Joint Face Representation Adaptation and Clustering in Videos: Supplementary Material

Zhanpeng Zhang[1], Ping Luo[2,1], Chen Change Loy[1,2], and Xiaoou Tang[1,2]

[1] Dept. of Information Engineering, The Chinese University of Hong Kong
[2] Shenzhen Key Lab of Comp. Vis. & Pat. Rec., Shenzhen Institutes of Advanced Technology, CAS, China

This supplementary material presents more clustering results, and mathematical details for the character label inference in MRF.

1 More Clustering Results

Fig. 1 and 2 show more clusters generated by our method in the Accio [1] and BF0502 dataset [2].

2 Details on Character Label Inference in MRF

2.1 The computation of pairwise term $\Psi(\cdot)$

As mentioned in Sec. 3.1 of the paper, given the face representation $\mathbf{x}_i$ and $\mathbf{x}_j$, we define the pairwise term $\Psi(\cdot)$ for the character label y_i and y_j by

$$\Psi(y_i, y_j) = \exp\left\{\alpha v(\mathbf{x}_i, \mathbf{x}_j) \cdot \left(\mathbf{1}(y_i, y_j) - \mathbf{1}(v(\mathbf{x}_i, \mathbf{x}_j) > 0)\right)\right\}, \qquad (1)$$

where $\mathbf{1}(\cdot)$ is an indicator function and α is a trade-off coefficient between unary term and pairwise term. This coefficient α will be updated as stated in the following Sec. 2.2. Here we give the details on the computation of face pair relation $v(\mathbf{x}_i, \mathbf{x}_j) \in V$, which includes two steps:

1. Compute a normalized affinity matrix V^a based on the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$.
2. Propagate the initial pairwise constraints $\mathbf{C}_0$[3] by the affinity matrix V^a and obtain the pair relation V.

For the first step, to compute the normalized affinity matrix V^a, we follow the method used in spectral clustering [3]. Firstly, we compute an affinity matrix A by $A_{ij} = \exp(-d^2(x_i, x_j)/\sigma_i\sigma_j)$ if x_j is within the h-nearest neighbors of x_i, otherwise we set $A_{ij} = 0$. We set $h = 10$ here as [4] (In fact, h is not a sensitive parameter. Usually $h \in [5, 100]$ produces similar clustering accuracy

[3] As stated in the paper, we obtain $\mathbf{C}_0$ by assuming all the face images in the same track have the same identity, *i.e.*, $c(I_i, I_j) = 1$. For faces appearing in the same frame of the video, their identities should be exclusive, *i.e.*, $c(I_i, I_j) = -1$. For other face pairs we have $c(I_i, I_j) = 0$.

Fig. 1. Example results generated by the proposed method on the Accio (*Harry Potter*) dataset. Every two rows denote a cluster.

Fig. 2. Example results generated by the proposed method on the "Buffy the Vampire Slayer" dataset [2]. Every row denote a cluster.

within 3% fluctuation). The term $d(x_i, x_j)$ is the L2-distance between x_i and x_j, and σ_i is the local scaling factor with $\sigma_i = d(x_i, x_m)$, where x_m is the m-th nearest neighbor of x_i. We set $m = 7$ as in [3]. Then the normalized affinity matrix is obtained by $V^a = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$, where D is a diagonal matrix with $D_{ii} = \sum_{j=1}^{n} A_{ij}$.

For the second step, we use the constraint propagation method in [5] to compute the pair relation V by:

$$V = (1-\lambda)^2 (1-\lambda V^a)^{-1} W (1-\lambda V^a)^{-1}, \tag{2}$$

where W is a matrix form of the pairwise constraints $\mathbf{C}_0$ (*i.e.*, $\mathbf{W}_{ij} = c(I_i, I_j)$). λ controls the propagation degree ($\lambda = 0.5$ in implementation). To this end, we can have V and compute the pairwise term $\Psi(\cdot)$.

