

Document downloaded from:

<http://hdl.handle.net/10251/66684>

This paper must be cited as:

Alonso Nanclares, JA.; Paredes Palacios, R.; Rosso, P. (2015). Data Mapping by Restricted Boltzmann Machines for Social Circles Detection. International Joint Conference on Neural Networks (IJCNN 2015). IEEE. doi:10.1109/IJCNN.2015.7280653.



The final publication is available at

<http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7256526>

Copyright IEEE

Additional Information

©2015 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Data Mapping by Restricted Boltzmann Machines for Social Circles Detection

Jesús Alonso, Roberto Paredes and Paolo Rosso
Pattern Recognition and Human Language Technologies Research Center
Universitat Politècnica de València
<https://www.prhlt.upv.es/>

Abstract—Social circles detection is a special case of community detection in social network that is currently attracting a growing interest in the research community. In this paper, we propose a two-step technique, making emphasis on the mapping of the data by Restricted Boltzmann Machines (RBMs). Social circles are subsequently inferred by k -means over the preprocessed data. We define different vectorial representations from both structural egonet information and user profile features, and perform a set of tests to adjust the optimal parameters of the RBMs. We study and compare the performance on the ego-Facebook dataset of social circles from Facebook from the Stanford Large Network Dataset Collection. We compare our results with several different baselines.

I. INTRODUCTION

Nowadays, users in social networks tend to organize the contacts in their personal networks by means of social circles, a tool already implemented by the major companies, like for instance Facebook lists or Google+ circles. However, this labelling is still mostly done manually and, therefore, a growing interest has risen in the automatic detection of these circles. In addition, this problem is related to the more general task of community detection in graphs, or the identification of subnetworks in a given network. The main difference between both problems is the use of information from users' profiles, apart from information from the network structure itself.

Despite the lack of a precise and well-accepted definition of community, there is a wide variety of methods and techniques designed to cope with community detection [1], [2]. Moreover, some techniques specifically designed for social circles detection are being developed currently [3], [4]. In this article, we present a novel two-step method to tackle this problem. First, we map the data by means of Restricted Boltzmann Machines (RBMs). Second, we cluster the preprocessed data using the classical k -means algorithm. The work takes into account different representations of both structural network information and user profile information, and a comprehensive set of tests has been designed to find the optimal parameters for the RBMs.

The rest of the paper is structured as follows. In Section Two, we present previous works on community detection and social circles detection. In Section Three, we describe the Restricted Boltzmann Machine, a generative architecture of which we make use in our research. In Section Four, we describe thoroughly our methodology, from the different data representations proposed to the RBM topologies we defined and the use of k -means for social circles detection. In Section

Five, we comment the experiments we have conducted, including a presentation of the dataset and the evaluation measure, and we discuss the obtained results. Finally, we draw some conclusions and propose future work.

II. PREVIOUS WORK

A. Community Detection in Networks

From an abstract point of view, a network is equivalent to a graph, defined by a set of nodes connected by edges. Nevertheless, from the point of view of researchers devoted to a diversity of fields, the concept of network has additional connotations. Networks can represent real structures such as social networks, biological networks (neural synaptic networks, metabolic networks), technological networks (the Internet, the World Wide Web), logistic networks (distribution networks), etc. There is no well-accepted formal definition of community in general networks. 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. The relation of membership in a community usually has an extra meaning, and the vertices in 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 network. A considerable number of methods have been developed to solve this problem [1], [2]. The most traditional methods include graph partitioning, hierarchical clustering, partitional clustering and spectral clustering. Graph partitioning consists in dividing the vertices in a fixed number of groups of predefined size, minimizing the number of edges lying between the groups. One of the earliest of those methods, still frequently used, is the Kerninghan-Lin algorithm [5], whereas another significant method is described in [6]. Hierarchical clustering [7] is adequate when a hierarchical structure underlies the graph, with small clusters falling recursively within larger clusters. Partitional clustering separates the nodes of the network into a fixed number of clusters by optimizing a cost function based on a predefined distance between points. The most popular partitional technique is k -means clustering [8], [9], in which we base our work. Spectral clustering techniques [10] are based on the use of the eigenvectors of matrices.

