Supplementary Material of Modeling the Uncertainty of Contextual-Connections between Tracklets for Unconstrained Video-based Face Recognition

Jingxiao Zheng[1] Ruichi Yu[1]* Jun-Cheng Chen[2] Boyu Lu[1] Carlos D. Castillo[1]
Rama Chellappa[1]
[1] UMIACS, University of Maryland, College Park [2] CITI, Academia Sinica, Taiwan
{jxzheng, yrcbsg}@umiacs.umd.edu, pullpull@citi.sinica.edu.tw, {bylu, carlos, rama}@umiacs.umd.edu

1. Derivation of Label Inference using Mean Field Algorithm in Section 3.3

Previously, we are given gallery-to-tracklet similarity $\mathbf{S}^{gt} = \left[s_{li}^{gt}\right]$, tracklet-to-tracklet similarity $\mathbf{S}^{tt} = \left[s_{ij}^{tt}\right]$ and cannot-link matrix $\mathbf{L}^{tt} = \left[L_{ij}^{tt}\right]$. We have the energy function

$$E(\mathbf{x}, \mathbf{y}) = \sum_{i \in \mathcal{V}} \psi_u^x(x_i) + \sum_{i \in \mathcal{V}, j \in \mathcal{N}(i)} \left[\psi_u^p(y_{i\to j}^p) + \psi_u^n(y_{i\to j}^n) + \psi_t^p(x_i, x_j, y_{i\to j}^p) + \psi_t^n(x_i, x_j, y_{i\to j}^n)\right] \tag{1}$$

where the potentials are defined as

$$\begin{aligned}
\psi_u^x(x_i = l) &= -T_{gt} \cdot s_{li}^{gt} \\
\psi_u^p(y_{i\to j}^p = 1) &= -T_{tt} \cdot s_{ij}^{tt} \\
\psi_u^n(y_{i\to j}^n = k) &= \begin{cases} 0 & \text{if } L_{ij}^{tt} = k \\ +\infty & \text{otherwise} \end{cases} \\
\psi_t^p(x_i, x_j, y_{i\to j}^p) &= \begin{cases} \alpha_p & \text{if } y_{i\to j}^p = 1 \text{ and } x_i \neq x_j \\ 0 & \text{otherwise} \end{cases} \\
\psi_t^n(x_i, x_j, y_{i\to j}^n) &= \begin{cases} \alpha_n & \text{if } y_{i\to j}^n = 1 \text{ and } x_i = x_j \\ 0 & \text{otherwise} \end{cases}
\end{aligned} \tag{2}$$

where T_{gt} and T_{tt} are corresponding temperature factors, $k \in \{0, 1\}$, α_p is the positive penalty and α_n is the negative penalty.

Directly looking for the label assignment that minimizes $E(\mathbf{x}, \mathbf{y})$ is a combinatorial optimization problem which is intractable. Instead, similar to [9], we use mean field method to approximate the distribution $P(\mathbf{X}, \mathbf{Y}) \propto \exp(-E(\mathbf{X}, \mathbf{Y}))$ by the product of independent marginals

$$Q(\mathbf{X}, \mathbf{Y}) = \prod_i Q_i(X_i) \prod_{j \in \mathcal{N}(i)} Q_{i\to j}^p(Y_{i\to j}^p) Q_{i\to j}^n(Y_{i\to j}^n) \tag{3}$$

Minimizing the KL-divergence $\mathbf{D}(Q||P)$ between $P(\mathbf{X}, \mathbf{Y})$ and $Q(\mathbf{X}, \mathbf{Y})$ yields the following updating equations:

*Currently working in Waymo.

