This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2533383, IEEE Transactions on Pattern Analysis and Machine Intelligence

An Efficient Joint Formulation for Bayesian Face Verification

Dong Chen, Xudong Cao, David Wipf, Fang Wen, and Jian Sun

Abstract—This paper revisits the classical Bayesian face recognition algorithm from Baback Moghaddam et al. and proposes enhancements tailored to face verification, the problem of predicting whether or not a pair of facial images share the same identity. Like a variety of face verification algorithms, the original Bayesian face model only considers the appearance difference between two faces rather than the raw images themselves. However, we argue that such a fixed and blind projection may prematurely reduce the separability between classes. Consequently, we model two facial images jointly with an appropriate prior that considers intra- and extra-personal variations over the image pairs. This joint formulation is trained using a principled EM algorithm, while testing involves only efficient closed-formed computations that are suitable for real-time practical deployment. Supporting theoretical analyses investigate computational complexity, scale-invariance properties, and convergence issues. We also detail important relationships with existing algorithms, such as probabilistic linear disciminant analysis (PLDA) and metric learning. Finally, on extensive experimental evaluations, the proposed model is superior to the classical Bayesian face algorithm and many alternative state-of-the-art supervised approaches, achieving the best test accuracy on three challenging datasets, Labeled Face in Wild (LFW), Multi-PIE, and YouTube Faces, all with unparalleled computational efficiency.

Index Terms—Bayesian face recognition, face verification, EM algorithm

1 Introduction

Face verification and face identification represent two sub-problems of face recognition. The former involves verifying whether or not two given faces belong to the same person, while the latter answers the question of which identity should be assigned to a probe face set given a gallery of candidates. In this paper, we focus on the verification problem, which is somewhat more widely applicable and lays the foundation for challenging identification problems.

Bayesian face recognition [1] by Baback Moghaddam et al. represents one successful algorithm that has been applied to face verification. It formulates the verification task as a binary Bayesian decision problem. Let H_I represents the intra-personal hypothesis that two faces x_1 and x_2 belong to the same subject, and H_E is the extra-personal hypothesis that two faces are from different subjects. Then, the face verification problem amounts to classifying the difference $\Delta = x_1 - x_2$ as intra-personal variation or extra-personal variation using the log-likelihood ratio statistic

$$r(x_1, x_2) = \log \frac{P(\Delta | H_I)}{P(\Delta | H_E)}. \quad (1)$$

This ratio can be considered as a probabilistic measure of similarity between x_1 and x_2 for the face verification problem. In [1], the two conditional probabilities in (1) are modeled as Gaussians and eigen-analysis is used for model learning and efficient computation.

• *D. Chen X. Cao, D. Wipf, F. Wen and J. Sun are with the Visual Computing Group, Microsoft Research, Beijing, China.*
E-mail: {doch, xudongca, davidwip, fangwen, jiansun}@microsoft.com

Because of the simplicity and competitive performance [2] of Bayesian face recognition, further progress has been made along this line of research. For example, Wang and Tang [3] propose a unified framework for subspace face recognition which decomposes the face difference into three subspaces: intrinsic difference, transformation difference, and noise. By excluding the transform difference and noise and retaining the intrinsic difference, better performance is obtained. In [4], a random subspace is introduced to handle the multi-model and high dimension problem. The appearance difference can be also computed in any feature space such as Gabor feature [5]. Instead of using a naive Bayesian classifier, a SVM is trained in [6] to classify the the difference face which is projected and whitened in an intra-person subspace.

One commonality of all of these Bayesian face methods is that discrimination is based solely on the difference between a pair of face images. As illustrated by a 2D example in Figure 1, modeling the difference can be viewed as first projecting all 2D points onto a 1D line (X-Y) and then performing classification in 1D. While such a projection may retain some important discriminative information, it may nonetheless reduce the separability of the classes. Therefore, the power of Bayesian face framework may be limited by discarding the discriminative information when we view two classes jointly in the original feature space.