Divisive algorithms form another important family of algorithms for the task of community detection, consisting its methodology on the removal of the edges that connect nodes of different communities, called local bridges. The most famous of them is the Girvan and Newman algorithm [11], [12], based

on a modularity measure introduced as a stopping criterion. As high values of modularity indicate good partitions, it has become the most used and best known quality function. As a consequence, modularity optimization would produce a good solution to the problem. Unfortunately, modularity optimization is an NP-complete problem [13], but there are several algorithms able to find fairly good approximations of the modularity maximum in a reasonable time [14], [15].

In real networks nodes are often shared among different communities. The most popular technique to detect overlapping communities is the clique percolation method [16]. 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. Despite of the good performance of this technique, clique percolation remains a hard computational problem: new and improved implementations still scale worse than some other overlapping community finding algorithms. Multi-assignment clustering (MAC) [17], [18] provides an alternative based on a decomposition of the data matrix into a matrix containing the clusters prototypes and a matrix representing the degree to which a particular user belongs to the different clusters. The model parameters are inferred by deterministic annealing [19], [20].

B. Application to Social Networks and Social Circles Detection

The study of social networks is a research topic with a history of decades and it has been recently revitalized by the appearance of new information and communication technologies which have opened new ways of interacting. Clustering of this social content has been studied designing several procedures. Some approaches base the clustering on the network links [2], while others consider the semantic content of social interactions [21]. In between both methodologies, there has also been work on combining the links and the content for doing the clustering [22], [23]. Very recently, a new technique studied the characteristics of community structures formed around topical discussion clusters, using modularity maximization algorithms [24].

Social circles detection is a special case of this framework. 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 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. Apart from the links of the egonet, user profile information may also be considered in this task. New techniques are currently being developed, being the one appearing in [3], [4] one of the most successful. In it, the authors propose a generative model that considers circle memberships and a circle-specific profile similarity metric. In addition, existing methods, like MAC [3], [4], [17], [18], are being adapted and tested for social circles detection [25].

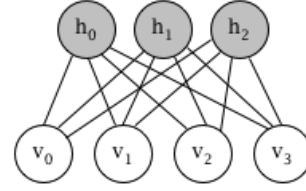


Fig. 1. Graphical representation of an RBM model. The grey and the white units correspond to the hidden and the visible layers, respectively.

III. RESTRICTED BOLTZMANN MACHINE

In this section, we are going to introduce the generative model employed to map the input data, the Restricted Boltzmann Machine. 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 $P(y|\mathbf{x})$. 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:

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

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 \mathbf{x} . Figure 1 shows a graphical representation of an RBM. Originally, RBMs were designed to work with binary visible and hidden variables. In this case, the energy function is defined by:

$$E_{RBM}(\mathbf{v}, \mathbf{h}) = - \sum_{i \in vis} a_i v_i - \sum_{j \in hid} b_j h_j - \sum_{i,j} v_i h_j w_{ij} \quad (2)$$

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:

$$p(h_j = 1|\mathbf{v}) = \sigma(b_j + \sum_i w_{ij}v_i) \quad (3)$$

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:

$$p(v_i = 1|\mathbf{h}) = \sigma(a_i + \sum_j w_{ij}h_j) \quad (4)$$

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:

$$\mathcal{L}(\theta, \mathcal{D}) = \sum_{\mathbf{x} \in \mathcal{D}} \log p(\mathbf{x}) \quad (5)$$

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 [26] for details):

$$\Delta w_{ij} = \epsilon (\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model}) \quad (6)$$

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, Eq. 4 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 [26]. 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 been demonstrated

some improvements recently are the Rectified Linear Units (ReLU) [27]. 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 [28].

IV. METHODOLOGY

A. Using Restricted Boltzmann Machines to Map the Input Data

The authors of [29] 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. In addition, the object of our study is not the comparison of different data mapping techniques, but to undertake the problem of social circles detection. For these reasons we decided to use RBMs to map the data in our experiments.

