K-MEAP: Generating Specified K Clusters with Multiple Exemplars by Efficient Affinity Propagation

Wang, Yangtao; Chen, Lihui

2014

Wang, Y & Chen, L. (2014). K-MEAP: Generating Specified K Clusters with Multiple Exemplars by Efficient Affinity Propagation. 2014 IEEE International Conference on Data Mining (ICDM), 1091-1096.

<https://hdl.handle.net/10356/81840>

<https://doi.org/10.1109/ICDM.2014.54>

© 2014 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. The published version is available at: [<http://dx.doi.org/10.1109/ICDM.2014.54>].

Downloaded on 26 Apr 2023 21:16:33 SGT

K-MEAP: Generating Specified K Clusters with Multiple Exemplars by Efficient Affinity Propagation

Yangtao Wang, Lihui Chen

School of Electrical and Electronic Engineering

Nanyang Technological University

Republic of Singapore

Email: wang0689@e.ntu.edu.sg, elhchen@ntu.edu.sg

Abstract—Recently, an attractive clustering approach named multi-exemplar affinity propagation (MEAP) has been proposed as an extension to the single exemplar based Affinity Propagation (AP). MEAP is able to automatically identify multiple exemplars for each cluster associated with a superexemplar. However, if the cluster number is a prior knowledge and can be specified by the user, MEAP is unable to make use of such knowledge directly in its learning process. Instead it has to rely on re-running the process as many times as it takes by tuning parameters until it generates the desired number of clusters. The process of MEAP re-running may be very time consuming. In this paper, we propose a new clustering algorithm called K-MEAP which is able to generate specified K clusters directly while retaining the advantages of MEAP. Two kinds of new additional messages are introduced in MEAP in order to control the number of clusters in the process of message passing. The detailed problem formulation, the derived updating rules for passing messages, and the in-depth analysis of the proposed K-MEAP are provided. Experimental studies demonstrated that K-MEAP not only generates K clusters directly and efficiently without tuning parameters, but also outperforms related approaches in terms of clustering accuracy.

I. INTRODUCTION

Clustering is an important unsupervised learning technique to find the pattern structure and information underlying the unlabelled data. It attracts much attention of the researchers from, to name a few, the areas of machine learning, data mining, information retrieval and knowledge discovery. Many different clustering algorithms based on various theories have been developed and successfully applied in a wide range of applications over the past decades [1], [2], [3]. The Affinity Propagation (AP) [4] clustering algorithm groups the data objects by identifying the most representative objects of the data set called exemplars based on the similarity between each pair of objects. AP has been widely used in many applications because of its efficiency and insensitiveness to initialization. However, identifying single exemplar for each cluster in AP may not be sufficient enough to capture the data structure hidden in the cluster. Moreover, in many applications, each cluster of a data set may have multiple subclusters. For example, the same person(cluster) could have different facial expressions(subclusters) and the same hand written digit could have different writing styles. To solve this kind of problem, recently a new approach called multi-exemplar affinity propagation (MEAP) [5] has been proposed. MEAP is able to identify multiple exemplars to represent the subclusters in each cluster automatically. Meanwhile each exemplar is assigned to its superexemplar. With these advantages, MEAP

achieves significant improvements in the applications of image categorization and handwritten digits clustering [5].

However, one drawback of both AP and MEAP is that they are not able to generate specified number of clusters directly when the number of clusters K is available. In many applications, the number of clusters is a prior knowledge or can be specified by the user. To achieve the desired number of clusters, the suitable parameters named preference and superexemplar preference are needed to be found for AP and MEAP respectively. Bisection method is suggested to use in AP to find a suitable preference for the specified cluster number [4]. The process of finding the parameter may be very time consuming because each change of the parameter will cause the algorithm re-run once. K-AP [6] has therefore been developed to address the issue. K-AP is able to directly generate K clusters as user specified by adding one constraint in the process of message passing to confine the number of clusters to be K.