In this paper, we propose to directly model the full joint distribution of $\{x_1, x_2\}$ for the face verification problem in a similar Bayesian framework. We begin by introducing an appropriate parametric prior on face representations in Section 2, where each face

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2533383, IEEE Transactions on Pattern Analysis and Machine Intelligence

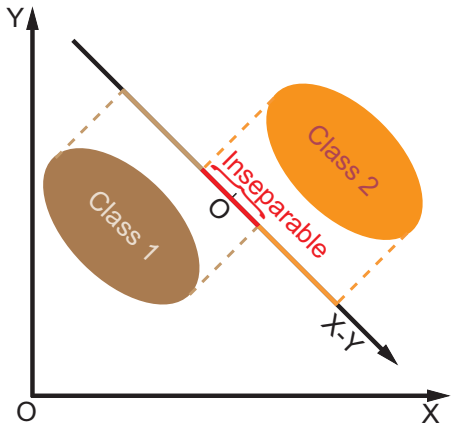

Fig. 1: The 2-D data is projected to 1-D by $x - y$. The two classes, which are separable in the original joint representation, become inseparable after the projection. In the context of face verification, "Class1" and "Class2" would refer to the two hypotheses H_I and H_E.

feature is expressed as the summation of two independent Gaussian latent variables, one related to identity, and another that captures intra-personal variations. This prior allows us to obtain the joint distributions of $\{x_1, x_2\}$ under either H_I or H_E, leading to a closed-form expression for a log-likelihood ratio classification rule analogous to (1).

Model parameters are estimated using the EM algorithm in Section 3, while low-rank and invariance properties germane to efficient training and testing are rigorously explored in Section 4. In Section 5, we next scrutinize similarities and differences between the proposed algorithm and probabilistic linear discriminant analysis (PLDA) [7], [8], a widely-used technique for face verification that is based on a related probabilistic factor analysis model. This is followed by further analytical comparisons with metric learning [9], [10], [11], [12] and reference-based methods [13], [14], [15], [16] in Section 6. Finally, extensive empirical validations are presented in Section 7, where we demonstrate state-of-the-art face verification performance on challenging datasets: WDRef [17], Labeled Face in Wild (LFW) [18], Multi-PIE [19], and YouTube Faces [20]. All proofs are deferred to Section 8. Overall our primary high-level contributions are summarized as follows:

- The derivation and theoretical analysis of a robust, practical face verification system that is well-suited for large-scale, real-time environments.
- The detailed situating of this algorithm in the context of existing methods and theory, pinpointing important distinctions that lead to improvements in both accuracy and efficiency. For example, we rigorously investigate the connection with PLDA, highlighting previously unexamined distinctions in the corresponding update rules that can impact performance. We envision that this analysis may have wider consequences in probabilistic models with related parameterizations.
- The empirical demonstration of face verification accuracy exceeding state-of-the-art algorithms on challenging benchmark data.

Note that portions of this work appeared in recent conference proceedings [17] and to a lesser extent [21]; however, here we include full proofs and algorithm derivations, expanded empirical results, and new theoretical analyses and comparisons.

2 Our Approach: A Joint Formulation

In this section, we first present a naive joint formulation and then introduce our proposed alternative and attendant model learning algorithm.

2.1 A naive formulation

A straightforward joint formulation is to directly model the joint distribution of $\{x_1, x_2\}$ as a Gaussian. Thus, we have $P(x_1, x_2|H_I) = N(0, \Sigma_I)$ and $P(x_1, x_2|H_E) = N(0, \Sigma_E)$, where covariance matrixes Σ_I and Σ_E can be estimated from a large canon of intra-personal pairs and extra-personal pairs respectively. The mean of all faces is first subtracted as a preprocessing step. At test time, the log-likelihood ratio between $P(x_1, x_2|H_I)$ and $P(x_1, x_2|H_E)$ is used as the similarity metric just as in (1). As will be seen in later experiments, such a naive formulation is moderately better than the original Bayesian face algorithm.

Estimating the covariance matrices Σ_I and Σ_E directly from the data may be ineffective though because of two factors. First, if face images x_1 and x_2 are represented by d-dimensional features, then the naive formulation requires the estimation of joint covariance matrices of size $2d \times 2d$. This size could be prohibitively large for robust estimation with limited training samples. Secondly, because training samples may not be entirely independent, the estimated Σ_E may not be block diagonal, although ideally it should be if x_1 and x_2 are truly statistically independent samples from different subjects. To address these issues, we next introduce a simple prior on the face representation to form a more robust joint Bayesian formulation.

