A Cloud-Assisted Framework for Bag-of-Features Tagging in Social Networks

Zhanming Jie, Ming Cheung, James She
HKUST-NIE Social Media Lab., The Hong Kong University of Science and Technology
{zjieaa, cpming, eejames}@ust.hk

***Abstract*—Recently, Bag-of-Features Tagging is proven to be an alternative to discover user connections from user shared images in social networks. This approach used unsupervised clustering to classify the user shared images and then correlate similar user, which is computationally intensive for real-world applications. This paper introduces a cloud-assisted framework to improve the efficiency and scalability of Bag-of-Features Tagging. The framework distributes the computation of the unsupervised clustering, the profile learning process and also the similarity calculation. The experiment proves how a scalable cloud-assisted framework outperforms a stand-alone machine with different parameters on a real social network dataset, Skyrock.**

I. Introduction

Social media is becoming prevalent among people in our daily live nowadays. Lots of social media applications have been deployed on the Internet based on the social graphs (SGs), e.g., item recommendation (jobs, movies, etc.) using friends' interests [1], and friendship recommendation [2] based on existing connections among users. Facebook makes recommendations based on implicit information that defines users kept by the system [3]. The SG of Facebook is formed by users explicitly adding other individuals as "Friends" in the social network [4]. Other social network sites also follow a similar way to form the SG. s Recently, connection discovery using user shared images is also proven to be effective by Bag-of-Features Tagging (BoFT) [5].

One simple method to discover connections is the Friends-of-Friends (FoF) approach, which calculates user similarity based on friendship information of users [6]. Connection discovery can also be based on the features that describe user profiles according to the principle that people establish their social contacts with others who have similar tastes. [5] proposes BoFT to discover connections based on the information of user shared images. However, BoFT has its limitation in that intensive computation for feature extraction and clustering of the images, especially when processing a large amount of data.

This paper proposes a cloud-assisted framework for the BoFT approach, which will solve the intensive computation issue in BoFT. Fig. 1 shows the general idea of BoFT on the cloud platform. The BoFT migrates to the cloud and the whole architecture is improved to help BoFT fit into the cloud. With the images shared by users, this cloud-based system will analyze the images and understand users' interests from their images. The similarity between users will then be calculated based on their profiles. A higher similarity between users means that they are more likely to be friends. The main contributions of this paper are summarized as follows:

Fig. 1: General idea of BoFT in the cloud

1) proposes a cloud-assisted framework for BoFT to enhance the scalability;
2) develops an implementation to prove true feasibility and effectiveness of the cloud-assisted BoFT;
3) proves how the cloud-assisted system is better than a stand-alone machine on a real social network dataset.

The rest of the paper is structured as follows: Section II introduces the BoFT approach for connection discovery and Section III presents the proposed cloud-assisted design for BoFT; Several experiments are conducted in Section IV to evaluate the proposed framework and Section V concludes the work.

II. Bag-of-Features Tagging for Connection Discovery

In BoFT, Bag-of-Features (BoF) is used to visually annotate the images with non-user-generated labels. BoF is a method to represent images as feature vectors of local image descriptors. The first step is feature extraction by Scale-Invariant Feature Transform (step 1 of Fig. 2). Secondly, Codebook generation referred in step 2 of Fig. 2 is a process to obtain visual words that can represent the features from feature extraction. In this part, the feature clustering process is used to group similar features. With set of visual words representing images, clustering can also be achieved to assign each image a non-user-generated label. These labels generated via the image analysis process are different from those generated using user

