Universitat Politècnica de València

Trabajo de Fin de Máster

Advances in Social Circles Detection

Autor:
Jesús Alberto Alonso Nanclares

Directores:
Dr. Roberto Paredes Palacios
Dr. Paolo Rosso

Máster en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital

Departmento de Sistemas Informáticos y Computación

Julio 2015

UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

“Lo que no es bueno para la colmena no puede ser bueno para las abejas.”

Marco Aurelio

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Departmento de Sistemas Informáticos y Computación

Máster en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital

Advances in Social Circles Detection

Abstract

Social circles arised out of a need to organize the contacts in personal networks, within the current social networking services. The automatic detection of these social circles still remains an understudied problem, and is currently attracting a growing interest in the research community. This task is related to the classical problem of community detection in networks, albeit it presents some peculiarities, like overlap and hierarchical inclusion of circles. The usual community detection techniques cease to be the most appropiate, due to these characteristics. Prediction is performed from two data sources: the network graph and node attributes corresponding to users' profile features. In this thesis, new approaches to this task are discussed and the results obtained from a thorough experimentation are presented. We provide a review of the state of the art in the fields of community detection in graphs, community detection in social networks and social circles detection. We describe the datasets employed in our experiments, both retrieved from Facebook, and we design a variety of feature representations, both from the structural network information and the users' profile information. We define and comment the prediction techniques in which our work is based: multi-assignment clustering, restricted Boltzmann machines and k-means. We describe some evaluation measures that have been proposed for social circles detection, and provide a critical commentary of some of them, as they present some flaws which lead to degenerate optimal performance. The core of this work is the presentation of the experiments that we have designed, along with the obtained results. There are two blocks of experiments, depending on the prediction technique employed: the first block considers multi-assignment clustering, a clustering method allowing for the inclusion of an element into several different clusters; whereas the second block considers a two-step method in which the data samples are mapped by restricted Boltzmann machines before feeding a k-means algorithm. We provide a discussion of the results, which have been satisfactory and have led to the publication of two articles, while a third one is awaiting revision. Our work opens the door to several lines of future work.

Acknowledgements

En primer lugar, agradecer a Roberto Paredes y Paolo Rosso la oportunidad de trabajar en el centro de investigación PRHLT, gracias a la cual se ha llevado a cabo la investigación que recoge este trabajo. Nuestra colaboración sigue y espero que sea duradera y fructífera.

Al resto de profesores del máster, por la formación recibida, y a los compañeros por hacer más llevadero el día a día. A mis compañeros de laboratorio, Emilio y Javi, por las conversaciones y por poder contar con ellos para pequeños y grandes favores.

A mis padres, Jesús y Lourdes, por su apoyo incondicional en todos los sentidos con todo lo que me proponga. Sin vosotros aquí no se habría podido escribir ni una letra y gran parte del mérito os corresponde. A mi hermana Lourdes y al resto de mi familia, con un recuerdo muy especial a mi abuela Pilar y a mi tía María Luisa, que estoy seguro estarán muy orgullosas de esto.

A Nieves por su cariño y apoyo constante día tras día, por interesarse siempre por cómo van las cosas, por felicitarme y animarme cuando corresponde. Por tu paciencia cuando tenemos que pasar tiempo sin vernos o sin hacer nada juntos. Estoy seguro de que las cosas irán mejorando y haciéndose más fáciles poco a poco.

A los amigos porque también se necesitan momentos en los que dejarlo todo atrás, divertirse y pasarlo bien, sin más. En especial a Ismael por demostrar que los buenos amigos no son a los que más se ve, sino para los que nada cambia tras un tiempo sin verse. Porque sé que estarás ahí si algún día todo lo demás se rompe en pedazos.

Finalmente quiero acordarme de los profesores de inglés que he tenido a lo largo de mi vida, ya que ellos también me han ayudado a escribir esta tesis.

This master's thesis was developed in the framework of the W911NF-14-1-0254 research project, funded by the US Army Research Office (ARO).

Contents

List of Figures

List of Tables

Abbreviations

MAC	Multi-Assignment Clustering
1	Structural network representation of *friendship rank* 1
2	Structural network representation of *friendship ranks* 1 and 2
3	Structural network representation of *friendship ranks* 1, 2 and 3
2w	Structural network representation of *weighted friendship ranks* 1 and 2
3w	Structural network representation of *weighted friendship ranks* 1, 2 and 3
2a	Structural network representation of *aggregated friendship ranks* 1 and 2
3a	Structural network representation of *aggregated friendship ranks* 1, 2 and 3
e	*Explicit* users' profile representation
i	*Intersection* users' profile representation
w	*Weighted* users' profile representation
LCP	Lossy Compression Problem
DA	Deterministic Annealing
RBM	Restricted Boltzmann Machine
pdf	Probability Density Function
CD	Contrastive Divergence
ReLU	Rectified Linear Units
SSE	Sum of Squared Errors
E_b	Best Matching evaluation measure
E_h	Hungarian Matching evaluation measure
E_d	Edit Distance evaluation measure

Chapter 1

Introduction

The study of social networks, understood as networks depicting social interactions among people, is a topic spanning decades of research history [1]. Recently, new information and communication technologies have opened novel ways of interacting. The most significant change of paradigm with regard to human relations has been the appearance of social networking services like Facebook, Google+ or Twitter. This has provoked a revitalization of social network analysis, with lots of new problems and challenges emerging.

Social circles detection is one of the tasks born out of the appearance of these technologies. Social circles are a tool already implemented in the major social networking services, whether called by circles, lists or other names. It is an aid to organise the contacts in personal networks that has proved to be fairly useful. However, circle labelling is still mostly done manually, which can become a tedious task. This results in an incomplete labelling of many personal networks. In this context, the problem of the automatic detection of social circles arises and attracts a growing interest. In addition, applications within sociology or other disciplines could be developed, as well.

A social network can be assimilated to a graph and, therefore, the problem of social circles detection constitutes a particular case of community detection in graphs, for which a wide variety of methods have been developed [2, 3]. In the following sections, I am going to review the state of the art in the field of community detection, and the particular case of community detection in social networks. After that, I am going to describe the general framework of the social circles detection task.

1.1 Community Detection in Graphs

From an abstract point of view, a network is equivalent to a graph, defined by a set of nodes connected by edges. Nevertheless, the concept of network has additional connotations. Networks can represent real structures such as social networks, biological networks (neural synaptic networks, metabolical networks), technological networks (the Internet, the World Wide Web), logistic networks (distribution networks), etc. There is no well-accepted formal definition of community in a general network. However, there is a consensus on the fact that it consists of a group of nodes that are more densely connected to each other than to the nodes outside. A more concrete definition usually depends on the specific study or system, and sometimes even an algorithmic definition is assumed, in other words, a community is defined as the output of a particular algorithm. The relation of membership in a community usually has an extra meaning, and the vertices pertaining to a community will probably share common properties or play similar roles within the graph.

Community detection is the task of automated identification of the communities of a given network. A considerable number of methods have been developed to solve this problem [2, 3], normally based on graph clustering algorithms. The classical graph clustering methods are classified in four families:

- *Graph partitioning:* These algorithms divide the vertices of the graph into a fixed number of groups of a predetermined size, minimizing the number of edges lying between different groups. Some examples of graph partitioning are the Kerninghan-Lin algorithm [4] and the method defined in [5].
- *Hierarchical clustering* [6]*:* These techniques are adequate when a hierarchical structure underlies the graph, with small clusters falling recursively within larger clusters.
- *Partitional clustering:* These methods provide a separation of the graph nodes into a fixed number of clusters, by optimizing a cost function based on a predefined distance between points. The most popular partitional technique is the well-known k-means algorithm [7, 8].

- *Spectral clustering* [9]*:* These algorithms are based on the spectrum of matrices. First, a matrix of similarity between nodes is calculated. Then, the eigenvectors of that matrix are employed to calculate the clusters.

Divisive algorithms constitute another important family of community detection algorithms. Their methodology consists in the removal of the edges that connect nodes of different communities, also called local bridges. The most famous divisive algorithm is the Girvan and Newman algorithm [10, 11]. It is based on a modularity measure, introduced as a stopping criterion. Having a particular division of a network into k communities, let $\mathbf{E}$ be a $k \times k$ symmetric matrix whose element E_{ij} is the fraction of all edges in the network that link vertices in community i to vertices in community j. Then, the modularity measure Q is defined by:

$$Q = \sum_i \left(E_{ii} - \left(\sum_j E_{ij} \right)^2 \right) \tag{1.1}$$

Higher values of the modularity measure indicate good partitions, and therefore modularity has become the most used and best known quality function. As a consequence, modularity optimization would produce a satisfactory solution to the problem. Unfortunately, modularity optimization is an NP-complete problem [12], although there are several algorithms able to find fairly good approximations of the modularity maximum in a reasonable time [13, 14].

For the moment, no mentioned prediction technique is able to detect overlapping communities. However, in real networks, this is a common phenomenon, nodes often belong to several different communities. The most popular technique to detect overlapping communities is the clique percolation method [15]. Given a graph, a k-clique is defined as a complete subgraph of size k. Clique percolation consists in the identification of k-clique communities, defined as the union of all k-cliques that can be reached from each other through a series of adjacent k-cliques.

The clique percolation algorithm is based on first locating all cliques of the network, in a decreasing order of their size, and then identifying the communities by carrying out a standard component analysis of the clique-clique overlap matrix [16]. Despite its good performance, this technique remains a hard computational problem, as new and improved implementations still scale worse than some other overlapping community finding algorithms. There are several recent alternatives that detect overlapping clusters. These methods include Multi-Assignment Clustering [17, 18].

1.2 Community Detection in Social Networks

Social networks provide extra sources of information that may be used for clustering, apart from just the network graph structure. Some of this additional information can be modelled as node attributes, for example the data present in the users' profiles found in any social networking service. The content of publications or interactions among users can feed the clustering algorithms, as well.