Inspired by the above idea, in this paper we propose a new clustering algorithm called K-MEAP which is able to directly generate K clusters specified by the user while retaining the advantages of MEAP. Same as MEAP, K-MEAP identifies multiple exemplars automatically for each cluster with no need to specify the number of exemplars. Different from MEAP, one more constraint which includes two new kinds of messages is added into the objective function to ensure the number of clusters to be K. Max-sum (the *log*-domain max-product) belief propagation algorithm [7], [8], [5] is applied to derive the updating rules of the messages passing on the factor graph [9]. The experiments of K-MEAP show that K-MEAP generates K clusters directly without any parameter tuning and also performs better than related algorithms in terms of clustering accuracy. Moreover, given a specified cluster number, K-MEAP is much more efficient than re-runned MEAP.

The rest of the paper is organized as follows: in the next section, the details of the proposed clustering approach K-MEAP are described including the objective function, the derived rules for passing messages and the algorithm of K-MEAP. Experiments on two image data sets are conducted and the results are presented and analyzed in section III. Finally, conclusions are drawn in section IV.

II. THE PROPOSED APPROACH

A. Objective Function of K-MEAP

The objective function of K-MEAP is formulated as follows:

$$J_{K-MEAP} = \sum_{i=1}^N \sum_{j=1}^N S_{ij}(b_{ij}) + \sum_{i=1}^N H_i(b_{i1}, \dots, b_{iN}) \\ + \sum_{j=1}^N E_j(b_{1j}, \dots, b_{Nj}) + \sum_{k=1}^N F_k(b_{11}, \dots, b_{NN}) \\ + G(b_{11}, \dots, b_{NN}) \quad (1)$$

Where b_{ij} is the element of an assignment matrix B . The non-diagonal and diagonal elements of B are defined respectively as follows:

$$b_{ij} = \begin{cases} 1 & \text{if } j \text{ is an exemplar of } i \\ 0 & \text{otherwise} \end{cases} \quad \forall i \neq j \quad (2)$$

$$b_{ii} = \begin{cases} k \in \{1, \dots, N\} & \text{if } k \text{ is a superexemplar of an exemplar } i \\ 0 & \text{if } i \text{ is not an exemplar} \end{cases} \quad (3)$$

S is a function matrix $[S_{ij}(b_{ij})]_{N \times N}$ shown as follows:

$$S_{ij}(b_{ij}) = \begin{cases} s_{ij}, & \text{if } i \neq j \text{ and } b_{ij} \neq 0 \\ s_{ii} + l_{ib_{ii}} & \text{if } i = j \text{ and } b_{ii} \notin \{0, i\} \\ s_{ii} & \text{if } i = j \text{ and } b_{ii} = i \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

Here $[s_{ij}]_{N \times N}$ is defined as the similarity matrix in which each element s_{ij} measures the similarity between an object i and object j , and $[l_{ij}]_{N \times N}$ is referred to as a linkage matrix in which each element l_{ij} measures the linkage of an exemplar i and its superexemplar j . The above definition of S aims to integrate two kinds of information as follows:

$$S = S_1 + S_2 = \sum_{i=1}^N \sum_{j=1}^N s_{ij} \cdot [b_{ij} \neq 0] + \sum_{i=1}^N l_{ib_{ii}} [b_{ii} \notin \{0, i\}] \quad (5)$$

Here, S_1 represents the sum of the similarities between all the data objects and its exemplars. S_2 represents the sum of the linkages between all the exemplars and its superexemplars. $[\cdot]$ is the Iverson notation with $[true] = 1$ and $[false] = 0$. Note that our definition of S is different from MEAP [5]. In MEAP, a suitable superexemplar preference l_{ii} need to be found to achieve the specified cluster number. While, in K-MEAP l_{ii} is not defined and used, instead the added constraint in message passing will help to generate specified cluster number automatically.

The objective of K-MEAP is to maximize S with assignment matrix B satisfying the four constraints as follows:

Constraint $H_i(b_{i1}, \dots, b_{iN})$ enforces that each object i only has one exemplar:

$$H_i(b_{i1}, \dots, b_{iN}) = \begin{cases} -\infty & \text{if } \sum_{j=1}^N [b_{ij} \neq 0] \neq 1, \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

Constraint $E_j(b_{1j}, \dots, b_{Nj})$ enforces that if an object j is selected as the exemplar of any other object i , then the object j must be an exemplar:

$$E_j(b_{1j}, \dots, b_{Nj}) = \begin{cases} -\infty & \text{if } b_{jj} = 0 \text{ but } \exists i : b_{ij} = 1 \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

