering	28
5.3	Single Level Fuzzy Clustering	28
5.3.1	Fuzzy c-means Clustering	28
5.3.2	Expectation Maximization	29
5.3.3	Fuzzy clustering regarding to our data	29
6	k-means	29
6.1	Optimization problem	29
6.2	Solving the minimization problem	30
6.3	The algorithm	31
7	Distance metric	31
7.1	Squared Euclidean distance	31
7.2	City Block metric	32
7.3	Cosine distance	32
7.4	Correlation distance	32
7.5	Hamming distance	33
8	Objective function	33
8.1	Formalizing a good clustering	33
8.1.1	Defining the silhouette function	33
8.1.2	Meaning of the silhouette function	34

8.1.3	Applying the silhouette function	34
9	Visualization	35
10	Results	36
10.1	Procedure	36
10.2	Euclidean Distance	36
10.2.1	Meaning	37
10.2.2	K-means is a heuristic method	37
10.3	Correlation and Cosine Distance	38
10.3.1	The data was too unrelated	41
10.3.2	The names of the groups	41
10.3.3	k-means is a hard assignment method	43
11	Conclusions	43
11.1	Data	43
11.2	Research questions	43
11.3	Methods	44
11.4	Results	44
12	Future work	45
12.1	Unrestricted users	45
12.2	Fuzzy clustering	45
A	MATLAB CODE	46
A.1	Statistic indicators	46
A.2	Clustering	49

1 Introduction

1.1 Internet evolution

1.1.1 Web

Web is Dead, Long Live the Web... For more than 20 years, the web has been evolving, offering more and more features based on new technologies. From static pages to dynamically created content pages, the internet presents more and more possibilities to users, and is now the most used communication medium in the world, delivering information more quickly, in a more reliable way, and often for free. On March 21st 2010, the number of Internet users was 1,966,514,816[31] which represents a penetration rate of 28,7% of the global population. Since this number is still increasing, infrastructures continue to expand in order to provide new systems that rely on social behaviors. Such systems enable ubiquitous communication among peoples, platforms and applications in a global manner.

1.1.2 Web2.0

Due to this evolution, user behavior also has greatly changed; progressing first from readers to actors, they have now become developers of the web. This new behavior is often said to be characterized by the appellation Web2.0[27] The primary characteristics of this new kind of web are user-generated content, participation platform, data as driving force, and collective intelligence.

Such a trend has lead to the development of social networks where users can generate and share content with each other. The first goal of this platform was to provide the user a window through which one could glimpse life and easily communicate with another. This idea quickly lead to social *human* behaviors and the notion of ties between users appeared. Such links, for instance, are called friends in *Facebook*[13] or followers in *twitter*[33].

The social networks then became an aggregation of users, each user being tied somehow to other users, everybody sharing something with parts or all of her linked users. As the number of users increased, some interesting properties such as the six degrees of separation[29] quickly surfaced.

Such properties caused researchers to think more about the social network as a relationship-web among users [18, 35, 39]. However, a new

trend emerged little by little and is now one of the fundamental points of any social network.

1.2 New mass communication channel

1.2.1 Social network evolution

Social network user numbers have been increasing exponentially over the last few years[12, 30]; this progression has been accelerated by the development of smart phones that enable access to the social network, anytime, from anywhere. With such platforms, the user is able to create, collaborate, edit, categorize and exchange or promote any kind of information. Nowadays, companies such as Facebook possess a great deal of decision-making power and an interesting influence on the world wide web evolution. They even have a huge impact on social events. *Despite its giant population (nldr :500 million users[11]), Facebook is not quite a sovereign state—but it is beginning to look and act like one*[9].

1.2.2 Emergence of groups

The emergence of such large frameworks has clearly created new channels for mass communication and event coordination. Groups create and tend to attract like-minded communities of interest, in order to transmit more or less passionate ideas. These kinds of gatherings, after they traverse political or ideological matters, may then lead to various forms of “*hacktivism*“.

Focusing on this new role of social networks in the evolution of trends in the world, may be a clever idea for someone who wants to understand mass movements or idea propaganda. Indeed, it has been frequently suggested that the Middle East revolutions have been made possible by these social networks[?] that have become either vectors of information, or simply the core systems that enable people to gather around the same political ideas. From what we have seen in these revolutions, people gathered through social networks into differing groups that were clearly not in physical contact with each other, but nonetheless were sharing a dream or political ideology.

1.2.3 A new framework for mass communication

These social networks have since become information media as well[19]; it really demonstrates that people are ready to use a framework such

as a social network in order to gather behind common interests. Other examples can be found easily when looking at the *Wikileaks*[6] case in 2011. Indeed a group of people called *Anonymous*[4] decided to use the social network *Facebook* in order to share their opinions about Wikileaks and provide tools to users so that they could participate in a huge *Denial of Service* attack[5, 20]. The social network then became a way of promoting and conveying ideas, information and tools. The platform once again gathered people behind a common idea, a common goal, a common interest.

1.3 New approach to the Social Network

After analyzing the Middle East revolutions and Anonymous cases, we actually figured out that these are just the visible part of the iceberg; indeed, while crawling the social network, we found thousands of engaged groups, some more or less extreme, others more or less mainstream, but all gathering people behind a common cause. This gave us the idea of considering the social network no longer as a graph where edges between vertices represent relationships between users, but rather as a concatenation of clusters, each cluster gathering people with common interests.

We do not focus any longer on the social link that might exist between two random people, but see the social network as the aggregation of many people sharing the same interests. From this idea we want to prove that there exists a new model of social networks based on interest convergence.

1.4 Consequences of a new approach

1.4.1 Information diffusion

As soon as we consider a social network as the concatenation of groups, we begin to focus on the information diffusion in such groups [34]. Indeed, one of the main problems in the mass diffusion of information is that once the information is released, no one can predict the impact it will have on the masses; this impact will depend on the analysis that is done by different protagonists who speak about the information. When spreading information to the masses, we have no access as to who will act on the information first, and so we surrender our own bias as to what the information means. With this process, we are then able to recognize patterns and preferences in the clusters so that it becomes possible to analyze which group can spread the information in such a way that it

keeps the sense we want to impart. As a consequence, this new approach might be able to help event coordination and mass manipulation.

1.4.2 Asymmetry

Moreover, this super fast coordination of like minded people will establish an asymmetry in any conflicts, since it will unleash passions and will lead to mass movements that are difficult to stop. It has been proven that in many online social systems, social ties between users play an important role in dictating their behavior[3, 40] —for instance, through social influence a user can induce his/her friends to behave in a similar way. Once again, examples can be derived from the Middle East revolutions and from the Wikileaks case study.

As a consequence, showing that the social network can be divided into different clusters of users that share the same interest may lead to huge safety concerns, which make this study even more interesting and legitimate.

1.5 Problematic

In this paper, we will try to focus on the following questions:

- Does there exist an interesting clustering of people rooted in shared interests?
- Based on this clustering, can we find a sort of fingerprint of the cluster?
- Is it possible to predict user interest based on the knowledge of his cluster?
- When a new user appears, is it possible to assign him to one of the existing clusters?

1.6 Reasoning

In order to answer these questions, we first have to choose a social network and define how we can model user interest based on this social network. Since we want to prove that the bilateral relationship-based view of a social network is old fashioned, we will try to take a social network that

provides such types of links. For instance, Twitter only provides a one-way relationship so it is not interesting for our study.

Next we need to find a way of defining what data best represent our model of user interest and then somehow we have to be able to retrieve this data from a number of users in order to make our study legitimate.

Once the data have been collected, they will have to be extracted and put into useful mathematical objects.

Last but not least, we will need to analyze the data we receive. First of all, some basics statistics will give an initial overview of user behavior. Then we can go a bit further and apply some clustering algorithms in order to examine the validity of our model.

Finally, we will provide some further work ideas in order to get better results, or suggest some applications that can use our work.

2 Description of the model and Implementation

2.1 Choosing a Social Network

2.1.1 Two-way relationship criteria

The first objective in the model construction was to determine which social network to use in order for our study to be legitimate. Since we wanted to develop a new way of seeing social networks that is based on two-way relationships, we needed to focus on those types of networks. For instance, social networks such as *Twitter* are based on the notion of *follower*, which is a one-way relationship. Those types of social networks are not interesting for our study.

2.1.2 Legitimacy

Moreover, we had to find a social network that lent legitimacy in terms of usage. Facebook immediately came to mind. Indeed, Facebook has been said to have played a pivotal role in events such as the Middle East revolutions. Facebook has almost 500 million users and Facebook is based on the bond of friendship, which is a two-way relationship. Facebook was actually the first social network based on this symmetrical relation between users. This relation is commutative, which means


$$\forall(i, j) \in \mathbb{N} U_i R U_j \Leftrightarrow U_j R U_i \quad (1)$$


Besides, Facebook provided easy access to a huge amount of information that could then be modeled and studied according to mathematical algorithms. Indeed, every user has the potential to be part of some group by setting preferences, books, music, education background, etc.

