

Merge or Not? Learning to Group Faces via Imitation Learning

Yue He^{1*}, Kaidi Cao^{1*}, Cheng Li¹, Chen Change Loy²

¹SenseTime Group Limited, {heyue,caokaidi,chengli}@sensetime.com

²The Chinese University of Hong Kong, ccloy@ie.cuhk.edu.hk

Abstract

Face grouping remains a challenging problem despite the remarkable capability of deep learning approaches in learning face representation. In particular, grouping results can still be egregious given profile faces and a large number of uninteresting faces and noisy detections. Often, a user needs to correct the erroneous grouping manually. In this study, we formulate a novel face grouping framework that learns clustering strategy from ground-truth simulated behavior. This is achieved through imitation learning (a.k.a apprenticeship learning or learning by watching) via inverse reinforcement learning (IRL). In contrast to existing clustering approaches that group instances by similarity, our framework makes sequential decision to dynamically decide when to merge two face instances/groups driven by short- and long-term rewards. Extensive experiments on three benchmark datasets show that our framework outperforms unsupervised and supervised baselines.

Introduction

Given a large number of unlabeled face images, face grouping aims at clustering the images into individual identities present in the data. This task is an actively researched computer vision problem due to its enormous potential in commercial applications. It not only allows users to organize and tag photos based on faces but also retrieve and revisit huge quantity of relevant images effortlessly.

The performance of face grouping significantly benefits from the recent emergence of deep learning approaches (Chen, Patel, and Chellappa 2016; Parkhi, Vedaldi, and Zisserman 2015; Schroff, Kalenichenko, and Philbin 2015; Sun et al. 2014; Taigman et al. 2014; Wen et al. 2016). Nevertheless, we still observe some challenges when we apply existing methods on real-world photo albums. In particular, we found that deeply learned representation can still perform poorly given *profile faces and false detections*. In addition, there is no obvious mechanism to disambiguate *large quantity of non-interested faces* (e.g. *faces in the background*) that are captured under the same condition with the person of interests.

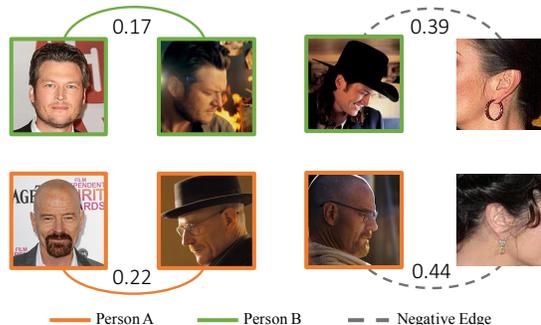


Figure 1: **Cosine angle in a deep feature space.** We measure the cosine angle between the deep feature vector of two faces. It is noted that even for two men with significantly different appearances, the angle between their profiles and noise faces (gray dash lines with 0.39~0.44) is much larger than one's frontal and his own profile (0.17 and 0.22).

We provide an illustrative example in Fig. 1, of which results were obtained from the Inception-v3 model (Szegedy et al. 2015) fine-tuned with MS-Celeb-1M (Guo et al. 2016) dataset. Despite the model achieves an accuracy of 99.27% on LFW (Huang et al. 2007), which is on par with the accuracy reported by a state-of-the-art method (Wen et al. 2016), its performance on the open-world face grouping task is unsatisfactory. Learning such an open-world model is still far from being solved due to highly imbalanced data (much more frontal faces compared to profile instances in existing datasets) and a large negative space to cover.

We human execute a visual grouping task in sequence with intermediate decision to govern our next step, like playing a jigsaw puzzle (Xie, Antle, and Motamedi 2008) with pieces of varying visual complexity. First we will link pieces with strong correlation and high confidence, then gain insights and accumulate visual evidence from these stable clusters. Consequently, a larger group can be formed through merging ambiguous positives and discarding uninteresting outliers. In the process, we may exploit contextual cues and global picture considering other samples. The above intuition motivates a novel face grouping framework. Our goal is not to design a better deep representation, but learning to make better merging/not-merging decision from expert's

demonstration using existing representation. In particular, we wish to introduce intermediate sequential decision between the clustering steps, *i.e.*, when to merge two samples or groups given the dynamic context. At each time step, an agent will choose from two possible actions, *i.e.*, to merge or not to merge a pair of face groups. The process responds at the next time step by moving to a new state and provides a reward to the agent. A sequence of good actions would lead to higher cumulative reward than suboptimal decisions.

Learning a decision strategy in our problem is non-trivial. In particular, the decision process is adversely affected by uninteresting faces and noisy detections. Defining a reward function for face grouping is thus not straightforward, which needs to consider the similarity of faces, group consistency, and quality of images. In addition, we also need to consider the operation cost involved, *i.e.*, the manual human effort spent on adding or removing a photo from a group. It is hard to determine the relative weights of these terms a-priori. This is in contrast to (first person) imitation learning of which the reward is usually assumed known and fixed, *e.g.*, using the change of game score (Mnih et al. 2015).

