COMMUNITY DETECTION IN THE COLLABORATIVE WEB

Lylia Abrouk, David Gross-Amblard and Nadine Cullot

LE2I, UMR CNRS 5158
University of Burgundy, Dijon, France
lylia.abrouk@u-bourgogne.fr, david.gross-amblard@u-bourgogne.fr,
nadine.cullot@u-bourgogne.fr

ABSTRACT

Most of the existing social network systems require from their users an explicit statement of their friendship relations. In this paper we focus on implicit Web communities and present an approach to automatically detect them, based on user's resource manipulations. This approach is dynamic as user groups appear and evolve along with users interests over time. Moreover, new resources are dynamically labelled according to who is manipulating them. Our proposal relies on the fuzzy K-means clustering method and is assessed on large movie datasets.

KEYWORDS

Clustering, Data sharing, Information networks, user distance, Web community.

1. INTRODUCTION

In the last decade, the basic Internet turns into a generic exchange platform, where any user becomes a content provider by using spreading technologies like comments, blogs and wikis. This new collaborative Web (called Web 2.0) hosts successful sites like Myspace, Facebook or Flickr, that allow to build social networks based on professional relationship, interests, etc. This so-called Social Web describes how people socialize or interact with each other. They suppose from the user an explicit description of his/her social network.

However, a large amount of users communities also appear implicitly in various domains. For example, any popular Web site about music will gather users with various musical tastes and preferences: this also forms a huge community. But this coarse-grain community is in fact composed of different pertinent and potentially disjoint sub-communities, all related to music (for example the pop community, the punk community, and so on). Identifying precisely these implicit communities would benefit to various actors, including Web site owners, on-line advertisement agencies and above all, users of the system.

In this work, we propose an automatic community detection method that relies on the resources manipulated by users. The method is generic as it depends only on a simple user-defined tagging of resources. The method is also dynamic: communities evolve over time as users change their resources annotations.

Finally, we also take into account the automatic tagging of a new resource, by analyzing how it is used by communities. A building block of our method is the unsupervised classification algorithm fuzzy-K-means [9].

DOI : 10.5121/ijmit.2010.2401

The rest of the paper is organized as follows: Section 2 introduces our approach for automatic community detection. This approach was implemented and tested on the movieLens data set, as shown is Section 3. The related work is presented in Section 4. Finally, conclusion and perspectives are presented in Section 5.

2. OUR APPROACH

Our method, in order to apply to a wide set of situations, in based on few hypothesis. We consider a set *P* of users (persons) and a set *R* of resources (for example music files, videos, news, etc.). First, we suppose that users express votes on some resources. This vote is not necessarily explicit and can be obtained by monitoring user's behavior (what item is listened, or bought, or annotated, or recommended, etc.)

Votes are illustrated in a matrix MP: $|P|x|R|$ defined as follows:

$$MP(p_i, r_j) = \begin{cases} 1 \text{ if } p_i \text{ likes } r_j: \\ 0 \text{ otherwise.} \end{cases} \qquad (1)$$

Where $p_i \in P$ and $r_j \in R$. Second, we suppose to have a finite set *T* of tags (like pop, rock, punk, etc.), and that each resource is annotated with a subset of these tags (potentially empty). We define $L(p_i,t_k)$ as a subset of *R*, where $p_i \in P$ and $t_k \in T$, the set of resources having tag t_k liked by user $p_{i.}$

The main goals of this work are (1) to automatically detect communities and (2) to automatically determine tags of new items. A community gathers persons having the same interests, in the sense that they like resources that are tagged almost the same way. Our approach deals with three key concepts that are presented below:

- *Users distance*: once a user has voted (implicitly) for resources, we define a user distance that represents similarity of users interests.
- *Community clustering*: based on users'similarity, we construct users communities. Each user belongs to one or several communities.
- *Tags detection: each new resource is tagged automatically.*

2.1. Users distance

Several works on collaborative recommendation systems and communities detection are based on a similarity distance. Our distance measure is based on the number of tags users have in common. Two users are considered closer if they appreciate the same resources, based on their tags. The distance between two users p_i and p_j is defined by:

$$d(p_i, p_j) = 1 - \frac{1}{|R|} \sum_k \frac{|L(p_i, t_k) \cap L(p_j, t_k)|}{|L(p_i, t_k) \cup L(p_j, t_k)|} \qquad (2)$$