Several clustering methods have been developed specifically for social networks, based on different sources of information. Some of them consider only the network structure [3]. Others are just based on the semantic content of social interactions [19]. A third group of methods combines the structure with node attributes [20, 21]. Finally, some algorithms are based on the combination of the network structure and the content of social interactions [22–24].

1.3 Social Circles Detection

Social circles detection is a particular case of community detection in social networks. Normally, the network structure and node attributes, particularly users' profile features, are used as data sources for the clustering.

Within a social network, an *ego network* or *egonet* is defined as the subgraph of the contacts of a particular user (called the *ego*). Thus, it includes all the contacts of the *ego* (named the *alters*) and the contact relationship between every pair of them. Then, the social circles of an *ego* can be considered as clusters of the *egonet*. Social circles may overlap (share nodes), for example university friends who were high school friends as well; and they may also present hierarchical inclusion (the nodes of a circle totally included into another), for example university friends into a generic friends category. A graphical depiction of a simple *egonet*, presenting these phenomena, is shown in Figure 1.1.

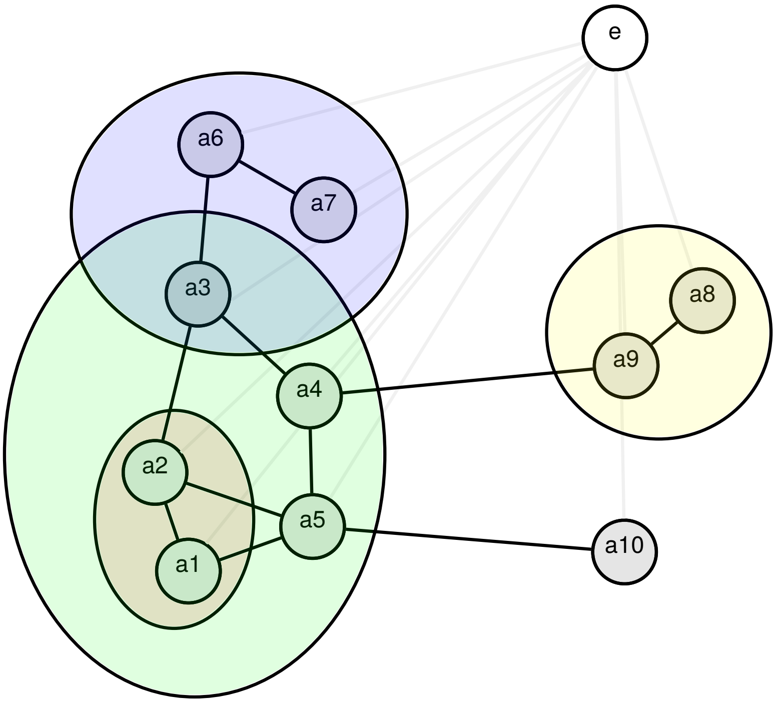


FIGURE 1.1: Graphical depiction of a simple *egonet* and its social circles. Note that it presents both overlap and hierarchical inclusion of circles.

Some current, successful work in social circles detection involves a generative model that considers circle memberships and a circle-specific profile similarity metric [20, 21], which are modelled as parameters to be learnt. This metric encodes what dimensions of profile similarity cause the circles to form. Other approaches are also being considered, such as the use of Multi-Assignment Clustering (MAC) [17, 18] for circle prediction. In [20, 21], some tests are performed with this idea, but using only the node attributes, discarding thus the information from the graph structure.

Chapter 2

Feature Construction

2.1 Datasets

We have used two datasets for the experiments presented in this thesis. They are different but similar, as they contain the same kind of information. Both sets are composed of data retrieved from Facebook. The contact (friendship) relationship within this particular social networking service is symmetric, which makes the underlying network graph undirected. The datasets contain information of several *egonets*, both from the network structure and the users' profiles. The structural network information consists of the adjacency within the *egonets* graphs. It is complete, meaning that every connection between two *alters* is specified. The users' profile information, within every *egonet*, is composed of up to 57 profile features for every *alter*, which can take discrete values from a finite set. In contrast to the structural network information, the users' profile information is scarce, as some features present a non-empty value only for a minority of the *alters*; redundant, as there are groups of features providing similar information (hometown ids and hometown names, for instance); and, sometimes, irrelevant for prediction, like first names. Within both datasets, the ground-truth circles were hand-labelled by the *egos* themselves, by means of a Facebook application.

The first dataset, named *Kaggle*, is the one published for the Kaggle competition on learning social circles in networks [25]. The whole published dataset is composed of 110 *egonets*. However, the ground-truth is available just for the so-called training subset, composed of 60 *egonets*, which is the part that we use. The second dataset forms part of the Stanford Large Network Dataset Collection [26], a collection of numerous and diverse datasets related to networks. Its name is *ego-Facebook*. Some basic statistics of both datasets are shown in Table 2.1.

	Kaggle	*ego-Facebook*
Egonets	60	10
Users in smallest *egonet*	45	59
Users in largest *egonet*	670	1045
Total users	14519	4167
Connections	348131	88234
Circles	592	193

Table 2.1: Basic statistics of the datasets.

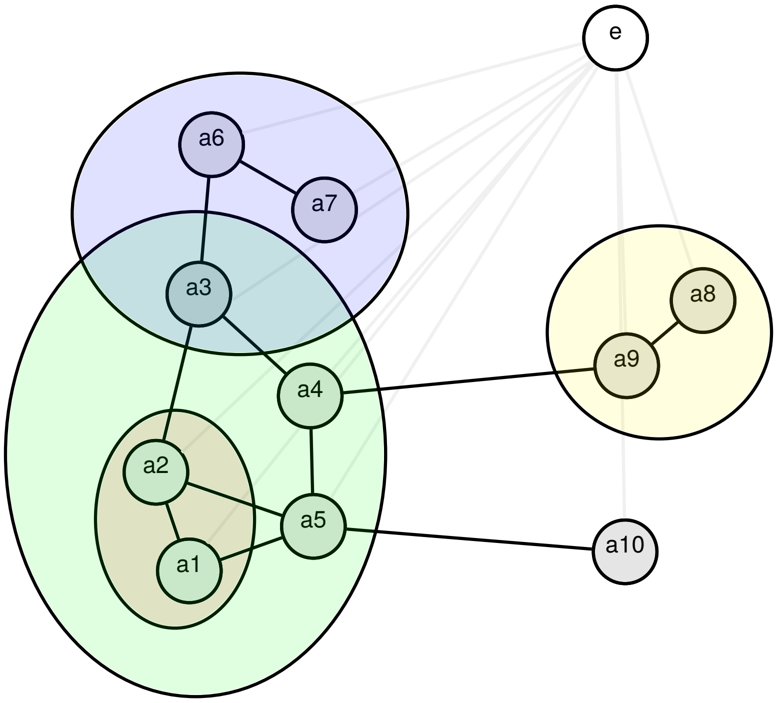
The data contained in the datasets needs some manipulation, in order to feed the prediction algorithms. All the algorithms employed in our studies receive a matrix $\mathbf{X}$ as input. This matrix is itself a horizontal concatenation of a matrix $\mathbf{S}$, containing structural network information; and a matrix $\mathbf{P}$, containing users' profile information: $\mathbf{X} = [\mathbf{S}|\mathbf{P}]$.

The rows of $\mathbf{X}$ represent the *alters* in the *egonet* and, therefore, for every *alter* a there is a row vector of structural network information, $\mathbf{S}_a$, and a row vector of profile features information, $\mathbf{P}_a$. Therefore, the number of rows of the matrix $\mathbf{X}$ is the number of *alters* $\mid a \mid$, and the number of columns of the matrix $\mathbf{X}$ is the total number of features used to represent structural and profile information of each *alter*. We have constructed several different representations of both kinds of information, which are going to be presented in the following sections.

2.2 Structural Network Information

The aim of every representation of the structural network information is to transform graph links into the matrices $\mathbf{S}$. Being $\mid a \mid$ the number of *alters* in the *egonet*, we define some concepts below. A graphical example of them is shown in Figure 2.1, as well.

- *Friendship ranks:* when there is a link between two *alters*, we say they are direct friends or *rank* 1 *friends*. When two *alters* are not direct friends but have a



	Rank 1					Rank 2				
	a1	a2	a3	a4		a1	a2	a3	a4	
a1	1	1	0	0	...	1	1	1	1	...
a2	1	1	1	0	...	1	1	0	1	...
a3	0	1	1	1	...	1	0	1	0	...
a4	0	0	1	1	...	1	1	0	1	...
	...	...	...	...	...	...	...	...	...	...

(a) *Non-weighted* structural network representation.

	Rank 1					Rank 2				
	a1	a2	a3	a4		a1	a2	a3	a4	
a1	1	1	0	0	...	0.5	0.5	0.5	0.5	...
a2	1	1	1	0	...	0.5	0.5	0	0.5	...
a3	0	1	1	1	...	0.5	0	0.5	0	...
a4	0	0	1	1	...	0.5	0.5	0	0.5	...
	...	...	...	...	...	...	...	...	...	...

(b) *Weighted* structural network representation.

	a1	a2	a3	a4	
a1	1	1	0.5	0.5	...
a2	1	1	1	0.5	...
a3	0.5	1	1	1	...
a4	0.5	0.5	1	1	...
	...	...	...	...	...

(c) *Aggregated* structural network representation.

Figure 2.1: Feature construction from the structural network information. *Friendship ranks* 1 and 2 are considered.

common direct friend, we say they are *rank* 2 *friends*. *Friendship ranks* of greater levels can be further defined. In this study we consider up to *rank* 3 *friends*. There is a column in **S** for every *friendship rank* and *alter* in the *egonet*. An element of **S** is 1 if the row *alter* and the column *alter* are *friends* of such *rank*, and 0 otherwise. We obtain in total up to $3\times \mid a \mid$ structural features for each *alter*. *Friendship ranks* can be seen in Figure 2.1.