Constraint $F_k(b_{11}, \dots, b_{NN})$ enforces that if an exemplar k is selected as the superexemplar of any other exemplar i , i.e., $b_{ii} = k$, then the exemplar k must be a superexemplar, i.e., $b_{kk} = k$:

$$F_k(b_{11}, \dots, b_{NN}) = \begin{cases} -\infty & \text{if } b_{kk} \neq k \text{ but } \exists i : b_{ii} = k \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

Constraint $G(b_{11}, \dots, b_{NN})$ restricts the number of clusters(superexemplars) to be the specific number K :

$$G(b_{11}, \dots, b_{NN}) = \begin{cases} -\infty & \text{if } \sum_{i=1}^N [b_{ii} = i] \neq K, \\ 0 & \text{otherwise.} \end{cases} \quad (9)$$

Note that the last constraint differentiates K-MEAP from MEAP. It ensures K-MEAP generate specified K clusters directly without tuning parameters.

B. Optimization based on Belief Propagation

Similar as MEAP and K-AP, the problem of maximizing the objective function J_{K-MEAP} can be represented by a factor graph [9] which is constructed by adding the constraint G onto the factor graph of MEAP [5]. Max-sum belief propagation algorithm is applied on the factor graph to maximize the objective function of K-MEAP. The following discussions are about the new messages of K-MEAP, the derived rules for message passing and the generation of assignment matrix.

1) *Messages of K-MEAP*: As shown in Fig. 1(a),(b), there are totally 14 types of messages passing in the factor graph of K-MEAP. Fig. 1(a) shows the messages on a variable node $b_{ij}, i \neq j$ which is the nondiagonal element of assignment matrix B . And Fig. 1(b) shows the messages on a variable node $b_{ii}, i = j$ which is the diagonal element of B . As we add the constraint G on variable node b_{ii} , two messages named ψ_{ii} and λ_{ii} are introduced consequently. These two messages are the key differences compared to MEAP to enable K-MEAP generate specified K clusters directly.

Fig. 1: Messages of K-MEAP passing between variable nodes and function nodes

Based on the working principle of max-sum belief propagation algorithm [5], the messages in Fig. 1(b) are calculated as follows. Noted that here we only show the messages in Fig. 1(b) as the messages in Fig. 1(a) are same as in MEAP. Let q denote $b_{ii} \in \{0, 1, \dots, N\}$:

$$\theta_{ii}(q) = S_{ii}(q) \quad (10)$$

$$\rho_{ii}(q) = \theta_{ii}(q) + \eta_{ii}(q) + \lambda_{ii}(q) + \sum_{k=1}^N \gamma_{ik}(q) \quad (11)$$

$$\alpha_{ii}(q) = \max_{\substack{b_{1i}, \dots, b_{i-1,i}, \\ b_{i+1,i}, \dots, b_{Ni}}} [E_i(b_{1i}, \dots, b_{ii} = q, \dots, b_{Ni}) + \sum_{i' \neq i} \rho_{i'i}(b_{i'i'})] \quad (12)$$

$$\beta_{ii}(q) = \theta_{ii}(q) + \alpha_{ii}(q) + \lambda_{ii}(q) + \sum_{k=1}^N \gamma_{ik}(q) \quad (13)$$

$$\eta_{ii}(q) = \max_{\substack{b_{1i}, \dots, b_{i-1,i}, \\ b_{i+1,i}, \dots, b_{Ni}}} [H_i(b_{1i}, \dots, b_{ii} = q, \dots, b_{Ni}) + \sum_{i' \neq i} \beta_{i'i}(b_{i'i'})] \quad (14)$$

$$\phi_{ik}(q) = \theta_{ii}(q) + \alpha_{ii}(q) + \eta_{ii}(q) + \lambda_{ii}(q) + \sum_{k': k' \neq k} \gamma_{ik'}(q) \quad (15)$$

$$\gamma_{ik}(q) = \max_{\substack{b_{11}, \dots, b_{i-1,i-1}, \\ b_{i+1,i+1}, \dots, b_{NN}}} [F_k(b_{11}, \dots, b_{ii} = q, \dots, b_{NN}) + \sum_{i' \neq i} \phi_{i'k}(b_{i'i'})] \quad (16)$$