Contributions. We make the following contributions to overcome the aforementioned challenges:

- 1) We formulate a novel face grouping framework based on imitation learning (IL) via inverse reinforcement learning (Ng, Russell, and others 2000). To our knowledge, this is the first attempt to address visual clustering via inverse reinforcement learning. Once learned, the policy can be well generalized to unseen photo albums with good performance.
- 2) We assume the reward as an unknown to be ascertained through learning by watching an expert’s behavior, which can be easily simulated from a set training photo albums of which the ground-truth partition of the photos is known. We formulate the learning such that both short- and long-term rewards are considered. The formal considers similarity, consistency and quality of local candidate clusters; whereas the latter measures the operation cost to get from an arbitrary photos partition to the final ground-truth partition. The reward system effectively handles the challenges of profile, noisy, and uninteresting faces, and works well with conventional face similarity under an open-world context.
- 3) We introduce a large-scale dataset called Grouping Faces in the Wild (GFW) to facilitate the research of real-world photo grouping. The new dataset contains 78,000 faces of 3,132 identities collected from a social network. This dataset is realistic, providing a large number of uninteresting faces and noisy detections.

Extensive experiments are conducted on three datasets, namely, LFW simulated albums, ACCIO dataset (Harry Potter movie) (Ghaleb et al. 2015), and the GFW introduced by us. We show that the proposed method can be adapted to a variety of clustering algorithms, from the conventional k-means and hierarchical clustering to the more elaborated graph degree linkage (GDL) approach (Zhang et al. 2012). We show that it outperforms a number of unsupervised and supervised baselines. Our codes and data are released¹ to facilitate future studies.

¹<https://github.com/bj80heyue/Learning-to-Group>

Related Work

Face Grouping. Traditional face clustering methods (Cao et al. 2015; Li and Tang 2004; Otto, Klare, and Jain 2015; Zhu, Wen, and Sun 2011) are usually purely data-driven and unsupervised. They mainly focus on finding good distance metric between faces or effective subspaces for face representation. For instance, Zhu *et al.* (Zhu, Wen, and Sun 2011) propose a rank-order distance that measures the similarity between two faces using their neighboring information. Zhang *et al.* (Zhang et al. 2012) propose agglomerative clustering on a directed graph to better capture global manifold structures of face data. There exist techniques that employ user interactions (Tian et al. 2007), extra information on the web (Berg et al. 2004) and prior knowledge of family photo albums (Xia, Pan, and Qin 2014). Deep representation is recently found effective for face clustering (Schroff, Kalenichenko, and Philbin 2015), and large-scale face clustering has been attempted (Otto, Wang, and Jain 2016). Beyond image-based clustering, most existing video-based approaches employ pairwise constraints derived from face tracklets (Wu et al. 2013; Xiao, Tan, and Xu 2014; Zhang et al. 2016) or other auxiliary information (El Khoury, Senac, and Joly 2010; Tang et al. 2015; Zhou et al. 2015) to facilitate face clustering in video. The state-of-the-art method by Zhang *et al.* (Zhang et al. 2016) adapts DeepID2+ model (Sun, Wang, and Tang 2015) to a target domain with joint face representation adaptation and clustering. In this study, we focus on image-based face grouping without temporal information. Our method differs significantly to existing deep learning-based methods (Zhang et al. 2016) although we use deep representation too. Specifically, our method learns from experts to make sequential decision on grouping considering both short- and long-term rewards. It is thus capable of coping with uninteresting faces and noisy detections effectively.

Clustering with Reinforcement Learning. There exist some pioneering studies that explored clustering with RL. Likas (Likas 1999) models the decision process of assigning a sample from a data stream to a prototype, *e.g.*, cluster centers produced by on-line K-means. Barbakh and Fyfe (Barbakh and Fyfe 2007) employ RL to select a better initialization for K-means. Our work differs to the aforementioned studies: (1) (Barbakh and Fyfe 2007; Likas 1999) are unsupervised, *e.g.*, their loss is related to the distance from data to a cluster prototype. In contrast, our framework guides an agent with a teacher’s behavior. (2) We consider a decision that extends more flexibly to merge arbitrary instances or groups. We also investigate a novel reward function and new mechanisms to deal with noises.

Overview

An illustration of the proposed framework is given in Fig. 2. We treat grouping as a sequential process. In each step during test time, two candidate groups C_i and C_j are chosen. Without loss of generality, a group can be formed by just a single instance. Given the two groups, we extract meaningful features to characterize their similarity, group consistency, and image quality. Based on the features, an agent will

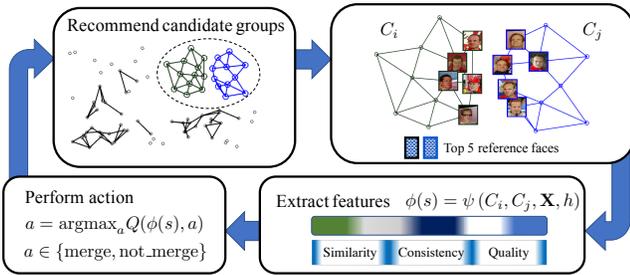


Figure 2: Face grouping by the proposed framework.

then perform an action, which can be either i) merging the two groups, or ii) not merging the two groups. Once the action is executed accordingly, the grouping proceeds to select the next pair of groups. The merging stops when there are no further candidate groups can be chosen, *e.g.*, the similarity between any groups is higher than a pre-defined threshold. Next, we define some key terminologies.

Recommender. At each time step we pick and consider the merging of two face groups. The action space is large with a complexity of $O(N^2)$, where N is the number of groups. This adds hurdles to both learning and test stages. To makes our approach scalable, we employ a recommender, M , which recommends two candidates cluster C_i and C_j at each time step. This reduces the $O(N^2)$ action space to a binary problem, *i.e.*, to merge or not to merge a pair of face groups. A recommender M can be derived from many classic clustering algorithms especially agglomerative-based algorithm like hierarchical clustering (HC), ranked-ordered clustering (Zhu, Wen, and Sun 2011) and GDL approach (Zhang et al. 2012). For instance, hierarchical clustering-based M always suggest two clusters that are nearest by some distance metric. In the experiment section, we perform rigorous evaluations on plausible choices of a recommender.

