A Particle Swarm Optimization Based Approach for Finding Similar Users on Facebook

Pratik S. Kabara*, Vishal Kaushal, Akshay Divekar, Mohit Kamdar, Devvrat Ganeriwal
Department of Computer Engineering, Vishwakarma Institute of Technology Pune, Maharashtra, India.

* Corresponding author. Tel.: +919422503775; email: kabrapratik28@gmail.com
Manuscript submitted April 19, 2015; accepted July 27, 2015.
doi: 10.17706/jcp.11.1.18-25

Abstract: With ever increasing size and use of social networks, there is a large amount of information which users implicitly or explicitly leave behind on social media. This information can be used to identify their personal traits and preferences. In this work, we have proposed a Particle Swarm Optimization (PSO) based approach for clustering users on Facebook (FB) data to identify similar users. Proposed method can be used to recommend people having similar interests. Our results indicate a lesser quantization error than k-means when used for 7 or less clusters. Possible applications of this approach include rope-in exercises for new hires, assigning a resource person to work team, etc.

Key words: Clustering method, facebook, particle swarm optimization (PSO), recommender systems, social network services.

1. Introduction

Online social networks like Facebook, My Space or LinkedIn are rapidly emerging as the most popular services on the Web. These systems are attracting a significant portion of Web users: for instance, in the February 2015, Facebook had 1.39 billion monthly active users [1]. Social networking sites (SNS) contain a lot of data left behind by the users either implicitly or explicitly in the form of published interests, text messages, likes, comments, and affiliations. Facebook, for example, allows users to publish online profiles describing both demographic data (e.g., place and date of birth) as well as interests.

The vast amount of data available in online social networking profiles of users has attracted researchers to mine them for valuable insights. For example, in their work, Mislove, Alan, *et al.* [2] has done analysis of online social networking data to show how structure of social networking is different from structure of the web. Thompson, Lindsay A., *et al.* [3] in their work studied social network data in context of medical professionalism. Similarly, Palmer *et al.* [4] used social network for direct marketing.

More recently, work has begun in mining online social networking profiles to identify the personal traits, preferences and interests of a user. This has several interesting applications like predicting job satisfaction, developing personalized user interfaces, facilitating rope-in exercises for new hires in a company or new students in a college. In the myPersonality project [5] Kosinski, Stillwell *et al*. have used Facebook data to identify personality traits of users.

Identifying traits and preferences of users allow us to evaluate similarities or differences among them. Finding similar people is useful in many applications like recommender systems, predicting stable marriage or relationship, recruiting employees, etc. In their work, Aviv Nisgav, Boaz Patt-Shamir [6] has used a query based classification algorithm to find similarity between two users. On the other hand, Pasquale De Meo *et*

al. [7] have proposed an approach based on the knowledge of social ties existing among users, and the analysis of activities in which users are involved, in order to estimate the similarity of two users. A similar approach is used by Chris Tanner *et al.* [8] in finding highly similar users and their inherent patterns. In our work presented herein, we find groups of similar users by clustering them on the basis of their Facebook usage available through their profiles. We have used Particle Swarm Optimization based technique for clustering and have evaluated the performance by measuring quantization error, which is lesser than what is achieved by k-means, when used for 7 or less clusters.

The rest of the paper is organized as follows: in Section 2 we present work related to finding similar users on Facebook. That is followed by Section 3, where we give a background on clustering and in particular, clustering using Particle Swarm Optimization. In Section 4 we describe our methodology followed by details about the experiments conducted, in Section 5 and the results and discussions in Section 6 and Section 7. Towards the end, in Section 8 we present our conclusion and we give directions for future research in Section 9.

2. Related Work

Finding similar users on Facebook has drawn attention from several researchers in the past. For example, Aviv Nisgav, Boaz Patt-Shamir [6] has considered each user as a vector of preferences (answers to queries). They consider two users similar if their preference vectors differ in only a few coordinates. The preferences are unknown to the system initially, and the goal of the algorithm is to classify the users into classes of roughly the same preferences by asking each user to answer the least possible number of queries. They present an "anytime" algorithm that asks each user at most one query in each round, while maintaining a partition of the users. The quality of the partition improves over time: for n users and time T, groups of $O(n/T)$ users with the same preferences will be separated (with high probability) if they differ in sufficiently many queries.