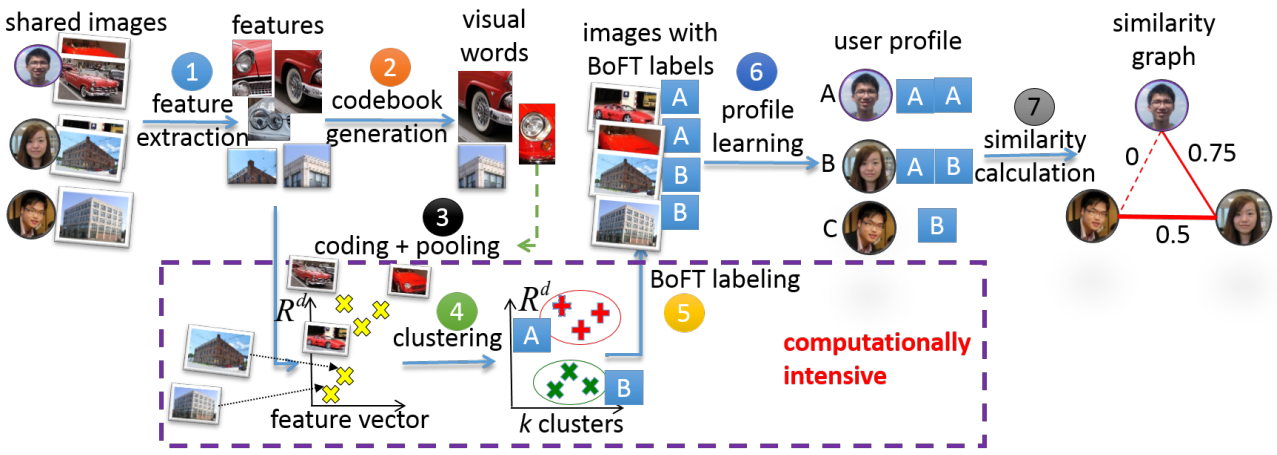

Fig. 2: Overview of BoFT-based Connection Discovery [5]

annotated tags. Particularly, these non-user-generated labels are called BoFT labels in BoFT approach.

User profiles are then learned through those labels and similarity between users is calculated based on their profiles. Connections are discovered based on the cosine similarity calculation. As shown in step 7 in Fig.2, similarity between users can be calculated using the generated labels of images that are posted by them. User pairs with more similar profiles will obtain a higher similarity, which means that these users are similar based on their profiles. Finally, the most similar m users will be chosen for each user u as discovered friends.

III. Proposed Cloud-assisted Bag-of-Features Tagging Approach

This section indicates the limitations of BoFT and presents the cloud-assisted design for the BoFT approach. As shown in step 4 of Fig. 2, BoFT-based connection discovery consumes many computational and storage resources because all the data need to be stored in memory. Besides the large amount of images, similarity calculation is also computationally intensive when there are millions of users in social networks. A stand-alone machine definitely cannot efficiently process billions of images in real-world social networks. Therefore, a cloud-assisted design is proposed to help to solve the above problem for the BoFT approach.

A. Proposed Cloud-assisted Framework

In the cloud-assisted design, as shown in Fig. 3, the architecture corresponds to the clustering, profile learning and similarity calculation (step 4, 6 and 7 of Fig. 2, respectively). The original image data, as represented by vectors, is first split into multiple blocks in the Hadoop Distributed File System (HDFS) and distributed to virtual machines (VMs). Then an unsupervised clustering process, k-means in this paper, is performed to obtain a label for each image under the Hadoop Mapreduce framework. The user profile, which is a distribution of the labels of each user, is learned once the image labels are obtained. Finally, similarity calculation is distributed in the framework and similarity graph is obtained as a result.

1) Distributed Unsupervised Clustering: In terms of the unsupervised clustering for the image vectors, this paper employs the MapReduce framework [7], the dominant framework nowadays, on k-means clustering. The images will be assigned different BoFT labels based on their Euclidean distance during the distributed k-means clustering and their labels will also be attached to the corresponding users. The BoFT label is referred by the cluster ID generated from the k-means clustering. The complexity of k-means is $O(lmkd)$, where l is number of iterations, m is number of images, k is number of clusters and d is the dimension of images. Thus k-means clustering has a high time complexity when the dataset is large. However, it is clear that the Euclidean distance computation among the points and centroids can be parallelized in the cloud through splitting the data into different groups processed by different VMs. Both Euclidean distance computation and recomputing the centroids can be run on multiple VMs. As shown in Fig. 3, the mapper is responsible for assigning image vectors to the nearest centroids and the reducer is responsible for recomputing the centroid for each cluster. Finally, the result of the MapReduce program is a set of labels and each image is attached with a corresponding label.