- *Weighting:* the data is *weighted* depending on the *friendship rank* it represents. *Rank* 1 *friendship* is left with 1, whereas *rank* 2 *friendship* is *weighted* to 0.5 and *rank* 3 *friendship* is *weighted* to 0.25. Like in the previous case, we obtain in total up to $3\times \mid a \mid$ structural features for each user. An example of *weighting* is shown in Figure 2.1b.

- *Aggregation:* for every user, the different *friendship ranks* are *aggregated* into just one value. This is obtained by calculating the maximum *weighted friendship rank*. In this case, we reduce the number of structural features to $\mid a \mid$. The information presented in Figure 2.1c is *aggregated*.

From the concepts defined above, we consider the structural network representations shown in Table 2.2.

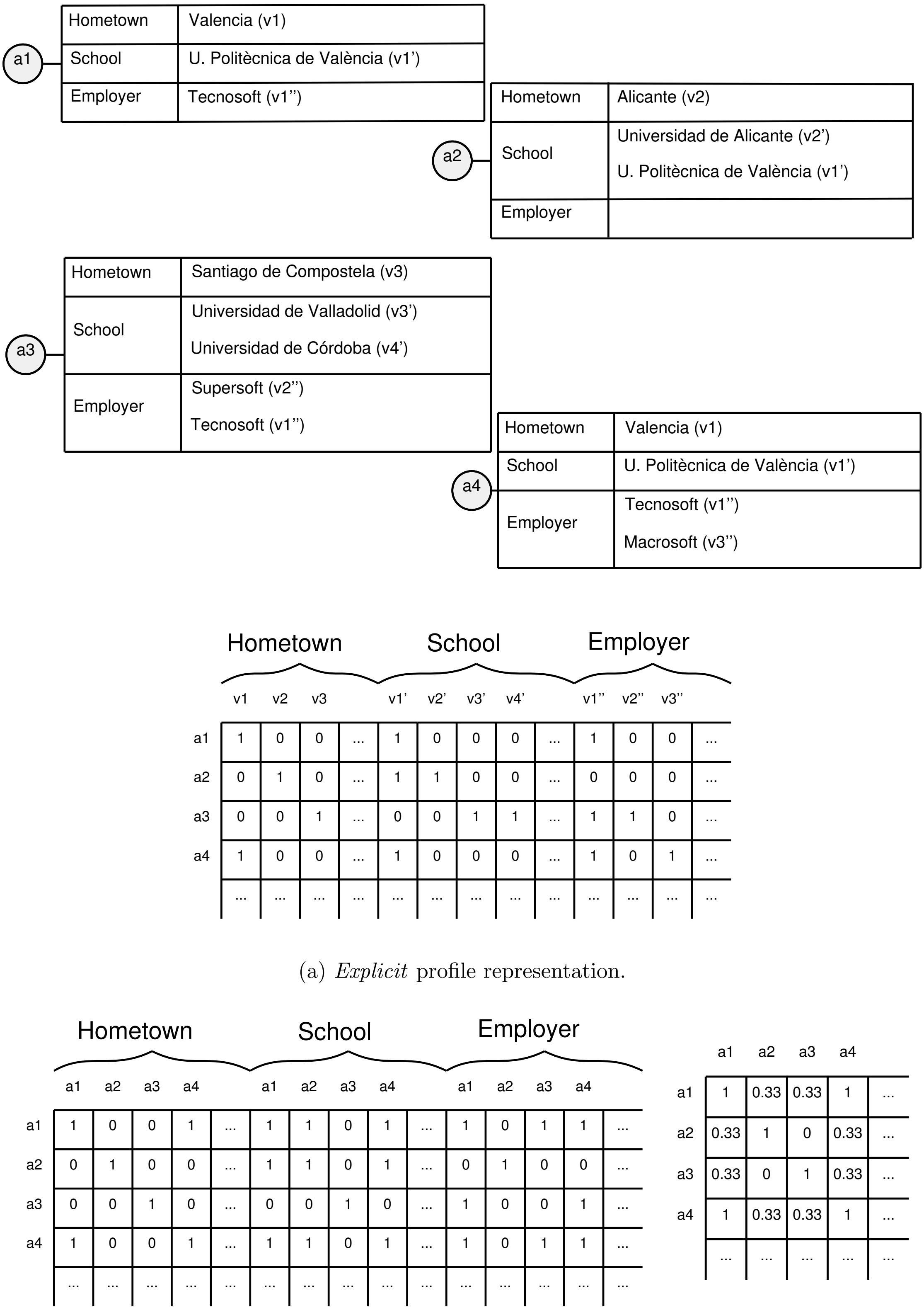
Representation	Definition
1	*Rank* 1
2	*Ranks* 1 and 2
3	*Ranks* 1, 2 and 3
2w	*Ranks* 1 and 2, *weighted*
3w	*Ranks* 1, 2 and 3, *weighted*
2a	*Ranks* 1 and 2, *aggregated*
3a	*Ranks* 1, 2 and 3, *aggregated*

Table 2.2: Representations of structural network information.

2.3 Users' Profile Information

Due to the problems inherent in the use of the users' profiles as a source of information, not every profile feature contained in the dataset can be employed for prediction. Therefore, a subset of them is to be chosen. Being $\mid f \mid$ the number of chosen features, and $\mid v \mid$ the total number of values of the chosen features that are taken by at least one *alter* in the *egonet*, we encode the users' profile information into the matrices $\mathbf{P}$. We have constructed the users' profile representations described below. A graphical example of these representations is shown in Figure 2.2.

- *Explicit* (`e`): There is a column of $\mathbf{P}$ for every different value of the considered features. An element of $\mathbf{P}$ is 1 if the row *alter* takes the column value for the respective feature, and 0 otherwise. We obtain thus in total $\mid v \mid$ profile features for each *alter*. This representation is the one shown in Figure 2.2a.
- *Intersection* (`i`): There is one column of $\mathbf{P}$ for every *alter* in the *egonet* and every considered profile feature. An element of $\mathbf{P}$ is 1 if the sets of values of the row *alter* and the column *alter*, for that particular feature, intersect. It is 0 otherwise. In this case, we obtain $\mid f \mid \times \mid a \mid$ profile features for each *alter*. An example of an *intersection* representation is presented in Figure 2.2b.
- *Weighted* (`w`): There is just one column of $\mathbf{P}$ for every *alter* in the *egonet*. An element of $\mathbf{P}$ represents the proportion of features for which the row *alter* and the column *alter* share at least one value. It is calculated as $\frac{|s|}{|f|}$, where $\mid s \mid$ is the number of features shared between both users. With this representation, we reduce the number of profile features to $\mid a \mid$. This representation can be seen in Figure 2.2c.



a1	Hometown	Valencia (v1)
	School	U. Politècnica de València (v1’)
	Employer	Tecnosoft (v1”)

a2	Hometown	Alicante (v2)
	School	Universidad de Alicante (v2’) U. Politècnica de València (v1’)
	Employer	

a3	Hometown	Santiago de Compostela (v3)
	School	Universidad de Valladolid (v3’) Universidad de Córdoba (v4’)
	Employer	Supersoft (v2”) Tecnosoft (v1”)

a4	Hometown	Valencia (v1)
	School	U. Politècnica de València (v1’)
	Employer	Tecnosoft (v1”) Macrosoft (v3”)

	Hometown				School					Employer			
	v1	v2	v3		v1’	v2’	v3’	v4’		v1”	v2”	v3”	
a1	1	0	0	...	1	0	0	0	...	1	0	0	...
a2	0	1	0	...	1	1	0	0	...	0	0	0	...
a3	0	0	1	...	0	0	1	1	...	1	1	0	...
a4	1	0	0	...	1	0	0	0	...	1	0	1	...
	...	...	...	...	...	...	...	...	...	...	...	...	...

(a) *Explicit* profile representation.

	Hometown					School					Employer				
	a1	a2	a3	a4		a1	a2	a3	a4		a1	a2	a3	a4	
a1	1	0	0	1	...	1	1	0	1	...	1	0	1	1	...
a2	0	1	0	0	...	1	1	0	1	...	0	1	0	0	...
a3	0	0	1	0	...	0	0	1	0	...	1	0	0	1	...
a4	1	0	0	1	...	1	1	0	1	...	1	0	1	1	...
	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

(b) *Intersection* profile representation.

	a1	a2	a3	a4	
a1	1	0.33	0.33	1	...
a2	0.33	1	0	0.33	...
a3	0.33	0	1	0.33	...
a4	1	0.33	0.33	1	...
	...	...	...	...	...

(c) *Weighted* profile representation.

FIGURE 2.2: Feature construction from the users’ profile information. 3 profile features are considered: *hometown*, *schools* and *employers*.

Chapter 3

Prediction Techniques

3.1 Multi-Assignment Clustering

Multi-Assignment Clustering (MAC) [17, 18] was developed as an unsupervised learning technique, with the intention to remove the restrictions on having disjoint clusters, inherent to every classical clustering method. The exclusive assignment of every data object into just one cluster often results severely strict, depending on the concrete problem being modelled. Instead of only providing a partition of the dataset, the objective of MAC is to infer the hidden structure responsible for generating the data. Within this generative approach, multiple clusters can simultaneously generate a data item. It presents some differences with fuzzy clustering, as well. In fuzzy clustering, an object is partially (softly) assigned to several clusters, obtaining fractional assignments which must sum up to 1. In contrast, given a data item, MAC provides hard assignments to several different clusters.