$$\psi_{ii}(q) = \theta_{ii}(q) + \alpha_{ii}(q) + \eta_{ii}(q) + \sum_{k=1}^N \gamma_{ik}(q) \quad (17)$$

$$\lambda_{ii}(q) = \max_{\substack{b_{11}, \dots, b_{i-1,i-1}, \\ b_{i+1,i+1}, \dots, b_{NN}}} [G(b_{11}, \dots, b_{ii} = q, \dots, b_{NN}) + \sum_{i' \neq i} \psi_{i'i'}(b_{i'i'})] \\ = \begin{cases} \max_{T=U^{K-1}(\{1, \dots, N\} \setminus \{i\})} \left(\sum_{j \in T} \psi_{jj}(j) \right) \\ + \sum_{\substack{i' \in \{1, \dots, N\} \setminus \{T, i\}, \\ b_{i'i'} \in \{0, \dots, N\} \setminus \{i'\}}} \psi_{i'i'}(b_{i'i'}) & q = i \\ \max_{T=U^K(\{1, \dots, N\} \setminus \{i\})} \left(\sum_{j \in T} \psi_{jj}(j) \right) \\ + \sum_{\substack{i' \in \{1, \dots, N\} \setminus \{T, i\}, \\ b_{i'i'} \in \{0, \dots, N\} \setminus \{i'\}}} \psi_{i'i'}(b_{i'i'}) & q \neq i \end{cases} \quad (18)$$

In (18), $U^{K-1}(\{1, \dots, N\})$ denotes the subset of $\{1, \dots, N\}$ with number of $K-1$ different elements. $U^K(\{1, \dots, N\})$ denotes the subset of $\{1, \dots, N\}$ with number of K different elements. $\lambda_{ii}(q)$ and $\psi_{ii}(q)$ are the two messages between constraint node G and variable node b_{ii} to help to generate specified K clusters (superexemplars). When $q = i$ which means object i is already a superexemplar. Therefore only $K-1$ objects are needed to select as superexemplars to form K clusters. Based on the same principle, K superexemplars are needed to select when $q \neq i$. Note that the selection of those superexemplars must satisfy the maximization condition as given in (18).

To simplify the message passing, we adopt the derivation methods applied in [5] and [4] by only focusing on the scale differences of messages. The simplified rules for message passing can be given as follows: For $i \neq j$

$$\begin{aligned} \tilde{\rho}_{ij} &= \rho_{ij}(1) - \rho_{ij}(0) \\ &= s_{ij} - \max_{j' \notin \{j, i\}} [s_{ij'} + \tilde{\alpha}_{ij'}], \\ &\quad \max_{q \in \{1, \dots, N\} \setminus \{i\}} (l_{iq} + \tilde{\gamma}_{iq}) + s_{ii} + \tilde{\alpha}_{ii}, s_{ii} + \tilde{\alpha}_{ii} + \tilde{\lambda}_{ii} + \tilde{\gamma}_{ii} \end{aligned} \quad (19)$$

$$\tilde{\alpha}_{ij} = \alpha_{ij}(1) - \alpha_{ij}(0) = \min[0, \max_{q \in \{1, \dots, N\}} [\tilde{\rho}_{ij}^q + \sum_{i' \notin \{i, j\}} [\tilde{\rho}_{i'j}, 0]]] \quad (20)$$

$$\forall i = 1, \dots, N, k = 1, \dots, N$$

$$\begin{aligned} \tilde{\rho}_i^k &= \rho_{ii}(k) - \rho_{ii}(0) \\ &= \begin{cases} s_{ii} + l_{ik} - \max_{i' \neq i} (s_{ii'} + \tilde{\alpha}_{ii'}) + \tilde{\gamma}_{ik} & k \neq i \\ s_{ii} - \max_{i' \neq i} (s_{ii'} + \tilde{\alpha}_{ii'}) + \tilde{\lambda}_{ii} + \tilde{\gamma}_{ii} & k = i \end{cases} \end{aligned} \quad (21)$$