David Tortel [Add as Friend](#)


🎓 Studying Information Security at [ETH Zurich](#) 🏠 Lives in [Austin, Texas](#) 🗣️ Knows English, Français, Español 📍 From [Zürich, Switzerland](#) 📅 Born on 05 September 1987


Education and work

University  **Telecom ParisTech**
School year 2011 · Information Security · Network

 **ETH Zurich**
School year 2012 · Information Security

Philosophy

Religious views  **Apostat Agnostique**

Political Views  **Full Disclosure**

Favourite quotations **Si vous ne pouvez être des saints de la connaissance, soyez en au moins des guerriers**


People who inspire David 

Figure 1: Available information on a public profile

Figures 1 and 2 show information that can be seen on the social network for the user David Tortel[10]

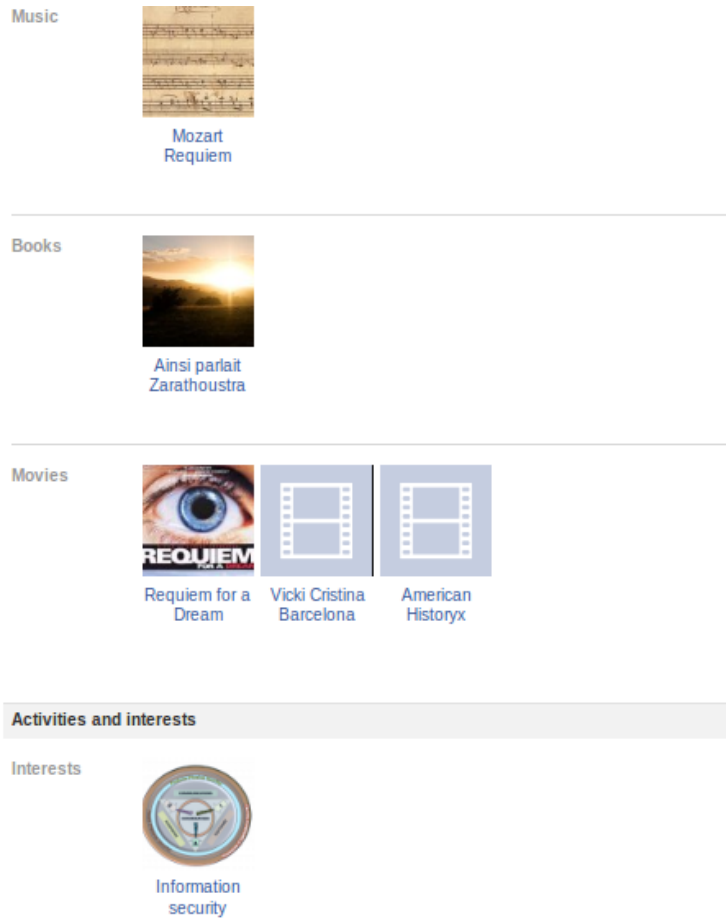


Figure 2: Available information on a public profile

2.1.3 Public API

Another important point when choosing the social network was the fact that it had to provide large documentation and an easy to use API that would enable us to get the data we were interested in. Recently, Facebook developed just such a public API for developers to create Facebook applications.

2.1.4 Local application

Last, but not least, the Facebook approach offered a real advantage. We were able to develop a local application that users could run in order to

provide us the information we needed to use. The Facebook application implementation will be developed further.

2.2 Data to get

2.2.1 Facebook information about users

Facebook information is stored in databases[14]. Some of these databases are accessible from the internet under certain conditions. Depending on the information it wants to access, the Facebook application has to query different tables, such as album application, checkin, comment, connection, cookies, developer, domain, event, family, friend, friend list, group...

Information relative to the user's interests can be found in the **user** table.

2.2.2 The user table

The first idea we had was to get all the data from the **user** table that would assist us in profiling him. This would enable us to gather as much information as possible about the user's interests. As a consequence, we examined the **user** table and then decided to take all data from the Facebook **user** table that are linked to the user. In the Facebook **user** table, we find the following information[15]:

user ID	first name
middle name	last name
full name	networks
time of update.	time zone
religion	birthday
gender	home town
genders the user wants to meet.	reasons to meet someone.
relationship for the user	user ID of the partner
political views	current location
activities	interests
favorite music	favorite television shows
favorite movies	favorite books
favorite quotes	information
high school	Post high school
work history	number of notes
number of Wall posts	current status
two-letter language code	URL to a user's profile.

2.2.3 Groups

Moreover, we decided to focus on the Facebook groups because, for us, they really demonstrated the users' interests. Therefore, we also culled references from all groups that are followed by the users.

2.3 Implementing the Facebook app'

In order to get all data from the users, we developed a Facebook application. This Facebook application is a Java servlet, that retrieves all interesting data from the user who is running the app as well as all his friends. This part aims to provide an overview of the development phase and to explain the different decisions we had to face during the implementation.

2.3.1 Registration process

Before allowing anyone to develop a Facebook application, Facebook requires a registration. The registration provides the developer with a key, a randomly generated token that is supposed to authenticate the application. Every time the application runs, this token is sent in all requests so that Facebook knows what application is trying to

access what data. The application was developed and run on a local server at the address `http://84.75.169.88:8080/Facebook/` Facebook gave us the identifier token 201038469931676. The application is called 851-0585-04L and is available at the address `http://www.facebook.com/apps/application.php?id=201038469931676`

2.3.2 Giving permissions

Whenever she connects to the application, the user is asked to login to Facebook with her username and password. After this is completed, the user is authenticated for Facebook. Then the application asks the user for permission to access information on her profile. This permission must be given by the user. When accepting, the user sends to Facebook her credentials as well as identifiers for the application that is allowed to access that data.

Figure 3 shows the permissions that are requested of the user to run the application. Most of them are related to data that are stored in the user.

Permissions	Dialogue impressions ▼	CTR	Dialogues accepted
ads_management, email, friends_about_me, friends_activities, friends_birthday, friends_checkins, friends_education_history, friends_events, friends_groups, friends_hometown, friends_interests, friends_likes, friends_location, friends_notes, friends_online_presence, friends_photo_video_tags, friends_photos, friends_relationship_details, friends_relationships, friends_religion_politics, friends_status, friends_videos, friends_website, friends_work_history, manage_friendlists, publish_stream, read_friendlists, read_insights, read_requests, read_stream, user_about_me, user_activities, user_birthday, user_checkins, user_education_history, user_events, user_groups, user_hometown, user_interests, user_likes, user_location, user_notes, user_online_presence, user_photo_video_tags, user_photos, user_relationship_details, user_relationships, user_religion_politics, user_status, user_videos, user_website, user_work_history, xmpp_login	50	86%	43

Figure 3: Needed permissions to run the application

2.3.3 FQL requests

When it has the correct permission, the application starts receiving data. To do so it sends some FQL –Facebook database language[14]– requests to the Facebook databases in order to access the data. The FQL request is the following:

```

SELECT uid,first_name,middle_name,last_name ,name,
affiliations,profile_update_time,timezone,religion,birthday,birthday_date,
sex,hometown_location,meeting_sex,meeting_for,relationship_status,
significant_other_id,political,current_location,activities,interests,
is_app_user,music,tv,movies,books,quotes,about_me,hs_info,education_history,
work_history,notes_count,wall_count,status,has_added_app,online_presence,
locale,proxied_email,profile_url,email_hashes,allowed_restrictions,
verified,profile_blurb,family,website ,is_blocked,contact_email,email
FROM user WHERE uid =me()

```

Then the same request is sent in order to retrieve the same data for friends of the user.

These two requests simply ask the `user` table to give back any information that can be linked with the user preferences and interests.

Then the two requests are run in order to retrieve all the groups that are followed by the user and by her friends.

```

SELECT gid,uid FROM group_member WHERE uid = me()

```

This request asks the system to go over the whole `group_member` table and look at the groups the user is part of. A list of group ID's and user ID's are returned by the application.

2.3.4 Translation

Facebook returns an XML document. When accessing this XML document, the application creates an XML tree and generates some data according to the XSLT sheet. This XSLT sheet enables the application to structure and exploit the XML document into a textfile document.

A new file is created for every user that runs the application.

2.4 Getting the data

2.4.1 First data

The application ran from May 6th to May 9th 2011. In this period, 39 users ran it from seven different countries. Figure 5 points out the repartition of users against their age and countries. When retrieving data from their friends, we recovered information on about 8000 users.

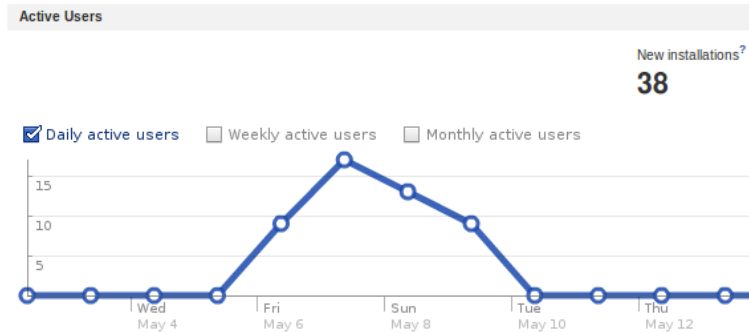


Figure 4: Application life

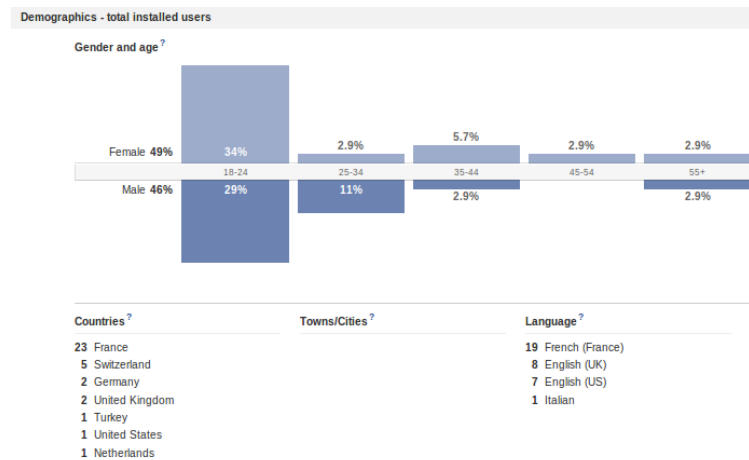


Figure 5: Repartition of users who ran the application

2.4.2 Anonymity

The first point when using this data was to insure anonymity. In fact, in our reasoning, one of the most important points was to insure that no person would be able to come back to the user through this same data. This privacy measure became one of our *sine qua non* conditions as we built our project. Therefore, we erased all data that was directly linkable to the user. As we still needed an identifier for every user, we decided to use the hash of the user ID and a secret value. We now had

$$user_{id} = HASH(uid + seed) \quad (2)$$

While the seed remained secret, this insured that nobody could deduce the uid and the user identity from the $user_{id}$ on the assumption that the

HASH algorithm was not broken.

