An AOP-RBPNN approach to infer user interests and mine contents on social media

Andrea Fornaia [a], Christian Napoli [a,*], Giuseppe Pappalardo [a], and Emiliano Tramontana [a]

[a] *Department of Mathematics and Informatics, University of Catania
Viale A. Doria 6, 95125 Catania, Italy
E-mail: {fornaia, napoli, pappalardo, tramontana}@dmi.unict.it*

Abstract. Users engaging in online social networks provide sparse data about themselves, e.g. by participating in *groups* to discuss some topics, linking to each other, etc. Such sparse data can be carefully used to build both user and group profiles, automatically. We put forward a multi-agent system that collects and analyses data scattered on an online social network. The analysis aims at characterising both users, by inserting them into categories, and groups, with a set of key words. The user classification technology is an especially devised neural network that extracts relevant characteristics from raw data characterising user behaviour, and then provides for unknown users the most likely category. Thanks to the said classification tool, some online activities performed by a given user that are unusual for such a user are automatically detected. Moreover, according to the user interests, contents inserted on public pages, which the user is unaware of, can be automatically found and suggested.

Keywords: Neural Networks, Online Social Networks, Content Mining, Knowledge Retrieval, Agent Oriented Programming

1. Introduction

Knowledge retrieval and information extraction on the contents available on online social networks are important concerns for both service providers and subscribers. Given the large size of a social network, in terms of subscribers, information exchanged, and number of links (such as friendship, following, membership to groups, endorsements, etc.), it is desirable to have an automatic way to efficiently analyse the editorial contents to ensure an efficient information spread by a proper selection of sources. User *feature* and *behavioural* analysis are two interesting and important means upon which a solution can be build.

The first step in this direction is to group users into *categories*. While there can be many ways for identifying user categories and computing the degree of mutual affinity between users, interesting performances have been achieved by systems analysing user interests, however, in general, such systems are only intended for a small context, or for analysing selected users. Even though statistical methods make it possible to characterise features and interests for a single user [1], it is difficult to build a proper analytical model for user interactions due to the vastness of data available in a social network, i.e. number of links, undetermined number of subscriber features, etc. A huge amount of features characterise subscribers, however a relevant portion of values for such features is missing for many subscribers in a real environment, hence a complete formulation of a comprehensive analytical model would be unfeasible [2,3].

Other limitations of an analytical model for representing an online social network, and restraining the possibility to analyse the behaviour of subscribers, are given by the dynamic changes of the state, i.e. the whole amount of textual contents and other data, such as e.g. “friendship”, that are continuously updated by users of the online social network itself. Moreover, the large amount of data and the frequency of changes make the numerous reiterations needed to formulate the analytical model very computationally costly. Finally, since data related to subscribers are continuously updated and modified, defining the appropriate vari-

[*] Corresponding author. E-mail: napoli@dmi.unict.it

ables and parameter size needed to solve the problem analytically can be rather difficult. In fact, it is humanly impossible to elaborate a new analytical model, considering all the possible variables, each time an update in the social network asset occurs. Moreover, when considering the great number of possible features characterising a user profile, it is a challenge to select only the minimum number of useful variables and correlations in order to get the needed prediction using a possibly simple analytical model.

Still it is highly desirable to have an automatic processing system that analyses the activities performed online and that can dynamically incorporate data available on the online social network over time. This is fundamental for building advanced services. Such an automated mechanism can take advantage of the soft computing approach, such as soft artificial intelligence. Neural networks have been proven effective for a large number of problems that cannot be solved in terms of a priori mathematical models, especially when used with hybrid architectures [4].

We propose an agent driven artificial intelligence system based on a specific Artificial Neural Network (ANN) architecture called Radial Basis Probabilistic Neural Network (RBPNN), which is well known for its capability to classify and generalise datasets and can be continuously trained to recognise novel features, hence can easily cope with changing data. The proposed neural network has been embedded into a *Classification Agent* that builds a model out of data coming from user profiles, and handled by other agents, such as a *Profiling Agent* and a *Crawler Agent*, which retain useful data from different parts of an online social network [5], such as Facebook(R).

When analysing a social network, as Facebook, the main difficulties are due to: the unknown number of subscribers, friendship relations, groups, followers, etc.; and the unknown size of data and features for each subscriber. We overcome such difficulties thanks to several agents, which gather data and retain a representation for them, after having performed an analysis (one of our analysers processes a big amount of data by resorting to a GPU based solution).

Specifically, our *Classification Agent*, according to the proposed RBPNN solution, can handle partial data, acting as a modeller for dynamically changing user's profiles. With our classification approach, we are able to assign a user to a category, according to his/her behaviour on the network in terms of profile features and post contents. Such categories identify sets of similarly acting users (e.g. users with similar interests or following/posting similar contents) even if such users do not know each others or do not belong to the same groups.

Moreover, the agent system can use the same classification approach to recommend new groups or posts that fit user interests: this is achieved by using group subscriptions as categories, instead of the ones specifically designed by the administrator to classify user behaviour. Our solution, comprising different collaborating agents, is then used to enhance the user experience by suggesting new groups they can subscribe to, according to their interests, or contents such as posts or pages.