State. Each state $s_t = (h_t, H_t) \in \mathcal{S}$, contains the current grouping partition h_t and recommender history H_t , at time step t . In each discrete state, the recommender M will recommend a pair of cluster $(C_i, C_j) = M(s)$ based on the current state.

Action. An action is denoted as a . An agent can execute two possible actions, *i.e.*, merge two groups or not. That is the action set is $\mathcal{A} = \{\text{merge, not_merge}\}$, and $a \in \mathcal{A}$.

Transition. If a merging action is executed, candidate groups C_i and C_j will be merged. The corresponding partition is updated as $h_{t+1} \leftarrow \{h_t \setminus \{C_i, C_j\}\} \cup \{C_i \cup C_j\}$. Otherwise, the partition remains unchanged, $h_{t+1} \leftarrow h_t$. The candidate information will be appended to the history H_{t+1} so that the same pair would not be recommended by M . The transition is thus represented as $s_{t+1} = T(s_t, a)$, where $T(\cdot)$ denotes the transition function, and $s_{t+1} = (h_{t+1}, H_{t+1})$ and $s_t = (h_t, H_t)$.

Learning Face Grouping by Imitation

The previous section explains the face grouping process at test time. To learn an agent with the desired behavior, we assume access to demonstrations by an expert. In our study, we obtain these demonstrations from a set training photo al-

bums of which the ground-truth partition of the photos is known. Consequently, given any two candidate groups, C_i and C_j , we know if merging them is a correct action or not. These ground-truth actions $\{a_{GT}\}$ represent the pseudo expert’s behavior.

Towards the goal of learning an agent from the expert’s behavior, we perform the learning in two stages: (1) we find a reward function to explain the behavior via inverse reinforcement learning (Ng, Russell, and others 2000), (2) with the learned reward function we find a policy that maximizes the cumulative rewards.

Formally, let $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ denotes the reward function, which rewards the agent after it executes action a in state s . And \mathcal{T} is a set of state transition probabilities upon taking action a in state s . For any policy π , a value function V^π evaluates the value of a state as the total amount of reward an agent can expect to accumulate over the future, starting from that state, s_1 ,

$$V^\pi(s_1) = \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t | \pi) \right], \quad (1)$$

where γ is a discount factor.

An action-value function Q^π is used to judge the value of actions, according to

$$Q^\pi(s, a) = R(s, a) + \gamma V^\pi(s' | s' = T(s, a)), \quad (2)$$

where the notation $s' = T(s, a)$ represents the transition to state s' after taking an action a at state s . Our goal is to first uncover the reward function R from expert’s behavior, and find a policy π that maximizes $Q^\pi(s, a)$.

Rewards. In our study, the reward function that we wish to learn consists of two terms, denoted as

$$R = R_{\text{short}} + \beta R_{\text{long}}. \quad (3)$$

The first and second term corresponds to short- and long-term rewards, respectively. The parameter β helps balance the scale of the two terms. The short-term reward is multi-variate. It considers how strong two instances/groups should be merged locally based on face similarity, group consistency, and face quality. A long-term reward captures more far-sighted clustering strategy through measuring the operation cost to get from an arbitrary photos partition to the final ground-truth partition. Note that during the test time, the long-term reward function is absorbed in our learned action-value function for a policy π , thus no ground-truth is needed during testing. We provide explanations on the short- and long-term rewards as follows.

Short-Term Reward

Before a human user decides a merge between any two face groups, he/she will determine how close the two groups are in terms of face similarity. In addition, he/she may consider the quality and consistency of images in each group to prevent any accidental merging of uninteresting faces and noisy detections. We wish to capture such a behavior through learning a reward function. The reward is considered short-term since it only examines the current groups’ partition. Specifically, we compute the similarity between two groups, the quality for each group and photos consistency in each

group as a feature vector $\phi(s)$, and we project this feature into a scalar reward,

$$R_{\text{short}}(s, a) = y(a) (\mathbf{w}^\top \phi(s) + b), \quad (4)$$

where $y(a) = 1$ if action $a = \text{merge}$, and $y(a) = -1$ if $a = \text{not_merge}$. Note that we assume the actual reward function is unknown and (\mathbf{w}, b) should be learned through IRL. We observe that through IRL, a powerful reward function can be learned. An agent can achieve a competitive result even by myopically deciding based on one step’s reward function rather than multiple steps. We will show that optimizing (\mathbf{w}, b) is equivalent to learning a hyperplane in support vector machine (SVM) in the ‘Finding the Reward and Policy’ section.

Next, we describe how we design the feature vector $\phi(s)$, which determines the characteristics an agent should examine before making a group merging decision. A feature vector is extracted considering the candidate groups, all faces’ representation \mathbf{X} in the groups, and current partition h , that is $\phi(s) = \psi(C_i, C_j, \mathbf{X}, h)$.

The proposed feature vector contains three kinds of features, so as to capture face similarity, group consistency, and image quality. All face representation are extracted from Inception-v3 model (Szegedy et al. 2015) fine-tuned with MS-Celeb-1M (Guo et al. 2016). More elaborated features can be considered given the flexibility of the framework.