Distance closer to zero represents closer user friendship. Based on this measure, we can construct users distance graph G_d:

$$G_d = <P, P\,X\,P\,X\,[0,1]> \qquad (3)$$

This graph is complete, undirected and each of its edges (p_i, p_j) is weighted by the similarity distance between p_i and $p_{j.}$

2.2. Community clustering

Different classification techniques aiming at building users clusters can be envisioned. There are two possible approaches for this classification: the supervised and the unsupervised one. The first approach requires initially classified users to classify a new user. Among possible algorithms used in this kind of approach, we can retain the k-nearest neighbors algorithm (k-NN) [8] based on closest training examples. But this supervised approach appears a little constraining because of the required manual construction of the social network. Indeed, it imposes on users to create initially their profiles and to invite friends. The friends community will then grow progressively.

Because of this strong constraint, we preferred the second type of classification - unsupervised classification - that allows for automatic classification (that is, does not require training examples).

Our goal is both to gather similar users (having the same interests) in the same community (class) and to increase the distance between theses communities (classes). From time to time, users votes can completely change, for example in musical items from Rock'n'roll to classical music. Moreover, a person's interests may be composed of different tags. For this reason, we chose an algorithm that allows a user to belong to several clusters simultaneously communities. We chose a fuzzy extension of the K-means algorithm: fuzzy K-means [9], [10].

In this method, the number K of awaited classes has to be defined (we discuss the choice of K later on). The method is based on the minimization of the following function (4), with *K* being the number of clusters and *N* the number of persons, P={ p_i | i ∈ [1..n]}, and *m* is a predefined constant (generally *m*=2):

$$J = \sum_{i=1}^{N} \sum_{j=1}^{K} u_{ij}^{m} \| p_i - c_j \| \tag{4}$$

with the constraint:

$$\sum_{j=1}^{k} u_{ij}^{m} = 1 \tag{5}$$

Coefficient u_{ij} ∈ [0,1] is the membership degree of person p_i in cluster *j*, and c_j is the center of cluster *j*.

The different steps of the fuzzy K-means algorithm are: (i) initialize matrix U=[u_{ij}], (ii) at step k: compute centers C*k*=[c_j] (Equation 6), (iii) update the membership degree and (iv), if $||U(k+1)\text{-}Uk||<\varepsilon$ then stop else return to (ii).

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} p_i}{\sum_{i=1}^{N} u_{ij}^{m}} \tag{6}$$

$$u_{ij} = \frac{1}{\sum_{k=1}^{K} \left(\frac{\| p_i - c_j \|}{\| p_i - c_k \|} \right)^{\frac{2}{m-1}}} \tag{7}$$

The choice of the value of K can be done by the Web site owner, according to the desired granularity. A natural option would be to choose K close to |*T*|, the size of the set of tags, if these tags are supposed to provide a rich enough description vocabulary. If pairs of tags are required for a convenient description, hence a value of K close to $|T|^2$ should be chosen, and so on. For the sake of simplicity in the sequel we fixed $K=|T|$. In is noteworthy that semantically related tags (synonyms) should be gathered in a precomputation phase. Otherwise, close users could be shattered in different clusters. The algorithm result is a matrix *MGp*: $|P|x|T|$ where each element of *MGp(i,j)* is the degree of membership of p_i in cluster j. Using this matrix, it is now possible, for example, to invite a user to meet new friends in the same community, starting with the closest user, according to the similarity distance. It is also possible to locate the closest user to the center of the cluster: this user is representative of the whole community (a so-called *trendsetter*). He/she can be the target of special attentions (access rights promotion on the Web site forums, special offers, advertisements).

