Surpassing Human-Level Face Verification Performance on LFW with GaussianFace (Supplementary Material)

Chaochao Lu **Xiaoou Tang**
Department of Information Engineering
The Chinese University of Hong Kong
{lc013, xtang}@ie.cuhk.edu.hk

Optimization

As discussed in the main text, learning the GaussianFace model can amount to minimizing the following marginal likelihood,

$$\mathcal{L}_{Model} = -\log p(\mathbf{Z}_T, \boldsymbol{\theta}|\mathbf{X}_T) - \beta\mathcal{M}. \quad (1)$$

For the model optimization, we first expand Equation (1) to obtain the following equation (ignoring the constant items)

$$\mathcal{L}_{Model} = -\log P_T + \beta P_T \log P_T + \frac{\beta}{S}\sum_{i=1}^{S}\big(P_{T,i}\log P_i - P_{T,i}\log P_{T,i}\big), \quad (2)$$

where $P_i = p(\mathbf{Z}_i, \boldsymbol{\theta}|\mathbf{X}_i)$ and $P_{i,j}$ means that its corresponding covariance function is computed on both $\mathbf{X}_i$ and $\mathbf{X}_j$.

To obtain the optimal $\boldsymbol{\theta}$ and $\mathbf{Z}$, we need to optimize Equation (2) with respect to $\boldsymbol{\theta}$ and $\mathbf{Z}$, respectively. We first present the derivations of hyper-parameters $\boldsymbol{\theta}$. It is easy to get

$$\begin{aligned}\frac{\partial\mathcal{L}_{Model}}{\partial\theta_j} =& \Big(\beta(\log P_T + 1) - \frac{1}{P_T}\Big)\frac{\partial P_T}{\partial\theta_j} \\ &+ \frac{\beta}{S}\sum_{i=1}^{S}\frac{P_{T,i}}{P_i}\cdot\frac{\partial P_i}{\partial\theta_j} \\ &+ \frac{\beta}{S}\sum_{i=1}^{S}(\log P_i - \log P_{T,i} - 1)\frac{\partial P_{T,i}}{\partial\theta_j}.\end{aligned}$$

The above equation depends on the form $\frac{\partial P_i}{\partial\theta_j}$ as follows (ignoring the constant items)

$$\begin{aligned}\frac{\partial P_i}{\partial\theta_j} =& P_i\frac{\partial\log P_i}{\partial\theta_j} \\ \approx& P_i\Big(\frac{\partial\log p(\mathbf{X}_i|\mathbf{Z}_i,\boldsymbol{\theta})}{\partial\theta_j} + \frac{\partial\log p(\mathbf{Z}_i)}{\partial\theta_j} + \frac{\partial\log p(\boldsymbol{\theta})}{\partial\theta_j}\Big).\end{aligned}$$

The above three terms can be easily obtained (ignoring the constant items) by

$$\begin{aligned}\frac{\partial\log p(\mathbf{X}_i|\mathbf{Z}_i,\boldsymbol{\theta})}{\partial\theta_j} \approx& -\frac{D}{2}\mathrm{Tr}\Big(\mathbf{K}^{-1}\frac{\partial\mathbf{K}}{\partial\theta_j}\Big) \\ &+ \frac{1}{2}\mathrm{Tr}\Big(\mathbf{K}^{-1}\mathbf{X}_i\mathbf{X}_i^\top\mathbf{K}^{-1}\frac{\partial\mathbf{K}}{\partial\theta_j}\Big), \\ \frac{\partial\log p(\mathbf{Z}_i)}{\partial\theta_j} \approx& -\frac{1}{\sigma^2}\frac{\partial J_i^*}{\partial\theta_j} \\ =& -\frac{1}{\lambda\sigma^2}\Big(\mathbf{a}^\top\frac{\partial\mathbf{K}}{\partial\theta_j}\mathbf{a} - \mathbf{a}^\top\frac{\partial\mathbf{K}}{\partial\theta_j}\tilde{\mathbf{A}}\mathbf{a} \\ &+ \mathbf{a}^\top\mathbf{K}\tilde{\mathbf{A}}\frac{\partial\mathbf{K}}{\partial\theta_j}\tilde{\mathbf{A}}\mathbf{K}\mathbf{a} - \mathbf{a}^\top\mathbf{K}\tilde{\mathbf{A}}\frac{\partial\mathbf{K}}{\partial\theta_j}\mathbf{a}\Big), \\ \frac{\partial\log p(\boldsymbol{\theta})}{\partial\theta_j} =& \frac{1}{\theta_j},\end{aligned}$$