2.2 A joint formulation with face prior

As assumed by previous models [8], [22], [23], [24], the appearance of a face can be well-approximated by two additive factors: identity and intra-personal variation. A face is represented by the sum of two independent Gaussian variables

$$x = \mu + \varepsilon, \tag{2}$$

where x is the observed face feature vector with the mean of all faces subtracted, μ represents and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2533383, IEEE Transactions on Pattern Analysis and Machine Intelligence

identity component, and ε is remaining intra-personal variations that reflect differences in lighting, pose, and expression within a given identity. The latent variables μ and ε are distributed independently as $N(0, S_\mu)$ and $N(0, S_\varepsilon)$ respectively, where S_μ and S_ε are two unknown covariance matrixes. Together these distributions constitute a prior distribution on faces.

Given this prior, the joint distribution of $\{x_1, x_2\}$ is Gaussian with zero mean under either H_E or H_I. Based on the linear form of (2) and the independent assumption between μ and ε, the covariance of two faces is

$$\begin{aligned}\mathbf{cov}(x_i, x_j) &= \mathbf{cov}(\mu_i, \mu_j) + \mathbf{cov}(\varepsilon_i, \varepsilon_j), \\ & i, j \in \{1, 2\}.\end{aligned} \tag{3}$$

Under hypothesis H_I, the identity μ_1, μ_2 of the pair are the same and their intra-person variations ε_1, ε_2 are independent. Consequently, $P(x_1, x_2|H_I)$ is a zero-mean Gaussian with covariance

$$\Sigma_I = \begin{bmatrix} S_\mu + S_\varepsilon & S_\mu \\ S_\mu & S_\mu + S_\varepsilon \end{bmatrix}.$$

In contrast, assuming H_E, both the identities and intra-person variations are independent. Hence, the covariance matrix of the distribution $P(x_1, x_2|H_E)$ is

$$\Sigma_E = \begin{bmatrix} S_\mu + S_\varepsilon & 0 \\ 0 & S_\mu + S_\varepsilon \end{bmatrix}.$$

Given these two conditional joint probabilities for H_I and H_E, the log-likelihood ratio $r(x_1, x_2)$ can be obtained in closed-form after a series of linear algebra manipulations resulting in

$$\begin{aligned} r(x_1, x_2) &= \log \frac{P(x_1, x_2|H_I)}{P(x_1, x_2|H_E)} \\ &= x_1^T A x_1 + x_2^T A x_2 - 2 x_1^T G x_2, \end{aligned} \tag{4}$$

where

$$A = (S_\mu + S_\varepsilon)^{-1} - (F + G),$$

and F and G satisfy

$$\begin{bmatrix} F + G & G \\ G & F + G \end{bmatrix} = \begin{bmatrix} S_\mu + S_\varepsilon & S_\mu \\ S_\mu & S_\mu + S_\varepsilon \end{bmatrix}^{-1}. \tag{5}$$

Note that constant factors have been omitted from (4) for simplicity.

From (4) we have a candidate test statistic for face verification which only depends on estimating the $d \times d$ covariance matrices S_μ and S_ε, which are of significantly lower dimension than the requirements from the naive joint formulation. However, we still require a careful learning procedure to ensure reliable estimates. In the forthcoming sections we will derive an efficient algorithm along with complementary analyses and justifications. Highlights include the following:

1) The matrices A and G in (4) produced by the proposed training pipeline will ultimately be negative semi-definite and low rank, enabling a highly efficient testing implementation (see Section 4.2).
2) Both the learning procedure for the matrices S_μ and S_ε, and the resulting log-likelihood ratio metric r, are invariant to any invertible linear transform of the face features, implying reduced manual intervention (see Section 4.3).
3) While perhaps deceptively simple upon first inspective, we carefully examine important characteristics that differentiate the proposed Joint Bayesian model from existing face verification algorithms including PLDA, metric learning, and reference-based methods (see Sections 5 and 6).

3 Parameters Estimation

As described in the previous section, the unknown parameters that we need to learn are the covariance matrices of identity and intra-person variation $\Theta = \{S_\mu, S_\varepsilon\}$. In this section, we develop an EM algorithm to estimate the covariances by maximizing the log-likelihood function.