2.4.3 Axe of study

Since the amount of data was really huge, and due to time constraints, we decided to focus on a particular axe. Our first idea was to focus on the particularities that are part of the `user` table, such as films, interest, books, network and so on. However, we realized that it would be really difficult and time consuming to gather all information about these interests and then attribute some of them to the appropriate users. Some issues of synonyms and misspellings also appeared, so we had to eliminate this train of thought.

2.4.4 Facebook group

The second idea we conceived was to focus on the group concept in Facebook. A Facebook group is an entity that gathers people around a common interest. That definition underscores the importance of such entities in our study, which aims to show that we can cluster people based on their interests. With such an entity, it is possible for the user to follow any news, since the group has a user interface with an official website, some pictures, and a wall that is available for comments and information.

Figure 6 shows the Facebook page of Anonymous' group. As Facebook developed more pleasing windows, a lot of information was disclosed on the group page so that people could join the page and be centered around a same spirit. As seen in this picture, and as we can see from the goal of a group, these entities enable people to gather and share ideas about common interests. As a consequence, this notion of group totally matches our perspective of clustering people according to their interests.

2.4.5 Group similarity

When deciding to focus on the groups, our first idea was to assemble similar ones together. However, it quickly turned out that this process had to be hand made; indeed, once again, misspellings or similar groups with different names made it impossible to automate the process. This is among the biggest concerns in today's internet, and we do not pretend to have a new idea for clustering the groups in a more efficient and

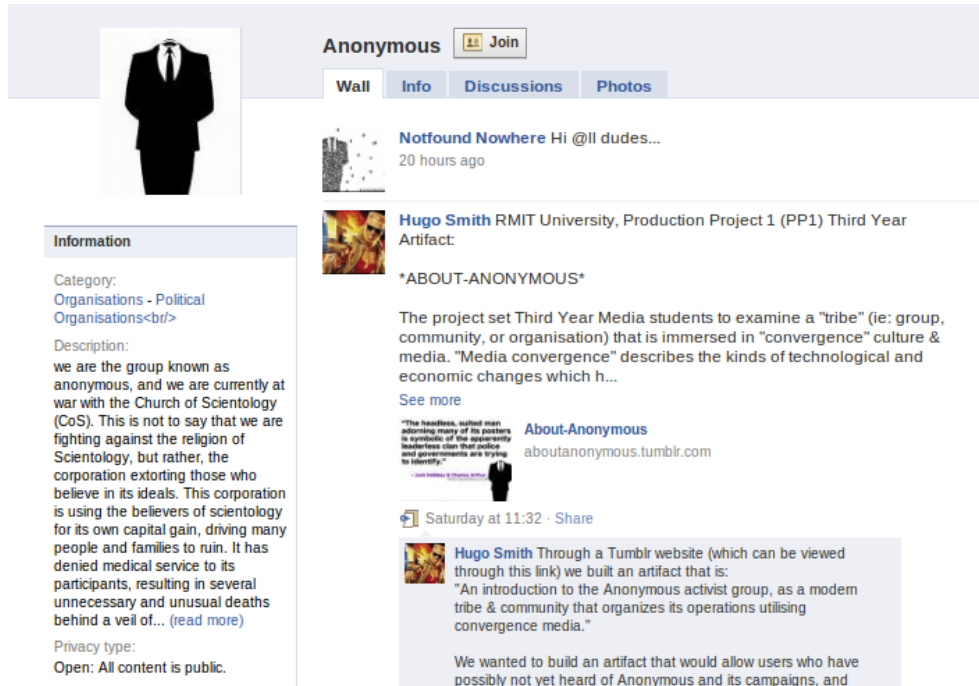


Figure 6: Anonymous group

clever way. Consequently, it means we have to group them in the *old fashioned* way. As we received about 344 000 different groups for our 8000 users, it turned out that this was far too much work and required far too much time to gather different groups in the same "group" appellation. Moreover, this approach would have penalized us for the last part of our task which is to suggest certain groups to the user. Therefore, we considered each group, based on its group id. As we have a bijection between a group and its group id, this approach totally made sense.

2.4.6 Modeling the data

Now that we knew the problematic we wanted to focus on, we wrote the Facebook application. When we received the information and decided which facts to use first, we had to figure out a way to model the data so that we could effectively use it. We needed to find a mathematical object that could carry the underlined structure in the notion of belonging to a group.

The natural approach that came to mind was to project the user

onto the space $vect < g_1, \dots, g_n >$ where g_1, \dots, g_n stands for the different groups. This space is the vectorial space that is generated by the vectors characterising the different groups. In this boolean projection, we have:

$$\forall(i, j) \in \mathbb{N}^2, \begin{cases} < User_i.Group_j > = 1 & \text{if } User_i \in Group_j \\ < User_i.Group_j > = 0 & \text{if } User_i \notin Group_j \end{cases} \quad (3)$$

Then we constructed the matrix of the projection of users onto space $vect < g_1, \dots, g_n >$ where g_1, \dots, g_n stands for the different groups. This projection gave us a boolean matrix that we could use from a mathematical point of view in a clustering perspective.

2.4.7 Creating the matrix

This boolean matrix is generated by a python script. This script first checks out the list of groups that are used by at least one of our users in order to generate the vectorial space. Then the script runs over all users and creates the matrix as follow:

$$\forall(i, j) \in \mathbb{N}^2, \begin{cases} Matrice[i][j] = 1 & \text{if } User_i \in Group_j \\ Matrice[i][j] = 0 & \text{if } User_i \notin Group_j \end{cases} \quad (4)$$

We have a $8000 * 344000$ matrix.

2.5 Reducing the data

2.5.1 The problem of huge datasets

Since we asked the matrix to work with the MATLAB tools, we focused on how to integrate it with MATLAB. Unfortunately, it turned out that such a matrix is far too big to be loaded in the MATLAB software[32].

Two solutions then occurred to us.

- Divide the matrix in several blocks, load the different blocks and work with them
- Sample the dataset to reduce the matrix

2.5.2 Dividing the datasets into sub datasets

If the first solution seemed to be more attractive at the beginning, it could not be implemented in our case since, for clustering, we needed to understand the behaviors of users one against another. For this reason, if we somehow split the dataset, we would then just be able to evaluate the user in her own dataset part, which would return incorrect results.

2.5.3 Sampling

Therefore, our application needs lead us to the second solution, reducing the dataset by sampling. For computational reasons, we were forced to choose a matrix with only 1700 users. In order to remain consistent in the whole work, we needed our sample to be representative of the entire Facebook sphere.

So we decided not to choose the users randomly, but, instead, took all users that were linked to a single user. Under these circumstances, we hoped to observe unique groups and preferences, because we could assume that people related to each other, would also tend to share certain interests. These people were quite representative of the Facebook sphere since we could find from public, political, religious, and private profiles, some people who are overactive on the social network, and others who are very quiet. Moreover, considering our sample, we knew that people are linked to each other in complex ways, which is, once again, a chief characteristic of the social network. Had we chosen people randomly, we would not have been able to prove that such social links still exist between people. We thus considered this sample to be indicative of the whole Facebook world. Some calculation would have demonstrated our assumption, but, once again for privacy reasons, we prevented ourselves from performing such analysis.

From this dataset of 1700 users and 65000 groups, we were able to start working on clustering algorithms and determine whether or not our assumptions were realistic.

3 Introduction and Research Question

3.1 Clustering

In algorithms terminology, partitioning a set of objects into groups such that the objects belonging to the same group are “similar” (*i.e.* share common features) is called *cluster analysis* or simply *clustering*. Figure 7 show such kind of clustering in 2 dimensions. The main goal of the cluster analysis is to detect underlying patterns existing in the data, which could not be determined by a superficial examination. As the labeling of the data is unknown and no prior identifiers are used throughout the algorithmic procedure, clustering is, from a machine learning point of view, an unsupervised learning method.

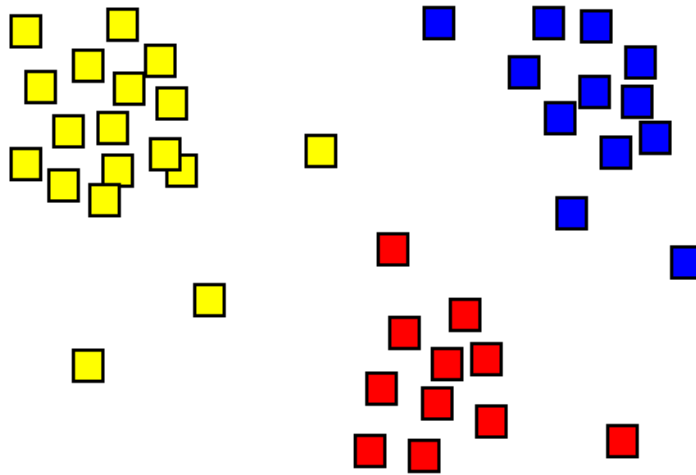


Figure 7: Example of clustering procedure in \mathbb{R}^2 : different colors represent different clusters[36]

3.2 Similarity

The method of defining the “similarity” measure and determining which objects are similar and which are not, depends heavily on the application of the clustering procedure. Similarly, there is no standard way to evaluate the performance of a given cluster configuration. Rather, a “good” clustering is determined by the context in which the analysis will be used. Also, even though in some cases the underlying idea might be the same, mathematically defining the “similarity” measure and the performance of a given cluster configuration is also dependent on the representation of the input data.