2.3. Tagging new resources

The previous clustering method can be invoked from time to time, and communities can be updated according to the current user's votes. This yields the dynamic flavor of the approach. Another aspect of dynamicity is the problem of tagging new resources uploaded on the Web site (by the Web site owner or by users). In this section we will tag a new resource according to the users who like this resource. Thus, we also attach to users their representative tags. We start by calculating user tags membership $m(p_i,t_k)$, based on users votes:

$$m(p_i, t_k) = \frac{|L(p_i, t_k)|}{|L(p_i)|} \qquad (8)$$

When a new resource r_j appears, we update the users votes matrix $MP(p_i, r_j)$ for each user p_i. Then, tags of the new resource are defined with regard to user's votes. We calculate the tag resource membership $v(r_j, t_k)$ that represents the membership of the new resource $r_j \in R$ to the tag $t_k \in T$ (it may be seen as probability that tag t_k represents resource r_j). Hence, for each $p_i \in P$ where $r_j \in L(p_i)$, we define:

$$v(r_j, t_k) = \frac{1}{|P|} \sum_{i, r_j \in L(p_i)} m(p_i, t_k), \qquad (9)$$

$$\sum_k v(r_j, t_k) = 1$$

Finally, among of all the potential tags for the resource, we select those tags that are representative, using Receiver Operating Characteristics (ROC) curves. ROC curves are used to evaluate classifiers: they provide information on the trade-off between the hit rate and the false hit rate. In our context, it tests the system validity and finds pertinent threshold for each potential tag.

Each item is represented by a vector, with items tags membership. ROC curve determines the sensibility according to 1 - SP for different thresholds. Based on training test, we calculate sensitivity (SE) and specificity (SP). Let:

- *th* be the threshold for tag $t_{i,}$
- *Rtp* be the set of items having tags t_i (annotated by expert) and having $v(r_j, t_k) >= th$, (true positive),

Rfp be the set of items having tags t_i (annotated by expert) and having $v(r_j, t_k) <= th$, (false positive),

- *Rtn* be the set of items which don't have tags t_i (annotated by expert) and have $v(r_j, t_k) <= th$, (true negative),

- *Rfn* be the set of items which don't have tags t_i (annotated by expert) such that v(r$_j$, t_k) >= *th*, (false negative).

Then

$$SE = \frac{|R_{tp}|}{|R_{tp} + R_{fn}|} \qquad (10)$$

$$SP = \frac{|R_{tn}|}{|R_{tn} + R_{fp}|} \qquad (11)$$

The best threshold is point in the upper left corner (coordinate (1-SP=0, SE=1)) of the ROC space, representing 100% sensitivity (no false negatives) and 100% specificity (no false positives). This approach is illustrated on the next section.

3. EXEPERIMENTS

Due to lack of space, we focus in this section on the new item tags detection algorithm. We tested our method on the MovieLens (ML)1 data set containing 100,000 ratings over 1,682 movies provided by 943 users. This data set contains movies rated with a numerical scale (1 to 5). Then, we transformed the value of this rating into a binary vote (where ratings greater than two become "like" and otherwise "don't like"). Then (1) we computed users distance matrix to construct users communities. In order to detect new item tags, (2) we compute the user tags membership based on the 843 first items. (3) We tested our approach on the 100 last movies, playing the role of new items. We present the results on six movie tags (1: comedy, 2: action, 3: crime, 4:drama, 5:romance, 6: thriller).

3.1. Tags detection

For the 100 new resources, we calculate for each tag the value of v(r_j, t_k) which represents the membership of the new resource r_j to the genre t_k .In order to detect tag, we calculate the threshold *th* for each tag t_i using ROC curves.

Table 1 represents Sensitivity (SE) and specificity (SP) for different threshold of "comedy" tag. The optimal threshold is 0,14. It is the point in the upper left corner in the ROC curve.

Table 1. Comedy tag ROC table.

Threshold	1-Sp	SE
0.08	0.957	1
0.1	0.903	0.968
0.12	0.772	0.937
0.14	**0.337**	**0.656**
0.16	0.196	0.343
0.18	0.087	0.06
0.2	0.044	0.06
0.22	0.033	0
0.24	0.011	0

[1] http://www.grouplens.org/node/73

Figure 1 represents ROC curves for "comedy" and "action" tags.