MAC, which was originally developed for Boolean data, tries to provide a decomposition of the input data matrix $\mathbf{X}$ into a matrix containing the clusters prototypes $\mathbf{Z}$ and a matrix representing the degree to which a particular data vector belongs to the different clusters $\mathbf{Y}$. Theoretically, an exact decomposition has been formulated as the set-cover problem [27, 28]:

$$\mathbf{X} = \mathbf{Z} \otimes \mathbf{Y}, \quad \text{where} \quad X_{ij} = \bigvee_{k} [Z_{ik} \wedge Y_{kj}] \tag{3.1}$$

It has been proven that the set-cover problem is NP-hard and the corresponding decision problem is NP-complete [29]. This makes approximation heuristics necessary. Finding an approximation of the matrix **X**, often more useful than an exact one, can be modelled as the Lossy Compression Problem (LCP), defined in [30, 31]. However, finding the optimal matrices **Z** and **Y** is NP-hard, as proven in [31]. The solution is to find a probabilistic representation, which allows us to drastically simplify the optimization problem.

In [17], the authors propose a mixture model where X_{ij} is either drawn from a signal or a noise component. The probability of X_{ij} under the signal model is the following, being β_{kj} independent random variables for the deterministic centroids **Y**:

$$p(X_{ij}|\mathbf{Z},\beta) = \left[1 - \prod_{k=1}^{K} \beta_{kj}^{Z_{ik}}\right]^{X_{ij}} \left[\prod_{k=1}^{K} \beta_{kj}^{Z_{ik}}\right]^{1-X_{ij}} \text{, where } \beta_{kj} := p(Y_{kj} = 0) \tag{3.2}$$

Alternatively, the probability of X_{ij} under the noise model, characterised as a Bernoulli distribution parameterized by r, is the following:

$$p(X_{ij}|r) = r^{x_{ij}}(1-r)^{1-x_{ij}} \tag{3.3}$$

All the model parameters are inferred by Deterministic Annealing (DA) [32, 33]. DA is a gradient descent optimization method that provides a smooth transition from the uniform distribution to a solution with minimal expected risk.

After running the algorithm, when the probabilistic decomposition of the matrix **X** is available, the matrix **Y** is the one indicating which data objects belong to the different clusters.

3.2 Restricted Boltzmann Machine

In this section, we are going to introduce a generative model known as Restricted Boltzmann Machine (RBM). Like in every generative model, its goal is to approximate the probability density function (pdf) of the data in an unsupervised manner, i.e. without any label information. Once the parameters of the generative model have been learned, the model can be used to obtain likely samples. RBMs make one of the simplest generative models, and are often used as the building block for more complex ones.

We part from the fact that it may be useful to extract data representations that capture the probability density function $p(\mathbf{X})$ of the available data. An RBM is a generative model that deals with such a probability distribution in order to extract feature representations that maximize the likelihood of the samples. In the context of a classification problem, these representations can facilitate the modelling of the real discriminative target distribution $P(C|\mathbf{X})$. However, they are useful on their own, not necessarily having to be in relation to a classification problem. Moreover, RBMs can be used as the elementary units of more complex Deep Learning architectures.

An RBM is an energy model with two different sets of variables. The visible variables, denoted by $\mathbf{v}$, are related to the data $\mathbf{X}$. On the other hand, the hidden variables, denoted by $\mathbf{h}$, are used to increase the expressiveness of the model. The RBM is characterized by a function that defines a probability distribution over all possible pairs of visible and hidden variables by assigning low energy values to high probability samples. This relation is defined by:

$$p(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} e^{-E(\mathbf{v},\mathbf{h})} \tag{3.4}$$

where the partition function Z is given by summing over all possible pairs of visible and hidden variables, so that $p(\mathbf{v}, \mathbf{h})$ is a probability distribution.

This generative model can be implemented as a neural network with two layers. The visible layer is the input of the network, so that each unit v_i represents the *i-th* component of a data sample. Figure 3.1 shows a graphical representation of an RBM.

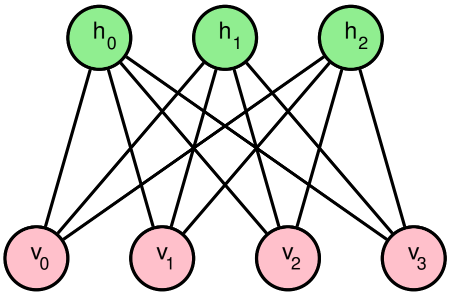


Figure 3.1: Graphical representation of an RBM model. The green and the pink units correspond to the hidden and the visible layers, respectively.

Originally, RBMs were designed to work with binary visible and hidden variables. In this case, the energy function is defined by:

$$E_{RBM}(\mathbf{v},\mathbf{h}) = -\sum_{i\in vis} a_i v_i - \sum_{j\in hid} b_j h_j - \sum_{i,j} v_i h_j w_{ij} \tag{3.5}$$

where v_i,h_j are the binary states of the visible unit i and hidden unit j, a_i,b_j are their respective biases and w_{ij} is the weight that connects both units.

A nice property of the RBM models is that the hidden units are mutually independent given the visible units and vice-versa. Therefore, the conditional distribution over the hidden units can be factorized given the visible units:

$$p(h_j = 1|\mathbf{v}) = \sigma(b_j + \sum_i w_{ij} v_i) \tag{3.6}$$

where $\sigma(x)$ is the transfer function. For binary units, $\sigma(x)$ takes the form of the sigmoid function $(1 + e^{-x})^{-1}$. Likewise, the conditional distribution over the visible units given the hidden units also factorizes:

$$p(v_i = 1|\mathbf{h}) = \sigma(a_i + \sum_j w_{ij} h_j) \tag{3.7}$$

During the training process, the parameters of the model are adjusted, so that the log-likelihood of the training data is maximized. Let $\mathcal{L}(\theta, \mathcal{D})$ be the log-likelihood of the data defined as:

$$\mathcal{L}(\theta, \mathcal{D}) = \sum_{\mathbf{x}\in\mathcal{D}} \log p(\mathbf{x}) \tag{3.8}$$

where θ are the parameters of the model and $\mathbf{x}$ is a sample of the training set $\mathcal{D}$. It is important to note that the log-likelihood definition does not require the samples to be labelled, and so the training process of the RBM model is completely unsupervised. The log-likelihood is maximized using stochastic gradient descent with a random initialization of the model parameters. In the case of an RBM, this leads to a very simple update rule (see [34] for details):

$$\Delta w_{ij} = \epsilon \left(\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model} \right) \tag{3.9}$$

where ϵ is a learning rate, $\langle v_i h_j \rangle_{data}$ is the frequency of the visible unit i and hidden unit j being jointly active when the model is driven by samples of the training set, and

$\langle v_i h_j \rangle_{model}$ is the corresponding frequency when the model is let free to generate likely samples (not driven by data). A simplified version of the same learning rule is also used for the biases. The first term, $\langle v_i h_j \rangle_{data}$ is very easy to obtain using Eq. 3.6, Eq. 3.7 and feeding the model with random training samples. However, it is much more difficult to obtain an unbiased sample of $\langle v_i h_j \rangle_{model}$ since the model has to be started with a random state of the visible layer and then perform alternating Gibbs sampling for a very long time until the model reaches equilibrium.

A much faster learning procedure called Contrastive Divergence (CD) was proposed by [34]. This method basically uses two tricks to speed up the learning process. On the one hand, the process is initialized by setting a training example in the visible layer. On the other hand, CD does not wait for the sampling process to converge, i.e. samples are obtained after only k-steps of the Gibbs sampling. In practice, $k = 1$ has been shown to work well for most applications.

As we have said before, the standard RBM model uses binary units in both visible and hidden layers with the sigmoid transfer function. However, many other types of units can be used as well. For instance, for image data, binary units are not adequate to represent pixel values. A solution to this problem is to replace the binary visible units with Gaussian units. The transfer function for Gaussian units is the identity function. Another type of units, that have recently shown some improvements are Rectified Linear Units (ReLU) [35]. In this case, the transfer function is given by $f(x) = \max(0, x)$, where x is the input of the neuron. The main advantage of these units is that they do not have more parameters than an ordinary binary unit, but they are much more expressive.

For additional details about the characteristics of the RBM model and its training procedure the reader is encouraged to check [36].

3.3 K-means

K-means [7] is the most famous and studied partitional clustering method. It divides the dataset into k clusters by minimizing the Sum of Squared Errors (SSE) between the samples x and their respective cluster means m_c.

$$\text{SSE} = \sum_{c} \sum_{x \in X_c} \|x - m_c\|^2 \tag{3.10}$$

K-means algorithm [37] is performed iteratively in two steps, until convergence:

1. Assignment Step: Each sample is assigned to the cluster whose mean yields the least within-cluster sum of squares. Intuitively, due to the assimilation of SSE to the squared Euclidean distance, this is understood as the nearest mean.

$$X_c^{(t)} = \{x : \|x - m_c^{(t)}\|^2 \leq \|x - m_i^{(t)}\|^2, \forall i = 1, \ldots, k\} \tag{3.11}$$

2. Update Step: The new means are the centroids of the samples in the new clusters.

$$m_c^{(t+1)} = \frac{1}{\mid X_c^{(t)} \mid} \sum_{x \in X_c^{(t)}} x \tag{3.12}$$

The algorithm converges to a local optimum, although there is no guarantee that the global optimum is found. Several methods exist to define the initial means, either randomly choosing k samples and using them as means, or making a random partition of the dataset and computing the means of the random clusters.

Chapter 4

Evaluation Metrics

The problem of social circles detection needs the definition of some evaluation metrics, in order to assess the performance of the proposed algorithms. The aim of any of those evaluation metrics $E(\mathcal{C},\bar{\mathcal{C}})$ is to measure the similarity between the set of predicted circles $\mathcal{C} = \{C_1, \ldots, C_K\}$ and the set of ground-truth circles $\bar{\mathcal{C}} = \{\bar{C_1}, \ldots, \bar{C_K}\}$. In this regard, two main approaches are found in the literature. The first one is based on the definition of a similarity score $s(C, \bar{C})$ between two circles, with a further calculation of the best alignment between $\mathcal{C}$ and $\bar{\mathcal{C}}$. The second is based on an edit distance between $\mathcal{C}$ and $\bar{\mathcal{C}}$. In each case described below, the final evaluation measure is the average of the evaluation measures obtained for all the *egonets* in the respective dataset.