The rest of this paper is structured as follows. Section 2 gives the background on the dynamics of a social network. Section 3 describes the proposed multi-agent system based on RBPNNs. Section 4 describes the Classification Agent. Section 5 explains the proposed content mining solution. Section 6 and Section 7 report the performed experiments and results, respectively. Finally, Section 9 draws our conclusions.

2. Social network dynamics

This work analyses data available on online social networks and Facebook is considered as a significant representing example. In social networks, the *small-world* properties are an important characteristic for the actual social dynamic of the network [6]. Moreover, social networks follow a *scale-free* behaviour [7], i.e. a few nodes (i.e. users) act as important hubs centralising a large number of links, hence data passing through such hubs are widely spread on the network.

2.1. Clusters of users in a social network

For online social networks, such as Facebook, we can identify two different kinds of relationships among users: (i) a bidirectional interaction between a pair of users, which occurs when such a pair exchanges a *friendship*, and (ii) a one-way interaction from a user to many, i.e. a user being in a (Facebook) *group* is given means to broadcast contents to all the members of the same group where s/he belongs to. We define the mutual exchange of *friendship* between a pair of users as a *strong* connection between the pair; whereas for a pair of users that are *members* of the same group, the membership provides a *weak* connection between such a pair. When a user posts a content into a group, then the resulting one-to-all interaction provides a *weak*, and sometimes *random*, connection with members of the group, who generally share a limited number of interests.

We define the *distance* between a pair of users as follows: when a pair has exchanged a friendship, then the distance is simply 1, otherwise the distance is the minimum count of hops between the pair by following friendship or group connections. Hence, weak connections (available to users belonging to the same group) provide means for information to rapidly flow across users belonging to portions of the online social network that have no direct friendship relationship. I.e., weak connections act as *bridges* between users having no friendship, by allowing their *distance* to become equal to 1. Therefore, even if group subscriptions are typically seen as weaker connections compered with the friendship ones, in our model we preferred to use only group memberships rather than a notion of distance. In fact, in such human-behaviour based contexts such as social networks, group memberships would be far more effective for providing a meaningful indication of the relation occurring among users rather than a simple distance measure.

From the friend list of each subscriber we identify *clusters* of users. Clusters consist of users having a higher number of friendships toward users within the same cluster rather than toward users not belonging to the cluster. As for the user distances, we define distance between a pair of clusters as the minimum count of hops between one user on the first cluster and one in the second cluster. Distant clusters can be considered as independent parts of the online social network that still satisfy the scale-free properties. Clusters generally consist of users sharing a set of interests and activities, and users of the same cluster form a sort of social neighbourhood [8].

Let us suppose that two users belong to different clusters, while being on the same group. When considering the relationship of users and groups, we can see that a group acts as a bridge for the contents to flow from a cluster to another (the clusters of the correspondent users). Hence, different parts of the network become mutually capable of exchanging contents, fostering the small-world behaviour of the social network [9]. In this way, clusters of users, representing different parts of the online social network, communicate by using weak connections rather than strong ones.

Thanks to the said properties of groups we can focus our analysis on a partition of the online social network (where a partition is one or several clusters of users), without loosing consistence and pertinence with the entire online social network.

2.2. *Existing online social networks*

The main difference between a formal scale-free graph and an online social network is given by the *percolation of links* [10], i.e. in real life, how worth a certain friend is tends to decrease if there is no good reason to maintain the relationship. This decrease of interest is still true even in an online social network, however it has no corresponding support in practice. From this absence there is a difficulty on accurately classifying links among subscribers when performing an automatic analysis. Moreover, in an online social network user features change steadily, thus it is difficult to determine the correlation between a user and his/her specific field of interests. Generally, for social networks that let users participate in a group, an average subscriber tends to sign into a large number of groups, while only a small amount of such groups are really interesting for the user.

The said wide-spread user behaviour would be difficult to generalise using traditional models and computational approaches, which are not noise robust. In turn, automatic selections and suggestions of posts provided by friends or groups become less useful, because of such inaccuracies. Even though the user profile can be potentially genuine, differently from online social networks, human relationships evolve following a homophily law [11], leading a person to connect with others having similar 'real' interests. Hence, the homophily law lets us detect and reason with small, though relevant, differences between social networks and theoretical scale-free networks.

Because of such differences, an existing online social network cannot adhere to a simple mathematical model, instead, since the stochastic behaviour typical of human beings is exhibited, an advanced nonlinear model is needed.

Due to the said untrustworthy, erratic, inconstant and unreliable user behaviour, we maintain that it is paramount to uncover hidden or un-explicit interests, therefore giving a representation of the effective relationships among users. Such (hidden) relationships are significant to find *categories* of users exhibiting some common traits. Such an identified category would unveil features that cannot be directly detected from the user profile.