3.3 Good clustering

Therefore, our research question is whether there exists a “good”, “representative” clustering of the users of a social network (in this case, Facebook), based solely on their interest, which is quantified by belonging to groups. More specifically, a “good” clustering would show a significant separation between objects in the same cluster and objects in different clusters, such that, in general, the belonging of two users to the same cluster represents a tight relationship between them (tighter than comparing two users from different clusters). The mathematical formalism and quantification of “good” and “representative” clustering, as well as what it means for two users to be “similar”, will be given in the following sections. Our final goal is to state, from a mathematical perspective, that the relationship between the users and their common interests represent meaningful criteria for grouping the users in an accurate manner.

4 Data statistics

4.1 The Data matrix

The input data are represented by objects (in our case, the users), which possess features (Boolean group belongings). To each user, a feature vector is assigned, having as its length the total number of existing groups; a value of 1 exists for the positions representing groups to which the user belongs and a value of 0 exists for the positions representing groups to which the user does not belong. As stated above, out of the initial data set obtained after use of the Facebook application for data collection, only

users with more than 15 groups have been considered. Therefore, the input structure is a matrix, with $m = 1449$ rows (users) and $d = 64391$ columns (features).

$$M = (x_{ij}) = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1d-1} & x_{1d} \\ x_{21} & x_{22} & \dots & x_{2d-1} & x_{2d} \\ \dots & \dots & \dots & \dots & \dots \\ x_{m-11} & x_{m-12} & \dots & x_{m-1d-1} & x_{m-1d} \\ x_{m1} & x_{m2} & \dots & x_{md-1} & x_{md} \end{pmatrix} \quad (5)$$

4.2 Data Analysis

We performed a basic statistical analysis on the data in two directions: the groups and the users.

4.2.1 Users Analysis

We analyzed 1449 users, all having at least 15 groups. Figure 8 underlines the plot of group per user. The plot on the left shows all the users and the number of groups they belong to. The maximum number of groups that a user has is $maxGroups = 1389$ and the minimum number is $minGroups = 16$. The average user possesses $meanGroups = 126.2$ groups and the median number of groups, among users, is $medianGroups = 88$. As can be seen on the right side of Figure 8, there are few users having many groups. Therefore, we also computed the trimmed average, by removing the outlying 10% of users, but we did not obtain a much lower estimate: $meanGroupsTrimmed = 113.8$. However, the group's data are quite spread out, with a standard deviation of 123.4.

4.2.2 Groups Analysis

The maximum number of groups we worked with equals the number of columns of the input matrix: 64391. Figure 9 points out the statistics of users per group. The plot on the left shows the groups and the number of users that belong to each group. The largest group has $maxUsers = 512$ users. As can also be seen on the histogram on the right side for Figure 9, there are many groups with very few users, therefore the mean number of users per group is small: $meanUsers = 2.84$. The trimmed average (without 10% outliers) is even smaller: $meanUsersTrimmed = 1.5$ users per group. In this case as well, the standard deviation is very high, meaning the data are very spread: $sdUsers = 9.7$.

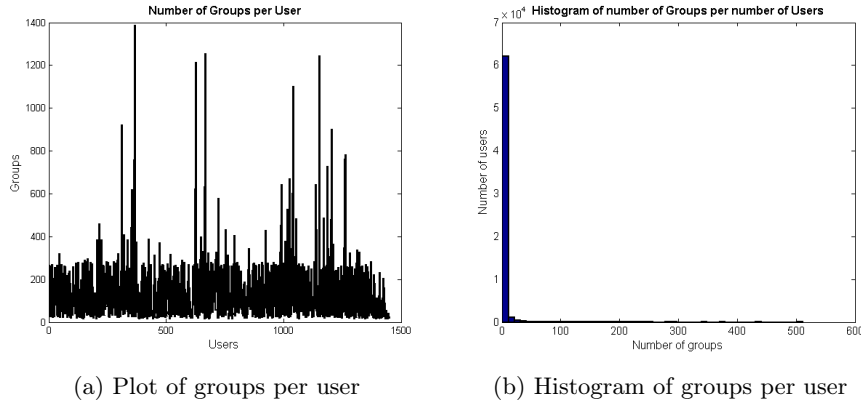


Figure 8: Statistics of groups per user

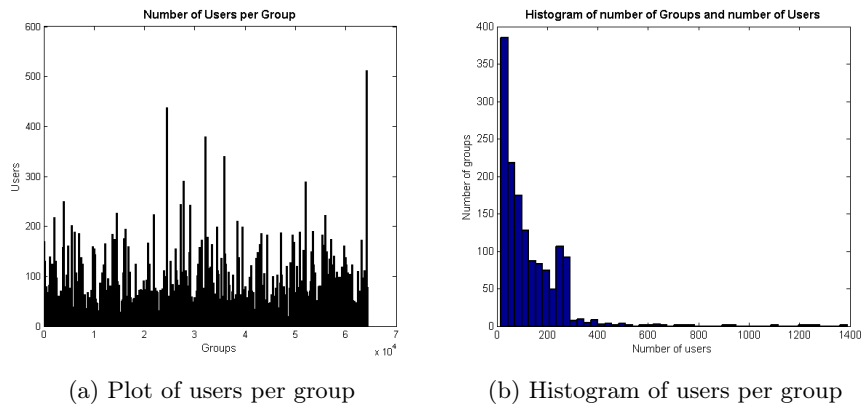


Figure 9: Statistics of users per group

5 Types of Clustering

In machine learning literature [7], there are three main types of clustering analysis.

5.1 Hierarchical Clustering

This method delivers a hierarchy of clusters, *i.e.* the new clusters are built on the basis of the existing ones. The starting condition is either the entire graph as a single cluster, which is progressively divided into multiple components until a certain stopping condition is reached (the

divisive or *top down* approach), or each point is assigned to separate clusters, that are progressively joined together, until a certain stopping condition is again reached (the *agglomerative* or *bottom up* approach). The results are visualized using a specific type of tree diagram called *dendogram*. In our case, as we aim to represent the relationship between the users on the same level rather than in a hierarchical manner, the hierarchical clustering technique is not suitable, since the resulting partitioning does not carry any interpretation in the social network context.

5.2 Single Level Hard Clustering

Unlike hierarchical clustering, in the single level clustering approach, the set of observations is partitioned into clusters, without building on existing structures. Therefore, there exists one single final partitioning of the objects. *Hard* means that the clusters are mutually exclusive (*i.e.* an object belongs to one and only one cluster). For interpretative reasons, the single level approach is more suitable than the hierarchical one for the analysis of real, large dimensional data. Since the general interpretation behind all the algorithms in this category is similar, we chose the most well studied single level hard clustering algorithm: *k-means*, to analyze our data.

5.3 Single Level Fuzzy Clustering

Unlike hard clustering, where an object belongs to only one group, in *fuzzy* (or *probabilistic*) clustering, an object has different degrees of belonging to groups (generally expressed as probabilities), which, if expressed as probabilities, sum up to 1. Two important approaches can be distinguished within the fuzzy clustering category [2]: Fuzzy c-Means Clustering and the Density Based Expectation Maximization Algorithm.

5.3.1 Fuzzy c-means Clustering

Fuzzy c-means Clustering is similar to k-means in the sense that it aims to partition the given set of objects into c clusters, while minimizing certain objective functions. As c-means is a fuzzy clustering technique, in contrast with k-means, it proposes a grouping in which any object can belong to more than one cluster. Therefore, the output of the algorithm is, for each user, a vector with length number of clusters, in which each element represents the degree of belonging to that respective cluster.

5.3.2 Expectation Maximization

In the case of the Expectation Maximization algorithm, the variables (the users) are represented as probability density functions, rather than as single points. One common case is working with *Gaussian Mixture Models*, in which the probability density functions are represented as a mixture of multivariate normal densities.

The Gaussian mixture distribution is a multivariate distribution that consists of a mixture of one or more multivariate Gaussian components. The Expectation Maximization Algorithm (EM) is used to fit the data, assigning posterior probabilities to each probability density with respect to each observation (group belonging). The posterior probability is maximized and the local optimum solution represents the cluster belongings (probability-wise).

5.3.3 Fuzzy clustering regarding to our data

Conceptually, fuzzy clustering assignment would make sense with respect to our settings, as it might provide us more insight into the grouping of the users than the hard assignment. However, the fuzzy techniques are much more computationally demanding than the hard ones and our input matrix has more observations per user (64391) than it does users (1449).

Therefore, it was impossible to run any of the fuzzy clustering techniques on our data set using any of the MATLAB implemented algorithms or even other versions of EM models, using other probability density function formulations. Moreover, none of the MATLAB implemented algorithms in the Fuzzy Clustering Toolbox [1] (*e.g. The Gustafson-Kessel algorithm, The Gath-Geva algorithm etc.*), with different input numbers of clusters, was able to complete the analysis of the data. For future purposes, this kind of technique could be employed if the number of considered groups is dramatically reduced.

6 k-means

6.1 Optimization problem

k-means clustering is a hard partitioning method which aims to locate each data point into one of the k clusters, while minimizing the within-cluster sum of squares (WCSS). The distance to be minimized (WCSS) is defined with respect to a fixed metric (*e.g. Euclidean, Correlation etc.*),

chosen prior to the optimization routine. Therefore, the goal of the minimization routine is the grouping of the objects, *i.e.* the clustering.

Given a set of n observations x_i , where each x_i is a user and is represented as a d -dimensional Boolean vector x_{ij} - the group belongings, with $j \in \{1, 2, \dots, d\}$, group these observations into k sets $\{S_1, S_2, \dots, S_k\}$ with $k \leq n$, such that each observation belongs to one and only one set and the within-cluster sum of squares is minimized:

$$\arg \min_S \sum_{i=1}^k \sum_{x_j \in S_i} \|x_j - \mu_i\|^2 \quad (6)$$

where μ_i is the mean distance (with respect to the defined metric) of all the points in S_i .