2) Distributed Profile and Similarity Learning: Once the corresponding labels of the images are obtained, user profiles can be learned based on their images' labels. Each VM is in charge of several users' profiles and computes the distribution of different labels for each user. The distribution will then be represented by a vector, which is the user profile. As shown in Fig. 3, the letters around a user are the labels generated in the image clustering process. The similarity calculation follows the same mechanism, in which each VM is responsible for calculating the cosine similarity for several users based on their profiles. The similarity graph is then obtained and connection discovery can be achieved based on this graph. Since the dataset in this paper only includes 722 users, a multi-thread program is used to simulate multiple VMs in the profile learning and similarity computation process.

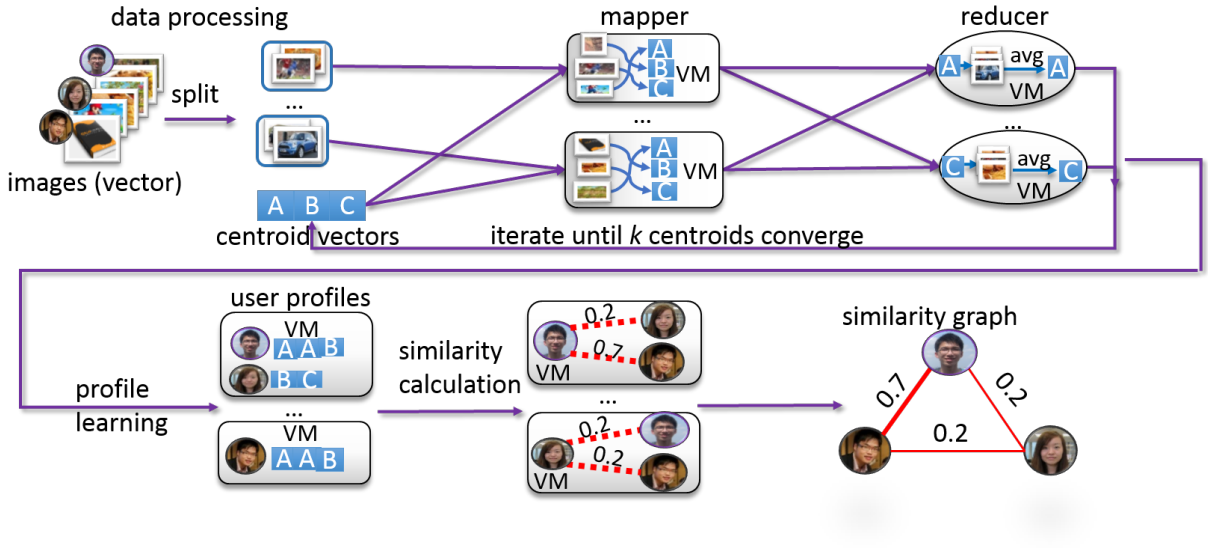

Fig. 3: Proposed Cloud-assisted Framework for Bag-of-Features Tagging

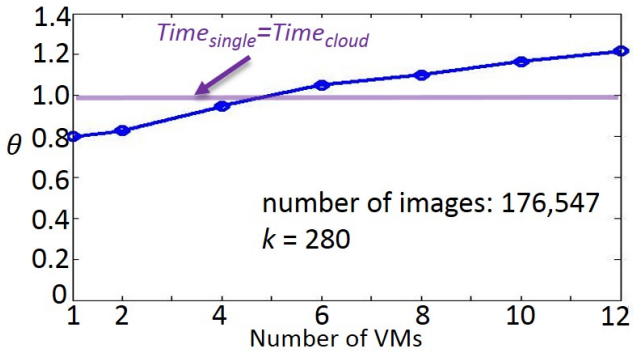

Fig. 4: Speedup of BoFT on different number of VMs

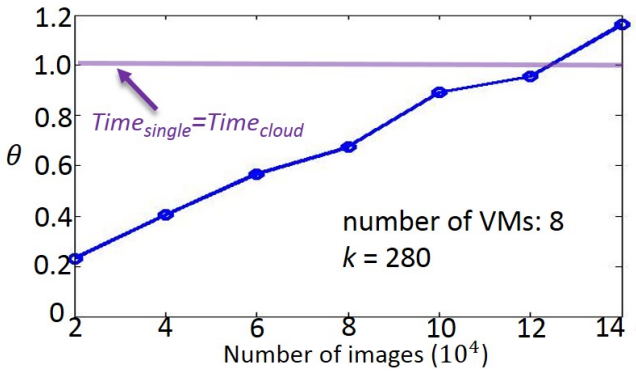

Fig. 5: Speedup of BoFT on different number of images

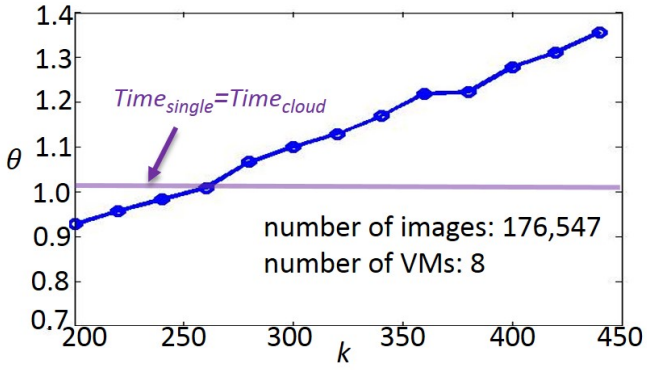

Fig. 6: Speedup of BoFT on different value of k

IV. Experimental Results

A. Experimental Setup

The dataset was collected from Skyrock, a general social network which allows users to post blogs and images. The dataset comprises of 176,547 images uploaded by 722 unique and randomly selected users. The dataset involves a total of 2,439,058 followee/follower connections, including 15,812 connections within these 722 users.

The Hadoop cluster consists of 13 VMs, one master node and 12 slave nodes. All the VMs are Amzon EC2 m3.xlarge instances with the following characteristics: 15 GB memory; 4 virtual compute units; 2x40 SSD storage and high network performance. For the single machine, experiments are run on stand-alone platform using one m3.xlarge instance.

B. Runtime Performance