3.1 The log-likelihood function

Due to the independence of each subject, the overall log-likelihood function is the summation of the log-likelihood function of each individual subject. Therefore our goal will be to solve

$$\begin{aligned} \min_{S_\mu, S_\varepsilon} \quad & -\textstyle\sum_i \log P(\mathbf{x}_i|S_\mu, S_\varepsilon) \\ \text{s.t.} \quad & S_\mu \succeq 0,\ S_\varepsilon \succeq 0 \end{aligned} \tag{6}$$

The likelihood term $P(\mathbf{x}_i|S_\mu, S_\varepsilon)$ for subject i is derived based on the following generative process. An identity factor μ_i is first drawn from $N(0, S_\mu)$. Subsequently m_i i.i.d. intra-person variations $[\varepsilon_{i1}; ...; \varepsilon_{im_i}]$ are drawn from $N(0, S_\varepsilon)$. The observed m_i samples are then given by the stacked vector $\mathbf{x}_i = [\mu_i + \varepsilon_{i1}; ...; \mu_i + \varepsilon_{im}]$. In matrix notation we have

$$\mathbf{x}_i = Q_i \mathbf{h}_i, \quad \text{where } Q_i = \begin{bmatrix} \mathbf{I} & \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{I} & \mathbf{0} & \mathbf{I} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{I} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I} \end{bmatrix}, \tag{7}$$

where the identity and intra-person variations are stacked into a column vector $\mathbf{h}_i = [\mu_i; \varepsilon_{i1}; \cdots; \varepsilon_{im_i}]$, the distribution of which is a Gaussian with a block-diagonal covariance matrix $\Sigma_{h_i} = \mathrm{diag}(S_\mu, S_\varepsilon, ..., S_\varepsilon)$. From this generative process, it follows that the likelihood function of the i^{th} subject is

$$P(\mathbf{x}_i|S_\mu, S_\varepsilon) = N(0, \Sigma_{x_i}), \quad \text{where } \Sigma_{x_i} = Q_i \Sigma_{h_i} Q_i^T, \tag{8}$$

where the covariance matrix Σ_{x_i} has been constructed with the unknown parameters $\Theta = \{S_\mu, S_\varepsilon\}$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2533383, IEEE Transactions on Pattern Analysis and Machine Intelligence

Minimizing (6) with $P(\mathbf{x}_i|S_\mu, S_\varepsilon)$ given by (8) is challenging both because of the constraints needed to enforce proper covariances and the underlying non-convexity of the problem. Consequently, we develop an EM algorithm for this purpose as described next.

3.2 EM algorithm

Instead of directly maximizing the log-likelihood using a brute force technique such as gradient descent, the EM algorithm introduces additional hidden or latent variables into the likelihood, the values of which, if known, would greatly simplify the optimization [25]. At each iteration, the expected log-likelihood of both observed and hidden variables is computed with respect to the conditional distribution of the hidden variables (E-step), and then the parameters are updated by maximizing the resulting functional (M-step). We now unpack each step in detail.

E-step. For our purposes, we choose the identity and intra-person variations $\mathbf{h}_i = [\mu_i; \varepsilon_{i1}; \cdots; \varepsilon_{im}]$ as the hidden variables and consider the joint distribution $P(\mathbf{x}_i, \mathbf{h}_i|\Theta)$. As the observations $\mathbf{x}_i$ are uniquely determined by the hidden variables $\mathbf{h}_i$ in (7), the joint distribution can be simplified into $P(\mathbf{h}_i|\Theta)$. Therefore the expected negative log-likelihood function over hidden and observed variables reduces to

$$-\sum_i E_{P(\mathbf{h}_i|\mathbf{x}_i,\Theta_t)} \log P(\mathbf{h}_i|\Theta_{t+1}), \tag{9}$$

where the expectation is computed on the conditional distribution of $P(\mathbf{h}_i|\mathbf{x}_i, \Theta_t)$ at iteration t. Note the parameters Θ_t in $P(\mathbf{h}_i|\mathbf{x}_i, \Theta_t)$ are known and used in the E-step for calculating the expectation. This is unlike the parameters Θ_{t+1} in the distribution $P(\mathbf{h}_i|\Theta_{t+1})$ which are assumed to be unknown and updated in the M-step below.

