2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)
July 6-11, 2014, Beijing, China

Stochastic Gradient Descent based Fuzzy Clustering for Large Data

Yangtao Wang, Lihui Chen and Jian-Ping Mei

***Abstract*— Data is growing at an unprecedented rate in commercial and scientific areas. Clustering algorithms for large data which require small memory consumption and scalability become increasingly important under this circumstance. In this paper, we propose a new clustering approach called stochastic gradient based fuzzy clustering(SGFC) which achieves the optimization based on stochastic approximation to handle such kind of large data. We derive an adaptive learning rate which can be updated incrementally and maintained automatically in gradient descent approach employed in SGFC. Moreover, SGFC is extended to a mini-batch SGFC to reduce the stochastic noise. Additionally, multi-pass SGFC is also proposed to improve the clustering performance. Experiments have been conducted on synthetic data to show the effectiveness of our derived adaptive learning rate. Experimental studies have been also conducted on several large benchmark datasets including real world image and document datasets. Compared with existing fuzzy clustering approaches for large data, the mini-batch SGFC shows comparable or better accuracy with significant less time consumption. These results demonstrate the great potential of SGFC for large data analysis.**

I. Introduction

LARGE data is much easier to be acquired and becomes prevalent and unprecedented due to the development of technologies such as the digit camera, distribution of various sensors, world wide web .etc. which are producing lots of data in the form of image, video or text every day. For example, there are over 50 billion pages indexed and more than 2 million queries per minute in Google, about 4.5 million photos uploaded every day in Flickr and about 48 hours of video uploaded every minute in YouTube. Mining valuable information by large data analysis techniques becomes critical for different organizations to get the competitive advantages and has the potential to transform many facets of society. Clustering as an unsupervised learning technique to find pattern structure underlying the unlabelled data plays a pivotal role in data analysis. Many different clustering algorithms based on various theories have been developed and successfully applied in different applications over the past decades[1], [2], [3]. For clustering large data, two main challenges are memory consumption and scalability. Traditional clustering methods that need the entire data matrix reside in the memory become infeasible when the data is too large for the memory. To handle these problems and accelerate the clustering, different strategies are used in large data clustering including random sampling[4], [5], summarization[6], [7], distributed methods[8], [9], approximation[10], [11] and incremental methods[12], [13], [14], [15]. Fuzzy clustering algorithms are also extended to handle large data because they may capture the natural structure of a dataset more closely as discussed in the literature [16], [17], [18], [19]. Two popular fuzzy clustering algorithms for large data are called single pass fuzzy c means(SPFCM) [20] and online fuzzy c means(OFCM) [21]. In the two approaches, data is processed in a chunk-based way which means data is loaded and processed partially in the computer to reduce the memory consumption. In this paper, we propose a new clustering approach called stochastic gradient based fuzzy clustering(SGFC) which is based on stochastic approximation to handle such kind of large data. Our approach has the properties with low memory consumption and also high scalability. Additionally, for the gradient descent algorithm, instead of using heuristic learning rate, we derive an adaptive learning rate which can be updated incrementally and maintained automatically during the clustering process. The experiments are conducted to show the effectiveness of our adaptive learning rate.

The rest of the paper is organized as follows: in the next section, a review on the related fuzzy clustering approaches is highlighted. In section 3, the details of the proposed stochastic gradient based fuzzy clustering called SGFC and its mini-batch and multi-pass version are presented. Experiments on several large datasets are conducted and the results are analyzed in section 4. Finally, conclusions are drawn in section 5.

Yangtao Wang and Lihui Chen are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore (email: wang0689@e.ntu.edu.sg, elhchen@ntu.edu.sg), Jian-Ping Mei is with the College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China (email: meijianping10@gmail.com).

II. Related work

In this section, two Fuzzy c means(FCM) [16] based clustering algorithms for large data are reviewed. The common and different characteristics of the two approaches are discussed.

A. SPFCM and OFCM

Both SPFCM [20] and OFCM [21] are designed for handling large data based on FCM which is a kind of batch algorithm. As we know, batch algorithm which needs load the entire dataset into the memory may not be suitable for large data when the data is too large for the memory. Therefore, to handle large data, both SPFCM [20] and OFCM [21] process the data in a mini-batch way to reduce the memory consumption. In other words, the entire dataset is considered as coming chunk by chunk in which a set of centroids is identified to represent each chunk. For identifying the centroids, weight FCM(wFCM) is applied in both approaches.

The main difference between SPFCM and OFCM is the way that how the identified centroids from each chunk are processed. In SPFCM, the centroids identified from the previous chunk are combined into next chunk and the final set of centroids for the entire data is generated after last chunk

978-1-4799-2072-3/14/$31.00 ©2014 IEEE

is processed. While in OFCM, the identification of centroids for every chunk is processed individually and an additional step is needed to generate the final set of centroids for the entire data.

As discussed previously, SPFCM and OFCM adopt the mini-batch way to handle large data. However, the time to identify the centroids for each mini-batch by using wFCM maybe long if the data in the mini-batch needs to iterate for many times to converge. To accelerate the clustering process, in our approach we adopt stochastic gradient method to identify the centroids for the entire dataset. Next, we propose our new fuzzy clustering approach based on stochastic gradient called SGFC including its extension named mini-batch SGFC and multiple-pass SGFC.