This paper first investigates the relationships between the number of VMs, the number of images and the value of k for BoFT on both the cloud platform and stand-alone machine. To quantify the effectiveness of the different platforms, T_c and T_s are used to denote the average running time of BoFT on the cloud and the stand-alone running time respectively. In order to compare them, θ is used to measure the speedup:

$$\theta = \frac{T_s}{T_c}. \quad (1)$$

The reference line $\theta = 1$ means that experiments on both the Hadoop cluster and the stand-alone machine have an equivalent performance. In Fig. 4, 176,547 images are processed and the k is fixed to 280. The results will have the same trend with any value of k and 280 is just an arbitrary choice. The speedup increases with increasing number of VMs and least number of VMs for BoFT to gain speedup from the cloud is 6. Otherwise, using a stand-alone machine is better for BoFT. In general, the cloud-assisted design allows faster computation time as more VMs are used. However, small number of VMs (1 to 4 in Fig. 4) lets the IO overhead dominates the running time of BoFT on cloud.

Fig. 5 shows the relationship between the speedup and the volume of the dataset. The number of VMs is fixed to 8, but any other number of VMs will show the same trend as those shown in Fig. 5 and Fig. 6. It is observed that the cloud-assisted framework can gain speedup when the dataset is large enough because overhead time in the cloud will be negligible compared to calculation time for the BoFT. The speedup also increases along with the number of images is increasing. In terms of the value of k, Fig. 6 shows the performance for different values of k from 200 to 440.

Intuitively, k-means will take more time to converge with increasing values of k value. With the fixed number of images and VMs, the speedup surpasses 1 when k reaches about 280. This phenomenon also reflects that cloud-assisted BoFT performs better than stand-alone BoFT although k is increasing. This section elaborates how to choose different platforms under different conditions of resources. The experimental results prove that with increasing number of VMs, dataset size and value of k, implementing BoFT on the cloud platform will gain more significant improvement in performance.