$$\tilde{\alpha}_{ii} = \alpha_{ii}(k) - \alpha_{ii}(0) = \sum_{i' \neq i} \max[\tilde{\rho}_{i'i}, 0] \quad (22)$$

$$\begin{aligned} \tilde{\phi}_{ik} &= \phi_{ik}(k) - \max_{q \neq k} \phi_{ik}(q) \\ &= \begin{cases} \min[l_{ik} - \max_{q \notin \{0, k, i\}} (l_{iq} + \tilde{\gamma}_{iq}), \tilde{\alpha}_{ii} + \tilde{\rho}_i^k - \tilde{\gamma}_{ik}, \\ l_{ik} - \tilde{\lambda}_{ii} - \tilde{\gamma}_{ii}] & k \neq i \\ \min[\tilde{\lambda}_{ii} - \max_{q \notin \{0, k\}} (l_{iq} + \tilde{\gamma}_{iq}), \tilde{\alpha}_{ii} + \tilde{\rho}_i^i - \tilde{\gamma}_{ii}] & k = i \end{cases} \end{aligned} \quad (23)$$

$$\begin{aligned} \tilde{\gamma}_{ik} &= \gamma_{ik}(k) - \gamma_{ik}(q : q \neq k) \\ &= \begin{cases} \min[0, \tilde{\phi}_{kk} + \sum_{i' \notin \{i, k\}} \max[0, \tilde{\phi}_{i'k}]] & k \neq i \\ \sum_{i' \neq i} \max[0, \tilde{\phi}_{i'k}] & k = i \end{cases} \end{aligned} \quad (24)$$

$$\tilde{\psi}_i^k = \psi_{ii}(k) - \psi_{ii}(0) = \begin{cases} \tilde{\rho}_i^k + \tilde{\alpha}_{ii} & k \neq i \\ \tilde{\rho}_i^i + \tilde{\alpha}_{ii} - \tilde{\lambda}_{ii} & k = i \end{cases} \quad (25)$$

$$\tilde{\lambda}_{ii} = \lambda_{ii}(q = i) - \lambda_{ii}(q \neq i) = f([\tilde{\psi}_i^k]_{N \times N}) \quad (26)$$

Note that β and η are eliminated to simplify the updating rules. In (26), $[\tilde{\psi}_i^k]_{N \times N}$ is a $N \times N$ matrix whose element is $\tilde{\psi}_i^k$ in row i and column k . As shown in (26), $\tilde{\lambda}_{ii}$ is defined as the scale difference of $\lambda_{ii}(q = i)$ and $\lambda_{ii}(q \neq i)$. By only focusing on calculating the scale difference, we can calculate $\tilde{\lambda}_{ii}$ based on $[\tilde{\psi}_i^k]_{N \times N}$ expressed as $f([\tilde{\psi}_i^k]_{N \times N})$ in (26). The detailed algorithm of calculating $\tilde{\lambda}_{jj}$ based on $[\tilde{\psi}_i^k]_{N \times N}$ is given in Algorithm- $\tilde{\lambda}$ as follows:

Algorithm- $\tilde{\lambda}$:

Input: $[\tilde{\psi}_i^k]_{N \times N}$, number of clusters K **Output:** $\tilde{\lambda}_{jj}$

- 1 Build $[\tilde{\psi}_i^k]_{N \times (N+1)}$ by adding one column with zeros to the left of $[\tilde{\psi}_i^k]_{N \times N}$
 - 2 Find maximum values and their positions for all rows except j_{th} row of $[\tilde{\psi}_i^k]_{N \times (N+1)}$
 - 3 Calculate Num -the number of maximum values whose positions are the diagonal of $[\tilde{\psi}_i^k]_{N \times N}$ and their corresponding row indexes I_D
 - 4 **if** $Num < K$
 Except rows $j \cup I_D$, calculate all the subtraction value V between each maximum value and the value in diagonal of $[\tilde{\psi}_i^k]_{N \times N}$ in the same row.
 $\tilde{\lambda}_{jj}$ is the m_{th} small value in V , $m = K - Num$.
else
 For each row in I_D except row j if it is in I_D , calculate all the subtraction value $V1$ between the second large value and the largest value
 $\tilde{\lambda}_{jj}$ is the m_{th} large value in $V1$, $m = Num - K + 1$.
end
-