III. The proposed approach

In this section, we firstly introduce the proposed approach SGFC and derive the self-adaptive learning rate. Then, to reduce the impact of noisy gradient, we extend SGFC to mini-batch SGFC in which the data is considered as coming in mini-batch. The detail steps of the algorithms are presented respectively.

A. Stochastic gradient based fuzzy clustering(SGFC)

Given n objects $\{x_1, x_2...x_n\} \in \mathbb{R}^d$ generated from an unknown probabilistic space $P \in \mathbb{R}^d$, fuzzy clustering can be regarded as identifying k centroids to represent the objects. The problem can be formulated as follows:

$$\min_{V \in \mathbb{R}^{d \times k}} J_n(V) = \frac{1}{n} \sum_{i=1}^{n} l(x_i, V) \tag{1}$$

where V is the cluster centroids and

$$l(x_i, V) = \sum_{c=1}^{k} u_{ci}^m \|x_i - v_c\|^2 \tag{2}$$

subject to

$$\sum_{c=1}^{k} u_{ci} = 1 \tag{3}$$

where u_{ci} is the membership of object x_i in cluster c, and v_c is the centroid of the c_{th} cluster. As pointed out by the authors in [22], for large scale learning instead of minimizing the empirical cost $J_n(V)$, one can pay more attention on minimizing the expected cost $J(V)$ as follow.

$$\min_{V \in \mathbb{R}^{d \times k}} J(V) = E_{x \in P}(l(x, V)) \tag{4}$$

where $E_{x \in P}$ is the expectation on probabilistic space P.

It is difficult to solve (4) directly because (4) is nonconvex. Similar to most fuzzy clustering approaches, alternating optimization(AO) is used to solve (4). In particular, (2) is optimized by fixing V and (4) is optimized by fixing u_{ci}. These two steps are recursively conducted. For optimizing (2) under the constraint of (3), we use Lagrangian Multiplier method to derive the updating rule for this step. The Lagrangian function with V fixed is given as follows:

$$L = \sum_{c=1}^{k} u_{ci}^m \|x_i - v_c\|^2 + \lambda_i (\sum_{c=1}^{k} u_{ci} - 1) \tag{5}$$

where the λ_i is the Lagrangian Multiplier. The updating rule of u_{ci} can be derived as:

$$u_{ci} = \left[\sum_{j=1}^{k} \left(\frac{\|x_i - v_c\|}{\|x_i - v_j\|} \right)^{\frac{2}{m-1}} \right]^{-1} \tag{6}$$

For optimizing (4), inspired by the work in [23], stochastic gradient method is used to approximately minimize (4) by moving the centroid along the negative direction of the gradient of (2). Each centroid v_c is updated after each object is processed whose increment can be calculated as follows:

$$\Delta v_c = \eta_t u_{ci}^m (x_i - v_c) \tag{7}$$

where $u_{ci}^m (x_i - v_c)$ is the negative gradient of (2) and η_t is the learning rate on iteration t which controls the amount of increment. According to [24], η_t is often set to be $\eta_t = \eta_0 / t$, and η_0 is a small positive value. Based on the previous discussion, we now give the algorithm called stochastic gradient based fuzzy clustering(SGFC) as follows. As shown in Algorithm 1, SGFC processes one object at a time. The k centroids for the entire dataset is initialized in step 1. Then after each object coming, the membership u_{ci} of each object and the centroid v_c to the c_{th} cluster are calculated in step 3 and 4 respectively. The steps continue until every object in the dataset is processed and the final set of centroids V is identified.

Algorithm 1: SGFC

Input: Dataset X with size n, Cluster Number k, learning rate η_t

Output: Cluster centroids V

Method:

1 Initialize centroids $\mathbf{v}_c$ by randomly select k objects.
  for $i = 1$ to n
2 **for** $c = 1$ to k
3 Update u_{ci} using equation (6);
4 Update $\mathbf{v}_c$ using equation (7);
  end for
 end for

B. Learning Rate in SGFC

As shown in Algorithm 1, the proper learning rate η_t need to be specified in the beginning. And the value of η_0 is always specified by heuristic. In this paper, a much better way of setting the self-adaptive learning rate is derived which can be determined automatically during the process of gradient-descent based optimization in SGFC. Based on Algorithm 1, when p_{th} object is processed, only the p_{th} column of membership matrix $U_{c \times n}$ is updated and the columns from 1 to $p-1$ are not changed. Under this circumstance, it can be shown as follow that the increment of centroid Δv_c is determined by the new observed data x_{p+1} and the previous

centroid v_c^p. To minimize (1) by using Lagrangian Multiplier method, the updating rule for v_c can be derived as:

$$v_c = \frac{\sum_{i=1}^{n} u_{ci}^m x_i}{\sum_{i=1}^{n} u_{ci}^m} \tag{8}$$

Based on (8), the increment of centroid Δv_c can be written as follows:

$$\begin{aligned}
\Delta v_c = v_c^{p+1} - v_c^p &= \frac{\sum_{i=1}^{p+1} u_{ci}^m x_i}{\sum_{i=1}^{p+1} u_{ci}^m} - \frac{\sum_{i=1}^{p} u_{ci}^m x_i}{\sum_{i=1}^{p} u_{ci}^m} \\
&= \frac{(\sum_{i=1}^{p} u_{ci}^m)(\sum_{i=1}^{p} u_{ci}^m x_i + u_{c(p+1)}^m x_{p+1})}{\sum_{i=1}^{p+1} u_{ci}^m \sum_{i=1}^{p} u_{ci}^m} \\
&\quad - \frac{(\sum_{i=1}^{p} u_{ci}^m x_i)(\sum_{i=1}^{p} u_{ci}^m + u_{c(p+1)}^m)}{\sum_{i=1}^{p+1} u_{ci}^m \sum_{i=1}^{p} u_{ci}^m} \\
&= \frac{(\sum_{i=1}^{p} u_{ci}^m) u_{c(p+1)}^m x_{p+1} - (\sum_{i=1}^{p} u_{ci}^m x_i) u_{c(p+1)}^m}{\sum_{i=1}^{p+1} u_{ci}^m \sum_{i=1}^{p} u_{ci}^m} \\
&= \frac{1}{\sum_{i=1}^{p+1} u_{ci}^m} u_{c(p+1)}^m (x_{p+1} - \frac{\sum_{i=1}^{p} u_{ci}^m x_i}{\sum_{i=1}^{p} u_{ci}^m}) \\
&= \frac{1}{\sum_{i=1}^{p+1} u_{ci}^m} u_{c(p+1)}^m (x_{p+1} - v_c^p)
\end{aligned} \tag{9}$$

From the result of the derivation, it is shown that Δv_c is determined by x_{p+1} and v_c^p which means the centroid v_c can be updated incrementally. Compared with equation (7), the learning rate η_t can be naturally specified as the coefficient $1/\sum_{i=1}^{p+1} u_{ci}^m$ in equation (9). Note that it also can be updated incrementally and maintained automatically since $\sum_{i=1}^{p} u_{ci}^m$ is calculated and stored in previous updating step.

C. Mini-batch SGFC

The performance of SGFC may be affected by stochastic noise, which is caused by the optimization mechanism introduced here. The gradient in SGFC is estimated by processing data as one object at each time. In other words, SGFC updates the centroids by computing a gradient descent step for each object. This may generate lower quality clustering results because of the deviation of gradient estimation. To reduce the influence of stochastic noise, we propose mini-batch SGFC which is given as follows. Here $|M_p|$ is the number of objects in p_{th} mini-batch. As shown in Algorithm 2, instead of processing data as one object at a time, mini-batch SGFC handles the dataset as one mini-batch at a time. After initializing the centroids in step1, mini-batch SGFC first updates the membership u_{ci} of the objects in the current mini-batch in step 3-7. Note that the centroids are cached and not updated in these steps. Then the centroids of all the clusters are updated in step 8-12. The algorithm stops when all the mini-batches have been processed.

Algorithm 2: Mini-batch SGFC

Input: Data of p_{th} mini-batch M_p, Cluster Number k
Output: Cluster centroids V
Method:

1 Initialize centroids $\mathbf{v}_c$ by randomly select k objects .
2 **for** $p = 1$ to P
3 **for** $i = 1$ to $|M_p|$
4 **for** $c = 1$ to k
5 Update u_{ci} using equation (6);
6 **end for**
7 **end for**
8 **for** $i = 1$ to $|M_p|$
9 **for** $c = 1$ to k
10 Update $\mathbf{v}_c$ using equation (9);
11 **end for**
12 **end for**
13 **end for**

D. Multi-Pass Mini-batch SGFC

The SGFC and Mini-batch SGFC are one-pass algorithms which means the dataset is accessed and processed only one time. However, in many applications the datasets are available for being accessed multiple times. The data which is too large for the memory can be stored in the disk and passed into memory in mini-batches. As shown in algorithms 1 and 2, the centroids in V and membership U are updated after each object or each mini-batch coming. The errors caused by the random initial centroids and stochastic noise may be high and not easy to correct in one pass algorithm. We propose multi-pass mini-batch SGFC in which U can be updated based on the V calculated in the previous pass. It is expected that we can achieve better centroids and memberships for the dataset and also get better clustering performance for large data. The clustering performance of the approaches are compared and discussed in next section.

IV. Experimental results

In this section, experimental studies of the proposed approach are presented on synthetic data and two real world datasets including image and document data. Two types of experiments are conducted and reported. First, on synthetic data we compare our derived learning rate with heuristic learning rate to see if using our learning rate produces better clustering results. Meanwhile, we compare SGFC and Mini-batch SGFC to show if mini-batch can help improve the performance of clustering. The multi-pass version of each algorithm is also investigated to see if multiple passes can

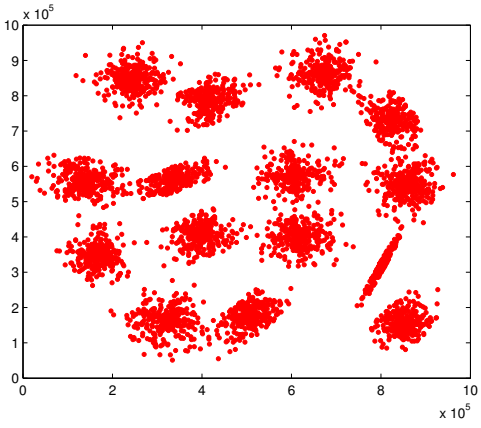


Fig. 1: 2D15 synthetic data

improve the clustering results. Second, we compare Mini-batch SGFC with SPFCM and OFCM which are both fuzzy clustering methods and handling data in mini-batch form to show the effectiveness and efficiency of our approach. The experiments implemented in Matlab were conducted on a PC with four cores of Intel I5-2400 with 24 gigabytes of memory.