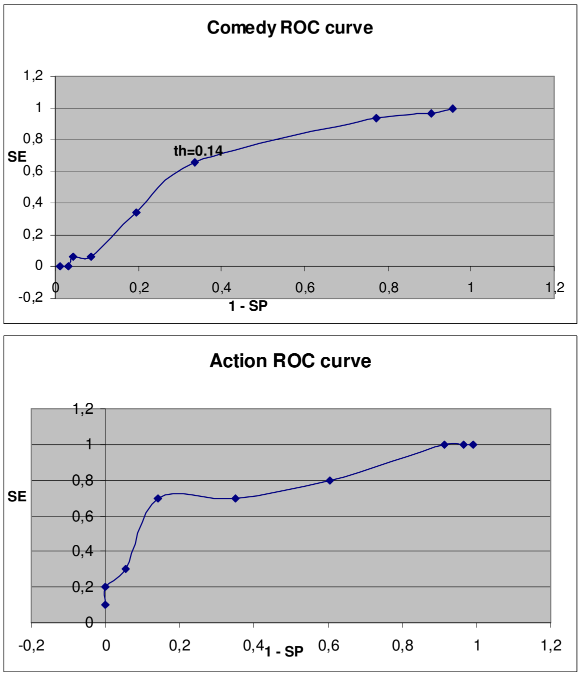

Fig 1: tags roc curve.

We compute a ROC curve for each tag (Table 2). We can observe that the thresholds that determine resource tags are generally between 0.1 and 0.2. The "Drama" tag is upper because half of the resources have this specific tag.

Table 2: Threshold tags.

Tag	Threshold
Comedy	0.14
Action	0.12
Crime	0.05
Drama	0.25
Romance	0.10
Thriller	0.10

3.2. Correlation between proposed and existing tags

Once our thresholds are calculated, we assess the correlation between our proposal and real data set tags. We calculate the linear correlation coefficient *r* where x_i represents the original data set item tags membership (1 or 0) and y_i represents our approach item tags membership.

$$r = \frac{n\sum x_i y_i - \sum x_i y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\,\sqrt{n\sum y_i^2 - (\sum y_i)^2}} \qquad (12)$$

Table 3: Correlation degrees.

Tag	Correlation
Comedy	0.19
Action	0.34
Crime	0.25
Drama	0.25
Romance	0.17
Thriller	0.23

The correlation coefficient defines linear dependencies between x and y.
We use the Fisher test for significant of correlation (table 4). Correlation is significant for result upper than 4. Significant result represents probability to have a false result less than 0.05.

Table 4: Fisher test result.

Tag	Fisher	Significant
Comedy	4.17	significant
Action	15.67	Very significant
Crime	7.98	significant
Drama	7.95	significant
Romance	3.49	No significant
Thriller	6.87	significant

"Romance" tag is on the limit of the significativity. This is explain by the association with other tags, it is appear with “comedy” or “drama” tags.

4. RELATED WORKS

Several works are devoted to community emergence. In this context, recommendation systems like Amazon [1] handle communities implicitly, recommending items to users based on the similarity between their interests.

Web sites generated by users are the cornerstone of Web 2.0 or collaborative Web: the goal of this new Web is to transform users into contributors. Users not only add contents, but also opinions and personal information. Another main aspect of this new Web is its social networks (social relationships) which connect friends, even geographically distant. Social networks are the grouping of individuals into specific communities. They make possible to look for comrades or family members, but also to discover new friends, generally by affinities. We distinguish two types of social networks: virtual network and social network online. The first one consists in the discovery of new friends; the second one is a meeting place for existing friends. There is a large number of social networking websites that focus on particular interests. For example, SixDegrees.com[2] was a social network which allowed users to expand their network based on user’s profiles, and permitted to target a user community for specific services (music, advertisement, etc.).

[2] From 1997 to 2001.

The Myspace[3] social network allows artists to upload their music and to create relations between network members.

4.1. Recommendation system

Based on user's behaviors, recommendation systems propose to the user a set of pertinent playlist according to his profile. We can distinguish two methods: (i) collaborative methods creating community of users with similar interests and recommend music listened by the same community. (ii) Content based methods.

Liu and al. [2] take into account the changes of user's interest in time by adding time parameter in order to improve the recommendation. The algorithm generates a decision tree to represents user's votes. The method is divided into three main steps: (1) users give to the system personal information, (2) the system constructs users communities and initial music lists, (3) formalized recommendations are generated by the system, using decision tree classification, in order to recommend music to the user at the current time.

This method solves the cold-start problem for new users but, in practice, this method is not completely satisfying because few users give all personal information like music genre.