where $\tilde{\mathbf{A}} = \mathbf{A}(\lambda\mathbf{I}_n + \mathbf{A}\mathbf{K}\mathbf{A})^{-1}\mathbf{A}$.

Similarly, the derivations of latent space $\mathbf{Z}$ can be also obtained.

GaussianFace Inference for Face verification

In the section of **GaussianFace Model for Face Verification**, when given a test feature vector $\mathbf{x}_*$ we need to estimate it latent representation $\mathbf{z}_*$. In this paper, we apply the same method in (Urtasun and Darrell 2007) to inference. For convenience a brief review is given here.

Given $\mathbf{x}_*$, $\mathbf{z}_*$ can be obtained by optimizing

$$\mathcal{L}_{Inf} = \frac{\|\mathbf{x}_* - \mu(\mathbf{z}_*)\|^2}{2\sigma^2(\mathbf{z}_*)} + \frac{D}{2}\ln\sigma^2(\mathbf{z}_*) + \frac{1}{2}\|\mathbf{z}_*\|^2, \quad (3)$$

where

$$\begin{aligned}\mu(\mathbf{z}_*) &= \mu + \mathbf{X}^\top\mathbf{K}^{-1}\mathbf{K}_*, \\ \sigma^2(\mathbf{z}_*) &= \mathbf{K}_{**} - \mathbf{K}_*^\top\mathbf{K}^{-1}\mathbf{K}_*,\end{aligned}$$

$\mathbf{K}_*$ is the vector with elements $\mathbf{k}_{\boldsymbol{\theta}}(\mathbf{z}_*, \mathbf{z}_i)$ for all other latent points $\mathbf{z}_i$ in the model, and $\mathbf{K}_{**} = \mathbf{k}_{\boldsymbol{\theta}}(\mathbf{z}_*, \mathbf{z}_*)$.

Further Validations: Shuffling the Source-Target

To further prove the validity of our model, we also consider to treat Multi-PIE and MORPH respectively as the target-domain dataset and the others as the source-domain

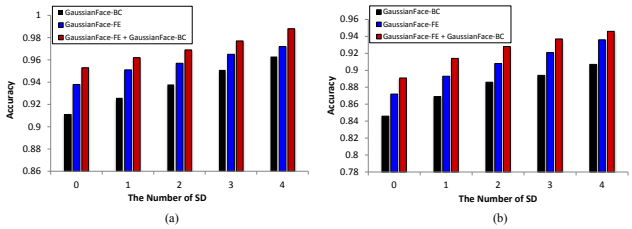

Figure 1: (a) The accuracy rate (%) of the GaussianFace model on Multi-PIE. (b) The accuracy rate (%) of the GaussianFace model on MORPH.

datasets. The target-domain dataset is split into two mutually exclusive parts: one consisting of 20,000 matched pairs and 20,000 mismatched pairs is used for training, the other is used for test. In the test set, similar to the protocol of LFW, we select 10 mutually exclusive subsets, where each subset consists of 300 matched pairs and 300 mismatched pairs. The experimental results are presented in Figure 1. Each time one dataset is added to the training set, the performance can be improved, even though the types of data are very different in the training set.

References

Urtasun, R., and Darrell, T. 2007. Discriminative gaussian process latent variable model for classification. In *ICML*.