A. Dataset

We compare the performance of the algorithms on the following datasets.

2D15[1]: This is a synthetic dataset which is composed of 5000 two dimensional points with 15 classes. The distribution of the points is shown in Fig. 1.

MNIST[2]: This dataset is composed of 10 classes which are 0 to 9 handwritten digit images. There are 70000 28×28 pixel images. We normalize the pixel value to [0,1] by dividing 255 and each image is represented as a 784 dimensional feature vector.

RCV1_5: This dataset is part of RCV1[25] in which we select 5 classes. It composed of 29008 documents with 47236 dimensions.

The basic characteristics of the three datasets are shown in the following Table. I.

TABLE I: Experimental DataSets

Name	No. of Objects	No. of Clusters	Dimension
2D15	5000	15	2
MNIST	70,000	10	784
RCV1_5	29,008	5	47,236

B. Evaluation criteria

Three external metrics *F-measure*, *Normalized Mutual Information*(NMI), and *Adjusted Rand Index*(ARI) are used to evaluate the clustering results, which measure the agreement of cluster results produced by an algorithm and the ground truth. If we refer *class* as the ground truth, and *cluster* as the results of a clustering algorithm, the NMI is calculated as follows:

$$NMI = \frac{\sum_{c=1}^{k}\sum_{p=1}^{m} n_c^p log(\frac{n\cdot n_c^p}{n_c\cdot n_p})}{\sqrt{(\sum_{c=1}^{k} n_c log(\frac{n_c}{n}))(\sum_{p=1}^{m} n_p log(\frac{n_p}{n}))}} \tag{10}$$

where n is the total number of objects, n_c and n_p are the numbers of objects in the c_{th} *cluster* and the p_{th} *class*, respectively, and n_c^p is the number of common objects in *class* p and *cluster* c. For F-measure, the calculation based on precision and recall is as follows:

$$F-measure = \frac{2\cdot \text{precision}\cdot \text{recall}}{\text{precision}+\text{recall}} \tag{11}$$

where,

$$\text{precision} = \frac{n_c^p}{n_c} \tag{12}$$

$$\text{recall} = \frac{n_c^p}{n_p} \tag{13}$$

Adjusted Rand Index(ARI) [26] is an adjusted form of Rand Index which is a measure of the similarity between two clustering results. The calculation of ARI is as follows:

$$ARI = \frac{\sum_{cp}\binom{n_c^p}{2} - \sum_c \binom{q_c}{2}\sum_p \binom{s_p}{2}/\binom{n}{2}}{\frac{1}{2}(\sum_c \binom{q_c}{2} + \sum_p \binom{s_p}{2}) - \sum_c \binom{q_c}{2}\sum_p \binom{s_p}{2}/\binom{n}{2}} \tag{14}$$

where, $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ is the binomial coefficient and

$$q_c = \sum_p n_c^p, \qquad s_p = \sum_c n_c^p \tag{15}$$

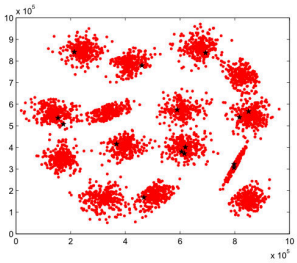
All the three criterions reflect better clustering results if the values of them are higher. The clustering result is same as the ground truth if their values equal to 1.

C. Results on 2D15

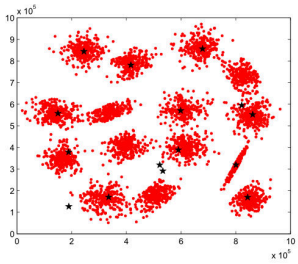
To better show and compare the effectiveness of different algorithms and provide the visual view of the clustering results, we first conduct experiments on the synthetic dataset 2D15 as shown in Fig. 1. For SGFC, two kinds of experiments are conducted. First, we compare the effectiveness of two kinds of learning rate discussed in previous section to show if our derived learning rate is better than the heuristic one. We also compare SGFC with mini-batch SGFC to see if mini-batch SGFC can improve the clustering performance. Second, multiple-pass SGFC is compared with one-pass SGFC to see if multiple-pass improves the clustering results. To fairly compare their performances, we initialize the centroids with same set of points for each algorithm as shown in Fig. 2(a) and the data is drawn in the same order. According to [24], the learning rate is often set to be $\eta_t = \eta_0/t$ in which t is the iteration number. We try η_0 with different values in the range [1,50] and select the value which produces the best clustering result. Based on the experimental

[1]This dataset was designed by Ilia Sidoroff and can be downloaded on http://www.uef.fi/en/sipu/datasets.