Face Similarity. We compute a multi-dimensional similarity vector to describe the relationship between two face groups C_i and C_j . Specifically, we first define the distance between the representation of two arbitrary faces $\mathbf{x}_i^u \in C_i$, and $\mathbf{x}_j^v \in C_j$ as $d(\mathbf{x}_i^u, \mathbf{x}_j^v)$. The subscript on \mathbf{x} indicates its group. In this study, we define the distance function as angular distance. We then start from C_i : for a face \mathbf{x}_i^u in C_i , we compute its distance to all the faces in C_j and select a median from the resulting distances. That is

$$d_{\text{med}}(\mathbf{x}_i^u, C_j) = \text{median} \{d(\mathbf{x}_i^u, \mathbf{x}_j^1), \dots, d(\mathbf{x}_i^u, \mathbf{x}_j^{n_j})\}, \quad (5)$$

where $n_j = |C_j|$. We select η number of instances with the shortest distances from $\{d_{\text{med}}(\mathbf{x}_i^u, C_j), \forall u\}$ to define the distance from C_i to C_j . Note that the distance is not symmetric. Hence, we repeat the above process to obtain another η shortest distances from $\{d_{\text{med}}(C_i, \mathbf{x}_j^v), \forall v\}$ to define the distance from C_i to C_j . Lastly, these 2η distances are concatenated to form a 2η -dimensional feature vector.

Group Consistency. Group consistency measures how close the samples in a group to each other. Even two groups have high similarity in between their respective members, we may not want to merge them if one of the group is not consistent, which may happen when there are a number of non-interesting faces inside the group. We define the consistency of a group as the median of pairwise distances between faces in the group itself. Given a group C_i :

$$\text{consistency}(C_i) = \text{median} \{d(\mathbf{x}_i^u, \mathbf{x}_i^v), u \neq v, \forall (u, v) \in C_i\}. \quad (6)$$

Consistency is computed for the two candidate groups, contribute a two-dimensional feature vector to $\phi(s)$.

Face Quality. As depicted in Fig. 1, profile faces and noises could easily confuse a state-of-the-art face recognition model. To make our reward function more informed on

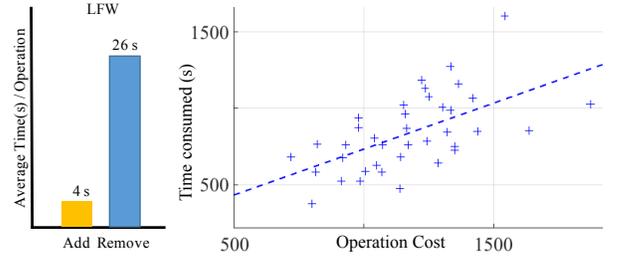


Figure 3: A user study on operation cost.

the quality of the images, we train a linear classifier by using annotated profile and falsely detected faces as negative samples, and clear frontal faces as positive samples. A total of 100k face images extracted from movies is used for training. The output of the classifier serves as the quality measure. Here, we concatenate the quality values of the top η faces in each of the two groups to form another 2η -dimensional features to $\phi(s)$.

Long-Term Reward

While the short-term reward R_{short} captures how likely two groups should be merged given the current partition, the long-term reward R_{long} needs to encapsulate a more far-sighted clustering strategy.

To facilitate the learning of this reward, we introduce the term ‘operation cost’, which measures the efforts needed to manipulate the images in the current partition to approach to ground-truth partition. Formally, given a partition $h \in \mathcal{V}$ and ground-truth partition $g \in \mathcal{V}$. A sequence of operations $o_i \in \mathcal{O} : \mathcal{V} \rightarrow \mathcal{V}$ can be executed to gradually modify the partition h to g . The cost function $c : \mathcal{O} \rightarrow \mathbb{R}_{>0}$ maps each type of operations into a positive time cost. then we define $Op(h, g)$ as the minimal cost for this change:

$$Op(h, g) = \min_{\Gamma, o_1, \dots, o_\Gamma} \sum_{t=1}^{\Gamma} c(o_t), \quad (7)$$

$$\text{s.t. } g = o_\Gamma \cdot \dots \cdot o_2 \cdot o_1 \cdot h$$

$$o_t \in \mathcal{O}$$

where Γ is the number of steps needed to get from h to g .

The cost function $c(\cdot)$ can be obtained from a user study. In particular, we requested 30 volunteers and show them a number of randomly shuffled images as an album. Their task is to reorganize the photos into a desired groups’ partition. We recorded the time needed for three types of operations: (1) adding a photo into a group, (2) removing a photo from a group, and (3) merging two groups. The key results are shown in Fig. 3. It can be observed that the ‘removing’ operation takes roughly $6\times$ longer than the ‘adding’ operation. The ‘merging’ operation is almost similar to ‘adding’. Consequently, we set the cost for these three operations as 1, 6, 1, respectively. The validity is further confirmed by the plot in Fig. 3 that shows a high-correlation between the time consumed and the computed operation cost.