The experiments presented in this paper rely on the use of RBMs to map the training data samples into new representations of a fixed dimension d . 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. The data samples are composed both of structural network information of the correspondent egonets \mathbf{s} and user profile features of its members \mathbf{p} . So, our data samples \mathbf{x} are formed as a concatenation of these kinds of information: $\mathbf{x} = (\mathbf{s}, \mathbf{p})$. Several representations were designed for both \mathbf{s} and \mathbf{p} . They are explained in high detail in the following subsections.

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 d . 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 formerly. 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 $|\mathbf{v}|$, provides a representation of dimension d . So we can obtain representations of dimension d 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 of \mathbf{v}_1 is the dimension of the data vectors, and the dimension of \mathbf{h}_1 is equal to $|\mathbf{h}_1| = \sqrt{|\mathbf{v}_1| \times d}$, being d 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 d . 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

B. Predicting Social Circles by k -means Clustering

Apparently, soft-clustering strategies such as fuzzy clustering are the most well-suited for social circles detection. However, in practice, some studies show that partitional or spectral clustering approaches provide better results [30], [31]. This especially applies with the error measure that we consider for our study. Motivated by these facts, we have chosen the partitional technique k -means as the clustering strategy for our work. In addition, it is illustrative for our aim to check whether or not the predictions improve when we perform the data mapping.

K -means is a classical partitional clustering method. It divides the dataset into k clusters by minimizing the Sum of Squared Errors, $SSE = \sum_c \sum_{x \in X_c} \|x - m_c\|^2$, between the samples x and their respective cluster means m_c . In the framework of this study, we perform k -means on the samples. As a result, we obtain a partition of the data set into k clusters, which we interpret as social circles.

We must notice that this technique classifies every user into a circle, whereas in reality some users are isolated and do not belong to any circle. In addition, it does not allow for overlap or hierarchical inclusion of circles, both being phenomena that are present in a number of egonets. However, as we have proclaimed, it is a well-suited technique for our purposes.

C. Structural network representation

In this subsection, we present the different representations of the structural network information that have been considered. All of them transform graph links into the vectors s . Being $|u|$ the number of users in the ego-network, we use the following concepts:

- *Friendship ranks*: when there is a link between two users, we say they are direct friends or rank 1 friends. When two users are not direct friends but have a 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 value in s for every friendship rank and user in the egonet. An element of s is 1 if the considered users are friends of such rank, and 0 otherwise. Obtaining in total $3 \times |u|$ structural features for each user.
- *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, obtaining in total $3 \times |u|$ structural features for each user.
- *Aggregation*: for every user, the different friendship ranks are aggregated into just one value. This is obtained by calculating the maximum weighted friendship rank. Reducing the number of structural features to $|u|$.

From these concepts we define the representations shown in Table I.

TABLE I. REPRESENTATIONS OF STRUCTURAL NETWORK INFORMATION

Representation	Definition
$r1$	Rank 1
$r12$	Ranks 1 and 2
$r123$	Ranks 1, 2 and 3
$r12w$	Ranks 1 and 2, weighted
$r123w$	Ranks 1, 2 and 3, weighted
$r12a$	Ranks 1 and 2, aggregated
$r123a$	Ranks 1, 2 and 3, aggregated

D. User profile representation

There are up to 27 profile features for every user in the data corpus we used for the experiments. Nonetheless, some of them are very seldom informed whereas others are redundant or not relevant for the task. As a consequence, we have selected the 3 most informative features which appear for every egonet and we use only these. The selected features are: *hometown*, *schools* and *employers*. Each of these features can take different discrete values from a finite set.