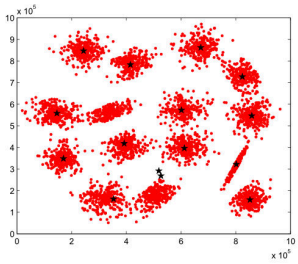
[2]This dataset can be downloaded on http://yann.lecun.com/exdb/mnist/.



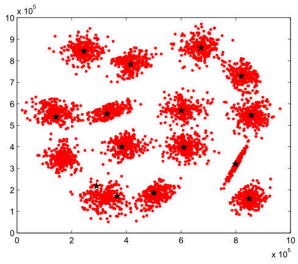
(a) 2D15 synthetic data with the random initialized centroids



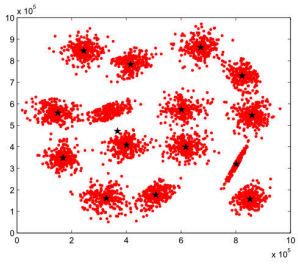
(b) SGFC with learning rate $\eta_t = 16/t$



(c) SGFC with derived learning rate



(d) Mini-batch SGFC with learning rate $\eta_t = 35/t$



(e) Mini-batch SGFC with derived learning rate

Fig. 2: (a) shows the same initialized centroids for different algorithms. (b), (c), (d), (e) show the final identified centroids of the four algorithms respectively. All the centroids are marked as pentagrams

results, the best values $\eta_t = 16/t$ and $\eta_t = 35/t$ are selected for SGFC and Mini-batch SGFC respectively. For Mini-batch SGFC, the mini-batch size is set to be 1% of size of the entire dataset. For each algorithm the same value of fuzzifier is set to be $m = 1.7$ for this dataset. The clustering results of one-pass algorithms on the dataset are shown in Fig. 2 and Table. II. By comparing sub-figures in Fig. 2 horizontally that is comparing Fig. 2(b) with (c) and (d) with (e) we can see that our derived learning rate identifies better centroids for the 15 classes than the selected learning rate η_t on both SGFC and Mini-batch SGFC. The results of F-measure, NMI and ARI in Table. II also show our derived learning rate produces better clustering results. While by comparing sub-figures in Fig. 2 vertically, we can see that Mini-batch SGFC always performs better than SGFC no matter what kind of learning rate used. The same property also shows in Table. II.

TABLE II: Results on 2D15 data with one-pass

Algorithm	F-measure	NMI	ARI
SGFC, $\eta_t = 16/t$	0.9065	0.9184	0.8440
SGFC, derived learning rate	0.9352	0.9497	0.8967
Mini-batch SGFC, $\eta_t = 35/t$	0.9474	0.9513	0.9124
Mini-batch SGFC, derived learning rate	0.9878	0.9793	0.9745

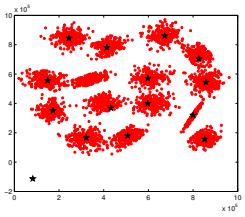
Next, we show the results of multiple-pass SGFC in Fig. 3 and Table. III. For 2D15 dataset, two-pass SGFC is conducted because the properties of the algorithms already shown apparently. From the results, we can easily see that the influence of multiple-pass to the algorithms with different learning rate is opposite. Note that in Fig. 3(a), (c), instead of improving the clustering performance, multiple-pass deteriorates the results with one identified centroid apart far from the correct position. While, the algorithms equipped with our derived leaning rate are able to take advantage of the multiple-pass and achieve the ideal set of centroids as shown in Fig. 3(b) and (d). And the identified centroids in (d) is slightly better than in (b) which is also shown in Table. III in which multiple-pass mini-batch SGFC achieve the best results.

TABLE III: Results on 2D15 data with Multiple-pass(two pass)

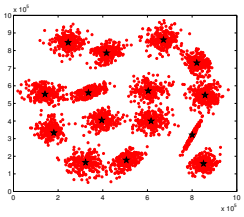
Algorithm	F-measure	NMI	ARI
SGFC, $\eta_t = 16/t$	0.9372	0.9512	0.9054
SGFC, derived learning rate	0.9934	0.9858	0.9859
Mini-batch SGFC, $\eta_t = 35/t$	0.9399	0.9554	0.9063
Mini-batch SGFC, derived learning rate	0.9938	0.9867	0.9868

D. Results on MNIST and RCV1_5

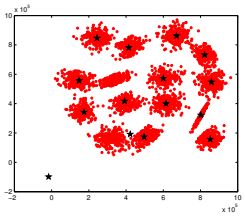
In this section, we compare mini-batch SGFC with two related fuzzy clustering SPFCM and OFCM on two large real world datasets. The fuzzifier m is set to be 1.7 for all the approaches. Cluster number k is set to be 10 and 5 for each mini-batch of MNIST and RCV1_5 respectively. We set mini-batch size as 1%, 2.5%, 5%, 10% and 25% of size of the entire dataset. The clustering results on MNIST are shown in Table. IV and Table. V respectively. Note



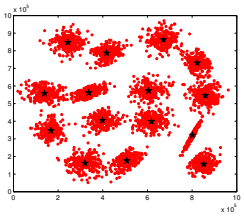
(a) SGFC with learning rate $\eta_t = 16/t$