1) For the *tracklet nodes*, we have

$$\begin{aligned}
&Q_i^{(t)}(x_i = l)\\
=&\frac{1}{Z_i}\exp\left\{-\psi_u^x(l) - \sum_{j\in\mathcal{N}(i)}\sum_{l'}\sum_{k\in\{0,1\}}\psi_t^p(l,l',k)Q_j^{(t-1)}(l')Q_{i\to j}^{p,(t-1)}(k) - \sum_{j\in\mathcal{N}(i)}\sum_{l'}\sum_{k\in\{0,1\}}\psi_t^n(l,l',k)Q_j^{(t-1)}(l')Q_{i\to j}^{n,(t-1)}(k)\right\}\\
=&\frac{1}{Z_i}\exp\left\{-\psi_u^x(l) - \alpha_p\sum_{j\in\mathcal{N}(i)}Q_{i\to j}^{p,(t-1)}(1)\sum_{l'\neq l}Q_j^{(t-1)}(l') - \alpha_n\sum_{j\in\mathcal{N}(i)}Q_{i\to j}^{n,(t-1)}(1)Q_j^{(t-1)}(l)\right\}\\
=&\frac{1}{Z_i}\exp\left\{T_{gt}s_{li}^{gt} + \alpha_p\sum_{j\in\mathcal{N}(i)}Q_{i\to j}^{p,(t-1)}(1)Q_j^{(t-1)}(l) - \alpha_n\sum_{j\in\mathcal{N}(i)}Q_{i\to j}^{n,(t-1)}(1)Q_j^{(t-1)}(l)\right\} \qquad (4)
\end{aligned}$$

where Z_i is the normalization factor and $Q^{(t)}(\cdot)$ is the approximated distribution at the t-th iteration. It is initialized by

$$Q_i^{(0)}(x_i = l) = \frac{1}{Z_i}\exp\{T_{gt}s_{li}^{gt}\} \qquad (5)$$

2) For the *positive gates*, we have

$$\begin{aligned}
Q_{i\to j}^{p,(t)}(y_{i\to j}^p = 1) =&\frac{1}{Z_{i\to j}^p}\exp\left\{-\psi_u^p(1) - \sum_{l,l'}\sum_{j\in\mathcal{N}(i)}\psi_t^p(l,l',1)Q_i^{(t-1)}(l)Q_j^{(t-1)}(l')\right\}\\
=&\frac{1}{Z_{i\to j}^p}\exp\left\{-\psi_u^p(1) - \alpha_p\sum_{l'\neq l}Q_i^{(t-1)}(l)Q_k^{(t-1)}(l')\right\}\\
=&\frac{1}{Z_{i\to j}^p}\exp\left\{T_{tt}s_{ij}^{tt} + \alpha_p\sum_{l}Q_i^{(t-1)}(l)Q_j^{(t-1)}(l) - \alpha_p\right\} \qquad (6)
\end{aligned}$$

For normalization purpose, we set the factor $Z_{i\to j}^p$ so that $\sum_{j\in\mathcal{N}(i)}Q_{i\to j}^{p,(t)}(1) = 1$. Thus we have

$$Q_{i\to j}^{p,(t)}(y_{i\to j}^p = 1) = \frac{1}{Z_{i\to j}^p}\exp\left\{T_{tt}s_{ij}^{tt} + \alpha_p\sum_{l}Q_i^{(t-1)}(l)Q_j^{(t-1)}(l)\right\} \qquad (7)$$

It is initialized by

$$Q_{i\to j}^{p,(0)}(y_{i\to j}^p = 1) = \frac{1}{Z_{i\to j}^p}\exp\{T_{tt}s_{ij}^{tt})\} \qquad (8)$$

3) For the *negative gates*, we have

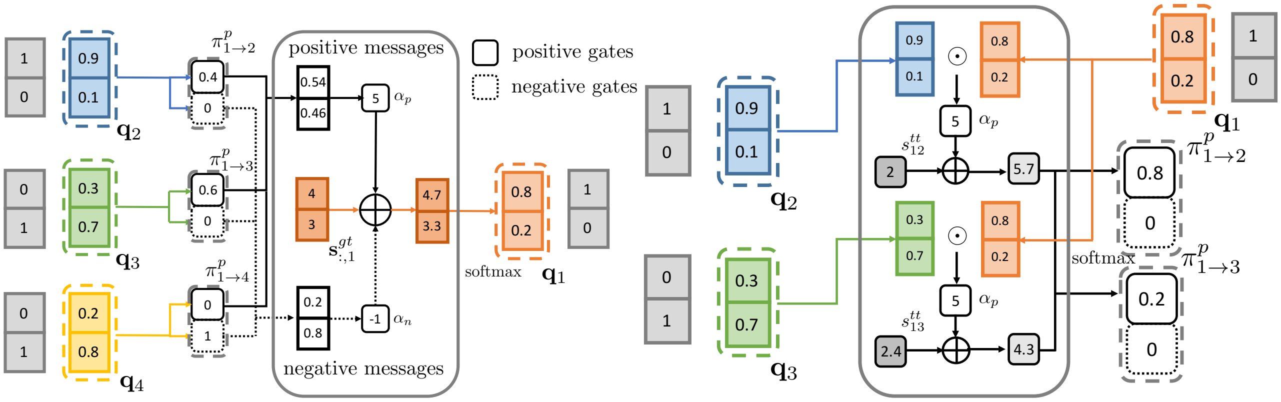
$$\begin{aligned}
Q_{i\to j}^{n,(t)}(y_{i\to j}^n = 1) =&\frac{1}{Z_{i\to j}^n}\exp\left\{-\psi_u^n(1) - \sum_{l,l'}\sum_{j\in\mathcal{N}(i)}\psi_t^n(l,l',1)Q_i^{(t-1)}(l)Q_j^{(t-1)}(l')\right\}\\
=&\frac{1}{Z_{i\to j}^n}\exp\left\{-\psi_u^n(1) - \alpha_n\sum_{l}Q_i^{(t-1)}(l)Q_j^{(t-1)}(l)\right\} \qquad (9)
\end{aligned}$$