Given Eqn. (7), we define the long-term reward as:

$$R_{\text{long}} = -\Delta Op^{(K)} = -(Op(h_{t-K}, g) - Op(h_t, g)), \quad (8)$$

Algorithm 1: Reward function learning via IRL.

input : Training albums $\Omega = \{\omega^{(i)}\}$, ground-truth partition $\{g^{(i)}\}$
output: Binary classifier (\mathbf{w}, b) for R_{short}
Initialization $\mathbf{w} \leftarrow \mathbf{w}_0, b \leftarrow b_0, \mathcal{L} \leftarrow \emptyset$;
repeat
 for $\omega^{(i)} \in \Omega$ **do**
 $t = 0$;
 Initialize partition h_t with each photo as a single group;
 Initialize history $H_t \leftarrow \emptyset$;
 repeat
 M recommends candidate groups (C_j, C_k) ;
 Compute action $a = \operatorname{argmax}_a R_{\text{short}}(s, a)$;
 Obtain ground-truth action a_{GT} based on $g^{(i)}$;
 if $a \neq a_{\text{GT}}$ **then**
 | Add $(\phi(s), a_{\text{GT}})$ into \mathcal{L}
 end
 if $a = a_{\text{GT}}$ **then**
 | $h_{t+1} \leftarrow \{h_t \setminus \{C_j, C_k\}\} \cup \{C_j \cup C_k\}$
 end
 Append (C_j, C_k, a) into H_{t+1} ;
 $t = t + 1$;
 until end of grouping;
 Retrain (\mathbf{w}, b) on \mathcal{L} ;
 end
until all albums are successfully partitioned;

which encodes the operation cost changes in K steps. The key benefit brought by R_{long} is that it provides a long-term reward for the grouping process. For any action that can hardly be decided (e.g., merging two noisy groups or merging a clean group with a noisy group), this term provides a strong evidence to the action-value function.

Finding the Reward and Policy

We assume the availability of a set training photo albums of which the ground-truth partition of the photos is known. Let $\Omega = \{\omega^{(i)}\}$ denotes a set of albums in a training set. The ground-truth partition for albums $\omega^{(i)}$ is given as $g^{(i)}$, from which we can derive the ground-truth actions $\{a_{\text{GT}}\}$ as an expert’s behavior. Our goal is to find a reward function based on this behavior. We perform the learning in two steps to ease the convergence of our method: (1) We employ IRL to find the reward function with a myopic or short-sighted policy. (2) We then use the ϵ -greedy algorithm (Watkins 1989) to find the optimal policy.

Step 1: Algorithm 1 summarizes the first step. Specifically, we set $\gamma = 0$ in Eqn. (2) and $\beta = 0$ in Eqn. (3). This leads to a myopic policy $Q^\pi(s, a|\pi) = R_{\text{short}}(s, a)$ that considers the current maximal short-term reward. This assumption greatly simplifies our optimization as (\mathbf{w}, b) of R_{short} (Eqn. (4)) are the only parameters to be learned. We solve this using a binary RBF-kernel SVM with actions as the classes. We start the learning process with an SVM of random weights and an empty training set $\mathcal{L} = \emptyset$. We execute the myopic policy repeatedly on albums. Once the agent chooses the wrong action w.r.t. the ground-truth, the representations of the in-

cluded groups and the associated ground-truth will be added to the SVM training set. Different albums constitute different games in which SVM will be continually optimized using the instances that it does not perform well. Note that the set \mathcal{L} is accumulated, hence each time we use samples collected from over time for retraining (\mathbf{w}, b) . The learning stops when all albums are correctly partitioned.

Step 2: Once the reward function is learned, finding the best policy π becomes a classic RL problem. Here we apply the ϵ -greedy algorithm (Watkins 1989). ϵ -greedy policy is a way of selecting random actions with uniform distribution from a set of available actions. Using this policy either we can select random action with ϵ probability and we can select an action with $1 - \epsilon$ probability that gives maximum reward in a given state. Specifically, we set $\gamma = 0.9$ in Eqn. (2) and $\beta = 0.8$ in Eqn. (3). We first approximate the action-value function Q^π in Eqn. (2) by a random forest regressor $Q(\phi(s), a)$ (Pyeatt, Howe, and others 2001). The input to the regressor is $(\phi(s), a)$ and the output is the associated Q^π value. The parameters of the regressor are initialized by $\phi(s)$, a , and Q^π value, which are obtained in the first step (Algorithm 1). After the initialization, the agent selects and executes an action according to Q , i.e., $a = \operatorname{argmax}_a Q(\phi(s), a)$, but with a probability ϵ the agent will act randomly so as to discover a state that it has never visited before. At the same time the parameters of Q will be updated directly from the samples of experience drawn from the algorithm’s past games. At the end of learning, the value of ϵ is decayed to 0, and Q is used as our action-value function for policy π .

Experiments

Training Data. We assume the availability of a set training photo albums of which the ground-truth partition of the photos is known. From the known partitions we can derive ground-truth actions as an expert’s behavior. The learned policy can be applied to other datasets for face grouping. Here we employ 2,000 albums simulated from MS-Celeb-1M (Guo et al. 2016) of 80k identities as our training source and generalize it to various test data below.

Test Data. We evaluate the proposed approach on three datasets of different scenarios exclusive from the training source. Example images are provided in Fig. 4.

1) *LFW-Album*: We construct a challenging simulated albums from LFW (Huang et al. 2007), MS-Celeb-1M (Guo et al. 2016), and PFW (Sengupta et al. 2016), with a good mix of frontal, profile, and non-interested faces. We prepare 20 albums and with exclusive identities. The MS-Celeb-1M samples used here are exclusive from the training data.

2) *ACCIO Dataset*: This dataset (Ghaleb et al. 2015) is commonly used in the studies of video face clustering. It contains face tracklets extracted from series of Harry Potter movie. Following (Zhang et al. 2016), we conduct experiments on the first instalment of the series, which contains 3243 tracklets from 36 known identities. For a fair comparison, we do not consider uninterested faces in this dataset following (Zhang et al. 2016). We discard the temporal information and used only the frames in our experiments.