2.2 MRF optimization process

Given the face representation $\mathbf{X}$, we infer character $\mathbf{Y}$ by maximizing the joint probability $p(\mathbf{X}, \mathbf{Y})$ with the simulated field algorithm [6, 4]. The algorithm is as follows: (1) Initialization step: we initialize $\mathbf{Y}$ by running K-means clustering on $\mathbf{X}$. In this case, for the Gaussian of each cluster ℓ, we obtain the the initial mean μ_ℓ and covariance matrix Σ_ℓ. We also set the initial trade-off coefficient $\alpha = 0$ in Eqn. (1). For simplicity, we denote the model parameter $\Omega = \{\mu_\ell, \Sigma_\ell, \alpha\}$ in the following text. (2) After the initialization step, we infer $\mathbf{Y}$ and update Ω by repeating the following two steps in each iteration q:

1. Simulate a new inferred $\widetilde{\mathbf{Y}}^q$ given the face representation $\mathbf{X}$ and current model parameter Ω^q.
2. Given $\widetilde{\mathbf{Y}}^q$, update Ω^q to maximize the log-likelihood of $p(\mathbf{X}, \mathbf{Y})$ by EM algorithm.

For the first step, we aim to obtain a new $\widetilde{\mathbf{Y}}^q$ given $\mathbf{X}$ and Ω^q. A natural way is to infer from the posterior:

$$p(\mathbf{Y}|\mathbf{X}, \Omega^q) = \frac{p(\mathbf{X}|\mathbf{Y}, \Omega^q) p(\mathbf{Y}|\Omega^q)}{p(\mathbf{X}|\Omega^q)}. \tag{3}$$

However the computation of the term $p(\mathbf{Y}|\Omega^q)$ involves the interaction of each y_i and its neighborhood. Thus, it is intractable. Here we employ the mean field-like approximation [6] for $p(\mathbf{Y}|\Omega^q)$, in which we assume each y_i is independent, and we set the value of its neighborhood $\mathcal{N}_i$ constant when we compute $p(y_i)$. In this case, we have

$$p(\mathbf{Y}|\Omega^q) = \prod_i p(y_i|\mathbb{Y}_{\mathcal{N}_i}, \Omega^q) = \prod_i \frac{p(y_i, \mathbb{Y}_{\mathcal{N}_i}, \Omega^q)}{\sum_{y_i=\ell}^{K} p(y_i, \mathbb{Y}_{\mathcal{N}_i}, \Omega^q)}. \tag{4}$$

where we denote the value of y_i's neighborhood as $\mathbb{Y}_{\mathcal{N}_i}$. For example, we can reuse the value in the previous iteration $q-1$ (*i.e.*, $\mathbb{Y}_{\mathcal{N}_i} = \widetilde{\mathbf{Y}}_{\mathcal{N}_i}^{(q-1)}$). Since $p(y_i, \mathbb{Y}_{\mathcal{N}_i}, \Omega^q) = \frac{1}{Z} \prod_{j \in \mathcal{N}_i} \Psi(y_i, y_j)$, the partition function Z can be eliminated in Eqn. (4) and we can compute $p(y_i|\mathbb{Y}_{\mathcal{N}_i}, \Omega^q)$. Combining Eqn. (3), Eqn. (4) and the mean field-like approximation, we have

$$p(\mathbf{Y}|\mathbf{X}, \Omega^q) = \prod_i p(y_i|\mathbb{Y}_{\mathcal{N}_i}, \mathbf{x}_i, \Omega^q) = \prod_i \frac{\Phi(\mathbf{x}_i|y_i, \Omega^q)p(y_i|\mathbb{Y}_{\mathcal{N}_i}, \Omega^q)}{\sum_{y_i=\ell}^{K} \Phi(\mathbf{x}_i|y_i, \Omega^q)p(y_i|\mathbb{Y}_{\mathcal{N}_i}, \Omega^q)} \quad (5)$$

Then the posterior $p(y_i = \ell|\mathbb{Y}_{\mathcal{N}_i}, \mathbf{x}_i, \Omega^q)$ can be computed directly for each face i and cluster ℓ. After that, we can simulate a new $\widetilde{y_i}^q$ based on this posterior (*i.e.*, the probability of setting $\widetilde{y_i}^q = \ell$ is proportional to $p(y_i = \ell)$). Then we obtain $\widetilde{\mathbf{Y}}^q = \{\widetilde{y_i}^q\}$.