Pasquale De Meo, Emilio Ferrara, Giacomo Fiumara [7], on the other hand, have proposed an approach in order to estimate the similarity of two users based on the knowledge of social ties (i.e., common friends and groups of users) existing among users, and the analysis of activities (i.e., social events) in which users are involved. For each of these indicators, authors draw a local measure of user similarity, which takes into account only their joint behaviors. After this, book considers the whole network of relationships among users along with local values of similarities and combines them to obtain a global measure of similarity. Applying the Katz coefficient, a popular parameter introduced in Social Science research, carries out such a computation. Finally, similarity values produced for each social activity are merged into a unique value of similarity by applying linear regression.

In their work, Xiangye Xiao *et al.* [9] estimate the similarity between two users in terms of the semantic location history learned from their historical GPS trajectories. They measure the similarity between different users' semantic location history by using maximal travel match algorithm. Semantic location history carries more semantic meaning of users' interest beyond low-level geographic position. Their approach can estimate the similarity between two users without overlaps in the geographic positions.

Fabrício Benevenuto *et al.* [10] in their work found that understanding how users behave when they connect to social networking sites creates opportunities for better interface design, richer studies of social interactions, and improved design of content distribution systems. There analysis of the clickstream data reveals key features of the social network workloads, such as how frequently people connect to social networks and for how long, as well as the types and sequences of activities that users conduct on these sites.

3. Background

Clustering can be considered the most important unsupervised learning problem; so, as every other problem of this kind, it deals with finding a structure in a collection of unlabeled data. A loose definition of clustering could be "the process of organizing objects into groups whose members are similar in some way". A cluster is therefore a collection of objects which are "similar" between them and are "dissimilar" to the objects belonging to other clusters [11]. There are several well defined algorithms for clustering like K-means, EM algorithm, Fuzzy C-Means algorithm, etc. [12].

Recently, lot of work has happened in applying Swarm Intelligence (SI) techniques to solve data mining tasks [13]. SI is the collective behavior of decentralized, self-organized systems, natural or artificial. The concept is employed in work on artificial intelligence. The expression was introduced by Gerardo Beni and Jing Wang in 1989, in the context of cellular robotic systems. Particle Swarm Optimization is a SI technique, which lends itself well to several optimization problems like clustering. Particle Swarm Optimization is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality.

Particle Swarm Optimization on data clustering was done by DW van der Merwe and AP Engelbrecht [14]. Each particle represents a solution space of cluster. Each solution is represented by the coordinates of a user pre-defined number of cluster centroids. Each data instance is assigned to the nearest cluster centroid, using Euclidean distance. The fitness function used is the quantization error, which can be seen as the average distance from a data point to its cluster centroid, averaged over the different clusters.

Quantization error explained in equation (3) is used to measure cluster quality. After every iteration, each particle cluster quality is analyzed using this equation. Thus the clustering problem gets converted to an optimization problem of optimizing quantization error and hence solvable by Particle Swarm Optimization.

4. Methodolody

A person's Facebook activities include posting status, liking posts made by other people commenting, etc. Sometimes this show a person's interest, for example when a person in passionate about cricket his recent post will surely include posts on ICC Cricket world cup 2015. They express their feelings through these posts. People comment on some statuses or photos also and express feeling about that post. People also like photos, status, and comment. For our work, we have gathered Facebook data of users via a Facebook horoscope application developed by us. We show user their Facebook usage statistics in reply to accepting permission to give data and same data and statistics is also used in proposed algorithm. For every user who agrees to use our application, we collect his id, name, birthday, photos (name, place, updated time, tags, comments, likes), status updates (message, place, updated time, tags, comments, likes), tagged places, albums, friends.

From the collected status updates we extract certain features, which have been shown to be correlated with user's personality. In particular we extract Linguistic Inquiry and Word Count features (LIWC) introduced by Mairesse and Walker *et al.* [15]-[17]. LIWC produces statistics on 81 different features of text in five categories. These include Standard Counts (word count, words longer than six letters, number of prepositions, etc.), Psychological Processes (emotional, cognitive, sensory, and social processes), Relativity (words about time, the past, the future), Personal Concerns (such as occupation, financial issues, health), and Other dimensions (counts of various types of punctuation, swear words). In addition to the LIWC linguistic features, we have considered several non-linguistic features as well for our analysis. Some of these – name, gender, age group, friend count, profile picture count, cover picture count - were directly available as collected data while others – number of likes of per status, number of likes per photo, average number status per week, average number photo per week, total status updated, total photo updated – were extracted as a result of some computations.