We define as $|f|$ the number of features considered, and as $|n|$ the total number of values of the considered features that are taken by at least one user in the egonet. We encode the profile features information in the vectors \mathbf{p} , for which the following representations have been defined:

- *Explicit*: There is an element of \mathbf{p} for every different value of the considered features. An element of \mathbf{p} is 1 if the considered user takes the column value for the respective feature, and 0 otherwise. Obtaining in total $|n|$ profile features for each user.
- *Intersection*: There is one element of \mathbf{p} for every user in the egonet and every considered profile feature. An element of \mathbf{p} is 1 if the sets of values of the considered users, for that particular feature, intersect. It is 0 otherwise. In this case, obtaining $|f| \times |u|$ profile features for each user.
- *Weighted*: There is just one element of \mathbf{p} for every user in the egonet. An element of \mathbf{p} represents the proportion of features for which the considered users share at least one value. It is calculated as $\frac{|s|}{|f|}$, where $|s|$ is the number of features shared between both users. Reducing the number of profile features to $|u|$.

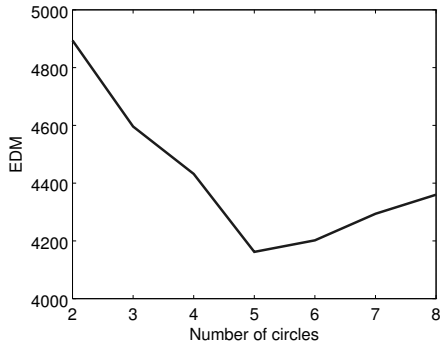
V. EXPERIMENTS AND RESULTS

The corpus used for the experiments is the ego-Facebook dataset of social circles from Facebook from the Stanford Large Network Dataset Collection [32]. The data consist of 10 hand-labelled friendship egonets from Facebook and a set of profile features for every node in those networks which varies depending on the ego. Out of the 10 egonets, the smallest one contains 59 users and the largest one contains 1,045 users. All of them altogether comprise 4,039 users, with 88,234 connections between them.

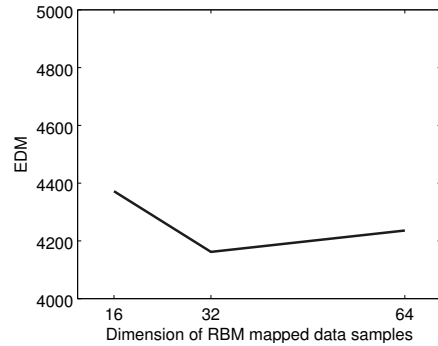
We have performed a battery of tests, not only considering the variations of the data representations, but also trying different values of the dimension $d \in \{16, 32, 64\}$, and the

TABLE II. RESULTS OF THE EXPERIMENTS

		Baseline		EDM			
		Results in [3], [4]		5024			
		Only k -means		4412			
Data representation		RBM parameters					
Structural	Profile	$d = 16$		$d = 32$		$d = 64$	
		$i = 500$	$i = 1000$	$i = 500$	$i = 1000$	$i = 500$	$i = 1000$
		r1	Not used	4416	4326	4252	4378
r1	Explicit	4410	4414	4490	4402	4366	4390
r1	Weighted	4786	4536	4834	4368	4720	4668
r12	Not used	4522	4258	4448	4430	4274	4242
r12	Explicit	4486	4408	4308	4288	4596	4346
r12	Weighted	4444	4554	4374	4250	4384	4572
r12w	Not used	4362	4314	4274	4278	4310	4358
r12w	Explicit	4374	4368	4294	4222	4290	4324
r12w	Weighted	4522	4424	4374	4220	4584	4478
r123w	Not used	4532	4466	4262	4366	4456	4354
r123w	Explicit	4320	4252	4310	4382	4510	4390
r123w	Weighted	4374	4476	4784	4926	4278	4234
r12a	Not used	4348	4320	4386	4324	4398	4296
r12a	Explicit	4372	4238	4162	4450	4236	4512
r12a	Weighted	4462	4594	4552	4400	4764	4514
r123a	Not used	4412	4316	4288	4232	4332	4256
r123a	Explicit	4412	4308	4336	4356	4310	4204
r123a	Weighted	4900	4760	4914	4682	4860	4788



(a) Variation of EDM in relation to the number of circles



(b) Variation of the EDM in relation to the new dimension d

Fig. 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 II

number of training cycles $i \in \{100, 500, 1000\}$. We do not include any prediction technique for the number of circles within the egonets. Instead, we predict a minimum of 2 circles and a maximum of 8 circles for every experiment and check which one works best. A lower number of circles seems to provide better results with this technique.