Within the first approach, the similarity measures s can perfectly be well-established similarity metrics between sets. The following have been used in the references:

- Jaccard Coefficient [38]:

$$J(C, \bar{C}) = \frac{\mid C \cap \bar{C} \mid}{\mid C \cup \bar{C} \mid} \tag{4.1}$$

- F-measure [21]:

$$F(C, \bar{C}) = 2 \times \frac{\text{precision}(C, \bar{C}) \times \text{recall}(C, \bar{C})}{\text{precision}(C, \bar{C}) + \text{recall}(C, \bar{C})} \tag{4.2}$$

$$\text{being } \text{precision}(C, \bar{C}) = \frac{\mid C \cap \bar{C} \mid}{\mid C \mid}, \text{ and } \text{recall}(C, \bar{C}) = \frac{\mid C \cap \bar{C} \mid}{\mid \bar{C} \mid}$$

- Balanced Error Rate [39]:

$$B(C,\bar{C}) = \frac{1}{2}\left(\frac{\mid C \setminus \bar{C} \mid}{\mid C \mid} + \frac{\mid \bar{C} \setminus C \mid}{\mid \bar{C} \mid}\right) \tag{4.3}$$

For this kind of metrics, the calculation of an optimal alignment between the sets of predicted circles and ground-truth circles is required. Two of them are found in the references. The first one is described in [40] as follows: every detected circle is matched with its most similar ground-truth community, and the performance is computed. After that, every ground-truth community is matched with its most similar predicted community, and the performance is computed again. The final evaluation function is the average of the two performance measures. This results in the Best Matching evaluation measure:

$$E_b(\mathcal{C},\bar{\mathcal{C}}) = \frac{1}{2|\bar{\mathcal{C}}|}\sum_{\bar{C}_i\in\bar{\mathcal{C}}}\max_{C_j\in\mathcal{C}} s(\bar{C}_i,C_j) + \frac{1}{2|\mathcal{C}|}\sum_{C_j\in\mathcal{C}}\max_{\bar{C}_i\in\bar{\mathcal{C}}} s(\bar{C}_i,C_j) \tag{4.4}$$

The average is calculated because matching only from one side leads to degenerate optimal performance (for example, outputting all possible subsets of nodes as detected communities would achieve perfect matching ground-truth communities to the detected ones). However, this measure is too optimistic, several ground-truth circles can be aligned to just one predicted circle or vice versa, without any penalization to non-aligned predicted or ground-truth circles.

In [20, 21] the alignment is defined as an optimal correspondence via linear assignment, found by means of the Hungarian algorithm [41], what leads to the Hungarian Matching evaluation measure:

$$E_h(\mathcal{C},\bar{\mathcal{C}}) = \max_{f:\mathcal{C}\to\bar{\mathcal{C}}}\frac{1}{|f|}\sum_{C\in\mathrm{dom}(f)}(1-s(C,f(C)) \tag{4.5}$$

This approach ensures that, unlike in the previous case, there are no cases of single-to-multiple circles alignment. Nevertheless, the use of the Hungarian algorithm makes the set having the smallest number of circles to have all its circles aligned, whereas the other set will always have a number of $\max(|\mathcal{C}|,|\bar{\mathcal{C}}|) - \min(|\mathcal{C}|,|\bar{\mathcal{C}}|)$ circles without being aligned at no cost. Sadly, this leads to degenerate optimal performance, as having all possible subsets of nodes as detected communities would achieve a perfect matching. In [20, 21], the authors state that this kind of undesirable behaviour only happens when the number of predicted circles is greater than the number of ground-truth circles. However, E_h presents other problems, as well. For instance, predicting only one perfect circle would have a perfect matching as well, forgetting that the rest of ground-truth circles remain without prediction.

The use of an edit distance as an evaluation measure for the task was introduced at the Kaggle competition on learning social circles in networks [25]. This Edit Distance evaluation measure, $E_d(\mathcal{C}, \bar{\mathcal{C}})$, has four basic edit operations, all of them at cost 1:

- Adding a user to an existing circle
- Creating a circle with one user
- Removing a user from a circle
- Deleting a circle with one user

We believe that this is the most complete and accurate evaluation measure of the ones described, as it is a global measure between sets of circles, it does not consider single-to-multiple circles alignments and it does not lead to degenerate optimal performance.

Chapter 5

Experiments

5.1 Experiments Using Multi-Assignment Clustering

In this section, we apply the Multi-Assignment Clustering (MAC) technique to predict social circles. It has been already employed and considered as a baseline method for this task in [20, 21], although using only the users' profile information. The authors of those works define an alternative method which outperforms all the proposed baselines, including MAC. However, we believe that this technique still has potential for the problem, given that the data feeding the algorithm is conveniently represented. Thus, we include the structural network information and feed the algorithm with the novel data representations defined in Chapter 2, with which we hope to improve the results. In addition, the evidence that MAC is a state of the art technique, having recent and influential publications, helped us making the choice over alternative soft-clustering strategies. Furthermore, MAC is more adequate than other methods with a very high computational cost, like clique percolation. Our final aim is to compare our results to the ones provided by the technique defined in [20, 21]. The experiments that we have conducted using MAC have led to a publication [42], while another article has been submitted to a conference and is awaiting revision.

5.1.1 First Set of Experiments

The first experiments that we have conducted are designed to discover which data representations are best suited for the task, and whether the combination of structural network and users' profile information provides an improvement of the results over each of the sources separately. For these experiments, we have used the *Kaggle* dataset and the Edit Distance evaluation measure (E_d), as we believe it is the most complete. In addition, we do not incorporate a prediction technique for the number of social circles within the *egonet*, leaving the focus of the study on the data representations.

For the structural network information, we use all the representations appearing in Table 2.2. With respect to the users' profile information, due to the problems that it presents, we have selected the 3 most informative features, which are: *hometown*, *schools* and *employers*. We consider the representations defined in Section 2.3, as well. Note that, unlike the original MAC, we allow the input to be real data in $[0,1]^n$ as a way to model a hierarchy of link levels in the case of structural information, or an aggregation of the number of feature values shared by two users profiles in the case of users' profile information.

The degree of a given user is defined as the number of different circles which it belongs to. MAC takes as a parameter the range of possible degrees of the *alters* in an *egonet*. In all our experiments, the minimum degree is set to 0 and we try several values for the maximum degree, up to 3. In this regard, we do not include any prediction technique for the number of circles within the *egonets*, using the number of circles of the ground-truth instead.

We compare our results to several different baselines:

- *MAC only structure.* Using only structural network information. The `1` representation defined in Section 2.2 is employed.
- *MAC only profile.* Using only users' profile information. The *explicit* (`e`) representation defined in Section 2.3 is employed.
- *Empty circles.* It consists in defining an empty set of circles, $\mathcal{C} = \emptyset$, and relies on the fact that the E_d evaluation measure heavily penalizes the misclassification of users into circles. Thus, defining no circle at all performs better than other possible simple baselines like connected components or classifying all the *alters* into just one circle.
- *Clique percolation.* We have considered a very high-performing baseline by using a 5-clique percolation algorithm. However, this cannot be done for every *egonet*

due to its exponential computational complexity. Therefore, we replace the clique percolation predictions by empty circles in those cases.

As the *Kaggle* dataset was borrowed from a competition, another interesting baseline would have been composed of results of the participants and, particularly, of the top ranking positions. Unfortunately, there are no publicly available rankings for the labelled subset employed in this thesis. Thus, there is no possibility to make this comparison.

Baseline	E_d
MAC only structure	311.32
MAC only profile	337.85
Empty circles	285.02
Clique percolation	255.83

Data representation		E_d		
Structural	Profile	deg. 1	deg. 2	deg. 3
2w	e	282.70	280.05	280.45
2w	i	267.20	272.67	265.45
2w	w	283.35	282.58	282.00
3w	e	285.10	284.22	284.70
3w	i	275.33	275.07	275.30
3w	w	283.23	284.42	284.58
2a	e	263.28	260.32	**259.50**
2a	i	273.88	280.67	278.23
2a	w	262.08	262.52	260.42
3a	e	280.07	279.50	278.38
3a	i	282.67	283.33	288.38
3a	w	277.23	275.70	275.97

Table 5.1: Baselines and results of the first set of experiments conducted using Multi-Assignment Clustering. The best performing of the structural network and users' profile representations defined in Chapter 2 are shown. The evaluation measure is the Edit Distance E_d defined in Chapter 4.

The values of E_d obtained by the baselines and our experiments are shown in Table 5.1. Only results obtained from *weighted* and *aggregated* structural network representations are presented, as *non-weighting* has always performed worse. The best results have been produced when considering *friendship* of *ranks* 1 and 2, *aggregated*; and an *explicit* representation of the profile features information, allowing MAC for a maximum degree of 3. This representation has provided a value of E_d close to that obtained from the clique percolation baseline. All the experiments using the structural network representation `2a` have given low values of E_d, outperforming the empty circles baseline in all the cases and most of the other representations as well. The combination of (*weighted*) structural network information and profile features has always performed better than structure or profile separately. Our best result outperforms all the baselines, excepting clique percolation. However, computationally this is a very demanding technique, and scaling it to bigger networks would be complicated, whereas MAC would be less problematic.

The results of these experiments have been published in [42].