(b) SGFC with derived learning rate



(c) Mini-batch SGFC with learning rate $\eta_t = 35/t$



(d) Mini-batch SGFC with derived learning rate

Fig. 3: (a), (b), (c), (d) show the final identified centroids of the four algorithms with multiple-pass(two-pass) on the data respectively. All the centroids are marked as pentagrams

that the results of mini-batch SGFC in Table. IV is based on single pass on the dataset. From the table we can see that the value of F-measure, NMI and ARI of mini-batch SGFC is comparable with SPFCM and OFCM. The time spent by mini-batch SGFC is much less than SPFCM and OFCM. To show the effectiveness of multi-pass mini-batch SGFC, the experiments are conducted with different passes on MINST dataset. The results of 1-pass, 3-pass and 5-pass are shown in Table. V. It is shown that the accuracy of the clustering results improves as the number of passes increases. The algorithm will certainly spend more time when the number of passes increases. However the total time spent shows that mini-batch SGFC with 5-pass on the data is still much faster than SPFCM and OFCM. Compared with SPFCM and OFCM, the time reduction of 5-pass mini-batch SGFC on average time over all the mini-batch sizes are 92.2% and 86.8%. In other words, the time spent by 5-pass mini-batch SGFC is less than $1/12$ and $1/7$ of SPFCM and OFCM respectively. Table. VI shows the results for RCV1_5 dataset and mini-batch SGFC is conducted with single pass on the dataset. As shown in Table. VI, mini-batch SGFC is much faster than SPFCM and OFCM with all the mini-batch sizes. More importantly, the clustering performance of mini-batch SGFC is much better than SPFCM and OFCM. Compared with SPFCM on the three evaluation criterions(F-measure, NMI and ARI), the improvement of mini-batch SGFC on average results over all mini-batch sizes are 49.1%, 161.2% and 179.8%, respectively. Compared with OFCM, the improvement of mini-batch SGFC are 82.2%, 573.7% and 658.1%, respectively.

TABLE IV: Results on MNIST dataset

(a) F-measure

Mini-batch size	Algorithm		
	SPFCM	OFCM	MiniBatch-SGFC
1%	0.57 ± 0.03	0.57 ± 0.01	0.53 ± 0.05
2.5%	0.57 ± 0.01	0.56 ± 0.01	0.53 ± 0.03
5%	0.57 ± 0.03	0.56 ± 0.01	0.52 ± 0.04
10%	0.57 ± 0.02	0.56 ± 0.02	0.50 ± 0.04
25%	0.56 ± 0.02	0.57 ± 0.01	0.50 ± 0.03

(b) NMI

Mini-batch size	Algorithm		
	SPFCM	OFCM	MiniBatch-SGFC
1%	0.48 ± 0.02	0.48 ± 0.01	0.45 ± 0.03
2.5%	0.48 ± 0.01	0.48 ± 0.01	0.44 ± 0.02
5%	0.48 ± 0.02	0.48 ± 0.01	0.43 ± 0.03
10%	0.48 ± 0.01	0.48 ± 0.01	0.42 ± 0.03
25%	0.48 ± 0.02	0.47 ± 0.01	0.41 ± 0.02

(c) ARI

Mini-batch size	Algorithm		
	SPFCM	OFCM	MiniBatch-SGFC
1%	0.37 ± 0.03	0.36 ± 0.01	0.33 ± 0.04
2.5%	0.37 ± 0.02	0.36 ± 0.01	0.33 ± 0.03
5%	0.37 ± 0.03	0.36 ± 0.01	0.32 ± 0.04
10%	0.37 ± 0.02	0.36 ± 0.01	0.31 ± 0.04
25%	0.36 ± 0.02	0.36 ± 0.01	0.30 ± 0.03

(d) Time

Mini-batch size	Algorithm		
	SPFCM	OFCM	MiniBatch-SGFC
1%	159.3 ± 31.4	357.0 ± 29.1	$\mathbf{9.7 \pm 0.2}$
2.5%	340.2 ± 73.2	444.7 ± 43.5	$\mathbf{9.7 \pm 0.1}$
5%	561.7 ± 104.4	363.4 ± 65.3	$\mathbf{9.7 \pm 0.2}$
10%	799.2 ± 219.6	358.6 ± 136.8	$\mathbf{9.9 \pm 0.2}$
25%	1241.9 ± 249.1	310.6 ± 157.0	$\mathbf{10.0 \pm 0.1}$

V. Conclusions

We have proposed a new stochastic gradient based fuzzy clustering approach called SGFC including its mini-batch and multi-pass version for large data analysis, and apply mini-batch SGFC on large real world datasets to demonstrate its effectiveness and scalability. Mini-batch SGFC processes large data by considering the data as coming mini-batch by mini-batch. Instead of using heuristic learning rate, an adaptive learning rate which can be updated and maintained automatically is derived for SGFC in this paper. Experimental results on a synthetic dataset show that our derived adaptive learning rate helps to achieve better clustering results than heuristic learning rate. Experiments conducted on two large datasets show that mini-batch SGFC outperforms related incremental algorithms with much less time consumption but comparable or higher clustering accuracy. The merits shown in the experiments indicate that SGFC including its mini-batch and multi-pass version has a great potential to be used for large data clustering.