Celma and al. [3] use the FOAF standard description and content based description to recommend music resources. The Friend of a Friend (FOAF) project[4] consists in creating a Web of machine-readable pages describing people in order to connect social Web sites.

The music recommendation system extracts users' interests from a FOAF profile, detects artists by relationships and finally selects similar artists by relevance.

Firan and al. [4] propose a recommendation algorithm based on user profiles (tags). The tag usage is analyzed on last.fm[5] music site. Authors define three types of algorithms: (i) collaborative filtering based on tracks where users rank tracks; the cold start problem appears in this algorithm type, (ii) collaborative filtering based on tags and search based on tags.

An hybrid method (collaborative aspect and content) proposed by Yoshi and al. [5] uses a probabilistic model to integrate rating and content data using a Bayesian network to perform classical methods.

4.2. Emergent community

Cattuto and al [6] present an approach experimented on del.icio.us[6] web site data where community structure exists in tagging data collection to construct weighted networks of resources. In this context the resources similarity is represented by the overlap of tag sets. To take into account tag frequency, the TF-IDF weight is used. In [6], authors propose to detect virtual communities of users with similar music interests in order to create a music channel for the community. They use Pearson correlation coefficient to define the similarity measure. Clustering methods are used and estimated.

Several techniques are applied in the collaborative Web to create users communities and recommendations systems. Measures are generally based on user profiles. Current methods take into account user needs and consider that the new resources contain metadata (tags, type, …).

[3] http://www.myspace.com/

[4] http://www.foaf-project.org/

[5] www.last.fm

[6] http://delicious.com/

5. CONCLUSION

In this work we proposed a dynamic method for automatic community detection and automatic tagging of new resources. This work is currently under integration into the NEUMA platform[7], an open system for communities manipulating music in the symbolic format (like MusicXML). Our future work will extend the notion of simple tags with more sophisticated ontology on the one side [11], and will take into account the dynamic of communities over time on the other side.

ACKNOWLEDGEMENTS

This work is partially supported by the French ANR fund Content & Interactions -- NEUMA (2009-2011).

REFERENCES

[1] Linden G and Smith B and York J “Amazon.com recommendations: item-to-item collaborative filtering”, Internet Computing, IEEE, Vol. 7, No. 1. pp. 76-80., 2003.

[2] Ning-Han Liu, Szu-Wei Lai, Chien-Yi Chen, Shu-Ju Hsieh,Adaptive “Music Recommendation Based on User Behavior in Time Slot”, Vol. 9 No. 2 pp. 219-227.

[3] Celma, O. and Ramairez, M. and Herrera, P.
“Getting music recommendations and filtering newsfeeds from FOAF descriptions”, 1st Workshop on Scripting for the Semantic Web co-located with the 2nd European Semantic Web Conference, 2005.

[4] Firan, C S and Nejdl, W and Paiu, R. “The Benefit of Using Tag-Based Profile”, Proceedings of the 2007 Latin American Web Conference (LA-WEB), 2007.

[5] Yoshii, K and Goto, M and Komatani, K and Ogata, T and Okuno, H, G. “Hybrid collaborative and content-based music recommendation using probabilistic model with latent user preferences”, Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR), 2006.

[6] Cattuto, C and Baldassarri, A and Servedio, D,P and Loreto, V. “Emergent Community Structure In Social Tagging Systems”. Advances in Complex Systems (ACS), pp 597-608, 2008.

[7] Anglade, A and Tiemann, M and Vignoli, F. “Virtual communities for creating shared music channels”', Proceedings of ISMIR, pages 95-100, 2007.

[8] Dasarathy, B. V. “Nearest Neighbor (NN) Norms--NN Pattern Classification Techniques”, Los Alamitos, CA: IEEE Computer Society Press. 1991.

[9] Dun, J. C. “A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters”, Journal of Cybernetics 3: 32-57.

[10] Bezdek, J. C. “Pattern Recognition with Fuzzy Objective Function Algoritms”, Plenum Press, New York.

[11] Abrouk, L and Gouaich, A “Automatic Annotation Using Citation Links and Co-citation Measure: Application to the Water Information System”, Proceedings of ASWC06, 2006.

[7] http://neuma.irpmf-cnrs.fr