Given that the distribution of the hidden variables is a Gaussian, we expand the expected log-likelihood to produce the equivalent

$$\sum_i \log|\Sigma_{h_i}| + \text{trace}(\Sigma_{h_i}^{-1} E[\mathbf{h}_i\mathbf{h}_i^T]), \tag{10}$$

where $E[\cdot]$ is an abbreviated form for representing the expectation computed on the conditional distribution $P(\mathbf{h}_i|\mathbf{x}_i, \Theta_t)$.

The expectation $E[\mathbf{h}_i\mathbf{h}_i^T]$ is the second-order moment of the conditional distribution $P(\mathbf{h}_i|\mathbf{x}_i, \Theta_t)$. As shown in Section 8.1, the conditional distribution $P(\mathbf{h}_i|\mathbf{x}_i, \Theta_t)$ is a Gaussian with first- and second-order moments given by

$$E[\mathbf{h}_i] = \Sigma_{h_i} Q_i^T \Sigma_{x_i}^{-1} \mathbf{x}_i \tag{11}$$

$$E[\mathbf{h}_i\mathbf{h}_i^T] = \Sigma_h - \Sigma_h Q_i^T \Sigma_x^{-1} Q_i \Sigma_h + E[\mathbf{h}_i]E[\mathbf{h}_i]^T. \tag{12}$$

The second-order moments can be simplified further using $E[\mathbf{h}_i\mathbf{h}_i^T] \approx E[\mathbf{h}_i]E[\mathbf{h}_i]^T$. While the full E-step without this approximation can actually be calculated using our model with limited additional computation, in most practical situations we choose not to include this extra term for several reasons. First, generalized EM algorithms (of which this approximation is a special case) enjoy similar convergence properties to regular EM and are widely used in machine learning and signal processing [26]. Secondly, we have observed empirically that the system performance is essentially identical with or without this additional covariance factor, largely because it tends to be negligibly small (several orders of magnitude smaller than $E[\mathbf{h}_i]E[\mathbf{h}_i]^T$, and in certain quantifiable conditions provably equal to zero). And finally, removing this covariance leads to much more transparent analysis as discussed more in Section 4.2.

M-step. In the M-step, we solve for the unknown parameters Θ_{t+1} by optimizing the expected log-likelihood in (10). As the covariance of the hidden variables Σ_{h_i} is a block diagonal matrix, the unknown S_μ and S_ε are effectively decoupled, and (10) can be further simplified to

$$\sum_i \log|S_\mu| + \text{trace}(S_\mu^{-1} E[\mu_i\mu_i^T]) \tag{13}$$
$$+ \sum_i \sum_j \log|S_\varepsilon| + \text{trace}(S_\varepsilon^{-1} E[\varepsilon_{ij}\varepsilon_{ij}^T])$$

where $E[\mu_i\mu_i^T]$ and $E[\varepsilon_{ij}\varepsilon_{ij}^T]$ are the block matrices at the diagonal of $E[\mathbf{h}_i\mathbf{h}_i^T]$ in (11). We can then derive the optimal parameters in a closed form by taking derivatives with respective to S_μ and S_ε and equating with zero, leading to

$$S_\mu = 1/n \sum_i E[\mu_i\mu_i^T] \tag{14}$$

$$S_\varepsilon = 1/k \sum_i \sum_j E[\varepsilon_{ij}\varepsilon_{ij}^T], \tag{15}$$

where n is the number of subjects, and $k = \sum_i m_i$ is the total number of all images of all subjects. The solution is therefore very intuitive: the optimal covariance matrices are the sample covariances of the corresponding expected second-order moments of the hidden variables. Overall then, the proposed EM algorithm handles the aforementioned difficulties in optimizing the raw log-likelihood of the observed data by decoupling the unknown parameters using hidden data. This ultimately produces straightforward, closed-form updates for the M-step.

3.3 Initialization

The proposed likelihood optimization problem (6) is non-convex, and hence it is difficult to guarantee a priori that the global solution can be found except in special situations. The limiting case of large, equally-distributed samples is one such example.