Let there be **b** linguistic features and **g** non-linguistic features. Therefore, total features n is $= b + g$. For n features, $f_1, f_2, \ldots, f_n$, each user is now described by a row of n columns with their values as the values of respective features for that user. If there are u users then we have a matrix M_{u*n} where u is number of user (rows) and n is the number of features (columns).

However, before the above matrix can be populated we need to calculate an average feature value for each of the linguistic features corresponding to all the status updates of a user. This need arises because there are several status updates for each user. Say user u_1 has t status updates $s_1, s_2, \ldots, s_t$ in the collected data. In feature extraction for user u_1 matrix of F_{t*b} will be created where t is no. of statuses of user u_1 and b is number of linguistic features to be extracted. Now we have to convert this matrix data into one row in above matrix M_{u*n} because each user gets one row. So we take average of feature in column major form as

$$f_j = \frac{\sum_{i=1}^{t} s_{ij}}{t}$$

where f_j is feature column which runs form 1 to n, t is total no of statuses by the user, and s_{ij} is i^{th} status value for j^{th} feature column. By this equation we are averaging over column for each feature for particular u_1 user. Now we have b values for user u_1. This is how we get the sub matrix M_{u1*b}. Remaining g non-linguistic features are directly added to M_{u1*g}. In this way we get total row for one user M_{u1*n}. The complete matrix M_{u*n} of features is then populated by doing similar processing for all users.

Particle Swarm Optimization based clustering algorithm [14] as described above in Section III is used for clustering users based on similarities or dissimilarities with respect to above features. We have used the Hybrid Particle Swarm Optimization algorithm [14] where inertia and acceleration (c_1, c_2, w) are constant and a random particle is initialized as K-means clustering output. Other remaining particles are initialized randomly.

On matrix of user-features M_{u*n}, Particle Swarm Optimization based clustering is applied. Each particle is initialized in solution space of dimension number of cluster $\times$ no of feature. Particle structure is array of centroid vectors as follows

$$x_i = \left(m_{i1}, m_{i2}, \ldots, m_{iN_c}\right)$$

where N_c is number of clusters, $\mathrm{m_{i1}}$ is the first cluster centroid in the i^{th} particle. Each particle is given a constant inertia and random amount of initial velocity. Each particle represents a possible solution in the solution space.

At every iteration, for each particle, velocity and position values are calculated using the following respectively.

$$V_{i,k}(t+1) = w * V_{i,k}(t) + c_1 r_{1,k}(t)\left(y_{i,k}(t) - x_{i,k}(t)\right) + c_2 r_{2,k}(t)\left(\hat{y}_k(t) - x_{i,k}(t)\right) \quad (1)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (2)$$

where,

$V_{i,k}(t+1)$ is new velocity of i^{th} particle

w is inertia weight constant, c is a constant and r is a random value

$y_{i,k}(t)$ is local best, and $\hat{y}_k(t)$ is global best.

Fig. 1 shows, new position of $\mathrm{i^{th}}$ particle at x_i^{k+1}, where x_i^k is old position of particle. Pbest_i^k , Gbest^k is respective personal best of particle and global best of swarm for current iteration.

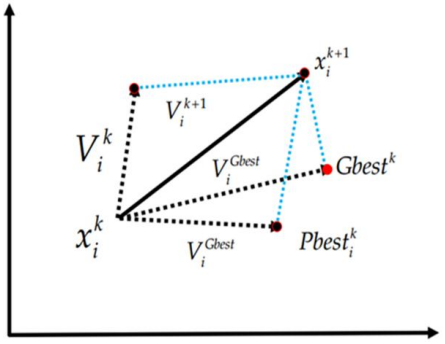

Fig. 1. Resultant new velocity of particle.

To check how good a solution is represented by a particle, i.e. fitness of particle, quantization error is calculated of that cluster solution as follows

$$J_e = \frac{\sum_{j=1}^{N}[\sum_{Z_p \in C_{ij}} \frac{d(z_p . m_j)}{|C_{ij}|}]}{N_c} \tag{3}$$

where N_c is number of clusters, Z_p is a data vector belonging to cluster C_{ij}, $|C_{ij}|$ is number of data points in that cluster, and

$$d(z_p, m_j) = \sqrt{\sum_{k=1}^{n} (Z_{pk} - m_{jk})^2} \tag{4}$$

Which computes the Euclidean distances of a data point z_p in a cluster j from its centroid m_j.