The evaluation measure of our experiments is the one proposed at the Kaggle competition on learning social circles in networks [30]. We preferred it over other possible evaluation measures, like those based on the calculation of the Balanced Error Rate (BER) [33] or the F_1 score of pairs of circles, as it penalizes more heavily the incorrect inclusion of users into circles. It is calculated as follows:

An evaluation measure for every egonet e is computed as an edit distance between the ground truth circles (g_e) and the predicted circles (p_e): $EDM_e = d(g_e, p_e)$. Four basic edit

operations are considered: adding a user to an existing circle, creating a circle with one user, removing a user from a circle and deleting a circle with one user; every one of them at cost 1. The evaluation measure of the whole dataset is the sum of the edit distances obtained for all the egonets.

$$EDM = \sum_{e \in E} EDM_e, \quad (7)$$

being E the set of the egonets in the corpus. The smaller EDM is, the better the performance of the prediction.

We compare our results to the ones obtained in [3], [4]. In this regard, we calculated the Kaggle measure of the predictions obtained using the authors' method. We must remark that the technique employed in this work does allow for overlapping and hierarchical inclusion of circles. In addition,

a user is not necessarily classified into any circle. So, a priori their results should be closer to the ground truth than the ones obtained by k -means. Moreover, we conducted all the experiments mentioned in the previous section without the RBM data mapping, as well. The best performing of these tests using k -means clustering over the original data serves us as the second baseline. 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.

The evaluation of the baselines and our experiments, in terms of the edit distance EDM, is shown in Table II. Note that only predictions of 5 circles, $k = 5$, are included, as they have given the best performance. The k -means only baseline is based on $k = 5$, as well. An example of this behaviour is found in Figure 2a, where the value of the evaluation measure is compared when considering different numbers of predicted circles. A similar comparison for the dimension d is shown in Figure 2b, although it is not so representative of all the cases. The structural network representation r123, the profile features representation intersection, and predictions obtained from 100 RBM training cycles are also omitted in the table due to the reason that their results are poorer than the ones shown.

Every result appearing in the table outperforms the one in [3], [4]. 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 profile features representation; and mapping the original data into representations of dimension $d = 32$ by an RBM with a 500 cycles training process.

VI. CONCLUSIONS AND FUTURE WORK

RBM mapping of data samples improves the results of k -means clustering for social circles detection. In particular, the combination of friendship information of first and second rank, and an explicit representation of the profile features, seems to give the best performance. After a thorough experimentation (we required to do an optimal adjustment of the parameters of the RBMs) we obtain results that beat the two proposed baselines. Our approach outperforms the generative model proposed in [3], [4], designed to produce clusters in principle more similar to social circles.

This work might be extended in several ways. We defined a fixed value of the dimension of the RBM mapped data representations for every egonet in the dataset. However, it could be made dependant on the size of the original data of each egonet, thus allowing for more flexibility in the samples mapping. A more complex extension would be to use the RBMs themselves for prediction and not only for obtaining new data representations.

ACKNOWLEDGEMENT

This work was developed in the framework of the W911NF-14-1-0254 research project Social Copying Community Detection (SOCOCODE), funded by the US Army Research Office (ARO).