Lemma 1: If the number of samples per subject $m_i = m$ is the same across all subjects, and $m \to \infty$, then

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2533383, IEEE Transactions on Pattern Analysis and Machine Intelligence

the global minimum of (6) is obtained by

$$S_\varepsilon = \frac{1}{mn}\sum_i \sum_j (x_{ij} - \bar{x}_i)(x_{ij} - \bar{x}_i)^T$$
$$S_\mu = \frac{1}{n}\sum_i \bar{x}_i \bar{x}_i^T,$$

where $\bar{x}_i = 1/m \sum_j x_{ij}$.

The proof is deferred to Section 8.2. The asymptotic closed-form solution is similar to the M-step of the aforementioned EM algorithm. The sample mean and sample residual can be viewed as the empirical approximation of the true expected hidden identity and intra-person variation. In the extreme case described by Lemma 1, the sample mean and residual gradually approach the true identity and intra-person variation, and the optimal estimation is the sample covariance.

However in practical scenarios it is intractable to collect infinite samples for each subject. Moreover, it is rare to have an equivalent number of samples from each subject, and inefficient to throw away extra samples to artificially create such a balanced set. Consequently, the proposed EM algorithm, even though non-convex, provides better estimates of the hidden data as well as the unknown covariances compared to raw empirical estimates. And we are always free to initialize with the raw empirical estimates and then press for improvements through the EM iterations. Intuitively, the *fewer* samples per subject, and the more *heterogeneous* the distribution of samples between subjects, the larger the margin of improvement afforded by the proposed EM algorithm.

Of course ultimately the EM algorithm, and the quality of the solutions it produces, is still potentially initialization dependent. However, we have carefully tested the robustness of Joint Bayesian to a wide variety of initialization schemes for both S_μ and S_ε, e.g., different combinations of sample between-class, sample within-class, and randomized covariance matrices). In all cases the algorithm converges to a solution with the same face verification accuracy, and therefore local minima do not appear to be a problem.

4 Training and Testing Considerations

In this section we detail several ways of improving efficiency such that Joint Bayesian can be scaled to large-scale, practical application domains. We also discuss an invariance property which simplifies the feature selection process.

4.1 Efficient Training

The main computational cost of the EM algorithm is from computing (11) in the E-step, which requires both the inverse of a large covariance Σ_{x_i} as well as expensive matrix multiplications. The complexity of computing the inverse of Σ_{x_i} is $O(d^3 m_i^3)$, where d is the dimension of feature and m_i is the number of samples of the i^{th} subject. For example, if $d = 2,000$ and $m_i = 40$, the size of covariance matrix Σ_{x_i} is 80,000. It takes 25GB memory to store such a large matrix and 9 hours to compute its inverse.

Herein, we present an algorithm to reduce the cost by taking the advantage of block-wise structure embedded in Σ_{x_i}. Given that

$$\Sigma_{x_i} = \begin{bmatrix} S_\mu + S_\varepsilon & S_\mu & \cdots & S_\mu \\ S_\mu & S_\mu + S_\varepsilon & \cdots & S_\mu \\ \vdots & \vdots & \ddots & \vdots \\ S_\mu & S_\mu & \cdots & S_\mu + S_\varepsilon \end{bmatrix}.$$

it follows that $\Sigma_{x_i}^{-1}$ has the same block-diagonal structure

$$\Sigma_{x_i}^{-1} = \begin{bmatrix} F + G & G & \cdots & G \\ G & F + G & \cdots & G \\ \vdots & \vdots & \ddots & \vdots \\ G & G & \cdots & F + G \end{bmatrix}$$

for some F and G. By plugging the above matrices into the equation $\Sigma_{x_i}\Sigma_{x_i}^{-1} = I$, we can solve for the unknown matrices giving

$$\begin{aligned} F &= S_\varepsilon^{-1} \\ G &= -(m_i S_\mu + S_\varepsilon)^{-1} S_\mu S_\varepsilon^{-1}. \end{aligned}$$

During training, we only need to compute and store F and G. Compared to directly evaluating $\Sigma_{x_i}^{-1}$, the computational complexity is reduced from $O(d^3 m_i^3)$ to $O\left(d^3\right)$ and the storage complexity from $O(d^2 m_i^2)$ to $O\left(d^2\right)$, where d is the feature dimension. This leads to substantial reductions, for example, if $d = 2,000$ and $m_i = 40$, the running time is reduced from 9 hours to 0.5 seconds and the storage cost is reduced from 25GB to 16MB.