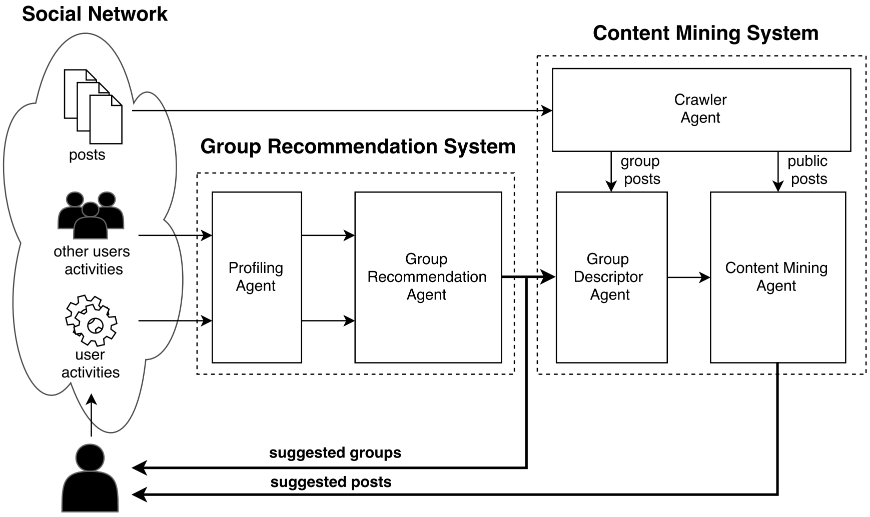

Fig. 1. Data flowing through the agents of the proposed system

3. The multi-agent system

The aim of the proposed multi-agent system is to provide an online social network with a practical and effective tool that infers real user interests, in order to suggest group subscriptions or relevant public posts to read. This in turn will enhance the user experience.

The individual and unpredictable behaviour of each single user makes the online social network a distributed and constantly changing environment. To cope with the intrinsic complexity of such an environment, a high-level modelling solution is needed. According to our model, a social network is characterised by many independent users, interacting with each other by means of e.g. different groups. Since multi-agent systems are known as an appropriate modelling solution for capturing flexible and autonomous behaviours, we associate each user and group to a separate and independent agent. Each of these agents autonomously gathers newly inserted data from the user profile and the group contents. Then, such data will be provided to other more complex agents that will manage respectively the learning system and the content mining algorithms.

Figure 1 shows the agents for our designed system that suggests new groups or posts for a specific social network user. To give such suggestions we analyse the type of contents or groups similar people are following. As shown in the said figure, the agents can be grouped in two different subsystems, which are the *Group Recommendation System* and the *Content Mining System*: the former will suggest new groups to the user starting from user activities inside the social network; the latter will use these group suggestions to mine the social network content to suggest new public posts to the user.

Firstly, a *Profiling Agent* (one for each user) autonomously and periodically gathers user-related data from his/her social network profile and activities, other than the list of current group subscriptions, which actually gives important information about a user interests. Of course, given the huge amount of data, we can select users according to some criteria (i.e. the value of some parameter on the profile, a combination of conditions on the actual state, or a list of ids, etc.), hence after a sequence of preprocessing tasks, the Profiling Agent will build a complete *interest profile* for the considered users, and complementing the descriptive features to the current group subscriptions. Thanks to our multi-agent approach we reduce the load on the social network servers by moving the Profiling Agent on the client side, and achieve a scalable solution that monitors and extracts relevant user activities when they are actually needed.

Gathered data are then given to the *Group Recommendation Agent* whose objective is to find a group that fits the selected user interests. This is actually a classification problem, where we will use the social network groups as a classification category. For this reason, from now on, we will often refer to a group as a classification category, or simply category, without losing precision on this sense. The classification concern is taken further by a RBPNN classifier, that is inside the Group Recommendation Agent. This neural network assigns the selected user profile to actual social network groups, according to the statistical model built according to group subscriptions made by users, and that the RBPNN learns during the previous training phases. Due to the intrinsic dynamics that the social network imposes, this model has to be constantly and incrementally updated.

The classification results, i.e. the group suggested by the RBPNN, can be directly sent to the selected user. Moreover, we use this information to suggest contents that have been inserted in to the online social network, e.g. posts that are related to the same topics discussed by the group, however belonging to other public sources. Starting from the group spotted by the Group Recommendation Agent, we can build a *"fingerprint"* for each group, according to the actual contents of the posts published in it.

A team of *Crawler Agents* gather the user textual posts from the social network, and then provide the contents published inside the suggested groups together with a list of all the other public contents that we may want to recommend to the selected social network users, according to their interests that have been implicitly unveiled by the Group Recommendation Agent. We decided to have more independent Crawler Agents running on server side in order to have a practical model to distribute the load on different server when actually needed. This is an important concern to consider knowing the great amount of contents that a real Social Network may have.

For each of the groups suggested in the previous stage, a *Group Descriptor Agent* will use the textual contents of the group to build it a descriptive profile (see Section 5 for more details). Using these profiles a *Content Mining Agent* will search inside the public posts provided by the Crawler Agents for the ones that better cover the topics discussed inside the recommended groups, using a textual analysis approach that will be shown in Section 5. At the end of this mining process, this agent will provide the selected posts to the user as a suggestion for further readings.