References

[1] A. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.

TABLE V: Results of Mini-batch SGFC with different passes on MNIST dataset

(a) F-measure

Mini-batch size	MiniBatch-SGFC		
	1-pass	3-pass	5-pass
1%	0.53 ± 0.05	0.55 ± 0.04	0.57 ± 0.02
2.5%	0.53 ± 0.03	0.55 ± 0.04	0.57 ± 0.03
5%	0.52 ± 0.04	0.55 ± 0.03	0.56 ± 0.03
10%	0.50 ± 0.04	0.55 ± 0.05	0.56 ± 0.03
25%	0.50 ± 0.03	0.54 ± 0.03	0.56 ± 0.03

(b) NMI

Mini-batch size	MiniBatch-SGFC		
	1-pass	3-pass	5-pass
1%	0.45 ± 0.03	0.46 ± 0.02	0.48 ± 0.02
2.5%	0.44 ± 0.02	0.47 ± 0.03	0.48 ± 0.02
5%	0.43 ± 0.03	0.46 ± 0.02	0.47 ± 0.02
10%	0.42 ± 0.03	0.46 ± 0.03	0.48 ± 0.02
25%	0.41 ± 0.02	0.45 ± 0.02	0.47 ± 0.02

(c) ARI

Mini-batch size	MiniBatch-SGFC		
	1-pass	3-pass	5-pass
1%	0.33 ± 0.04	0.35 ± 0.03	0.36 ± 0.02
2.5%	0.33 ± 0.03	0.35 ± 0.03	0.36 ± 0.03
5%	0.32 ± 0.04	0.35 ± 0.03	0.36 ± 0.03
10%	0.31 ± 0.04	0.35 ± 0.04	0.36 ± 0.03
25%	0.30 ± 0.03	0.33 ± 0.03	0.36 ± 0.03

(d) Time

Mini-batch size	MiniBatch-SGFC		
	1-pass	3-pass	5-pass
1%	9.7 ± 0.2	29.1 ± 0.3	47.4 ± 0.7
2.5%	9.7 ± 0.1	28.3 ± 0.3	48.3 ± 0.7
5%	9.7 ± 0.2	29.6 ± 0.7	48.2 ± 0.7
10%	9.9 ± 0.2	30.8 ± 0.6	49.1 ± 0.5
25%	10.0 ± 0.1	30.5 ± 0.4	49.1 ± 0.6

TABLE VI: Results on RCV1_5 dataset

(a) F-measure

Mini-batch size	Algorithm		
	SPFCM	OFCM	MiniBatch-SGFC
1%	0.47 ± 0.07	0.33 ± 0.01	$\mathbf{0.70} \pm 0.07$
2.5%	0.43 ± 0.06	0.32 ± 0.02	$\mathbf{0.65} \pm 0.04$
5%	0.46 ± 0.04	0.35 ± 0.04	$\mathbf{0.69} \pm 0.09$
10%	0.45 ± 0.06	0.44 ± 0.02	$\mathbf{0.71} \pm 0.08$
25%	0.45 ± 0.05	0.41 ± 0.04	$\mathbf{0.62} \pm 0.08$

(b) NMI

Mini-batch size	Algorithm		
	SPFCM	OFCM	MiniBatch-SGFC
1%	0.23 ± 0.08	0.01 ± 0.004	$\mathbf{0.53} \pm 0.09$
2.5%	0.18 ± 0.04	0.02 ± 0.01	$\mathbf{0.46} \pm 0.05$
5%	0.20 ± 0.04	0.05 ± 0.04	$\mathbf{0.57} \pm 0.08$
10%	0.18 ± 0.06	0.17 ± 0.04	$\mathbf{0.56} \pm 0.10$
25%	0.19 ± 0.06	0.13 ± 0.04	$\mathbf{0.44} \pm 0.09$

(c) ARI

Mini-batch size	Algorithm		
	SPFCM	OFCM	MiniBatch-SGFC
1%	0.19 ± 0.08	0.01 ± 0.002	$\mathbf{0.52} \pm 0.09$
2.5%	0.15 ± 0.05	0.01 ± 0.004	$\mathbf{0.42} \pm 0.06$
5%	0.18 ± 0.04	0.03 ± 0.01	$\mathbf{0.50} \pm 0.09$
10%	0.17 ± 0.07	0.15 ± 0.04	$\mathbf{0.53} \pm 0.10$
25%	0.15 ± 0.05	0.11 ± 0.04	$\mathbf{0.38} \pm 0.09$

(d) Time

Mini-batch size	Algorithm		
	SPFCM	OFCM	MiniBatch-SGFC
1%	1019.3 ± 10.3	561.1 ± 2.3	$\mathbf{263.7} \pm \mathbf{3.0}$
2.5%	1216.1 ± 43.7	500.4 ± 4.2	$\mathbf{202.8} \pm \mathbf{3.7}$
5%	1422.1 ± 64.5	495.8 ± 5.4	$\mathbf{177.1} \pm \mathbf{0.5}$
10%	1664.7 ± 157.9	478.9 ± 2.5	$\mathbf{177.5} \pm \mathbf{1.1}$
25%	2061.9 ± 301.2	507.6 ± 22.9	$\mathbf{202.3} \pm \mathbf{2.2}$

[2] M. Filippone, F. Camastra, F. Masulli, and S. Rovetta, "A survey of kernel and spectral methods for clustering," *Pattern recognition*, vol. 41, no. 1, pp. 176–190, 2008.