Since

$$\psi_u^n(y_{i\to j}^n = k) = \begin{cases}0 & \text{if } L_{ij}^{tt} = k\\ +\infty & \text{otherwise}\end{cases} \qquad (10)$$

for $k \in \{0, 1\}$, we have

$$Q_{i\to j}^{n,(t)}(y_{i\to j}^n = k) = \begin{cases}k & \text{if } L_{ij}^{tt} = 1\\ 1-k & \text{otherwise}\end{cases} \qquad (11)$$



(a) Update Sample Distributions

(b) Update Gate Probabilities

Figure 1: (a) shows the update of $\mathbf{q}_1$. Distribution of the neighbors are weighted by the probability of opening gates and collected as positive and negative messages, respectively. The new marginal distribution is updated by the sum of messages and the unary scores. Grey boxes are the ground truth labels of samples. (b) shows the update of gate $\pi^p_{1\to 2}$ and $\pi^p_{1\to 3}$. Distributions of sample node pairs are used to modify the marginal probability of positive gates. We can see that the connection between sample 1 and 3 is misleading since s^{tt}_{13} is large but they belong to different identities. After updating the probability of gates by utilizing the information from neighboring nodes, $\pi^p_{1\to 3}$ drops comparing to (a), results in less positive information passing between sample 1 and 3 in the next iteration. $\odot$ is inner product operation.

for $k \in \{0, 1\}, t = 0, \ldots, K$.

Let $\mathbf{q}_i^{(t)} = \left[Q_i(1)^{(t)} \quad \cdots \quad Q_i(C)^{(t)}\right]^T$ be the identity distribution vector of node i at the t-th iteration. $\pi^{p,(t)}_{i\to j} = Q^{p,(t)}_{i\to j}(1)$ and $\pi^{n,(t)}_{i\to j} = Q^{n,(t)}_{i\to j}(1)$ be the probability of opened positive and negative gates on edge $i \to j$ respectively, we have the following message passing equations

$$\begin{aligned}\mathbf{q}_i^{(0)} &= \text{softmax}(T_{gt}\mathbf{S}^{gt}_{:,i})\\ \mathbf{q}_i^{(t)} &= \text{softmax}(T_{gt}\mathbf{S}^{gt}_{:,i} + \alpha_p \sum_{j\in\mathcal{N}(i)} \pi^{p,(t-1)}_{i\to j}\mathbf{q}_j^{(t-1)} - \alpha_n \sum_{j\in\mathcal{N}(i)} \pi^{n,(t-1)}_{i\to j}\mathbf{q}_j^{(t-1)})\end{aligned} \tag{12}$$

where $\mathbf{S}^{gt}_{:,i}$ is the ith column of $\mathbf{S}^{gt}$. And

$$\begin{aligned}\pi^{p,(0)}_{i\to j} &= \underset{\mathcal{N}(i)}{\text{softmax}}(T_{tt}s^{tt}_{ij})\\ \pi^{p,(t)}_{i\to j} &= \underset{\mathcal{N}(i)}{\text{softmax}}(T_{tt}s^{tt}_{ij} + \alpha_p\mathbf{q}_i^{(t-1)} \cdot \mathbf{q}_j^{(t-1)})\\ \pi^{n,(t)}_{i\to j} &= L^{tt}_{ij}\end{aligned} \tag{13}$$

Thus the marginal probability of negative relation is fixed and not updated in the iterations.

Two illustrations of message passing and node update are shown in Figure 1.