	Album#	Identities /album	Face /album	Frontal Face	Profile Face	“No Name” Face
LFW-Album	20	55 ±30	1650 ±900			10%-30%
ACCIO	8	35	3243			39%
GFW	60	54 ±50	1300 ±1000			20%-70%

Figure 4: Overview of test datasets.

3) *Grouping Face in the Wild (GFW)*: To better evaluate our algorithm for real-world application, we collect 60 real users’ albums with permission from a Chinese social network portal. The size of an album varies from 120 to 3600 faces, with a maximum number of identities of 321. In total, the dataset contains 84,200 images with 78,000 faces of 3,132 different identities. All faces are automatically detected using Faster-RCNN (Ren et al. 2015). False detections are observed. We annotate all detections with identity/noise labels. The images are unconstrained, taken in various indoor/outdoor scenes. Faces are naturally distributed with different poses with spontaneous expression. In addition, faces can be severely occluded, blurred with motion, and differently illuminated under different scenes.

Given the limited space, we exclude results on traditional grouping datasets like Yale-B (Georgiades, Belhumeur, and Kriegman 2001), MSRA-A (Zhu, Wen, and Sun 2011), MSRA-B (Zhu, Wen, and Sun 2011) and Easyalbum (Cui et al. 2007). Yale-B were captured in controlled condition with very few profile faces and noises. The number of albums is limited in the other three datasets.

Implementation Details. All face representation are extracted from Inception-v3 model (Szegedy et al. 2015) fine-tuned with MS-Celeb-1M (Guo et al. 2016). We set $\beta = 0.8$ in Eqn. (3) to balance the scales of short- and long-term rewards. We fixed the number of faces $\eta = 5$ to form the similarity and quality features. The five shortest distances is a good trade-off between performance and feature complexity. If a group has fewer than five faces (to the extreme only one face exists), we pad the distance vector with the farthest distance.

Evaluation Metrics. We employ multiple metrics to evaluate the face grouping performance, including the B-cubed precision, recall, and F_1 score suggested by (Zhang, Kalashnikov, and Mehrotra 2013) and (Zhang et al. 2016). Specifically, B-cubed recall measures the average fraction of face pairs belonging to the ground truth identity assigned to the same cluster. And B-cubed precision is the fraction of face pairs assigned to a cluster with matching identity labels. The F_1 score measures the harmonic means of these two metrics. We also use *operation cost* introduced in the ‘Long-Term Reward’ section. To facilitate comparisons across datasets of different sizes, we compute the operation cost normalized by the number of photos as our metric, *i.e.*, $\overline{Op} = Op/N$. We believe that this metric is more important than the others since it directly reflects how much effort per image a user needs to spend to organize a photo album.

Comparison with Unsupervised Methods

We compare our method with classic and popular clustering approaches: 1) K-means, 2) Graph Degree Linkage (GDL) (Zhang et al. 2012), 3) Hierarchical Clustering (HC), and 4) Affinity Propagation (AP) (Frey and Dueck 2007). Note that we also compare with (Zhang et al. 2016) on its reported precision, recall, and F_1 scores on the ACCIO-1 dataset. Note that these baselines use the same features as our approach. To verify if the proposed imitation learning (IL) framework helps existing clustering methods, we adapt K-means, GDL and HC into IL-K-means, IL-GDL and IL-HC to equip them with the sequential decision capability. This is achieved by using the respective algorithm as the recommender.

Table 1 summarizes the results on three datasets. We observed that imitation learning consistently improves the different clustering baselines. For instance, on LFW-Album, the F_1 score and \overline{Op} of HC improves from 76.6% and 0.35 to 91.1% and 0.14. Notably, IL-HC outperforms other variants based on the proposed IL, although our framework is not specifically developed to work only with hierarchical clustering. Qualitative results are provided in the supplementary material, showing that imitation learning is capable of yielding more coherent face groupings with exceptional robustness to outliers.

Comparison with Supervised Methods

We compare our framework with two supervised baselines, namely a SVM classifier and a three-layer Siamese network. The three layers of the Siamese network have 256, 64, 64 hidden neurons, respectively. A contrastive loss is used for training. To train the baselines, each time we sample two subsets of identities from MS-Celeb-1M as the training data. SVM and the Siamese Network are used to predict if two groups should be merged or not. Features are extracted following the method presented in the methodology section. These supervised baselines are thus *strong* – the features include face similarity vector that is derived from Inception-v3 face recognition model fine-tuned with MS-Celeb-1M dataset. The deep representation achieves 99.27% on LFW, which is better than (Sun et al. 2014) and on-par with (Wen et al. 2016). The results of the baseline are presented in Table 1. It is observed that the IL-based approach outperforms the supervised baselines by a considerable margin.

Ablation Study

Further Analysis on Recommender. In previous experiment, we tested three different recommenders based on different clustering methods, namely K-means, GDL, and HC. In this experiment, we further analyze the use of a random recommender that randomly chooses a pair to recommend. Figure 5 shows the F_1 score comparisons between a Hierarchical Clustering (HC) recommender and a random recommender. In comparison to the recommender based on HC, which always recommends the nearest groups, the random recommender exhibits a slower convergence and poorer results. It is worth pointing out that the random recommender still achieves a F_1 score of 61.9% on GFW, which outperforms the unsupervised baseline, which only achieves 15%.

Table 1: Face grouping results on LFW-Album, ACCIO-1, and GFW.