2) *Generating Assignment Matrix:* Each element b_{ij} in assignment matrix B is evaluated by calculating the value $\hat{b}_{ij}$ which maximizes the sum of all the messages passing to the node b_{ij} .

$$\hat{b}_{ij} = \begin{cases} 1 & \text{if } \tilde{\alpha}_{ij} + \tilde{\rho}_{ij} \geq 0 \quad \forall i \neq j \\ 0 & \text{otherwise} \end{cases} \quad (27)$$

$$\hat{b}_{ii} = \begin{cases} \argmax_k \tilde{\rho}_i^k & \text{if } \tilde{\alpha}_{ii} + max_k \tilde{\rho}_i^k \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (28)$$

Based on the assignment matrix B obtained by (27) and (28), the superexemplars and exemplars can be identified and then clustering labels are generated.

C. K-MEAP Algorithm

According to the above discussions, the K-MEAP algorithm can be outlined as follows. First, all the simplified messages are initialized to 0. Then they are iteratively updated according to (19)-(25) and Algorithm- $\tilde{\lambda}$ until the assignment matrix B stays unchanged for $T_{unchange}$ times or the maximum iteration number T_{max} is achieved.

Note that the same as AP and MEAP, a damping is introduced to message updating to prevent oscillation and ensure convergence. The damping is defined as follows [4]:

$$v = uv^{old} + (1 - u)v^{new} \quad (29)$$

Here, v denotes any of the results of updating rules on the left-hand side of (19)-(26) and u is the damping factor which is recommended in [10] to be 0.9 for most cases.

The time complexity of K-MEAP is $O(6 \cdot N^2 + N + p \cdot N)$ considering the object number N and cluster number K . $O(6 \cdot N^2)$ is the time complexity of updating 6 kinds of messages $\tilde{\rho}_{ij}$, $\tilde{\rho}_i^k$, $\tilde{\alpha}_{ij}$, $\tilde{\gamma}_i^k$, $\tilde{\phi}_i^k$ and $\tilde{\psi}_i^k$. $O(N)$ is the time complexity of

updating $\tilde{\alpha}_{ii}$. $O(p \cdot N)$ is the time complexity of calculating $\tilde{\lambda}_{ii}$ where $p = \max\{K - Num, Num - K + 1\}$. Here Num is the number of maximum values whose positions are on the diagonal of $[\tilde{\psi}_i^k]_{N \times N}$. In real application, p is always much less than object number N . Therefore the time complexity of K-MEAP can be considered as $O(N^2)$ which is as the same as that of AP, K-AP and MEAP. It is obvious that K-MEAP is more efficient than re-running MEAP in order to generate specified K clusters using the same bisection method suggested in [4].

Algorithm: K-MEAP

Input: Similarity matrix $[s_{ij}]_{N \times N}$,linkage matrix $[l_{ij}]_{N \times N}$,number of clusters K ,maximum iteration number T_{max} ,number of times B stays unchanged $T_{unchange}$ **Output:** Assignment matrix B

- 1 Initialize all the simplified messages to 0,
set iteration number $m \leftarrow 0$,
number of B keeps unchanged $n \leftarrow 0$

2 **Repeat** $m \leftarrow m + 1$ Update $\tilde{\rho}_{ij}^{(m)}$, $\tilde{\rho}_i^{k(m)}$, $\tilde{\alpha}_{ij}^{(m)}$, $\tilde{\alpha}_{ii}^{(m)}$, $\tilde{\gamma}_i^{k(m)}$, $\tilde{\phi}_i^{k(m)}$, $\tilde{\psi}_i^{k(m)}$ using (19)~(25)Update $\tilde{\lambda}_{ii}^{(m)}$ using Algorithm- $\tilde{\lambda}$ based on $\tilde{\lambda}_{ii}^{(m-1)}$ Calculate $B^{(m)}$ using (27), (28)**if** $B^{(m)} = B^{(m-1)}$ **then** $n = n + 1$ **else** $n = 1$ **Until** ($m = T_{max}$ or $n = T_{unchange}$)3 **Output** B

III. EXPERIMENTAL RESULTS

In this section, experimental studies of the proposed approach are conducted two image data sets. The experiments are conducted to show the performance of K-MEAP and its ability of identifying subclusters comparing to the related approaches including K-medoids, AP, K-AP and MEAP. Note that other multi-exemplars based approaches for example MPC [11] is not included here because MEAP outperforms them as shown in [5] on the same data sets. The experiments implemented in Matlab were conducted on a PC with eight cores of Intel I7-3770 with 8 gigabytes of memory ¹.