5.1.2 Second Set of Experiments

The satisfactory results of the former experiments made us want to explore further the possibilities of MAC for social circles detection. In this regard, we have designed and conducted an additional set of experiments, with the objective of comparing our results to the ones provided by a state of the art technique. The novelties of these experiments are:

- Along with the *Kaggle* dataset, the experiments are conducted on the *ego-Facebook* dataset, as well.
- We report the three evaluation metrics defined in Chapter 4: Best Matching (E_b), Hungarian Matching (E_h) and Edit Distance (E_d). E_b and E_h need the definition of a similarity score s between circles. In this regard, the performance of the Jaccard coefficient, the F-measure and the balanced error rate is similar; and we have decided to employ the F-measure.
- Only the `2a` structural network representation defined in Section 2.2 has been used, as it has obtained the best results in the former experiments.
- The same 3 profile features from the former subsection have been used. However, we employ 3 different subsets of them: the most important feature (*hometown*), the 2 most important features (*hometown* and *schools*), and the 3 of them (*hometown*, *schools* and *employers*). We do not consider the *intersection* representation `i`, as

it has provided the worst results in the former experiments. We want to study how the number of features used in prediction affects the results, and if they improve monotonously according to this number.

- Again, we do not include any prediction technique for the number of circles within the *egonets*, but, in this case, we predict in any case a fixed number of 35 circles. We rely on MAC to leave empty the extra circles.

We compare our results again to several different baselines:

- *The method employed in* [20, 21]. In this regard, we cannot replicate the results reported in those works, as we do not have the best performing parameterization. This is the most important baseline, as our main objective is to compare our results to the ones provided by this technique.
- *Empty circles.* Defined as in the previous subsection, it can be evaluated only by the E_d evaluation measure, as there is no possible alignment between the set of ground-truth circles $\bar{\mathcal{C}}$ and an empty set. However, it is interesting to report this baseline, as the E_d evaluation measure heavily penalizes the misclassification of users into circles.
- *All in one circle.* The last baseline is constructed by defining an only circle composed by all the *alters* in the *egonet.* This baseline performs especially well when evaluated by the E_h measure. The reason is that the greatest grountruth circle gets aligned with the provided circle and the F-measure between them is the reported result. The larger this ground-truth circle, the better the performance of this baseline. The rest of ground-truth circles, without an aligned predicted circle, bring no penalty to that result.

The results obtained by the baselines and our experiments are shown in Table 5.2. Our best results have been obtained when considering the set of 3 *weighted* profile features for the *Kaggle* dataset, and the set of 1 *weighted* profile feature for the *ego-Facebook* dataset. Our results, evaluated by the E_b and E_h measures, have never obtained a better performance than the baseline. However, when using the Edit Distance E_d, the results of our method have always been the best-performing.

		Dataset					
		Kaggle			*ego-Facebook*		
Baseline		E_b	E_h	E_d	E_b	E_h	E_d
Method in [20, 21]		0.4714	0.5739	267.23	0.3899	0.5335	502.40
Empty circles		*	*	285.02	*	*	423.30
All in one circle		0.3771	0.5318	352.67	0.3330	0.5242	570.80
Profile features		E_b	E_h	E_d	E_b	E_h	E_d
Representation	Number						
`e`	1	0.2719	0.3260	263.68	0.1652	0.2579	416.2
	2	0.2871	0.3405	265.47	0.1442	0.1807	417.2
	3	0.2862	0.3394	264.48	0.1631	0.2277	409.4
`w`	1	0.3137	0.3470	270.02	**0.2576**	**0.3509**	**395.9**
	2	0.2985	0.3653	262.72	0.1633	0.2291	415.2
	3	**0.3244**	**0.4076**	**258.63**	0.1954	0.2836	397.7

Table 5.2: Baselines and results of the second set of experiments using Multi-Assignment Clustering. The users' profile representations `e` and `w` are the ones defined in Section 2.3. The evaluation measures E_b, E_h and E_d are defined in Chapter 4.

The datasets show an enormous difference with respect to the number of profile features employed. The *Kaggle* dataset obtains better results with the greatest number of features, whereas the *ego-Facebook* dataset performs best with just one feature. It seems that the *ego-Facebook* dataset contains a solid structural component, while the profile component of the *Kaggle* dataset is richer and provides more information. The results of [20, 21] with E_d classify better than the empty circles baseline for the *Kaggle* dataset, and are similar to the results obtained using MAC. However, they fall under empty circles for the *ego-Facebook* dataset, and the distance to the MAC results is large. This method is more accurate when the profile information is rich, whereas MAC has a higher performance with strong structural components. A fusion of the two methods would possibliy exploit the benefits of both.

5.2 Experiments Using Restricted Boltzmann Machines

From the first moment that we approached the social circles detection task, we were attracted by the idea of using Restricted Boltzmann Machines (RBMs) as a prediction technique. In this regard, we conducted some preliminary experiments, composed of several steps. First of all, an RBM was defined and trained, fed by the whole dataset. After that, the data samples were introduced in the RBM, obtaining as output a new representation of them, with a dimension equal to the desired number of circles. The hidden units could have sigmoid or linear transfer functions. If they were sigmoid, exact 1 components were interpreted as memberships of the sample in the corresponding circle. On the other hand, in the case of using linear transfer functions, we defined a threshold, being interpreted the components greater than that threshold as memberships. Unfortunately, the results were not competitive, but from the insights gained we decided to change the focus and use the RBMs to map the data samples into representations of a fixed dimensionality, instead. These experiments have led to a publication [43].

The authors of [44] have already highlighted the adequacy of RBMs as an unsupervised data mapping technique. One of their main advantages is that, in contrast to other approaches which only permit a reduction of dimensionality, RBMs can project the data into spaces of higher dimensionality, as well. Our experiments rely on the use of RBMs to map the training data samples into new representations of a fixed dimension *dim*. These representations will be later supplied to a k-means clustering algorithm. This 2-step method will finally provide the predictions of the social circles of the given *egos*. A graphical representation of this system is shown in Figure 5.1. The data, as in the experiments using MAC, are composed both of structural network information and users' profile information, employing the different representations defined in Chapter 2.

As a step towards this objective, we design RBMs with a visible layer ($\mathbf{v}$) composed of a number of units equal to the dimension of the data vectors, and a hidden layer ($\mathbf{h}$) composed of a number of units equal to the desired, given dimension *dim*. Both visible and hidden units are binary, with a sigmoid transfer function. Once the topology of the network is defined, the RBMs are trained for a certain number of cycles, using the Contrastive Divergence process defined in Section 3.2. In this regard, we perform only 1 step of the Gibbs sampling (CD_1). After the training process, we have an RBM that, given a data vector of dimension $\mid \mathbf{v} \mid$, provides a representation of dimension *dim*. So we can obtain representations of dimension *dim* of the original data.

Additional experiments were conducted using a system of two stacked RBMs. In this case, we define a first RBM with a visible layer $\mathbf{v_1}$ and a hidden layer $\mathbf{h_1}$. The dimension

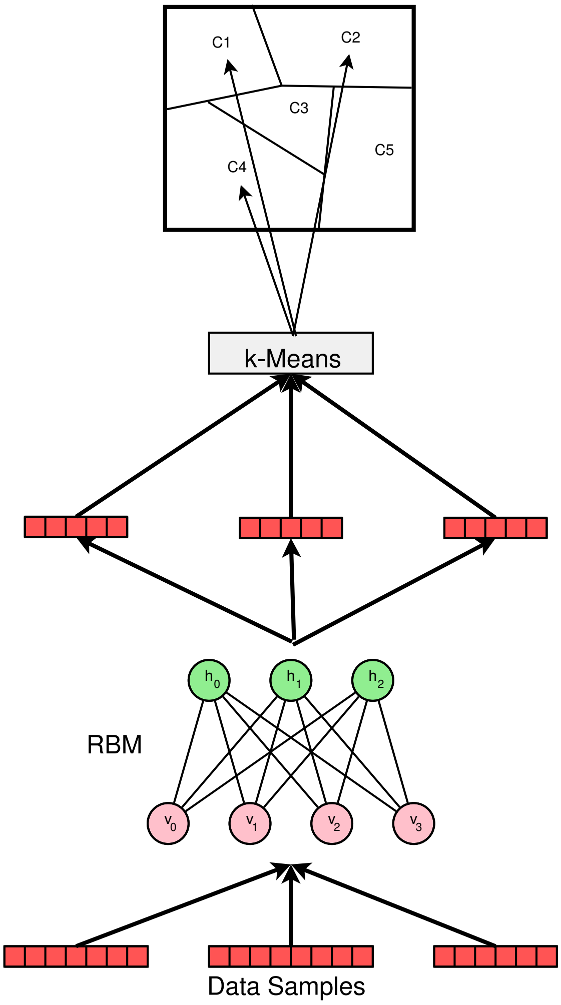


Figure 5.1: Graphical depiction of the 2-step prediction system composed of RBM data mapping and k-means clustering.

of $\mathbf{v_1}$ is the dimension of the data vectors, and the dimension of $\mathbf{h_1}$ is equal to $\mid \mathbf{h_1} \mid = \sqrt{\mid \mathbf{v_1} \mid \times dim}$, being *dim* the final dimension we want to achieve. After training, the output representation of this RBM will be the visible layer $\mathbf{v_2}$ of the second RBM, whose hidden layer $\mathbf{h_2}$ has dimension *dim*. Both RBMs are trained independently for the same number of cycles. This can be interpreted as an RBM with 2 hidden layers $\mathbf{h_1}$ and $\mathbf{h_2}$.

In the framework of this study, we perform k-means on the samples. As a result, we obtain a partition of the dataset into k clusters, which we interpret as social circles. We must notice that this technique will classify every *alter* into a circle, whereas in reality some *alters* are isolated and do not belong to any circle. In addition, it does not allow for overlap or hierarchical inclusion of circles, both phenomena present in a number of *egonets*. Apparently, soft-clustering strategies such as fuzzy clustering or techniques allowing for the hard assignment of an object into different clusters are more well-suited for social circles detection. However, in practice, in some cases partitional or spectral clustering approaches have provided similar or even better results [25, 45], which especially applies when the predictions are assessed by the Edit Distance evaluation measure (E_d). Motivated by these facts, we have chosen the partitional technique k-means as the clustering strategy. In addition, it is illustrative for our aim to check whether or not the predictions improve when we perform the data mapping, in comparison to a clustering over the original, unmapped data.