Dataset	LFW-Album				ACCIO-1				GFW			
	P(%)	R(%)	F_1 (%)	Op	P(%)	R(%)	F_1 (%)	Op	P(%)	R(%)	F_1 (%)	Op
K-means	73.6	86.6	79.3	1.12	72.2	34.4	46.6	0.65	66.6	35.7	41.1	1.47
GDL (Zhang et al. 2012)	66.5	92.2	76.4	1.21	18.1	91.1	30.2	3.51	67.4	59.4	55.9	1.30
HC	74.2	80.8	76.6	0.35	17.1	91.9	28.9	3.28	77.5	22.3	15.0	0.81
AP (Frey and Dueck 2007)	76.7	71.1	73.7	1.07	82.2	9.6	17.1	0.59	69.7	25.3	32.7	0.86
Deep Adaptation (Zhang et al. 2016)	-	-	-	-	71.1	35.2	47.1	-	-	-	-	-
IL-Kmeans	76.7	87.8	81.6	0.95	82.8	34.1	48.3	0.54	53.4	43.6	43.3	1.17
IL-GDL	79.9	90.1	84.5	0.54	88.6	46.3	60.8	0.78	78.4	76.2	74.5	0.68
IL-HC	97.8	85.3	91.1	0.14	90.8	78.6	84.3	0.52	96.6	53.7	67.3	0.17
SVM + Deep Features	82.7	87.4	85.0	0.45	89.0	61.3	72.6	0.74	84.3	46.4	56.3	0.33
Siamese Network + Deep Features	87.1	87.6	87.3	0.44	59.7	88.1	71.2	0.79	49.9	92.3	62.8	0.33

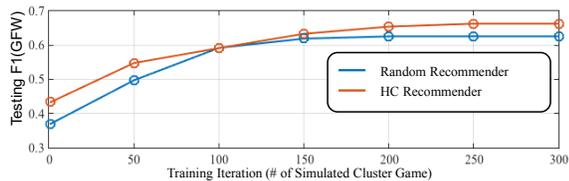


Figure 5: The F_1 score on using different recommenders along with different training iterations. The red curve is obtained by using Hierarchical Clustering (HC) as the recommender, while the blue curve is obtained by using the random recommender.

The results suggest the usefulness of deploying a recommender.

Reward Function Settings. We evaluate the effect of two reward terms in the reward function defined in Eqn. (3).

- 1) R_{short} & R_{long} : The full reward setting with $\beta \neq 0$.
- 2) w/o R_{long} : Without the long-term reward based on operation cost, i.e., $\beta = 0$.
- 3) w/o R_{short} : In this setting, we discarded R_{short} learned by IRL, and redefined it to take a naïve ± 1 loss, i.e., $R_{\text{short}} = \mathbb{1}(a = a_{GT})$, where $\mathbb{1}(\cdot)$ is an indicator function that outputs 1 if the condition is true, and -1 if it is false.

The results reported in Table 2 shows that both short- and long-term rewards are indispensable to achieve good results. Comparing the baselines “w/o R_{short} ” against the full reward, we observed that IL learned a more powerful short-term reward function than the naïve ± 1 loss. Comparing the baselines “w/o R_{long} ” against the full reward, albeit removing R_{long} only reduces the F_1 score slightly, the number of false positive and false negative merges actually increase for noisy and hard cases. Figure 6 shows some representative groups that were mistakenly handled by IL-HC w/o R_{long} .

Discard the Face Quality Feature. If we remove the face quality feature from the feature vector $\phi(s)$, the F_1 score achieved by IL-HC of LFW-Album, ACCIO-1, and GFW will drop from 91.1%, 84.3%, and 67.3%, to 89.5%, 65.0%, and 48.4%, respectively. The results suggest that the importance of quality measure depends on the dataset. Face quality feature is essential on the GFW dataset but less so on others, since GFW consists more poor-quality images.

Table 2: Different settings of reward function. We use IL-HC in this experiment.

Dataset	LFW-Album		GFW	
Metric	F_1 (%)	Op	F_1 (%)	Op
R_{short} & R_{long}	91.1	0.14	67.3	0.17
w/o R_{long}	90.7	0.14	62.6	0.17
w/o R_{short}	73.0	0.54	17.1	0.65

Candidate Groups	Ground-Truth	IL-HC	IL-HC (w/o R_{long})	HC

Merge Not-Merge

Figure 6: Example of merging decisions made by different algorithms. Each image represents a group they belong to. It is observed that IL-HC w/o R_{long} tends to produce false negative (FN) and false positive (FP) mistakes in comparison to IL-HC with full reward.

Conclusion

We have proposed a novel face grouping framework that makes sequential merging decision based on short- and long-term rewards. With inverse reinforcement learning, we learn powerful reward function to cope with real-world grouping tasks with unconstrained face poses, illumination, occlusion, and abundant of uninteresting faces and false detections. We have demonstrated that the framework benefits many existing agglomerative-based clustering algorithms.

Acknowledgement. This work is supported by SenseTime Group Limited and the General Research Fund sponsored by the Research Grants Council of the Hong Kong SAR (CUHK 14241716, 14224316, 14209217).