A. Data sets

The performance of the algorithms are compared on the following data sets.

JAFFE(Japanese female facial expression database) [12]: This data set has 10 classes which are 213 facial images of 10 Japanese females. Each person poses 7 facial expressions including happiness, sadness, surprise, anger, disgust, fear, and one neutral expression. Each expression is one subclass of the same person. The goal is to cluster the 213 images into 10 clusters and each cluster contains the images that belong to the same person.

MNIST [13]: This data set is composed of 10 classes which are 0 to 9 handwritten digits images. There are 70,000 28×28

¹The Matlab code of K-MEAP is available via email request to the author.

Fig. 2: The NMI result on the JAFFE data set.

pixel images. We use a subset of the data set which contains 1000 images with 100 images in each digit category. Each image is represented as a 784 dimensional feature vector. Each element in the vector is the pixel value which is normalized to $[0, 1]$ by dividing 255.

B. Construction of Similarity matrix

For JAFFE and MNIST data sets, the similarity matrix is constructed by applying the same methods as used in MEAP [5].

C. Evaluation criterion

Two popular external metrics *Normalized Mutual Information*(NMI)[14] and *Adjusted Rand Index*(ARI) [15] are used to evaluate the clustering results, which measure the agreement of the clustering results produced by an algorithm and the ground truth. The higher the values of the two metrics are, the better the clustering result is. The values are equal to 1 only when the clustering result is same as the ground truth.

D. Results on data sets with subclusters (JAFFE and MNIST)

Fig. 2 shows the results of NMI on JAFFE data set with different cluster numbers for different approaches. Limited by the space, we only show the results of NMI here. The results of ARI have the similar pattern. Note that the different cluster numbers of AP and MEAP are achieved by using the same bisection method to find the suitable parameters. For K-medoids, K-AP and K-MEAP, the number of cluster is specified beforehand. Moreover, for MEAP and K-MEAP, the same exemplar preference s_{ii} is set to compare them fairly. The damping factor u is set to be 0.9 for AP, MEAP, K-AP and K-MEAP. The results of K-medoids is the mean value of 1000 runs with a random initialization of each run. As shown in the figure, K-MEAP not only generates K clusters directly, but also achieves the best accuracy compared to the other four algorithms. When the specified cluster number K is equal to the ground truth($K=10$), K-MEAP performs perfectly with 100% clustering accuracy. We plot the K-MEAP results on JAFFE($K=10$) in Fig. 3. As shown in the figure, the superexemplar and exemplars are the representative faces for each cluster and subclusters respectively. Moreover, all the faces belong to the same person are grouped into the same cluster correctly. We also observed that the facial expressions in each subcluster are similar to each other. Four clusters(persons) named KM, MK, NA, YM in Fig. 3 are almost perfectly divided into 7 subclusters with each one represents a type of

Fig. 4: The NMI result on the MNIST data set.

(a) JAFFE

(b) MNIST

Fig. 5: The computational time of the five approaches with different number of clusters on JAFFE and MNIST data sets.

a facial expression correctly, while only one cluster(person) named MK is perfectly divided into 7 subclusters which is reported on Fig.8 in the paper of MEAP [5].

Fig. 4 shows the results of NMI on MNIST data set with different cluster numbers for different approaches respectively. The experimental setting is same as that of JAFFE. As shown in the figure, K-MEAP also performs better than the other four algorithms with almost all the cluster numbers in the experiment. The running time of the five approaches on JAFFE and MNIST data sets are shown in Fig. 5(a),(b) respectively. As shown in the figure, given a specified cluster number K, K-MEAP is much more efficient than re-runned MEAP. The efficiency is more obvious as shown in Fig. 5(b) because the size of MNIST data set is larger. MEAP may be re-runned for many times to find suitable parameters to generate specified K clusters. For example, when cluster number is specified as 8 for MNIST data set, MEAP needs be re-runned 34 times. K-MEAP is slower than K-medoids and K-AP because more messages need to be updated to identify multiple exemplars.