We have performed a battery of tests on the *ego-Facebook* dataset, assessed by the Edit Distance evaluation measure E_d. All the data representations defined in Chapter 2 have been tested, using the 3 most informative profile features: *hometown*, *schools* and *employers*. In addition, we have tried several values for the following parameters:

- *Number of predicted circles.* We do not include any prediction technique for them. As a lower number seems to provide better results, we have tried the following values: 2, 3, 4, 5, 6, 7, 8
- *Dimension of the RBM-mapped data representations*: 16, 32, 64
- *Number of hidden layers of the RBMs*: 1, 2
- *Number of training cycles of the RBMs*: 100, 500, 1000

The baselines for these experiments are described below:

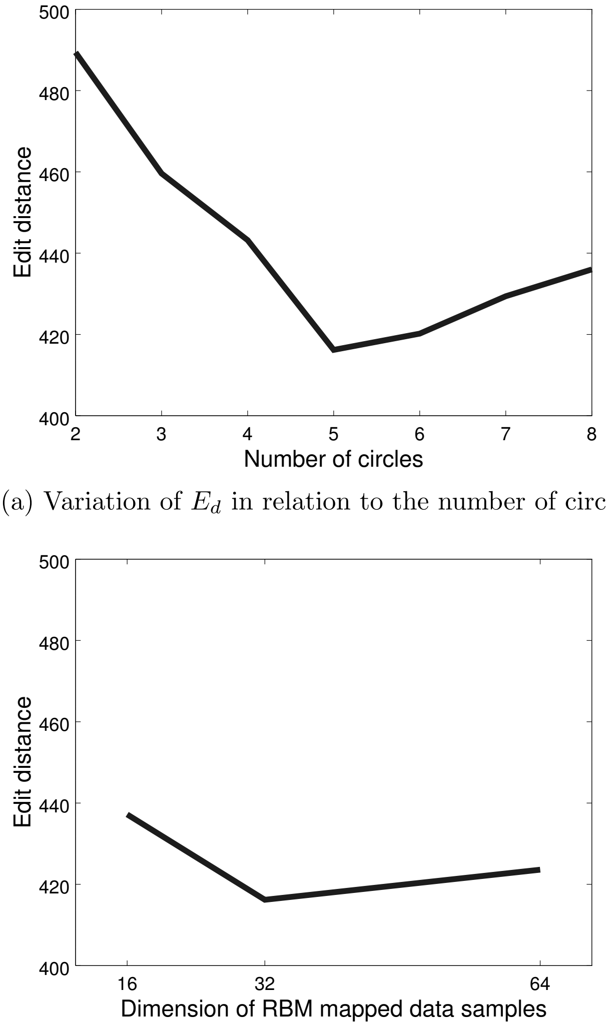
- *The method in* [20, 21]. Again, our main aim is to compare our results to the ones provided by this technique. In contrast to our method, this one does allow for overlap and hierarchical inclusion of clusters; and an *alter* may not be included into any circle. So, apparently, their results should be closer to the ground-truth than the ones obtained by k-means.
- *K-means on the original data.* We conducted all the experiments, employing the same data representations and parameter variation described before, but without mapping the data by RBMs. The best performing of these tests serves us as a second baseline. Its objective is to check whether or not RBM mapping improves the results over just a clustering on the original data.

The evaluation of the baselines and our experiments, in terms of E_d, is shown in Table 5.3. The results obtained using RBMs with 2 hidden layers are not better than the ones resulting from the use of only 1 hidden layer, and so we have discarded them. Moreover, only predictions of 5 circles, $k = 5$, are shown, as they have given the best performance. The k-means only baseline is based on $k = 5$, as well. An example of this behaviour, in relation to the number of circles k, is found in Figure 5.2a, where the value of E_d is compared when considering different values for k. A similar comparison for the dimension *dim* is shown in Figure 5.2b, although it is not so representative of all the cases. The structural network representation 3, the *intersection* users' profile representation i, and predictions obtained from 100 RBM training cycles are also omitted in the table, due to their results being poorer than the ones shown.

Baseline	E_d
Method in [20, 21]	502.4
Only k-means	441.2

Data representation		RBM parameters					
Structural	Profile	$dim = 16$		$dim = 32$		$dim = 64$	
		$it = 500$	$it = 1000$	$it = 500$	$it = 1000$	$it = 500$	$it = 1000$
1	None	441.6	432.6	425.2	437.8	439.6	461.2
1	e	441.0	441.4	449.0	440.2	436.6	439.0
1	w	478.6	453.6	483.4	436.8	472.0	466.8
2	None	452.2	425.8	444.8	443.0	427.4	424.2
2	e	448.6	440.8	430.8	428.8	459.6	434.6
2	w	444.4	455.4	437.4	425.0	438.4	457.2
2w	None	436.2	431.4	427.4	427.8	431.0	435.8
2w	e	437.4	436.8	429.4	422.2	429.0	432.4
2w	w	452.2	442.4	437.4	422.0	458.4	447.8
3w	None	453.2	446.6	426.2	436.6	445.6	435.4
3w	e	432.0	425.2	431.0	438.2	451.0	439.0
3w	w	437.4	447.6	478.4	492.6	427.8	423.4
2a	None	434.8	432.0	438.6	432.4	439.8	429.6
2a	e	437.2	423.8	**416.2**	445.0	423.6	451.2
2a	w	446.2	459.4	455.2	440.0	476.4	451.4
3a	None	441.2	431.6	428.8	423.2	433.2	425.6
3a	e	441.2	430.8	433.6	435.6	431.0	420.4
3a	w	490.0	476.0	491.4	468.2	486.0	478.8

Table 5.3: Baselines and results of the experiments using RBMs and k-means. The best performing of the structural network and users' profile representations defined in Chapter 2 are shown. The evaluation measure is the Edit Distance E_d defined in Chapter 4.



(a) Variation of E_d in relation to the number of circles.

(b) Variation of the E_d in relation to the new dimension *dim*.

FIGURE 5.2: Comparison of the performance of the method varying the number of circles and the dimension of the new data representations. The rest of variables are set as in the best performing experiment of Table 5.3.

Every result appearing in the table outperforms the one in [20, 21]. A selection of them are also more accurate than the best one obtained without the preprocessing of data by RBMs. The best result has been obtained when considering *friendship* of *ranks* 1 and 2, *aggregated*; an *explicit* users' profile representation (`e`); and mapping the original data into representations of dimension $dim = 32$ by an RBM with a 500 cycles training process.

The results of these experiments have been published in [43].

Chapter 6

Conclusions and Future Work

The Kaggle competition on learning social circles in networks [25] took place from May to October 2014, and our participation was our first approach to the problem of social circles detection. Our team (named PRHLT4ARO because of the collaboration between our research group and the US Army Research Office) was not at the top positions, but remained in the first third of the teams participating in the contest, as can be seen in Table 6.1. However, the competition piqued our curiosity to perform a deeper study of the problem and try to find better solutions than the ones available at the moment.

Position	Team	Eval. measure
1	tom denton	6637
2	Adrien	6665
3	Misha Siverski (PZAD, MSU, Russia)	6710
...		
61	**PRHLT4ARO**	7999
...		
201	Justin123	12921
202	yang	13305
203	wjwolf	25341

TABLE 6.1: Ranking of the Kaggle competition on learning social circles in networks. The evaluation measure is based on an edit distance, such as the measure E_d defined in Chapter 4.

One of the first insights that we gained is that social circles detection is a fairly recent topic, not extensively studied. Some state of the art studies exist, such as the one in

[20, 21], but there is still room to improve. The problem is assimilated to the older task of community detection in networks, although it presents some peculiarities that make other previously defined community detection procedures suboptimal. It has been later generalized as the problem of community detection in networks with node attributes [40]. The cited works contain the description of some successful prediction techniques, although the evaluation measures in which they are based present some flaws, leading in some cases to degenerate optimal performance.

We base our experiments on two divergent approaches. The first one is focused on the use of Multi-Assignment Clustering (MAC), a method allowing for the detection of overlap or hierarchical inclusion of clusters. It had already been tested as a prediction technique for community detection in social networks in [20, 21]. In that work, only the users' profile information was used for prediction, and MAC was outperformed by the method proposed by the authors. The results presented in this thesis show that, given that the algorithm is fed with the information modelled in the right way, MAC can also constitute a valid technique for predicting social circles.

The second approach that we have followed is based on the application of a classical clustering technique on data representations mapped by Restricted Boltzmann Machines (RBMs). We must notice that this method does not detect the overlap and hierarchical inclusion present in social circles. However, our experiments indicate that it can outperform some other techniques designed to produce clusters in principle more similar to social circles. We have chosen k-means as the clustering technique and conducted an empirical adjustment of the parameters. The main conclusion is that performing the clustering on the mapped data improves the results of a k-means clustering on the original data.

We have constructed several representations for both the structural network information and the users' profile information, and have conducted a thorough experimentation with the aim to understand which of these representations have a strongest predictive power for detecting social circles. In this regard, the combination of both kinds of information has performed better than the use of either the network structure or the users' profiles separately, in most cases. Thus, network structure and profile features are complementary sources of information for this task. In addition, *weighting* of the structural network information with respect to friendship levels is crucial to improve the results, as *non-weighting* has always performed worse. With respect to the number of profile features employed for prediction, it seems dependent on the dataset. Of the two datasets that we have considered, *ego-Facebook* has a greater structural component, and the use of a smaller number of profile features performs better; whereas the *Kaggle* dataset contains more informative profiles, and obtains better results with the addition of extra profile features. Further experiments can be conducted including a greater number of them, to check whether this tendency is preserved.