2. Implementation Details for Section 4.2

2.1. Cast Search in Movies

2.1.1 Pre-processing details

For the CSM dataset, we use the 256-dimensional facial and body features provided by [7]. We first flatten both facial and body features in each tracklet by average pooling. Denote facial features for galleries as $\mathbf{F}^g_F$, flattened facial features for tracklets as $\mathbf{F}^t_F$ and flattened body features for tracklets as $\mathbf{F}^t_B$. Three linear embedding matrices $\mathbf{W}^{gt}_F$, $\mathbf{W}^{tt}_F$, $\mathbf{W}^{tt}_B$, all with size 256 × 256 are applied on the features respectively for more discriminative representation.

We use the cosine similarity between $\mathbf{W}^{gt}_F\mathbf{F}^g_F$ and $\mathbf{W}^{gt}_F\mathbf{F}^t_F$ as the gallery-to-tracklet similarity $\mathbf{S}^{gt} = \mathbf{S}^{gt}_{F,cos}$. To improve the reliability of positive connections, we use the fusion of the cosine similarities between $\mathbf{W}^{tt}_F\mathbf{F}^t_F$ and between $\mathbf{W}^{tt}_B\mathbf{F}^t_B$

as the tracklet-to-tracklet similarity $\mathbf{S}^{tt} = \lambda_f \mathbf{S}^{tt}_{F,cos} + (1 - \lambda_f)\mathbf{S}^{tt}_{B,cos}$, with fusion weight λ_f. No detection information is provided in this dataset so the cannot-link matrix $\mathbf{L}^{tt}$ is all-zero. We feed $\mathbf{S}^{gt}$, $\mathbf{S}^{tt}$ and $\mathbf{L}^{tt}$ into the proposed UGG module. The module iterates for K iterations and produce the output similarity $\tilde{\mathbf{S}}^{gt}$.

2.1.2 Testing details

For testing, we use all the tracklets in each movie to build the graph. The neighborhood $\mathcal{N}(i)$ for tracklet i is defined as the top 10% of the tracklets in the movie with the largest tracklet-to-tracklet similarity score to tracklet i. We apply identity embedding matrices on the features and compute similarities. Then the UGG module is used to produce the output similarity scores $\tilde{\mathbf{S}}^{gt}$. Using the validation set, we choose parameters $T_{gt} = 10$, $T_{tt} = 15$, $\alpha_p = 5$, $K = 2$, $\lambda = 0.1$ and $\lambda_f = 0.1$ for the IN protocol and $T_{gt} = 20$, $T_{tt} = 30$, $\alpha_p = 15$, $K = 2$, $\lambda = 0.1$ and $\lambda_f = 0.1$ for the ACROSS protocol.

2.1.3 Training details

For end-to-end training, we train the embedding matrices $\mathbf{W}^{gt}_F$, $\mathbf{W}^{tt}_F$, and $\mathbf{W}^{tt}_B$, together with temperatures T_{gt} and T_{tt} in the UGG module, implemented in PyTorch [10]. For each movie, we use all the galleries and randomly pick $1/8$ of the tracklets to construct the graph. The overall loss is computed by (14) in the paper. The network is trained using Adam solver [8] for 20 epochs with batch size 2 (2 movies in each batch). The initial learning rate is 1×10^{-4}. All embedding matrices are initialized as identity matrix. We initialize T_{gt} and T_{tt} by 10 and 15 respectively and fix other parameters as $\alpha_p = 5$, $K = 2$, $\lambda = 0.1$ and $\lambda_f = 0.1$ during training.

2.2. IARPA Janus Surveillance Video Benchmark

2.2.1 Pre-processing details

For the IJB-S dataset, we follow the pre-processing steps in [16]. We employ the multi-scale face detector DPSSD [12] to detect faces in surveillance videos. We use the facial landmark branch of All-in-One Face [13] as the fiducial detector. Face alignment is performed using the seven-point similarity transform. Similar to [16], we use a ResNet-101 [6] and a Inception-ResNet-v2 [15], both trained on the union of the MSCeleb-1M dataset [4], the UMDFaces dataset [1], and the UMDFaces Video dataset with the crystal loss [11], to represent the faces. A triplet probabilistic embedding (TPE) [14] trained on the UMDFaces dataset is applied on face features for dimensionality reduction to 128.