4. Proposed RBPNN based Classification Agent

Classical models suffer of the incompleteness of the initial input dataset. On the contrary, neural networks have been largely used to uncover data classification and find probabilistic categories for data. Therefore, we use Radial Basis Probabilistic Neural Networks (RBPNN), managed by an independent agent, to automatically find *categories* of users, where a category reveals common traits for users. Note that *group* of social networks, such as Facebook, can be seen as categories, which the RBPNN finds.

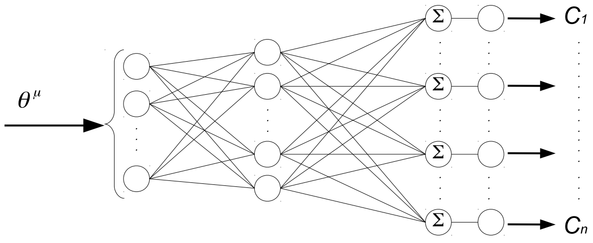

Fig. 2. A representation of a Radial Basis Probabilistic Neural Network

RBPNNs have a topology similar to that of ordinary FeedForward Neural Networks (FFNN) with Back-Propagation Training Algorithms (BPTA): the primary difference only lies in the activation function that, instead of being a sigmoid function or a similar activation function, is a radial basis function.

Our neural network, after being correctly trained, generates a model for the latent user features, and finds users having such features. This is usually considered both an interesting and difficult task [12]. However, the activation functions used for RBPNNs have to meet some important properties required to preserve generalisation abilities and the decision boundaries of Probabilistic Neural Networks (PNN) [13].

The selected RBPNN architecture (Figure 2) takes advantage from both PNN topology and Radial Basis Neural Networks (RBNN) used in [14].

In a RBPNN both the input and the first hidden layer exactly match the PNN architecture: the input neurones are used as distribution units that supply the same input values to all the neurones in the first hidden layer that, for historical reasons, are called *pattern units*. In a PNN, each hidden layer neuron performs the dot product of the input vector $\mathbf{u}$ by a weight vector $\mathbf{W}^{(0)}$, and then performs a nonlinear operation on the result. This nonlinear operation gives output $\mathbf{x}^{(1)}$ that is provided to the following summation layer.

While a common sigmoid function is used for a standard FFNN with BPTA, in a PNN the activation function is an exponential, such that, for the j-esime neurone the output is

$$\mathbf{x}_j^{(1)} \propto \exp\left(\frac{||\mathbf{W}^{(0)} \cdot \mathbf{u}||}{2\sigma^2}\right) \tag{1}$$

where σ represents the statistical distribution spread.

The given activation function can be modified or substituted while the condition of Parzen (window function) is still satisfied. In this case, while preserving the PNN topology, to obtain the RBPNN capabilities, the activation function is a radial basis function (RBF); an RBF still verifies all the conditions stated before. It then follows the equivalence between the $\mathbf{W}^{(0)}$ vector of weights and the centroids vector of a radial basis neural network, which, in this case, are computed as the statistical centroids of all the input sets given to the network. We name f the chosen RBF, so the output of the first hidden layer for the j-esime neuron is

$$\mathbf{x}_j^{(1)} \triangleq f\left(\frac{||\mathbf{u} - \mathbf{W}^{(0)}||}{\beta}\right) \tag{2}$$

where β is a parameter that is intended to control the distribution shape, quite similar to the σ used in (1).

The second hidden layer in a RBPNN is identical to that of a PNN, it just computes weighted sums of the values received from the preceding neurons. This second hidden layer is called, indeed, summation layer: the output of the k-esime summation unit is

$$\mathbf{x}_k^{(2)} = \sum_j \mathbf{W}_{jk}\mathbf{x}_j^{(1)} \tag{3}$$

where $\mathbf{W}_{jk}$ represents the weight matrix. Such weight matrix consists of a weight value for each connection from the j-esime pattern units to the k-esime summation unit. These summation units work as in the neurones of a linear perceptron network. The training for the output layer is performed as in a classic RBNN, however since the number of summation units is very small and in general remarkably less than in usual RBNNs, training becomes simplified and speed greatly increased [15].

The devised topology enables us to distribute different parts of the classification task to different layers (see Figure 3). While the pattern layer is just a nonlinear processing layer, the summation layer selectively sums the output of the first hidden layer.

The first hidden layer of the RBPNN is responsible to perform the fundamental task expected from a neural network, i.e. generalise and build an implicit model [16]. The second hidden layer selectively sums the output of the first hidden layer. The output layer fulfils the nonlinear mapping, such as classification, approximation and prediction.

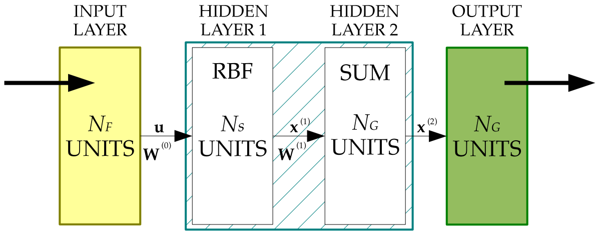

Fig. 3. RBPNN setup values: N_F is the number of considered features, N_S number of analysed subscribers, and N_G desired number of categories.