6.2 Solving the minimization problem

As can be noticed, the minimization problem is separated in two directions:

1. Finding k , the number of clusters;
2. Finding the partitioning of the objects, *i.e.* the clusters themselves.

The potential number of clusters can be assigned any value varying from 1 to the number of observations. Still, there is no analytical way of optimizing the number of clusters, while also optimizing their content. This is the reason why, in the k-means framework, the number of clusters, k , is given as an input and it is considered fixed. The problem then becomes finding the partitioning of the input set into a given number of clusters. Regarding computational complexity, the problem is NP-hard in the general Euclidean space d (number of observations) even for 2 clusters [2, 8] and NP-hard for a general number of clusters k even in the plane [22]. The algorithmic complexity, for a fixed number of clusters k , objects n and observations d is $O(n^{dk+1} \log n)$ [17]. Therefore, for clustering data sets in this manner, heuristic solutions are employed, and the guarantee of a global solution cannot be obtained.

6.3 The algorithm

The algorithm used is an iterative refinement technique, that minimizes the distances from each point to its cluster centroid over all clusters. The initial *centroids* (means) $\{m_1^{(1)}, m_2^{(1)}, \dots, m_k^{(1)}\}$ are chosen according to the Forgy method [16], as random sample points from the input data. The algorithm then proceeds by alternating between two distinct steps [37, 21]:

1. **Assignment step:** assign each observation to the cluster with the closest mean:

$$S_i^{(t)} = \{x_j : \|x_j - m_i^{(t)}\| \leq \|x_j - m_{i^*}^{(t)}\|, \text{ for all } i^* = 1, \dots, k\} \quad (7)$$

2. **Update step:** Calculate the new means to be the centroid of the observations in the cluster:

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j \quad (8)$$

The algorithm is deemed to have converged when the assignments no longer change.

7 Distance metric

The distance metric is an important factor in clustering, since the optimization routine is assigning all the users to the nearest cluster, while “nearest” is defined with respect to a metric. The most common metrics used in clustering are [23, 24]:

Given an m -by- d data matrix M , representing m d -dimensional vectors: x_1, x_2, \dots, x_m , the different distance metrics between any two elements x_s and x_t are defined as follows:

7.1 Squared Euclidean distance

$$d_{st}^2 = (x_s - x_t)(x_s - x_t)' \quad (9)$$

This is the most common metric encountered in the analysis of Euclidean space. In our case, as the vectors are represented solely through Boolean values, the Euclidean distance actually measures the (square root of the)

number of bits (groups) which differ (are not common) between the two users. Therefore, even if it makes sense to use this with our data, it is not very specific, since it does not take into account *which* are the groups that differ as well.

When this metric is employed for clustering, each centroid is the mean of the points in its cluster.

7.2 City Block metric

$$d_{st} = \sum_{j=1}^n |x_{sj} - x_{tj}| \quad (10)$$

This metric represents the sum of absolute differences between the two vectors, *i.e.* the *L1 distance*. Since we are working with Boolean data, the city block metric is nothing other than the square of the Euclidean distance. Therefore, since this metric does not provide any additional information, we discarded its use in favor of using the Euclidean distance.

7.3 Cosine distance

$$d_{st} = 1 - \frac{x_s x_t'}{\sqrt{(x_s x_s')(x_t x_t')}} \quad (11)$$

This metric is calculated as 1–the cosine of the angle between the two objects, treated as vectors. Since our data are high-dimensional, this distance measure is suitable for our analysis.

The centroid values are calculated as the mean of the points in the cluster, after the points are normalized to the Euclidean length unit.

7.4 Correlation distance

$$d_{st} = 1 - \frac{(x_s - \bar{x}_s)(x_t - \bar{x}_t)'}{\sqrt{(x_s - \bar{x}_s)(x_s - \bar{x}_s)'}\sqrt{(x_t - \bar{x}_t)(x_t - \bar{x}_t)'}} \quad (12)$$

where $\bar{x}_s = \frac{1}{n} \sum_j x_{sj}$ and $\bar{x}_t = \frac{1}{n} \sum_j x_{tj}$.

This metric measures the correlation between the two objects, treated as sequences of variables. As our final goal was to group users that are highly “similar” or *correlated* to each other, this metric is suitable for our analysis. We expect the clustering results obtained after using this function, to be similar to the ones obtained in the case of the Cosine distance, since both are very close as to method and as to interpretation.

In this case, each centroid is the mean of the points in the cluster (componentwise), after centering and normalizing those points to $mean = 0$ and $standard\ deviation = 1$.

7.5 Hamming distance

$$d_{st} = (\#(x_{sj} \neq x_{tj})/n) \quad (13)$$

The Hamming distance is generally suitable only for binary data. However, it only represents the percentage of bits that differ between the two vectors. As mentioned above, since the length of all our feature vectors is fixed, this metric returns the same information as the Euclidean distance and as the City Block distance. Since we are already using the Euclidean distance, employing this metric as well would not return any additional information.

8 Objective function

8.1 Formalizing a good clustering

8.1.1 Defining the silhouette function

Even after the above formalism, it is still unclear what constitutes a “good” clustering. In the beginning of the section, we stated that a “good” clustering would mean tight inter-cluster connections as compared to loose inter-cluster connections. We will now define this formally, using as a basis the *silhouette* function [25, 28, 38].

The *silhouette* function is defined on the set of users and has as codomain the closed real interval $[-1, 1]$. In each point, the value of the function is a measure of how similar the point is to points in its own cluster, compared to points in other clusters; more specifically, compared to the points in the cluster it is the closest to. Using the same matrix notation as before, this can be formulated as follows:

$$s : [1, n] \rightarrow [-1, 1]; \quad s(i) = \frac{\min b(i, :) - a(i)}{\max(a(i), \min b(i, :))} \quad (14)$$

where $a(i)$ is the average distance from the i^{th} point to the other points in its cluster and $b(i, k)$ is the average distance from the i^{th} point to points in another cluster k .

The numerator of the function represents the difference between the minimum distance of the user to any other cluster (by *minimum distance*, we mean the average distance between the user and all the users in the other cluster) and the average distance from the user to other users in its own cluster.

The denominator represents the maximum between these two terms: the minimum average inter-cluster distance and the average intra-cluster distance.

The sign of the silhouette value is determined by the numerator. If the average distance between the user and the other users in its cluster $a(i)$ is smaller than the minimum average distance between the user and users in other clusters $b(i, :)$, then the silhouette value will be positive. Otherwise, it will be negative.

Taking these two cases into account, here is how the function looks:

$$s(i) = \begin{cases} 1 - \frac{a(i)}{\min b(i, :)} & \text{if } a(i) < \min b(i, :) \\ 0 & \text{if } a(i) = \min b(i, :) \\ \frac{\min b(i, :)}{a(i)} - 1 & \text{if } a(i) > \min b(i, :) \end{cases} \quad (15)$$

8.1.2 Meaning of the silhouette function

For the silhouette function, a value of 1 means highest proximity to the points in the same cluster, while a value of -1 means highest proximity to the points in other clusters. A value close to 0 means that the point is located on the natural border between the two clusters. Of course, a value as high as possible (as close to 1 as possible) is desired.

8.1.3 Applying the silhouette function

Still, the silhouette function is defined on the objects to be clustered (the users). What we propose, as a measure of the quality of the entire clustering process, is to calculate the mean silhouette value for each cluster and, afterwards, to average these values for all the clusters. The final value will offer us a hint about how “good” the clustering is. Mathematically this means:

Having k clusters $\{S_1, S_2, \dots, S_k\}$, we define the function s' on the clusters' set, with values in the closed real interval $[-1, 1]$:

$$s' : \left\{ \begin{array}{l} [1, k] \longrightarrow [-1, 1] \\ i \longmapsto \text{mean}(s(k)) \forall k \in S_i \end{array} \right.$$

$$\text{score}(\text{clustering}, \text{metric}) = \text{mean}(s'(i)), \text{ for } i \in \{1, 2, \dots, k\} \quad (16)$$

Because there is no analytical way of determining the natural number of clusters in the data, our objective function (depending on the clustering and on the chosen metric) can help in this aspect. We expect to see variations in the score of the objective function, as introduced by variations in the number of clusters and in the metric we choose. Therefore the parameters for which we obtain a high score of the objective function will show us the hidden number of clusters in our data set, as well as the most natural way of calculating the distance between the users.

9 Visualization

Since our data are very high-dimensional (64391 dimensions/user), visualization is a problem. The usual clustering images, in which points belonging to different clusters are colored differently, are located either in the two or three-dimensional plane. One option, sometimes employed, is to reduce the dimensions of the data in a more meaningful direction (e.g. PCA - Principal Component Analysis), therefore mapping the 64391 dimensional space to a lower dimensional one. We did not employ this option for two reasons.

First, we always worked with the raw, unprocessed data, because we did not want to influence in any way the final results by eliminating any of the information we had. We also thought that reducing the data, even if only in the visualization procedure, might bias our interpretation of the results.

The second reason was that reducing the 64391 dimension to 2 or 3 would mean leaving the data without any physical interpretation, as too large a fraction of it would no longer exist. Given these two reasons, it was clear that reducing the data was not a sensible procedure.

The solution we chose was to use the silhouette plot[28], since it was also in very good accordance with our choice for objective function. The silhouette plot charts the silhouette values for each individual user and groups them by clusters. Therefore, in each cluster, every point is represented by its silhouette value and each cluster becomes a set of silhouette values for all the individual users it contains.

10 Results

10.1 Procedure

The statistical analysis on the users and on the groups, as well as the clustering procedure and the interpretation of the results, was conducted with MATLAB.

As previously noted, what we varied were the number of clusters, from 2 to 20, and the metric (Euclidean distance, Correlation distance, Cosine distance), both given as inputs to the clustering routine. For all the runs, we calculated the score of the objective function and, in this manner, assessed the quality of the clustering. Moreover, for each run, a silhouette plot was produced.