After every iteration of Particle Swarm Optimization, particle's personal best and overall global best is updated. After certain iterations, particles are converged. Finally, global best is optimal solution for quantization error i.e. solution for clustering.

In the optimal solution, each cluster can be thought of as a group of people on Facebook social network having similar characteristics (in terms of the chosen features). Whenever a new user is to be recommended a group of users similar to him, his features are extracted and by using equation number (4) its distance from each cluster centroid is computed. The user is then recommended the group of people who belong to that cluster centroid.

5. Experiments

We developed a Facebook Horoscope application to gather Facebook usage data of students residing in a student hostel. Our dataset thus collected consists of 65 users. Data consists of status and status related data, photo and photo related data, place, tagged place, groups connected to, etc. as described above. We used LIWC feature extraction tool [17] to extract 85 linguistic related features. In addition we have extracted 30 non linguistic features as mentioned in the methodology above.

As per recommendations by Van der Merwe, D. W. *et al.* [14] we have chosen the following values for the parameters - Number of particles taken in Particle Swarm Optimization is 10, Global weight constant c_1, and local weight constant c_2 is set to 1.49 in equation (1), Inertia constant w is taken as 0.72 in equation (1), Number of iterations carried over Particle Swarm Optimization is 1000. According to Van der Merwe, D. W. *et al.* these values of c_1, c_2, and w give good convergence rate in Particle Swarm Optimization. We vary the

number of clusters in our experiments from 2 to 9. Quantization error is reported as in following section.

6. Results

Experimentally the following 10 features give best results. They are, number of places went to, talking about I, negations, numbers, positive emotions, negative emotions, cognitive process, sensory process, social, time. For the sake of brevity we report results of our experiments corresponding to these features only.

All results shown in following table are averaged over 10 executions.

Table 1. Clustering Quantization Error

No. of Clusters	K-Means	PSO
2	9.613855	9.600989
3	13.23120	9.57369
4	11.9885	9.07875
5	11.65219	9.691163
6	7.868131	7.868131
7	8.197161	6.570301
8	6.025002	6.025002
9	7.905946	7.905946

7. Discussions

As number of clusters increase, the quantization error decreases till 8 number of clusters and then again starts increasing. This seems reasonable. With only one or two clusters, users who may be quite different from each other may also be forced to be grouped together in the same cluster. But as the number of clusters increase, only similar users would naturally belong to a group. However, when number of clusters increase too much, similar users may be forced to settle with different groups. In our experiments, we get least quantization error when number of clusters equal to 8. We observed that Particle Swarm Optimization performed better than k-means in 5 cases and equally performed in 4 cases.

8. Conclusion

We have proposed a method for finding similar users on Facebook using Particle Swarm Optimization technique. It always forms better groups than k-means. In worst case it forms groups with quantization error equal to k-means. This method can be used in college admission scenarios. Say for example, if a new student comes to an institution and he doesn't know whom should he meet and make friends, using our technique and algorithm he can suggested groups of other students he can meet of the same interests he has. This approach can also be used in companies where new employee joins a company and the company doesn't know in which project he would fit well. The company can then make groups based on mining professional network data (for e.g. from LinkedIn) and suggest the new employee a group of people with similar interests. Another possible application includes conducting rope-in exercises for new hires. A new hire's similarity can be computed from various existing groups in organization and can be assigned "buddy"/mentors accordingly.

9. Future Scope

One of the limitations of this work, as with k-means, is that the number of clusters has to be predefined. However, Particle Swarm Optimization can be very well adapted to overcome this limitation [18]. Further, our implementation is not optimized for working in very large dimensional space relative to the number of clusters. Again, variants of Particle Swarm Optimization have already been proposed by the research

community [19] to handle this limitation as well. Going forwards we also plan to extend our work to include more number and a wider variety of features. Another extension of this work could be to select features from a particular domain thereby getting clusters which will have some real world interpretation.

Acknowledgment

We wish to thank Van der Merwe, D. W., and Andries Petrus Engelbrecht for answering our queries on their work [14].

References

[1] Zephoria Inc. The top 20 valuable facebook statistics. From: https://zephoria.com/social-media/top-15-valuable-facebook-statistics/

[2] Mislove, A., *et al.* (2007). Measurement and analysis of online social networks. *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measuremen*t.