In order to have a proper classification of the input dataset, i.e. users in categories, the size of the input layer matches the number N_F of *features*, i.e. the labelled elements of the dataset (see Section 6), given to the RBPNN, whereas the size of the RBF units matches the number of examined subscribers N_S. The number of units in the second hidden layer is equal to the number of output units, this matches the number of categories N_G to be found for the subscribers.

5. The content mining system

In Section 4 we have shown how to suggest new group subscriptions starting from the user features, concerning their interests, behaviour and characteristics. This group suggestion is achieved by using an RBPNN classification agent that associates the most suitable category (a social network discussion group) to the user profile. This classification process leverages the user profile affinity to suggest new groups: it will suggest a new group to a user if his/her profile is actually compatible with the ones of other subscribed users. Hence, a group becomes a user category, mainly based on user interests, and regardless of the contents that are published inside the suggested group.

By considering that a group is not merely a set of users, instead it is a repository of published contents typically focused on a few discussion topics, we can accordingly suggest a group compatible with a user interests, then we are discovering the topics that a target user can find of interest. It is then paramount to find the topics that are discussed inside groups. Such results allow us to extend the proposed group recommendation system in order to search inside the social network contents for the public posts and news that seem to be related to the same involved topics. In this way, we are able to leverage a group recommendation system, which is based on users interest compatibility, to design a content mining system for public elements that will probably meet user interests.

As a configurable feature, building on the said analysis, we can limit suggestions to the contents of a certain group, or extend the mining to the public contents published in different groups, pages or public user profiles. In the former case, we can start with the automatic classification system to find the appropriate group first.

There are situations in which it is more desirable to have content suggestions, instead of group suggestions. Even though the social network provides some privacy settings to allow a user hiding his/her group subscriptions, this information is commonly public, or anyway visible to a subset of users that we trust (i.e. our social network friends). Anyway, knowing the public implication that a group subscription can have, some user can be tempted to refuse a group suggestion, even if appropriate, hindering the possibility to enhance their user experience on the social network. To overcome this limitation, it is then appropriate to have a more discrete selection of public contents that the user will probably like. Still, internally, the system has been leveraging the RBPNN group classification approach.

Since a group of users is generally based on a common ground of interests, it is possible to characterise such shared topics by means of a lexical analysis of the contents published on the group itself. The first step in the content mining process will be building a *descriptor* for each of the groups suggested by the classification agent, and then using it to define a scoring function for the public posts gathered from the social network, mining and suggesting the ones having the highest score.

By analysing the post publicly available on Facebook, the proposed textual analysis approach aims at inferring a semantical tag for each post, therefore enabling us to understand whether some different groups could host such a textual content.

5.1. Group profiling model

Let V be a vocabulary of words, and S a set of stop words, which are excluded from our consideration (e.g. because too common or not semantically rich). We will then consider each word x in a reduced subset V^* representing the vocabulary without the selected stop words, so that

$$x \in V^* = V \smallsetminus S \tag{4}$$

We are interested in analysing the text, deprived of stop words, contained in the posts of a group. In our analysis, for the i-th post of the g-th group we count the number of repetitions c_x^i for a certain word x. It follows that the set P_i^g of the pairs (x, c_x^i) is a subset of $V^* \times \mathbb{N}_0$. Such a set P_i^g is a projection of the textual content in a certain post within a Facebook group. It is then possible to obtain a significative measure of the semantic characteristics of a group by means of a function which expresses the number of recurrences of a certain word on the entire set of posts of a group. For this reason we defined a function $\phi : V^* \to \mathbb{N}_0$ so that

$$\phi(x) = \sum_i c_x^i \tag{5}$$

By means of this function it is possible to find a set Ω^g representing the words used in a group. This set is partially and decreasingly ordered with respect to their total recurrences count as measured by the function ϕ as in (5):

$$\Omega^g = \left\{ (x_\alpha, \phi(x_\alpha)) : \phi(x_\alpha) > \phi(x_{\alpha+1}) \right\}_{\alpha=1}^{|V^*|} \tag{6}$$

It is noticeable that $|\Omega^g| = |V^*|$ since V^* is the reference vocabulary, and since we want to order the words in V^* according to their recurrence on the posts of the g-th group. Finally, we can obtain a semantic descriptor for the group by selecting a maximum number K of features, taken as the most recurrent words in the group, and then characterising such a group by means of a set

$$D^g = \left\{ x / \left(x_\alpha, \phi(x_\alpha)\right) \in \Omega^g|_{\alpha<K} \right\} \tag{7}$$