All the 60 plots can be found, as separate files, in the archive joining the report.

Overall, we did not obtain any significant correlation/“closeness” between objects in the same cluster, as compared to objects in different clusters, even with all the variations we introduced in the parameters. The scores we obtained, for the objective function, were slightly above 0 in almost all cases. Below, we look in closer detail at all the types of metrics we tested.

10.2 Euclidean Distance

We already stated, in the above sections, that the Euclidean distance will not be such a good measure for grouping the users. The reason behind this was that, in the case of Boolean data, when comparing two users, the distance represents nothing more than the number of groups in which they differ. Therefore, users will be grouped more according to the *number* of groups they belong to, than according to the *groups* themselves.

Nevertheless, we performed *k-means* with Euclidean Distance as input as well. In almost all the cases, what we see is that one of the clusters is very good (has a silhouette value of between 0.4 and 0.5), while the other clusters have negative silhouette values (meaning that they were “not well” assigned). This is due to two reasons:

10.2.1 Meaning

The users were clustered just according to the number of groups in which they differ.

As there are many users with not - so - many groups, it was expected that those users are clustered together. The users that are left have a higher variation in the number of groups they belong to (*e.g.*, the maximum number of groups a user has is 1389), and therefore cannot be properly clustered. One additional argument supporting this explanation is that, indeed, in the case of the Euclidean Distance, the best (and pretty good by itself) clustering is obtained when using as input only 2 clusters. The interpretation of that is *the users which have few groups (the big and tight cluster) versus the users which have not - so - few groups (the small cluster)*. As we increase the number of clusters, the *users which have few clusters* begin to be separated, therefore the result decreases in quality (objective function score).

Figure 10 shows the k-means results for the 2 clusters case. As can be seen, there is one large, good cluster and another small, not-so-good cluster

10.2.2 K-means is a heuristic method

The users are assigned, at every iteration, to the closest cluster, until a **local** optimum is reached. In other words, users which agree in the number of groups they differ are well clustered together, whereas the remaining ones, which were not *that* good, are left aside. The new clusters formed are obviously much worse, as the users do not agree among themselves and the variations they induce are very high.

Figure 11 shows the k-means results for the 15 clusters case. One can see how the “trend” is retained, some of the users are well clustered together, whereas the others are not. It is worth pointing out though that the number of users which are well clustered (the length of the big cluster) has shortened. This is due to the fact that the data has to be

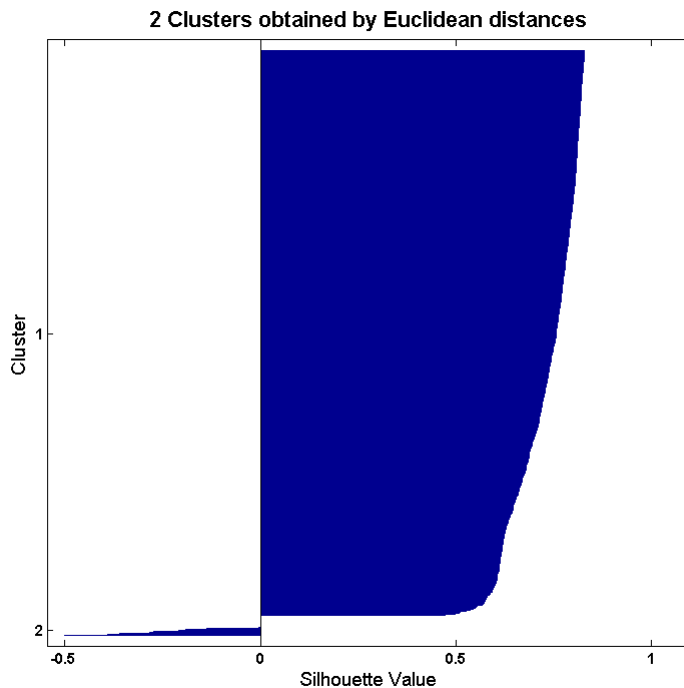


Figure 10: k-means results with 2 clusters and Euclidean distance

split in 15 groups rather than 2, therefore the users with few groups have to be further differentiated.

Figure 12 shows the objective function scores we obtained for all the number of input clusters, using Euclidean distances. Given the fact that, in most of the situation, the majority of the clusters had negative *silhouette* values, the objective function score is negative as well. One can also notice that there is no monotone tren (ascending/descending) among the values, as we increase the input number of clusters in which the data will be grouped.

10.3 Correlation and Cosine Distance

We treat these two types of distances together because, as we stated when describing the metrics, their interpretation is similar. As expected, we obtained similar results when using them.

The values of the *silhouette* function, for both metrics, are just

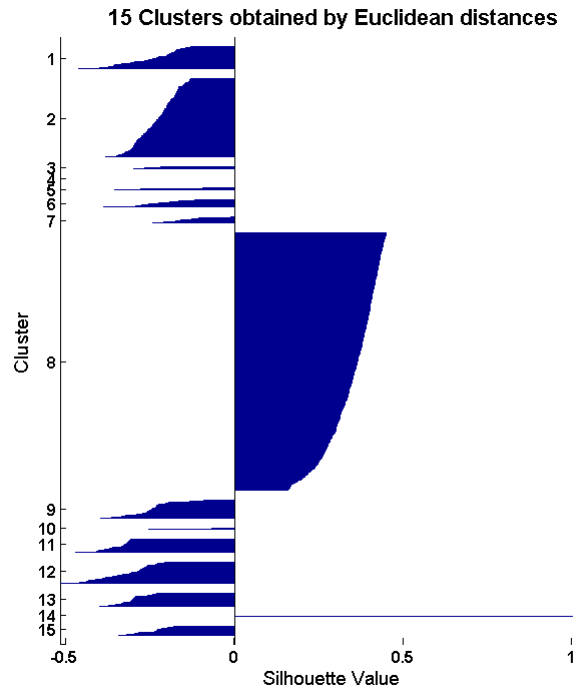


Figure 11: k-means results with 15 clusters and Euclidean distance

```
scoreE =
Columns 1 through 7
    0    0.2246   -0.0330   -0.0851   -0.1389   -0.1658   -0.1447
Columns 8 through 14
 -0.1961   -0.0592   -0.0691   -0.0877   -0.0989   -0.0153   -0.1154
Columns 15 through 20
 -0.0381    0.0342    0.0016    0.0269   -0.0021   -0.0424
```

Figure 12: The scores of the objective function for Euclidean Distance

slightly above 0, for almost all clusters (see Figures 13 and 14). This is unlike the Euclidean distance case, where one of the clusters had significant positive value and the others had negative values. Also, in these two cases, the values are always positive, which shows that, overall, the users clustered together are indeed more tightly connected than with users

```

scoreCR =

Columns 1 through 8
    0    0.0164    0.0131    0.0122    0.0115    0.0142    0.0150    0.0142

Columns 9 through 16
    0.0190    0.0182    0.0150    0.0162    0.0145    0.0161    0.0152    0.0135

Columns 17 through 20
    0.0153    0.0139    0.0152    0.0139

```

Figure 13: The scores of the objective function for Correlation Distance

```

scoreCS =

Columns 1 through 8
    0    0.0166    0.0131    0.0121    0.0115    0.0112    0.0208    0.0198

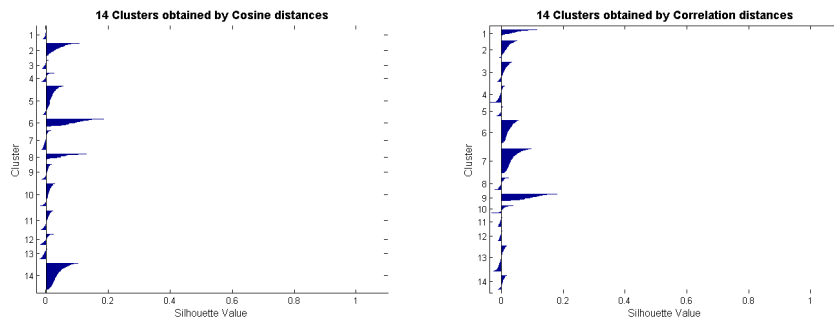
Columns 9 through 16
    0.0188    0.0180    0.0177    0.0161    0.0170    0.0165    0.0141    0.0144

Columns 17 through 20
    0.0147    0.0141    0.0132    0.0140

```

Figure 14: The scores of the objective function for Cosine Distance

from other clusters. However, the correlation is, in our opinion, too small. Moreover, one can notice, in these cases as well, that there is no monotone trend (ascending/descending) in the values of the objective function, as we increase the input number of clusters.



(a) 14 Clusters, Cosine distances

(b) 14 Clusters, Correlation distance

Figure 15: Sample run for 14 clusters

Figure 15 shows a sample run, with 14 clusters, for both these metrics. The clusters do not appear meaningful. For all cluster numbers, as can be seen by the objective function scores in Figures 13 and 14 as well, the “tightness” of the grouping was similar. However, all other images can be found as separate files joining the report.

In our opinion, both the Cosine and the Correlation metrics represent the “tightness” inside the clusters and should have grouped the users in a meaningful manner. The fact that we see no significant grouping can be due to the following reasons:

10.3.1 The data was too unrelated

The data set we are working with is not the entire original one, which we were unable to load into MATLAB because it was too large. We selected therefore all the users which possess a number greater than 15 groups. As the application obtained information from any person who ran it and all the friends of that person, the data in the original matrix was likely to have some hidden underlying patterns in it, as it was already slightly spatially “grouped”. By selecting, out of this data, only users which fulfill certain criteria, one of the possible effects was that the data will lose the hidden patterns it has, therefore the clusters would not be noticed anymore.