C. Scalability

Scaleup [8] is a common metric to evaluates the scalability of a system when both the number of VMs and the size of the dataset grow. Scaleup is defined as the ability of an m-times larger system to perform on m-times larger datasets in the same running time as the stand-alone machine:

$$Scaleup(m) = \frac{T_1}{T_m}, \tag{2}$$

where m represents m-times larger datasets and also the number of VMs, and T_m stands for the experiment time for m VMs to perform on m-times larger datasets. To demonstrate the effectiveness of the cloud-assisted design in handling larger datasets when more VMs are available, the scaleup experiment is performed with the increasing size of the datasets in direct proportion to the number of VMs in the cloud-assisted system. The whole dataset is divided into 12 parts, where the m^{th} part contains an m-times larger dataset. Fig. 7 shows the performance results of the datasets, where m means m-times larger datasets are performed on m VMs. Ideally, the curve is a horizontal line, which represents that a constant response time is maintained as the size of the problem and system grow incrementally. The scaleup first drops down to about 0.8 when the scale is small because of the overhead in the cloud platform. Then it decreases steadily until m is 9 and finally remains stable after that point. Clearly, the cloud-assisted design for BoFT scales very well.

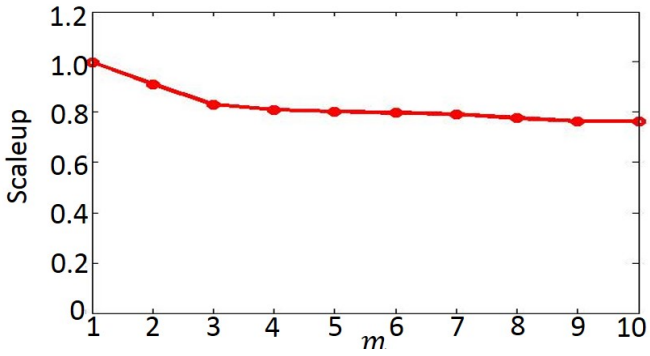

Fig. 7: Scaleup

V. Conclusion

In this paper, a cloud-assisted framework is proposed to improve the efficiency and scalability of BoFT. Specifically, the clustering process in BoFT is distributed by the Hadoop MapReduce framework and the computation for both profile learning and similarity calculation is parallelized in the cluster. The experimental results show that speedup performance of BoFT on the cloud platform is better than the performance on the stand-alone platform when the size of the dataset, the number of VMs and the value of k are large enough. Furthermore, this paper has also proved the high scalability of the proposed cloud-assisted framework on BoFT.

Acknowledgment

This work is supported by HKUST-NIE Social Media Lab., HKUST

References

[1] I. Konstas, V. Stathopoulos, and J. M. Jose, "On social networks and collaborative recommendation," in *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2009, pp. 195–202.

[2] E. Zheleva, L. Getoor, J. Golbeck, and U. Kuter, "Using friendship ties and family circles for link prediction," in *Advances in Social Network Mining and Analysis*. Springer, 2010, pp. 97–113.

[3] J. B. Walther, B. Van Der Heide, S.-Y. Kim, D. Westerman, and S. T. Tong, "The role of friends appearance and behavior on evaluations of individuals on Facebook: Are we known by the company we keep?" *Human Communication Research*, vol. 34, no. 1, pp. 28–49, 2008.

[4] J. Ugander, B. Karrer, L. Backstrom, and C. Marlow, "The anatomy of the facebook social graph," *arXiv preprint arXiv:1111.4503*, 2011.

[5] M. Cheung and J. She, "Bag-of-features tagging approach for a better recommendation with social big data," in *IMMM 2014, The Fourth International Conference on Advances in Information Mining and Management*, 2014, pp. 83–88.

[6] L. Lü and T. Zhou, "Link prediction in complex networks: A survey," *Physica A: Statistical Mechanics and its Applications*, vol. 390, no. 6, pp. 1150–1170, 2011.

[7] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[8] W. Zhao, H. Ma, and Q. He, "Parallel k-means clustering based on MapReduce," in *Cloud Computing*. Springer, 2009, pp. 674–679.