We are now able to associate a group with a semantical descriptor. The next step is to create the set of words in V^* used in a public post. I.e., when considering the j-th post publicly available on Facebook, we are interested on the set of $x \in P_j \subset V^*$. Starting from the j-th post represented as P_j we can compute a measure of semantical similarity σ_j^g, with respect to the content of the g-th group as characterised by its descriptor D^g, as

$$\sigma_j^g = |D^g \cap P_j|! \tag{8}$$

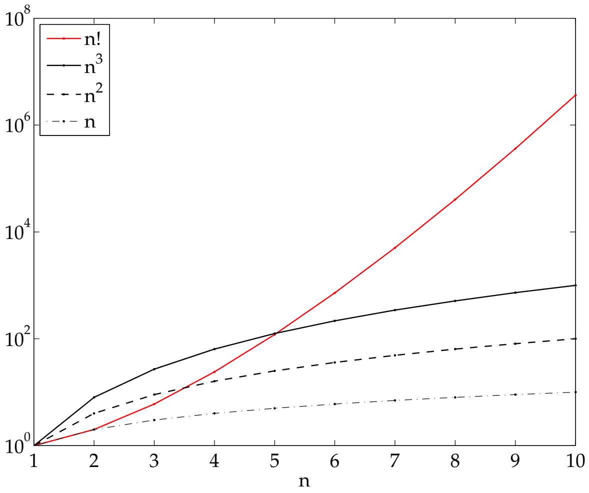

Fig. 4. Several possible ranking functions using (from top to bottom) factorial form, cubic form, squared form or linear form.

In equation (8) we used a factorial form due to its incremental ratio since it is a superlinear form that best suits our purposes with respect to other possible forms such as cubic or squared. The factorial form $n!$ is slowly incremental for low values of n while definitively boosting up for higher values of n (see Fig. 4), then it gives us a fair increasing rate for our semantical similarity measure.

The meaning of the scoring function for a post, defined as in equation (8) is as follows. We firstly count how many times a word of the group descriptor appears inside a post: the more a describing word appears inside the post, the more relevant the post becomes. Generally, a publicly available post contains a few lines of text, then once the stop words have been removed the data on which a similarity score is computed would be a few, since word repetitions will not be so frequent. Therefore, linearly relying on a recurrence approach would lead to low scores, and most importantly, this approach would not be resilient to noise, since we would give the same score to a post P_1 with a single descriptive word repeated n times and to a post P_2 having n different descriptive words appearing only once. Among the said two posts, post P_1 is the one offering a better coverage of the topics discussed inside the selected group, and for this reason we should associate it with a higher score. Therefore, we count the number of different descriptive words that appear inside the post, and then compute a factorial function, as a result the post that better covers the group topics will clearly emerge from the other ones.

5.2. *The content mining algorithm*

Basing on the group profiling model previously defined, we have implemented two different solutions to mine the social network contents for public posts that will meet user interests, according to the group recommendation system. Algorithm 1 builds a meaningful descriptor for a group, whereas Algorithm 2 uses this group descriptor to assign a score for an input post gathered from the pool of public contents available on the social network. As a result, we can suggest only the posts that obtained the highest score with respect to the selected groups, and therefore that are more likely to be among the user interests.

Algorithm 1 gets as input a group, with all its posts, a vocabulary and a set of stop words that we want to ignore during the textual analysis, because they are so frequent that they cannot be useful to characterise the group contents. The output of this algorithm is a descriptor for the input group, in terms of a list of words that frequently appear inside its posts and, for this reason, they are meaningful to describe the topics covered in the group. After the initialisation of the data structures required for the recurrence counting of each word inside the group, for each of its post the algorithm lists the inner text word by word: if the word is a stop word it is simply skipped, otherwise, the corresponding counter is increased. This recurrences counting is then ordered and the K words (where K is a configurable parameter) with the highest recurrence are chosen to define the group descriptor.

This descriptor is then given to the Algorithm 2, together with a single public post, in order to assign it a relevance score with respect to the group topics. This is achieved by retrieving from the post the list of different words it consists of, considering them only once, and then counting how many of such words appear inside the group descriptor. Let n be the number of words of the descriptor covered by the considered post: the returned score will be $n!$.

6. Experimental setup

Given the utmost importance of the classification component in the proposed multi-agent solution, we have deeply tested the performance of the conceived RBPNN classifier used by the Classification Agent. We used a dataset consisting of features, i.e. a trace of the user activities and their preferences, coming from real Facebook profiles. Data for the features that we

Algorithm 1 Group descriptor generator

```
Input: a group $G$, a vocabulary $V$, stop words $S$.
Output: a descriptor $D$ for the group G.
$R = (x, C[x]) \ \forall \, x \in V \setminus S$
$L$ = list of posts $P$ in $G$
for each $P \in L$ do
    $T$ = text of $P$
    for each $x \in T$ do
        if $x \notin S$
            $C[x]$ ++
        end if
    end for
end for
sort $R$ according to $C[x]$
$D = \{x\}$ from the first $K$ pairs of sorted $R$
return $D$
```

Algorithm 2 Scoring function

```
Input: a descriptor $D$, a public post $P$.
Output: a score $s$.
$W$ = empty list of words
$T$ = text of $P$
for each $x \in T$ do
    if $x \notin W$
        add $x$ to $W$
    end if
end for
$n = 0$
for each $x \in W$ do
    if $x \in D$
        $n$++
    end if
end for
$s = n!$
return $s$
```

have been given have a label which is a numerical ID, i.e. the feature itself can not be recognised, however this does not affect the scope of this work nor the analysis performed.