We also use the Mask R-CNN [5] implemented on Detectron [3] to detect the bodies in the videos and match each body to the face with the highest overlap ratio. The detected bodies are represented by a re-id network with ResNet-50 architecture trained on the Market1501 dataset [17], implemented on [18]. The network produces 2048-dimensional feature for each body.

We use SORT [2] to construct tracklets for every face appearing in the videos. Facial and body features are first flattened by average pooling for each gallery and tracklet. $\mathbf{S}^{gt}$ and $\mathbf{S}^{tt}$ are computed in the same way as the CSM dataset, except there is no embedding matrices applied since no training set available on IJB-S. We use the bounding box information from the detector to build the co-occurence cannot-link matrix $\mathbf{L}^{tt}$ such that all the tracklets with distinct bounding boxes appear in the same frame will have cannot-links between them.

2.2.2 Testing details

For the IJB-S dataset we empirically use the hyperparameter configuration of $T_{gt} = 15$, $T_{tt} = 15$, $\alpha_p = 10$, $\alpha_n = 2$, $K = 4$, $\lambda = 0.1$ and $\lambda_f = 0.1$ in the UGG module for testing. All the other details are the same as the CSM dataset.

3. Implementation Details for Section 4.6

For the semi-supervised training experiments, we follow the training settings on the CSM dataset in general. The differences are

- For each movie, we use all the galleries and randomly pick about $1/4$ of the tracklets to construct the graph. Then we randomly pick 25% tracklets in the graph as labeled samples, and the rest 75% as unlabeled samples.
- We only train the 256×256 linear embedding matrix $\mathbf{W}^{gt}_F$ on the face features. Other embeddings are fixed as identity matrices.

- During training, we fix the parameters as $T_{gt} = 10$, $T_{tt} = 15$, $\alpha_p = 5$, $K = 2$, $\lambda = 0$ and $\lambda_f = 0.1$. $\lambda = 0$ because we are not training the pairwise embeddings.

References

[1] Ankan Bansal, Anirudh Nanduri, Carlos D. Castillo, Rajeev Ranjan, and Rama Chellappa. Umdfaces: An annotated face dataset for training deep networks. *IEEE International Joint Conference on Biometrics (IJCB)*, 2017. 4

[2] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *ICIP*, pages 3464–3468, 2016. 4

[3] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. Detectron. https://github.com/facebookresearch/detectron, 2018. 4

[4] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. MS-Celeb-1M: A dataset and benchmark for large scale face recognition. In *ECCV*, 2016. 4

[5] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017. 4

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *arXiv prepring arXiv:1506.01497*, 2015. 4

[7] Qingqiu Huang, Wentao Liu, and Dahua Lin. Person search in videos with one portrait through visual and temporal links. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 437–454, Cham, 2018. Springer International Publishing. 3

[8] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015. 4

[9] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 109–117. Curran Associates, Inc., 2011. 1

[10] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017. 4

[11] Rajeev Ranjan, Ankan Bansal, Hongyu Xu, Swami Sankaranarayanan, Jun-Cheng Chen, Carlos D. Castillo, and Rama Chellappa. Crystal loss and quality pooling for unconstrained face verification and recognition. *CoRR*, abs/1804.01159, 2018. 4

[12] Rajeev Ranjan, Ankan Bansal, Jingxiao Zheng, Hongyu Xu, Josh Gleason, Boyu Lu, Anirudh Nanduri, Jun-Cheng Chen, Carlos D. Castillo, and Rama Chellappa. A fast and accurate system for face detection, identification, and verification. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 1(2):82–96, 2019. 4

[13] Rajeev Ranjan, Swami Sankaranarayanan, Carlos D. Castillo, and Rama Chellappa. An all-in-one convolutional neural network for face analysis. In *12th IEEE FG*, volume 00, pages 17–24, May 2017. 4

[14] Swami Sankaranarayanan, Azadeh Alavi, Carlos D. Castillo, and Rama Chellappa. Triplet probabilistic embedding for face verification and clustering. In *BTAS*, 2016. 4

[15] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, 2017. 4

[16] Jingxiao Zheng, Rajeev Ranjan, Ching-Hui Chen, Jun-Cheng Chen, Carlos D. Castillo, and Rama Chellappa. An automatic system for unconstrained video-based face recognition. *CoRR*, abs/1812.04058, 2018. 4

[17] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Computer Vision, IEEE International Conference on*, 2015. 4

[18] Kaiyang Zhou. Deep-person-reid. https://github.com/KaiyangZhou/deep-person-reid. 4