Fig. 3: Clustering result by K-MEAP on JAFFE dataset. The superexemplars and exemplars are annotated by thick and thin red box, respectively. The linkage between every exemplar and its corresponding superexemplar is represented by black arrow.

Table. I shows the results of all the approaches when the specified cluster number matches the ground truth. From the table we can see that K-MEAP performs the best on both data sets. Compared to MEAP on the two evaluation criterions(NMI and ARI), the improvements of K-MEAP on JAFFE are 2% and 3.1%, respectively. On MNIST data set the improvements are 14.8% and 18.5%, respectively. Compared to MEAP, the computational time of K-MEAP on JAFFE and MNIST are reduced by 19.9% and 73.4%, respectively. K-medoids and K-AP are faster than K-MEAP, however their accuracy are lower.

TABLE I: NMI, ARI and Time results of the five approaches on JAFFE and MNIST with true cluster number

Algorithms	JAFFE			MNIST		
	NMI	ARI	Time(s)	NMI	ARI	Time(s)
K-medoids	0.7771	0.6824	0.3043	0.3450	0.2111	1.6216
AP	0.9000	0.8700	1.5715	0.3922	0.2436	124.8396
K-AP	0.9802	0.9701	1.5615	0.3785	0.2501	47.5044
MEAP	0.9832	0.9716	4.0028	0.4407	0.2836	410.1594
K-MEAP	1	1	3.2066	0.5060	0.3361	109.2109

IV. CONCLUSION

We have proposed a new exemplar based clustering approach called K-MEAP for data analysis, and apply K-MEAP on several real world data sets to demonstrate its feasibility and potential. K-MEAP not only identifies multiple exemplars and superexemplars automatically, but also generate the specified K clusters directly without tuning parameters and re-running algorithm. Two new kinds of messages are added to replace the function of the parameter in MEAP specified by the user to control the number of clusters. The experimental results show that given a specified cluster number K, K-MEAP not only identifies multiple exemplars and generate K clusters directly and efficiently, but also outperforms related approaches in terms of clustering accuracy on both data sets.

In the future, to make K-MEAP scale for large data sets, further studies may be conducted to reduce the time complexity of K-MEAP by combining the strategies of large data analysis.

REFERENCES

- [1] A. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.
- [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] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *science*, vol. 315, no. 5814, pp. 972–976, 2007.
- [5] C.-D. Wang, J.-H. Lai, C. Suen, and J.-Y. Zhu, "Multi-exemplar affinity propagation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, pp. 2223–2237, Sept 2013.
- [6] X. Zhang, W. Wang, K. Nørvag, and M. Sebag, "K-ap: generating specified k clusters by efficient affinity propagation," in *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pp. 1187–1192, IEEE, 2010.
- [7] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *Information Theory, IEEE Transactions on*, vol. 47, no. 2, pp. 498–519, 2001.
- [8] Y. Weiss and W. T. Freeman, "On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs," *Information Theory, IEEE Transactions on*, vol. 47, no. 2, pp. 736–744, 2001.
- [9] H.-A. Loeliger, "An introduction to factor graphs," *Signal Processing Magazine, IEEE*, vol. 21, no. 1, pp. 28–41, 2004.
- [10] D. Dueck, *Affinity propagation: clustering data by passing messages*. PhD thesis, University of Toronto, 2009.
- [11] M. Liu, X. Jiang, and A. C. Kot, "A multi-prototype clustering algorithm," *Pattern Recognition*, vol. 42, no. 5, pp. 689–698, 2009.
- [12] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, "Coding facial expressions with gabor wavelets," in *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, pp. 200–205, IEEE, 1998.
- [13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [14] A. Strehl and J. Ghosh, "Cluster ensembles—a knowledge reuse framework for combining multiple partitions," *The Journal of Machine Learning Research*, vol. 3, pp. 583–617, 2003.
- [15] L. Hubert and P. Arabie, "Comparing partitions," *Journal of classification*, vol. 2, no. 1, pp. 193–218, 1985.