The trickiest issue within social circles detection is the design of evaluation metrics providing an adequate comparison between the set of ground-truth circles and the set of predicted circles. In this regard, they should penalize incorrect predictions, not only the disagreements between pairs of aligned circles, but also the extra unaligned predicted circles and the ground-truth circles remaining without prediction. The alignments defined for the Best Matching and Hungarian Matching evaluation measures defined in Chapter 4, E_b and E_h, are very favourable and optimistic in these last cases, permitting multiple alignments of one circle, allowing for prediction of extra circles at no cost, or leaving ground-truth circles without prediction at no cost, as well. There is a wide disparity in the results, depending on the evaluation measure employed. On the one side, most baselines are beaten when using the Edit Distance evaluation measure E_d. Only the first experiments using clique percolation remain better performing, by a low margin. Nevertheless, it would be impossible to consider clique percolation for bigger networks, due to its low scalability, in contrast to our methods. On the other side, other state of the art techniques are more well-suited if the evaluation measures E_b and E_h are used.

Over the course of this research, we have thought of some ideas that could inspire future experiments related to social circles detection. They are the following:

- *Use of a greater set of profile features.* Due to the problems inherent to users' profile information, we decided to conduct our experiments on the 3 most informative profile features, according to our criterion: *hometown*, *schools* and *employers*. However, extra tests could be done incorporating some of the less informative features, or using them alone. This could give better results, especially for the *Kaggle* dataset, in which the users' profiles are themselves more informative. Moreover, new representations of this kind of information might be developed or a new dataset including better retrieved profiles could be constructed.

- *Novel representations of the network structure.* Structural network information could be enhanced, for instance, employing representations able to capture cycles. In addition, some new features could be extracted from the network structure. They include node centrality measures such as the eigenvector and betweenness.

- *Variable dimension of mapped data representations.* We defined a fixed value of the dimension of the RBM mapped data representations for every *egonet* in the dataset. However, this dimension could be made dependent on the size of the original data of each *egonet*, thus allowing for more flexibility in the samples mapping.

- *A more in-depth study of MAC.* It would give us a deeper understanding of the intrinsic procedure of MAC. The objective would be to modify the method and adjust it better to social circles detection.

- *Fusion of the method in* [20, 21] *and MAC.* The method in [20, 21], as MAC, performs differently depending on the dataset and the evaluation measure. An adequate fusion of both methods would possibly generalize better.

- *Use of RBMs for prediction.* Despite that the tests done until the moment have not been satisfactory, we still believe that RBMs can constitute a powerful technique for prediction. For this, new ways to use them need to be designed and experiments need to be conducted to check whether the new developments are adequate for the task.

- *Adoption of other prediction techniques.* Finally, other community detection techniques might be adapted and tested. Furthermore, novel methods specifically designed for social circles detection could be developed.

In conclusion, during this research we have opened new perspectives in social circles detection. We have developed new data representations, we have adapted prediction techniques and we have provided a discussion on the adequacy of the evaluation metrics designed for the task. The results are positive and promising. We have drawn some conclusions and listed several lines of future work. Our research has resulted in the publication of two articles, being a third one in revision.

As future work, and based on the insights that we have gained during this research, we will apply the described developments to the detection of social copying communities, in the framework of the research project funded by the US Army Research Office (ARO).

Appendix A

Publications

The research described in this master's thesis has led to the publication of the following articles:

The first article gathered the results of the first set of experiments using Multi-Assignment Clustering:

- J. Alonso, R. Paredes, and P. Rosso. Empirical evaluation of different feature representations for social circles detection. In *Pattern Recognition and Image Analysis*, volume 9117 of *Lecture Notes in Computer Science*, pages 31–38. Springer International Publishing, 2015

The second article gathered the results of the experiments using Restricted Boltzmann Machines and k-means clustering:

- J. Alonso, R. Paredes, and P. Rosso. Data mapping by Restricted Boltzmann Machines for social circles detection. In *Proc. International Joint Conference on Neural Networks (IJCNN'15)*, IEEE, 2015

A third article, including the results of the second set of experiments using Multi-Assignment Clustering, has been submitted to another conference, and is awaiting revision.

Bibliography

[1] J. Scott. *Social Network Analysis: A Handbook.* SAGE Publications, London, UK, 2000.

[2] S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010.

[3] M. A. Porter, J. P. Onnela, and P. J. Mucha. Communities in networks. *Notices Amer. Math. Soc.*, 56(9):1082–1097, 2009.

[4] B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell Syst. Tech. J.*, 49(2):291–307, 1970.

[5] P. R. Suaris and G. Kedem. An algorithm for quadrisection and its applications to standard cell placement. *IEEE Transactions on Circuits and Systems*, 35(3): 294–303, 1988.

[6] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning.* Springer-Verlag, 2009.

[7] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. Fifth Berkeley Symp. on Math. Statist. and Prob.: Volume 1*, pages 281–297. 1967.

[8] S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[9] W. E. Donath and A. J. Hoffman. Lower bounds for the partitioning of graphs. *IBM Journal of Research and Development*, 17(5):420–425, 1973.

[10] M. Girvan and M. E. Newman. Community structure in social and biological networks. *Proc. Natl. Acad. Sci.*, 99(12):7821–7826, 2002.

[11] M. E. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E Stat. Nonlin. Soft. Matter. Phys.*, 69(2):026113, 2014.

[12] U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikoloski, and D. Wagner. On modularity-NP-completeness and beyond. Technical Report 2006-19, ITI Wagner, Faculty of Informatics, Universität Karlsruhe (TH), Germany, 2006.

[13] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *J. Stat. Mech. Theor. Exp.*, 2008:P10008, 2008.

[14] M. E. Newman. Modularity and community structure in networks. *Proc. Natl. Acad. Sci.*, 103(23):8577–8582, 2006.

[15] G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043): 814–818, 2005.

[16] M. G. Everett and S. P. Borgatti. Analyzing clique overlap. *Connections*, 21(1): 49–61, 1998.

[17] A. P. Streich, M. Frank, D. Basin, and J. M. Buhmann. Multi-assignment clustering for boolean data. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 969–976, 2009.

[18] M. Frank, A. P. Streich, D. Basin, and J. M. Buhmann. Multi-assignment clustering for boolean data. *The Journal of Machine Learning Research*, 13(1):459–489, 2012.

[19] D. Zhou, I. Councill, H. Zha, and C. L. Giles. Discovering temporal communities from social network documents. In *Seventh IEEE International Conference on Data Mining*, pages 745–750, 2007.

[20] J. Leskovec and J. Mcauley. Learning to discover social circles in ego networks. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 539–547. Curran Associates, Inc., 2012. URL http://papers.nips.cc/paper/4532-learning-to-discover-social-circles-in-ego-networks.pdf.

[21] J. McAuley and J. Leskovec. Discovering social circles in ego networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 8(1):4, 2014.

[22] N. Pathak, C. DeLong, A. Banerjee, and K. Erickson. Social topic models for community extraction. In *The 2nd SNA-KDD Workshop*, 2008.

[23] M. Sachan, D. Contractor, T. A. Faruqie, and L. V. Subramaniam. Using content and interactions for discovering communities in social networks. In *Proceedings of the 21st international conference on World Wide Web*, pages 331–340, 2012.

[24] K. Dey and S. Bandyopadhyay. An empirical investigation of like-mindedness of topically related social communities on microblogging platforms. In *International Conference on Natural Languages*, 2013.

[25] Kaggle. Learning social circles in networks. http://www.kaggle.com/c/learning-social-circles.

[26] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, June 2014.

[27] J. F. Gimpel. The minimization of spatially-multiplexed character sets. *Communications of the ACM*, 17(6):315–318, 1974.

[28] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms, 2nd ed.* MIT Press, 2001.

[29] L. J. Stockmeyer. The set basis problem is NP-complete. Technical Report RC5431, IBM Watson Research, 1975.

[30] P. Miettinen, T. Mielikäinen, A. Gionis, G. Das, and H. Mannila. The discrete basis problem. In *Proceedings of Principles and Practice of Knowledge Discovery in Databases*, pages 335–346, 2006.

[31] J. Vaidya, V. Atluri, and Q. Guo. The role mining problem: Finding a minimal descriptive set of roles. In *Proceedings of the 12th ACM Symposium on Access Control Models and Technologies*, pages 175–184, 2007.

[32] K. Rose, E. Gurewitz, and G. C. Fox. Vector quantization by deterministic annealing. *IEEE Transactions on Information Theory*, 38(4):1249–1257, 1992.

[33] J. Buhmann and H. Kuhnel. Vector quantization with complexity costs. *IEEE Transactions on Information Theory*, 39(4):1133–1145, 1993.

[34] G. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.

[35] V. Nair and G. E. Hinton. Rectified linear units improve Restricted Boltzmann Machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814, 2010.

[36] G. Hinton. A practical guide to training Restricted Boltzmann Machines. *Momentum*, 9(1):926, 2010.

[37] D. J. C. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003.

[38] P. Jaccard. Nouvelles recherches sur la distribution florale. *Bulletin de la Société Vaudoise des Sciences Naturelles*, 44(163):223–270, 1908.

[39] Y. Chen and C. Lin. Combining SVMs with various feature selection strategies. In *Feature Extraction*, pages 315–324. 2006.

[40] J. Yang, J. McAuley, and J. Leskovec. Community detection in networks with node attributes. In *IEEE 13th International Conference on Data Mining (ICDM)*, pages 1151–1156. IEEE, 2013.

[41] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial & Applied Mathematics*, 5(1):32–38, 1957.

[42] J. Alonso, R. Paredes, and P. Rosso. Empirical evaluation of different feature representations for social circles detection. In *Pattern Recognition and Image Analysis*, volume 9117 of *Lecture Notes in Computer Science*, pages 31–38. Springer International Publishing, 2015. http://dx.doi.org/10.1007/978-3-319-19390-8_4.

[43] J. Alonso, R. Paredes, and P. Rosso. Data mapping by Restricted Boltzmann Machines for social circles detection. In *Proc. International Joint Conference on Neural Networks (IJCNN'15)*. IEEE, 2015.

[44] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[45] T. Denton. Kaggle social networks competition. http://inventingsituations.net/2014/11/09/kaggle-social-networks-competition/.