[3] R. Xu, D. Wunsch *et al.*, "Survey of clustering algorithms," *Neural Networks, IEEE Transactions on*, vol. 16, no. 3, pp. 645–678, 2005.

[4] L. Kaufman and P. J. Rousseeuw, *Finding groups in data: an introduction to cluster analysis*. Wiley-Interscience, 2009, vol. 344.

[5] S. Guha, R. Rastogi, and K. Shim, "Cure: an efficient clustering algorithm for large databases," in *ACM SIGMOD Record*, vol. 27, no. 2. ACM, 1998, pp. 73–84.

[6] R. T. Ng and J. Han, "Efficient and effective clustering methods for spatial data mining," in *Proceedings of the 20th International Conference on Very Large Data Bases*, ser. VLDB '94. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1994, pp. 144–155. [Online]. Available: http://dl.acm.org/citation.cfm?id=645920.672827

[7] T. Livny, "Birch: an efficient data clustering method for very large databases," in *ACM SIGMOD international Conference on Management of Data*, vol. 1, 1996, pp. 103–114.

[8] A. Ene, S. Im, and B. Moseley, "Fast clustering using mapreduce," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2011, pp. 681–689.

[9] R. L. Ferreira Cordeiro, C. Traina Junior, A. J. Machado Traina, J. López, U. Kang, and C. Faloutsos, "Clustering very large multi-dimensional datasets with mapreduce," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2011, pp. 690–698.

[10] R. Chitta, R. Jin, T. Havens, and A. Jain, "Approximate kernel k-means: Solution to large scale kernel clustering," in *Proc. ACM SIGKDD*, 2011, pp. 551–556.

[11] R. Chitta, R. Jin, and A. K. Jain, "Efficient kernel clustering using random fourier features," in *Data Mining (ICDM), 2012 IEEE 12th International Conference on*. IEEE, 2012, pp. 161–170.

[12] F. Can, "Incremental clustering for dynamic information processing," *ACM Transactions on Information Systems (TOIS)*, vol. 11, no. 2, pp. 143–164, 1993.

[13] F. Can, E. A. Fox, C. D. Snavely, and R. K. France, "Incremental clustering for very large document databases: Initial marian experience," *Information sciences*, vol. 84, no. 1, pp. 101–114, 1995.

[14] H. Ning, W. Xu, Y. Chi, Y. Gong, and T. Huang, "Incremental spectral clustering by efficiently updating the eigen-system," *Pattern Recognition*, vol. 43, no. 1, pp. 113–127, 2010.

[15] Y. Wang, L. Chen, and J.-P. Mei, "Incremental fuzzy clustering with multiple medoids for large data," *Fuzzy Systems, IEEE Transactions on*, accepted.

[16] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. Norwell, MA, USA: Kluwer Academic Publishers, 1981.

[17] H.-C. Huang, Y.-Y. Chuang, and C.-S. Chen, "Multiple kernel fuzzy clustering," *Fuzzy Systems, IEEE Transactions on*, vol. 20, no. 1, pp. 120–134, 2012.

[18] J.-P. Mei and L. Chen, "A fuzzy approach for multitype relational data clustering," *Fuzzy Systems, IEEE Transactions on*, vol. 20, no. 2, pp. 358–371, 2012.

[19] L. F. Coletta, L. Vendramin, E. R. Hruschka, R. J. Campello, and W. Pedrycz, "Collaborative fuzzy clustering algorithms: Some refinements and design guidelines," *Fuzzy Systems, IEEE Transactions on*, vol. 20, no. 3, pp. 444–462, 2012.

[20] P. Hore, L. Hall, and D. Goldgof, "Single pass fuzzy c means," in *Fuzzy Systems Conference, 2007. FUZZ-IEEE 2007. IEEE International*. IEEE, 2007, pp. 1–7.

[21] P. Hore, L. Hall, D. Goldgof, and W. Cheng, "Online fuzzy c means,"

in *Fuzzy Information Processing Society, 2008. NAFIPS 2008. Annual Meeting of the North American*. IEEE, 2008, pp. 1–5.

[22] L. Bottou and O. Bousquet, "The tradeoffs of large scale learning," in *Advances in Neural Information Processing Systems*, J. Platt, D. Koller, Y. Singer, and S. Roweis, Eds. Cambridge, MA: MIT Press, 2008, vol. 20, pp. 161–168.

[23] L. Bottou, "Online learning and stochastic approximations," *On-line learning in neural networks*, vol. 17, p. 9, 1998.

[24] T. Kohonen, "Self-organization and associative memory," *Self-Organization and Associative Memory, Springer Series in Information Sciences, volume 8*, vol. 1, 1988.

[25] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li, "Rcv1: A new benchmark collection for text categorization research," *The Journal of Machine Learning Research*, vol. 5, pp. 361–397, 2004.

[26] L. Hubert and P. Arabie, "Comparing partitions," *Journal of classification*, vol. 2, no. 1, pp. 193–218, 1985.