[3] Thompson, L. A., *et al.* (2008). The intersection of online social networking with medical professionalism. *Journal of General Internal Medicine, 23(7)*, 954-957.

[4] Palmer, A., & Nicole, K.-L. (2009). An experiential, social network-based approach to direct marketing. *Direct Marketing: An International Journal, 3(3),* 162-176.

[5] Kosinski, M., Stillwell, D. J., & Graepel. T. (2013). Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the National Academy of Sciences (PNAS)*.

[6] Nisgav, A., & Boaz, P.-S. (2009). Finding similar users in social networks. *Proceedings of the Twenty-First Annual Symposium on PARALLELISM in Algorithms and Architectures*.

[7] De, M., Pasquale, E. F., & Giacomo, F. (2011). Finding similar users in facebook. *Social Networking and Community Behavior Modeling Qualitative and Quantitative Measurement*, 304-323.

[8] Tanner, C., Irina, L., & Amruta, J. (2008). Social networks: Finding highly similar users and their inherent patterns. *Social Netw*.

[9] X.-Y., Xiao, *et al.* (2010). Finding similar users using category-based location history. *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*.

[10] Benevenuto, F., *et al.* (2009). Characterizing user behavior in online social networks. *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement*.

[11] Cluster analysis. (March 25, 2015). *Wikipedia, the Free Encyclopedia*. Retrieved April 4, 2015, from http://en.wikipedia.org/w/index.php?title=Cluster_analysis&oldid=653528689

[12] Azad, N. Data clustering algorithms. From: https://sites.google.com/site/dataclusteringalgorithms/

[13] Martens, D., Bart, B., & Tom, F. (2011). Editorial survey: Swarm intelligence for data mining. *Machine Learning, 82(1)*, 1-42.

[14] Van der Merwe, D. W., & Andries, P. E. (2003). Data clustering using particle swarm optimization. *Proceedings of the 2003 Congress on Evolutionary Computation*: *Vol. 1.*

[15] Mairesse, F., & Marilyn, W. (2006). Words mark the nerds: Computational models of personality recognition through language. *Proceedings of the 28th Annual Conference of the Cognitive Science Society*.

[16] Mairesse, F., *et al.* (2007). Using linguistic cues for the automatic recognition of personality in conversation and text. *Journal of Artificial Intelligence Research*, 457-500.

[17] Pennebaker, J. W., Booth, R. J, & Francis, M. E. (2007). *LIWC2007: Linguistic Inquiry and Word Count*, Austin, Texas.

[18] Omran, M. G. H., Ayed, S., & Andries, P. E. (2006). Dynamic clustering using particle swarm optimization with application in image segmentation. *Pattern Analysis and Applications, 8(4),* 332-344.

[19] Y. P., Lu, *et al.* (2011). Particle swarm optimizer for variable weighting in clustering high-dimensional data. *Machine learning, 82(1),* 43-70.

Pratik Shrikant Kabara is a citizen of India and is living in a town called Yeola. He is currently a final year student of the Computer Engineering Department at Vishwakarma Institute of Technology (VIT), Pune. He has secured the 1st rank in the department. In Fall 2013, he completed one semester exchange program in Ryerson University, Canada. He has contributed to Python "Faker Library", which is an open source project.

Vishal Kaushal is a citizen of India and is currently staying in Pune. In 2004, he has received the bachelor of technology (BTech) in computer science and engineering from Indian Institute of Technology (IIT) Kharagpur, where he was the department topper. After graduating from IIT Kharagpur, he has worked in Oracle India Pvt. Ltd. and BMC Software for several years in various capacities. Currently he is working as an assistant professor in the Department of Computer Engineering, Vishwakarma Institute of Technology (VIT) Pune. He has done his master degrees from University of Pune.

Akshay Divekar is a citizen of India and is living in Pune. He is currently a final year student of the Computer Engineering Department at Vishwakarma Institute of Technology (VIT), Pune.

Mohit Kamdar is a citizen of India and is living in Pune. He is currently a final year student of the Computer Engineering Department at Vishwakarma Institute of Technology (VIT), Pune.

Devvrat Ganeriwal is a citizen of India and is living in Pune. He is currently a final year student of the Computer Engineering Department at Vishwakarma Institute of Technology (VIT), Pune.