For the second step, we aim to maximize the log-likelihood of $p(\mathbf{X}, \mathbf{Y})$ by updating the model parameter Ω in an EM algorithm. We define

$$\mathcal{Q}(\Omega|\Omega^q) = \mathbb{E}_{\mathbf{Y}|\mathbf{X},\Omega^{q-1}}(\log(p(\mathbf{X}, \mathbf{Y}|\Omega^q))), \quad (6)$$

where $\mathbb{E}$ denotes the expected value. So we have $\Omega^{q+1} = \arg\max_{\Omega} \mathcal{Q}(\Omega|\Omega^q)$.

Recall that (1) $\Omega = \{\mu, \Sigma, \alpha\}$ and $p(\mathbf{X}, \mathbf{Y}) = \frac{1}{Z}\prod_i \Phi(\mathbf{x}_i|y_i)\prod_i\prod_{j\in\mathcal{N}_i}\Psi(y_i, y_j)$, (2) μ and Σ are only related to the unary term Φ, (3) α is only related to the pairwise term Ψ. As in [6, 4], Eqn. (6) can be decomposed and we can update $\{\mu, \Sigma\}$ and α separately:

$$\mu^{q+1}, \Sigma^{q+1} = \arg\max_{\mu,\Sigma} \sum_i \sum_{y_i=\ell}^{K} p(y_i|\mathbb{Y}_{\mathcal{N}_i}, \mathbf{x}_i, \Omega^q) \log \Phi(\mathbf{x}_i|y_i\mu, \Sigma), \quad (7)$$

$$\alpha^{q+1} = \arg\max_{\alpha} \sum_i \sum_{y_i=\ell}^{K} p(y_i|\mathbb{Y}_{\mathcal{N}_i}, \mathbf{x}_i, \Omega^q) \log p(y_i|\mathbb{Y}_{\mathcal{N}_i}, \alpha). \quad (8)$$

For Eqn. (7), since $\Phi(\mathbf{x}_i|y_i\mu, \Sigma)$ is a Gaussian distribution and $p(y_i|\mathbb{Y}_{\mathcal{N}_i}, \mathbf{x}_i, \Omega^q)$ is constant, we can have a closed form solution for μ, Σ. As for Eqn. (8), we find a local optimal value for α, by the local search method as in [4].

The optimization of the above two steps ends when the posterior $p(y_i = \ell|\mathbb{Y}_{\mathcal{N}_i}, \mathbf{x}_i, \Omega^q)$ converged. The output character label $\mathbf{Y}^* = \arg\max_{\mathbf{Y}} p(\mathbf{Y}|\mathbf{X}, \Omega)$.

References

1. Ghaleb, E., Tapaswi, M., Al-Halah, Z., Ekenel, H.K., Stiefelhagen, R.: Accio: A data set for face track retrieval in movies across age. In: ACM International Conference on Multimedia Retrieval. (2015)
2. Everingham, M., Sivic, J., Zisserman, A.: Hello! My name is... buffy –automatic naming of characters in TV video. In: BMVC. (2006)
3. Zelnik-Manor, L., Perona, P.: Self-tuning spectral clustering. In: NIPS. (2004) 1601–1608
4. Wu, B., Zhang, Y., Hu, B.G., Ji, Q.: Constrained clustering and its application to face clustering in videos. In: CVPR. (2013)
5. Lu, Z., Ip, H.H.: Constrained spectral clustering via exhaustive and efficient constraint propagation. In: ECCV. (2010)
6. Celeux, G., Forbes, F., Peyrard, N.: EM procedures using mean field-like approximations for markov model-based image segmentation. Pattern recognition **36**(1) (2003) 131–144