References

- Barbakh, W., and Fyfe, C. 2007. Clustering with reinforcement learning. In *IDEAL*.
- Berg, T. L.; Berg, A. C.; Edwards, J.; Maire, M.; White, R.; Teh, Y.-W.; Learned-Miller, E.; and Forsyth, D. A. 2004. Names and faces in the news. In *CVPR*.
- Cao, X.; Zhang, C.; Fu, H.; Liu, S.; and Zhang, H. 2015. Diversity-induced multi-view subspace clustering. In *CVPR*.
- Chen, J.-C.; Patel, V. M.; and Chellappa, R. 2016. Unconstrained face verification using deep CNN features. In *WACV*.
- Cui, J.; Wen, F.; Xiao, R.; Tian, Y.; and Tang, X. 2007. Easyalbum: an interactive photo annotation system based on face clustering and re-ranking. In *ACM SIGCHI*.
- El Khoury, E.; Senac, C.; and Joly, P. 2010. Face-and-clothing based people clustering in video content. In *ICMIR*.
- Frey, B. J., and Dueck, D. 2007. Clustering by passing messages between data points. *Science*.
- Georghiadis, A. S.; Belhumeur, P. N.; and Kriegman, D. J. 2001. From few to many: Illumination cone models for face recognition under variable lighting and pose. *TPAMI* 23(6):643–660.
- Ghaleb, E.; Tapaswi, M.; Al-Halah, Z.; Ekenel, H. K.; and Stiefelhagen, R. 2015. Accio: A data set for face track retrieval in movies across age. In *ICMR*.
- Guo, Y.; Zhang, L.; Hu, Y.; He, X.; and Gao, J. 2016. MS-Celeb-1M: A dataset and benchmark for large scale face recognition. In *ECCV*.
- Huang, G. B.; Ramesh, M.; Berg, T.; and Learned-Miller, E. 2007. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst.
- Li, Z., and Tang, X. 2004. Bayesian face recognition using support vector machine and face clustering. In *CVPR*.
- Likas, A. 1999. A reinforcement learning approach to online clustering. *Neural computation* 11(8):1915–1932.
- Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518(7540):529–533.
- Ng, A. Y.; Russell, S. J.; et al. 2000. Algorithms for inverse reinforcement learning. In *ICMR*.
- Otto, C.; Klare, B.; and Jain, A. K. 2015. An efficient approach for clustering face images. In *ICB*.
- Otto, C.; Wang, D.; and Jain, A. K. 2016. Clustering millions of faces by identity. *arXiv preprint arXiv:1604.00989*.
- Parkhi, O. M.; Vedaldi, A.; and Zisserman, A. 2015. Deep face recognition. In *BMVC*.
- Pyeatt, L. D.; Howe, A. E.; et al. 2001. Decision tree function approximation in reinforcement learning. In *ISAS*.
- Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*.
- Schroff, F.; Kalenichenko, D.; and Philbin, J. 2015. Facenet: A unified embedding for face recognition and clustering. In *CVPR*.
- Sengupta, S.; Chen, J. C.; Castillo, C.; Patel, V. M.; Chellappa, R.; and Jacobs, D. W. 2016. Frontal to profile face verification in the wild. In *WACV*.
- Sun, Y.; Chen, Y.; Wang, X.; and Tang, X. 2014. Deep learning face representation by joint identification-verification. In *NIPS*.
- Sun, Y.; Wang, X.; and Tang, X. 2015. Deeply learned face representations are sparse, selective, and robust. In *CVPR*.
- Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2015. Rethinking the inception architecture for computer vision. *arXiv preprint arXiv:1512.00567*.
- Taigman, Y.; Yang, M.; Ranzato, M.; and Wolf, L. 2014. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*.
- Tang, Z.; Zhang, Y.; Li, Z.; and Lu, H. 2015. Face clustering in videos with proportion prior. In *IJCAI*.
- Tian, Y.; Liu, W.; Xiao, R.; Wen, F.; and Tang, X. 2007. A face annotation framework with partial clustering and interactive labeling. In *CVPR*.
- Watkins, C. J. C. H. 1989. *Learning from delayed rewards*. Ph.D. Dissertation, University of Cambridge England.
- Wen, Y.; Zhang, K.; Li, Z.; and Qiao, Y. 2016. A discriminative feature learning approach for deep face recognition. In *ECCV*.
- Wu, B.; Zhang, Y.; Hu, B.-G.; and Ji, Q. 2013. Constrained clustering and its application to face clustering in videos. In *CVPR*.
- Xia, S.; Pan, H.; and Qin, A. 2014. Face clustering in photo album. In *ICPR*.
- Xiao, S.; Tan, M.; and Xu, D. 2014. Weighted block-sparse low rank representation for face clustering in videos. In *ECCV*.
- Xie, L.; Antle, A. N.; and Motamedi, N. 2008. Are tangibles more fun?: Comparing children’s enjoyment and engagement using physical, graphical and tangible user interfaces. In *International conference on tangible and embedded interaction*, 191–198. ACM.
- Zhang, W.; Wang, X.; Zhao, D.; and Tang, X. 2012. Graph degree linkage: Agglomerative clustering on a directed graph. In *ECCV*.
- Zhang, Z.; Luo, P.; Loy, C. C.; and Tang, X. 2016. Joint face representation adaptation and clustering in videos. In *ECCV*.
- Zhang, L.; Kalashnikov, D. V.; and Mehrotra, S. 2013. A unified framework for context assisted face clustering. In *ICMR, ICMR '13*, 9–16. New York, NY, USA: ACM.
- Zhou, C.; Zhang, C.; Fu, H.; Wang, R.; and Cao, X. 2015. Multi-cue augmented face clustering. In *ACM MM*.
- Zhu, C.; Wen, F.; and Sun, J. 2011. A rank-order distance based clustering algorithm for face tagging. In *CVPR*.