We took this possibility into account also when creating the working data, but, at that time, we supposed that the inner grouping of the data was strong enough, such that it can be seen as well only on the users which have more than 15 groups. Moreover, as MATLAB has memory restrictions when working with very large matrices, our other option would have been to only select a compact slice out of the initial matrix. This would have, indeed, guaranteed the spatial connectivity still. However, the new problem would have been that the data was not informative enough, as, in the initial data set, most of the users had only one group.

10.3.2 The names of the groups

The Facebook application we ran judged every group by its unique ID. It is widely known that, in Facebook, many groups have the same name,

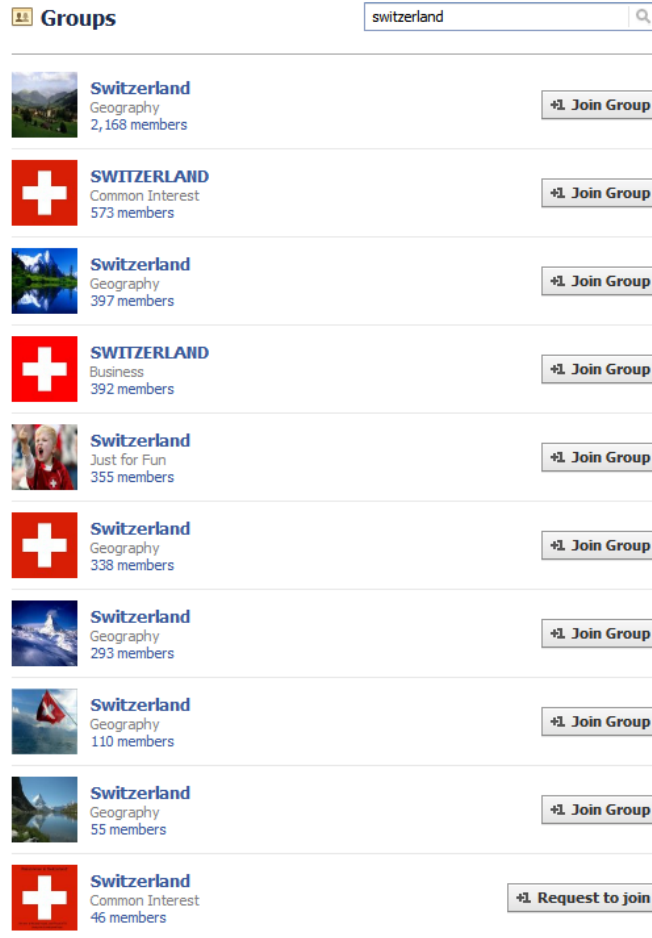


Figure 16: Groups search results for ‘Switzerland’

same meaning, same function, but different IDs. *e.g.*, if one likes Switzerland and wants to express this opinion through Facebook groups, there are at least 100 Facebook groups named exactly the same: “Switzerland”, some even using the exact same profile image: the Swiss flag. Figure 16 shows the first page of results obtained when searching for Facebook Groups about Switzerland.

Our Facebook application judges every group by its unique ID, therefore, assuming the example only described, all these groups would be

indexed differently. The application doesn't know if the groups have same meaning or are completely different as, in both cases, they would have different IDs.

To conclude, if two users actually like the same group and are conceptually connected through their preferences, this connection is likely not to be seen in the IDs of the Facebook groups, since there are many Facebook groups sharing the same name.

10.3.3 k-means is a hard assignment method

Even though k-means is widely used in working with real data, in some cases the assumption of any object belonging to one and only one cluster can be unrealistic. From this aspect, the fuzzy partitioning methods could render more meaningful results. Unfortunately, our dataset was too large (the number of features per user) to allow us using this type of methods.

11 Conclusions

11.1 Data

In the present work, we analyzed a very dense, popular and complex social network: Facebook. We obtained the data by running a Facebook application that we wrote ourselves, which gathered information about any user which clicked on the application and all her/his friends. Initially, the application obtained all the information the user had publicly available. However, for confidentiality reasons and also for better - defining our research questions, we focused only on the groups that every user likes and only indexed this information.

We initially obtained a lot of information, as many users ran the application. However, MATLAB was unable to cope with the large size of the input matrix, so we had to restrict it. In the end, we had as input data a Boolean matrix with 1449 users and 64391 groups. Each row represents a user and each column represents the user's binary group belongings.

11.2 Research questions

The research questions we aimed to answer were the following:

- Does there exist an interesting clustering of people rooted in shared interests?
- Based on this clustering, can we find a sort of fingerprint of the cluster?
- Is it possible to predict user interest based on the knowledge of his cluster?
- When a new user appears, is it possible to assign him to one of the existing clusters?

11.3 Methods

In order to answer our research questions, we analyzed the existing clustering methods in the literature [7] and it turned out that, in our case, their application was subject to the following two constraints:

1. Interpretability;
2. Computational demand.

Our possibilities were limited by these constraints and, in the end, we chose to perform the analysis with the hard partitioning method *k-means clustering* [7]. We varied the kind of metric used (Euclidean, Correlation and Cosine) and the number of clusters in which the data should be grouped (from 1 to 20).

11.4 Results

However, we did not find any significant clustering of the users in our input data, based on their interests represented as Facebook Groups. Therefore, unfortunately, 3 of our 4 research questions did not make sense anymore, in the light of our results.

The underlying reasons we propose as being the basis of our findings are:

- The data lost its underlying patterns when we restricted it.
- The groups are identified by the application solely on the basis of their ID, therefore multiple groups with the same name and meaning will be seen as completely different.

- *k-means* is a hard assignment method, which might be unrealistic. Better results could have been obtained when using fuzzy clustering techniques. Unfortunately, the large size of our data prevented us from doing so.

12 Future work

However, we still think that there exists a clustering of Facebook users solely based on their interests, expressed as belongings to Facebook groups. Some improvements to the present work can be imagined:

12.1 Unrestricted users

Working with blocks of unrestricted users, out of the initial dataset. However, when doing this, it should be taken into account that the data might lose its informative power, as there might be too many users which belong to only one group. Therefore, a suitable dataset should be selected, following these two constraints.

12.2 Fuzzy clustering

Developing a suitable and fast fuzzy - clustering algorithm, capable of coping with large sets of data. This idea could make the analysis more realistic and chances are that the resulting clusters are more significant, once any user is allowed to belong to more than one cluster.

A MATLAB CODE

A.1 Statistic indicators

```
% this function returns some basic statistics indicators of our data:
% users and groups
% written by:
% Simona Constantinescu, csimona@student.ethz.ch
% David Tortel, dtortel@student.ethz.ch
% ETH Zurich, May 2011

function [statsUsers statsGroups] = computeSimpleStatistics()
% load the boolean matrix of users and groups:
% rows: users;
% columns: groups;
% m(i,j) = 1 if user i belongs to group j and otherwise

load matrix.m;
% remove from the matrix the columns (groups) with no users
% this can happen because of the trimming, when obtaining the data
m = matrix(:,logical(any(matrix)));

% USERS ANALYSIS

% number of groups for each user
statsUsers.numGroups = sum(m,2);
% maximum number of groups per user
statsUsers.maxGroups = max(statsUsers.numGroups);
% minimum number of groups per user
statsUsers.minGroups = min(statsUsers.numGroups);
% mean number of groups per user
statsUsers.meanGroups = mean(statsUsers.numGroups);
% trimmed mean of groups per user (excluding 10% outliers)
statsUsers.meanGroupsTrimmed = trimmean(statsUsers.numGroups,10);
% median number of groups per user
statsUsers.medianGroups = median(statsUsers.numGroups);
% standard deviation for number of groups/user
statsUsers.sdGroups = std(statsUsers.numGroups);
```

```

%GROUPS ANALYSIS
% number of users in each group
statsGroups.numUsers = sum(m,1);
% maximum number of users per group
statsGroups.maxUsers = max(statsGroups.numUsers);
% minimum number of users per group
statsGroups.minUsers = min(statsGroups.numUsers);
% mean number of users per group
statsGroups.meanUsers = mean(statsGroups.numUsers);
% trimmed mean of users per group (excluding 10% outliers)
statsGroups.meanUsersTrimmed = trimmean(statsGroups.numUsers,10);
% median number of users per group
statsGroups.medianUsers = median(statsGroups.numUsers);
% standard deviation for number of groups/user
statsGroups.sdUsers = std(statsGroups.numUsers);

% the number of Users per Group as plot
figure();
plot(statsGroups.numUsers);
title('Number of Users per Group');
xlabel('Groups');
ylabel('Users');
figure();
hist(statsGroups.numUsers');

% the number of Users per Group as histogram
figure();
hist(statsGroups.numUsers,50);
title('Histogram of number of Users and number of Groups');
xlabel('Number of groups');
ylabel('Number of users');

% the number of Groups per Users as plot
figure();
plot(statsUsers.numGroups);
title('Number of Groups per User');
xlabel('Users');
ylabel('Groups');

```

```
% the number of Groups per User as histogram
figure();
hist(statsUsers.numGroups,50);
title('Histogram of number of Groups and number of Users');
xlabel('Number of users');
ylabel('Number of groups');
```