As far as the feature list is concerned, data provide boolean values. The presence or absence of a specific value is expressed as a boolean flag, e.g. 1 if the user has declared his job or 0 if no job information is given in the profile. Among such boolean values there are mutually exclusive values such as the gender, e.g. 1 if male or 0 if female.

The intrinsic structure of the dataset prevents us from considering only a reduced portion of the feature list for a user. A piece of information is usually largely spread over a certain number of features, e.g. a boolean variable could express if the gender is stated or not, and only if stated another variable could report if the user is male or female; then in case the profile does not state the gender, the latter feature has no meaning and should not be considered. However, since our dataset gives no labels, we can not exclude any feature.

Although data are anonymised, users are identified with a unique ID. Moreover, the memberships of users to groups has been identified from the list of subscribers to each group.

Data intended to be given as input for our RBPNN have been passed through a preprocessing filter that pairs each user feature to the list of group memberships. This gives to our statistically driven classifier the ability to correctly identify the relationships between user features and groups.

7. RBPNN findings

Both user profiles, consisting of features, and user memberships to groups were provided to our RBPNN classifier during the training phase. Therefore, the RBPNN classifier has learnt how to reproduce the correct paths that associate lists of profile features with groups.

Initially, we have asked our RBPNN to reconstruct the groups for 250 users. The RBPNN was able to correctly assign users to the proper groups with only a 5.67% of missing assignments: as a remarkable side effect while a few groups were not found, no false positive was given. Moreover, we have visually compared the features for such unclassified users and the average features of their groups, and found relevant differences with respect to the average (and correctly classified) user. Just for validation purposes, we have performed the same comparison for users with an almost empty profile, and noted that the RBPNN could not insert into any category, which is a highly desirable behaviour for the classifier. The results of our experiments have been summarised in Table 1, and in it we note the high success rate for the classification of users into categories (or groups), dubbed as 'correct answers'.

Additionally, we have used our RBPNN in order to identify categories for new users, who have not expressed any preference for them. For an appreciable percentage of users, i.e. about 20%, the proposed RBPNN has indicated a group that (unknown to the RBPNN) users had membership to (of course we had isolated data beforehand to perform a controlled exper-

Table 1

Number of analysed users and groups and the number of categories (answers) that have been suggested to unaware users by our RBPNN-based classifier

	Number of users	Number of groups	Number of memberships	Correct answers	Wrong answers	Success rate	Relative error
Groups Reconstruciton	250	32	476	449	27	95.33%	5.67%
Groups Suggestion	100	32	85	68	17	20%	80%

iment). Indeed, a relevant number of the other 80% of user profiles is (almost) empty, therefore no classifier, not only our RBPNN, would manage. Instead, expressing some suggestions in such cases would be similar to a random guess.

We finally note that it is not relevant to count how many features suffice for a user to build a classifier, because the model built as the RBPNN depends on the relationship between each features and groups, and features, though have been anonymised, are not equally relevant.

8. Related Works

Several generative models can be used to characterise datasets that determine properties and allow grouping data into *classes*. Generative models are based on stochastic block structures [17], on 'Infinite Hidden Relational Models' [18], etc. The main issue of class-based models is the type of relational structure that such solutions describe. Since the definition of a class is attribute-dependent, generally the reported models risk to replicate the existing classes for each new attribute added.

Such models would be unable to efficiently organise (inherit) similarities between (from) the classes 'cats' and 'dogs' as child classes of the more general class 'mammals'. Such attribute-dependent classes would have to be replicated as the classification generates two different classes of 'mammals': the class 'mammals as cats' and the class 'mammals as dogs'. Consequently, in order to distinguish between the different races of cats and dogs, it would be necessary to further multiply the 'mammals' class for each one of the identified race. As a consequence, such models quickly lead to an explosion of classes. In addition, we would either have to add another class to handle each specific use or a mixed membership model, as for crossbred species.

Another paradigm concerns the Non-Parametric Latent Feature Relational Model [19] i.e. a Bayesian non-parametric model in which each entity has boolean valued latent features that influence the model's relations. Such relations depend on well-known covariant sets, which are neither explicit or known in the case of a social network during the initial analysis.

With the recent growth of social networks usage, a keen interest for data analysis has been spawn, with the aim to perform sentiment analysis, suggest posts, etc. As far as privacy is concerned, in [20], authors describe the results of an extensive comparison between two important social networks such as Facebook and MySpace, showing that the interaction of trust and privacy concerns in social networking sites is not yet understood to a sufficient degree. Moreover, in [21], authors explore the preservation of privacy and propose a novel method to avoid *neighbourhood attacks*. The authors show that anonymised data can be used to answer aggregate queries accurately.

Other previous data analyses concerning user profiling have taken into account the category of words appearing in texts [22], as well as the user behaviour on-line.