REFERENCES

- [1] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3, pp. 75–174, 2010.
- [2] M. A. Porter, J. P. Onnela, and P. J. Mucha, "Communities in networks," *Notices Amer. Math. Soc.*, vol. 56, no. 9, pp. 1082–1097, 2009.
- [3] J. Leskovec and J. McAuley, "Learning to discover social circles in ego networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 539–547. [Online]. Available: <http://papers.nips.cc/paper/4532-learning-to-discover-social-circles-in-ego-networks.pdf>
- [4] J. McAuley and J. Leskovec, "Discovering social circles in ego networks," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 8, no. 1, p. 4, 2014.
- [5] B. W. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," *Bell Syst. Tech. J.*, vol. 49, no. 2, pp. 291–307, 1970.
- [6] P. R. Suaris and G. Kedem, "An algorithm for quadrisection and its applications to standard cell placement," *IEEE Trans. Circuits Syst.*, vol. 35, no. 3, pp. 294–303, 1988.
- [7] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer-Verlag, 2009.
- [8] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. Fifth Berkeley Symp. on Math. Statist. and Prob.: Volume 1*, 1967, pp. 281–297.
- [9] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–137, 1982.
- [10] W. E. Donath and A. J. Hoffman, "Lower bounds for the partitioning of graphs," *IBM Journal of Research and Development*, vol. 17, no. 5, pp. 420–425, 1973.
- [11] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proc. Natl. Acad. Sci.*, vol. 99, no. 12, pp. 7821–7826, 2002.
- [12] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E Stat. Nonlin. Soft. Matter. Phys.*, vol. 69, no. 2, p. 026113, 2014.
- [13] U. Brandes, D. Delling *et al.*, "On modularity-NP-completeness and beyond," ITI Wagner, Faculty of Informatics, Universität Karlsruhe (TH), Germany, Tech. Rep. 2006-19, 2006.
- [14] V. D. Blondel, J. L. Guillaume *et al.*, "Fast unfolding of communities in large networks," *J. Stat. Mech. Theor. Exp.*, vol. 2008, p. P10008, 2008.
- [15] M. E. Newman, "Modularity and community structure in networks," *Proc. Natl. Acad. Sci.*, vol. 103, no. 23, pp. 8577–8582, 2006.
- [16] G. Palla, I. Derényi *et al.*, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, 2005.
- [17] A. P. Streich, M. Frank *et al.*, "Multi-assignment clustering for boolean data," in *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009, pp. 969–976.
- [18] M. Frank, A. P. Streich *et al.*, "Multi-assignment clustering for boolean data," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 459–489, 2012.
- [19] K. Rose, E. Gurewitz, and G. C. Fox, "Vector quantization by deterministic annealing," *IEEE Trans. Inf. Theory*, vol. 38, no. 4, pp. 1249–1257, 1992.
- [20] J. Buhmann and H. Kuhnel, "Vector quantization with complexity costs," *IEEE Trans. Inf. Theory*, vol. 39, no. 4, pp. 1133–1145, 1993.
- [21] D. Zhou, I. Councill *et al.*, "Discovering temporal communities from social network documents," in *Seventh IEEE International Conference on Data Mining*, 2007, pp. 745–750.
- [22] N. Pathak, C. DeLong *et al.*, "Social topic models for community extraction," in *The 2nd SNA-KDD Workshop*, 2008.
- [23] M. Sachan, D. Contractor *et al.*, "Using content and interactions for discovering communities in social networks," in *Proceedings of the 21st international conference on World Wide Web*, 2012, pp. 331–340.

- [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] J. Alonso, R. Paredes, and P. Rosso, "Empirical evaluation of different feature representations for social circles detection," in *Proceedings of the 7th Iberian Conference on Pattern Recognition and Image Analysis*, 2015, to appear.
- [26] G. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, vol. 14, no. 8, pp. 1771–1800, 2002.
- [27] 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)*, 2010, pp. 807–814.
- [28] G. Hinton, "A practical guide to training restricted boltzmann machines," *Momentum*, vol. 9, no. 1, p. 926, 2010.
- [29] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [30] Kaggle. Learning social circles in networks. [Online]. Available: <http://www.kaggle.com/c/learning-social-circles>
- [31] T. Denton. Kaggle social networks competition. [Online]. Available: <http://inventionsituations.net/2014/11/09/kaggle-social-networks-competition/>
- [32] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection," <http://snap.stanford.edu/data>, Jun. 2014.
- [33] Y.-W. Chen and C.-J. Lin, "Combining SVMs with various feature selection strategies," in *Feature Extraction*, 2006, pp. 315–324.