A.2 Clustering

```
% this function performs the clustering of the data, using kmeans
% various distance metrics (Euclidean, Correlation, Cosine), as well as
% various numbers of clusters (1-n) are used
% inputs:
%   m: the input data
%   n: the maximum number of clusters to be tried
% outputs:
% scoring structures for all the distance metrics (scoreE, scoreCR,
% scoreCS), with number of elements equals number of input clusterings to
% be tried. each element of each vector represents the score obtained under
% our formulation of the objective function (see report).

% authors:
%   Simona Constantinescu: csimona@student.ethz.ch
%   David Tortel: dtortel@student.ethz.ch
%
% ETH Zurich, May 2011

function [scoreE scoreCR scoreCS] = computeAndPlotClusters(m,n)

scoreE = zeros(1,n);
scoreCR = zeros(1,n);
scoreCS = zeros(1,n);

for i=2:20

    clear('-regexp','^I|^C|^s|^h|^D');
    index = num2str(i);

    % EUCLIDEAN DISTANCE

    % save the data structures separately, maybe they will be useful
    % afterwards
    IDXname = strcat('IDX', index,'E');
    Cname = strcat('C', index,'E');
    sumname = strcat('sumd',index,'E');
    Dname = strcat('D',index,'E');
```

```

% perform k - means
[IDX,C,sumd,D] = kmeans(m,i);
assignin('caller', IDXname, IDX);
assignin('caller', Cname, C);
assignin('caller', sumdname, sumd);
assignin('caller', Dname, D);
filename = strcat('/home/csimona/Desktop/Facebook/Data','kmeans',index,'E');
figure();
% compute and plot the silhouette value for each user
[s h] = silhouette(m,IDX);
tit = strcat(index,' Clusters obtained by Euclidean distances');
title(tit);
% compute and save the score of our objective function
for j=1:size(D,2)
    score(j) = mean(s(find(IDX==j)));
end
scoreE(i) = mean(score);
save(filename);

% CORRELATION DISTANCE

% save the data structures separately, maybe they will be useful
% afterwards
clear('-regexp','^I|^C|^s|^h|^D');
IDXname = strcat('IDX', index,'CR');
Cname = strcat('C', index,'CR');
sumdname = strcat('sumd',index,'CR');
Dname = strcat('D',index,'CR');
% perform k - means
[IDX,C,sumd,D] = kmeans(m,i,'distance','correlation');
assignin('caller', IDXname, IDX);
assignin('caller', Cname, C);
assignin('caller', sumdname, sumd);
assignin('caller', Dname, D);
filename = strcat('/home/csimona/Desktop/Facebook/Data','kmeans',index,'CR');
figure();
% compute and plot the silhouette value for each user
[s h] = silhouette(m,IDX,'correlation');

```

```

tit = strcat(index,' Clusters obtained by Correlation distances');
title(tit);
% compute and save the score of our objective function
for j=1:size(D,2)
    score(j) = mean(s(find(IDX==j)));
end
scoreCR(i) = mean(score);
save(filename);

% COSINE DISTANCE

% save the data structures separately, maybe they will be useful
% afterwards
clear('-regex','^I|^C|^s|^h|^D');
IDXname = strcat('IDX', index,'CS');
Cname = strcat('C', index,'CS');
sumdname = strcat('sumd',index,'CS');
Dname = strcat('D',index,'CS');
% perform k - means
[IDX,C,sumd,D] = kmeans(m,i,'distance','cosine');
assignin('caller', IDXname, IDX);
assignin('caller', Cname, C);
assignin('caller', sumdname, sumd);
assignin('caller', Dname, D);
filename = strcat('/home/csimona/Desktop/Facebook/Data','kmeans',index,'CS');
figure();
% compute and plot the silhouette value for each user
[s h] = silhouette(m,IDX,'cosine');
tit = strcat(index,' Clusters obtained by Cosine distances');
title(tit);
% compute and save the score of our objective function
for j=1:size(D,2)
    score(j) = mean(s(find(IDX==j)));
end
scoreCS(i) = mean(score);
save(filename);

end

```

References

- [1] Janos Abonyi. Clustering toolbox. Website, May 2011. <http://www.mathworks.com/matlabcentral/fileexchange/7486>.
- [2] Daniel Aloise, Amit Deshpande, Pierre Hansen, and Preyas Popat. Np-hardness of euclidean sum-of-squares clustering. *Mach. Learn.*, 75:245–248, May 2009.
- [3] Aris Anagnostopoulos, Ravi Kumar, and Mohammad Mahdian. Influence and correlation in social networks. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 7–15, New York, NY, USA, 2008. ACM.
- [4] Anonymous. Anonymous website. Website, May 2011. <http://www.whyweprotest.net/>.
- [5] Anonymous. Wikileaks. Website, May 2011. <http://www.whyweprotest.net/freedom-of-information/wikileaks/>.
- [6] Julian Assange. Wikileaks webpage. Website, May 2011. <http://www.wikileaks.ch/>.
- [7] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1st ed. 2006. corr. 2nd printing edition, October 2007.
- [8] S. Dasgupta and Y. Freund. Random projection trees for vector quantization. *Information Theory, IEEE Transactions on*, 55(7):3229–3242, july 2009.
- [9] Economist. Cyberwar. Website, Jul 1st 2010. <http://www.economist.com/node/16481504>.
- [10] Facebook. David tortel profile. Website, May 2011. <http://www.facebook.com/profile.php?id=100001029444752>.
- [11] Facebook. Facebook statistics. Website, May 2011. <http://www.facebook.com/press/info.php?statistics>.
- [12] Facebook. Facebook timeline. Website, May 2011. <http://www.facebook.com/press/info.php?timeline>.

- [13] Facebook. Facebook website. Website, May 2011. www.facebook.com.
- [14] Facebook. Fql documentation. Website, May 2011. <http://developers.facebook.com/docs/reference/fql/>.
- [15] Facebook. User table. Website, May 2011. <http://developers.facebook.com/docs/reference/fql/user/>.
- [16] Greg Hamerly and Charles Elkan. Alternatives to the k-means algorithm that find better clusterings. In *Proceedings of the eleventh international conference on Information and knowledge management, CIKM '02*, pages 600–607, New York, NY, USA, 2002. ACM.
- [17] Mary Inaba, Naoki Katoh, and Hiroshi Imai. Applications of weighted voronoi diagrams and randomization to variance-based k-clustering: (extended abstract). In *Proceedings of the tenth annual symposium on Computational geometry, SCG '94*, pages 332–339, New York, NY, USA, 1994. ACM.
- [18] Ravi Kumar, Jasmine Novak, and Andrew Tomkins. Structure and evolution of online social networks. In Philip S. S. Yu, Jiawei Han, and Christos Faloutsos, editors, *Link Mining: Models, Algorithms, and Applications*, pages 337–357. Springer New York, 2010. 10.1007/978-1-4419-6515-8_13.
- [19] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is Twitter, a social network or a news media? In *WWW '10: Proceedings of the 19th international conference on World wide web*, pages 591–600, New York, NY, USA, 2010. ACM.
- [20] John Leyden. Payback for torrent tracker attack. Website, May 2011. http://www.theregister.co.uk/2010/09/20/4chan_ddos_mpa_riaa/.
- [21] David MacKay. Chapter 20. an example inference task: Clustering. In *Information Theory, Inference and Learning Algorithms*, pages 284–292. Cambridge University Press, 2003.
- [22] Meena Mahajan, Prajakta Nimbhorkar, and Kasturi Varadarajan. The planar k-means problem is np-hard. In Sandip Das and Ryuhei Uehara, editors, *WALCOM: Algorithms and Computation*, volume

- 5431 of *Lecture Notes in Computer Science*, pages 274–285. Springer Berlin / Heidelberg, 2009. 10.1007/978-3-642-00202-1_24.
- [23] Mathworks. Distance measures. Website, May 2011. <http://www.mathworks.com/help/toolbox/stats/pdist.html>.
- [24] Mathworks. K means clustering. Website, May 2011. <http://www.mathworks.com/help/toolbox/stats/kmeans.html>.
- [25] Mathworks. Silhouette plot. Website, May 2011. <http://www.mathworks.com/help/toolbox/stats/silhouette.html>.
- [26] Chair of Sociology in particular of Modeling and Simulation. Lecture with computer exercises: Modeling and simulating social systems with matlab. Website, May 2011. <http://www.soms.ethz.ch/teaching/MatlabSpring11>.
- [27] Tim O'Reilly. What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software. *Communications & Strategies*, No. 1, p. 17, First Quarter 2007.
- [28] P. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20(1):53–65, November 1987.
- [29] MILGRAM S. The small world problem. *Psychology Today*, 1967.
- [30] Stan Schroeder. The web in numbers: The rise of social media. Website, 2009. <http://mashable.com/2009/04/17/web-in-numbers-social-media/>.
- [31] Internet World Stats. World internet users and population stats. Website, May 2011. <http://www.internetworldstats.com/stats.htm>.
- [32] MathWorks Product Support. Maximum matrix size by platform. Website, May 2011. <http://www.mathworks.com/support/tech-notes/1100/1110.html>.
- [33] Twitter. Twitter website. Website, May 2011. <http://twitter.com>.
- [34] Thomas W. Valente. Network models of the diffusion of innovations. *Computational & Mathematical Organization Theory*, 2:163–164, 1996. 10.1007/BF00240425.

- [35] Bimal Viswanath, Alan Mislove, Meeyoung Cha, and Krishna P. Gummadi. On the evolution of user interaction in facebook. In *Proceedings of the 2nd ACM workshop on Online social networks*, WOSN '09, pages 37–42, New York, NY, USA, 2009. ACM.
- [36] Wikipedia. Clustering analysis. Website, May 2011. http://en.wikipedia.org/wiki/Cluster_analysis/.
- [37] Wikipedia. K means clustering. Website, May 2011. http://en.wikipedia.org/wiki/K-means_clustering.
- [38] Wikipedia. Silhouette plot. Website, May 2011. <http://en.wikipedia.org/wiki/Silhouette>.
- [39] Rongjing Xiang, Jennifer Neville, and Monica Rogati. Modeling relationship strength in online social networks. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 981–990, New York, NY, USA, 2010. ACM.
- [40] Shuang-Hong Yang, Bo Long, Alex Smola, Narayanan Sadagopan, Zhaohui Zheng, and Hongyuan Zha. Like like alike: joint friendship and interest propagation in social networks. In *Proceedings of the 20th international conference on World wide web*, WWW '11, pages 537–546, New York, NY, USA, 2011. ACM.