9. Conclusion

We have proposed a multi-agent system for the automatic analysis of data on an online social network and have shown that interesting results can be obtained in terms of the knowledge on the user behaviour. The proposed solution is based on the elicitation of meaningful texts and classification tools.

In our approach an automatic analysis finds for a user the most similar likely social network group s/he could belong to. Once the above solution would be integrated with the servers handling user data, higher levels of precision can be reached for proposing contents to users.

Acknowledgements

This work has been partially supported by project PRISMA PON04a2 A/F funded by the Italian Ministry of University within PON 2007-2013 framework, and by project PRIME funded by the Italian Ministry of University and Research within POR FESR Sicilia 2007-2013 framework.

References

[1] C. Kiss, A. Scholz, and M. Bichler, “Evaluating centrality measures in large call graphs,” in *Proceedings of IEEE Enterprise Computing, E-Commerce, and E-Services*, 2006.

[2] G. Cybenko, “Approximation by superpositions of a sigmoidal function,” *Mathematics of Control, Signals and Systems*, vol. 2, no. 4, pp. 303–314, 1989.

[3] M. T. Hagan, H. B. Demuth, M. H. Beale, *et al.*, *Neural network design*. Pws Pub. Boston, 1996.

[4] R. Vaidyanathan and V. Venkatasubramanian, “Representing and diagnosing dynamic process data using neural networks,” *Engineering Applications of Artificial Intelligence*, vol. 5, no. 1, pp. 11–21, 1992.

[5] C. Jones and E. H. Volpe, “Organizational identification: Extending our understanding of social identities through social networks,” *Journal of Organizational Behavior*, vol. 32, no. 3, pp. 413–434, 2011.

[6] Y.-Y. Ahn, S. Han, H. Kwak, S. Moon, and H. Jeong, “Analysis of topological characteristics of huge online social networking services,” in *Proceedings of World Wide Web*, pp. 835–844, ACM, 2007.

[7] A.-L. Barabási, “Scale-free networks: a decade and beyond,” *Science*, vol. 325, no. 5939, pp. 412–413, 2009.

[8] M. Granovetter, “The Strength of Weak Ties,” *The American Journal of Sociology*, vol. 78, no. 6, pp. 1360–1380, 1973.

[9] S. Schnettler, “A structured overview of 50 years of small-world research,” *Social Networks*, vol. 31, pp. 165–178, July 2009.

[10] N. Schwartz, R. Cohen, D. ben Avraham, A.-L. Barabási, and S. Havlin, “Percolation in directed scale-free networks,” *Phys. Rev. E*, vol. 66, p. 015104, Jul 2002.

[11] M. McPherson, L. Smith-Lovin, and J. M. Cook, “Birds of a feather: Homophily in social networks,” *Annual Review of Sociology*, vol. 27, no. 1, pp. 415–444, 2001.

[12] D. Liben-Nowell and J. Kleinberg, “The link-prediction problem for social networks,” *Journal of the American society for information science and technology*, vol. 58, no. 7, pp. 1019–1031, 2007.

[13] S. O. Haykin, *Neural networks and learning machines (3rd Edition)*, vol. 3. Prentice Hall, 2009.

[14] F. Bonanno, G. Capizzi, G. Graditi, C. Napoli, and G. Tina, “A radial basis function neural network based approach for the electrical characteristics estimation of a photovoltaic module,” *Applied Energy*, vol. 97, pp. 956–961, 2012.

[15] H. Deshuang and M. Songde, “A new radial basis probabilistic neural network model,” in *Proceedings of Conference on Signal Processing*, vol. 2, IEEE, 1996.

[16] W. Zhao, D.-S. Huang, and L. Guo, “Optimizing radial basis probabilistic neural networks using recursive orthogonal least squares algorithms combined with micro-genetic algorithms,” in *Proceedings of Neural Networks*, vol. 3, IEEE, 2003.

[17] K. Nowicki and T. A. B. Snijders, “Estimation and prediction for stochastic blockstructures,” *Journal of the American Statistical Association*, vol. 96, no. 455, pp. 1077–1087, 2001.

[18] Z. Xu, V. Tresp, K. Yu, and H. peter Kriegel, “Infinite hidden relational models,” in *Proceedings of Uncertainity in Artificial Intelligence (UAI)*, 2006.

[19] L. Getoor, *Introduction to statistical relational learning*. MIT press, 2007.

[20] C. Dwyer, S. Hiltz, and K. Passerini, “Trust and privacy concern within social networking sites: A comparison of facebook and myspace,” in *Proceedings of Americas Conference on Information Systems*, pp. 339–351, 2007.

[21] B. Zhou and J. Pei, “Preserving privacy in social networks against neighborhood attacks,” in *Proceedings of Data Engineering*, IEEE, 2008.

[22] C. Napoli, G. Pappalardo, and E. Tramontana, “An agent-driven semantical identifier using radial basis neural networks and reinforcement learning,” in *Proceedings of XV Workshop "Dagli Oggetti agli Agenti"*, vol. 1260, CEUR-WS, 2014.