We can also accelerate the required matrix multiplication by taking the advantage of this block-wise structure. The expected hidden variables can be efficiently computed as

$$E[\mu_i] = S_\mu (F + m_i G) \sum_j x_{ij}$$
$$E[\varepsilon_{ij}] = S_\varepsilon F x_{ij} + S_\varepsilon G \sum_j x_{ij}.$$

Here the multiplication of large square matrices (of size dm_i) have been decomposed into the multiplication of the small block matrices (of size d), again leading to significant computational savings. With all of these considerations in mind, the overall training pipeline of Joint Bayesian is shown in Table. 1.

4.2 Efficient testing

As described in Section 2.2, the closed-form solution of the log-likelihood ratio test statistic is,

$$r(x_1, x_2) = x_1^T A x_1 + x_2^T A x_2 - 2 x_1^T G x_2. \quad (16)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2533383, IEEE Transactions on Pattern Analysis and Machine Intelligence

Input

- The training samples $\{x_{ij}\}$; The number of subjects n; The number of the samples of the i^{th} subject m_i; The total number of the samples of all subjects k.

Output

- Joint Bayesian model parameters $\Theta = \{S_\mu, S_\varepsilon\}$.

Joint Bayesian Learning

1) Initialize S_μ and S_ε with the sample covariances.
2) Calculate the expectation of the hidden variables of each subject (E step):

$$\begin{aligned} F &= S_\varepsilon^{-1} \\ G &= -(m_i S_\mu + S_\varepsilon)^{-1} S_\mu S_\varepsilon^{-1} \\ E[\mu_i] &= S_\mu (F + m_i G) \sum_j x_{ij} \\ E[\varepsilon_{ij}] &= S_\varepsilon F x_{ij} + S_\varepsilon G \sum_j x_{ij} \\ E[\mu_i \mu_i^T] &= E[\mu_i] E[\mu_i]^T \\ E[\varepsilon_{ij} \varepsilon_{ij}^T] &= E[\varepsilon_{ij}] E[\varepsilon_{ij}]^T \end{aligned}$$

3) Update the model parameter (M step):

$$\begin{aligned} S_\mu &= \frac{1}{n} \sum_i E[\mu_i \mu_i^T] \\ S_\varepsilon &= \frac{1}{k} \sum_{i,j} E[\varepsilon_{ij} \varepsilon_{ij}^T] \end{aligned}$$

4) Repeat step 2) and step 3) until convergence (5 iterations is usually sufficient).

TABLE 1: Efficient training pipeline of Joint Bayesian algorithm. Note the matrix F and G can be pre-computed to speed up the training. These matrices only depend on the number of samples of a given subject, and we can pre-compute all possible F and G (for all unique values of m_i) at the beginning of each iteration, and retrieve the right one to use according to the number of samples of each subject. This saves considerable computation whenever many of the subjects share the same number of training instances.

Input

- Joint Bayesian model parameters $\Theta = \{S_\mu, S_\varepsilon\}$, and two testing samples x_1 and x_2.

Output

- The similarity of two testing samples $r(x_1, x_2)$.

Pre-computation

1) Calculate the matrices A and G used in the log-likelihood ratio function:

$$A = (S_\mu + S_\varepsilon)^{-1} - \left[(S_\mu + S_\varepsilon) - S_\mu (S_\mu + S_\varepsilon)^{-1} S_\mu\right]^{-1}$$

$$G = -(2S_\mu + S_\varepsilon)^{-1} S_\mu S_\varepsilon^{-1}$$

2) Decompose $-A$ and $-G$ into two low rank matrices P_A and P_G ($\{P_A, P_G\} \in R^{d \times s}$ and $s < d$) using SVD to produce:

$$\begin{aligned} P_A P_A^T &\approx -A \\ P_G P_G^T &\approx -G \end{aligned}$$

Computing Log-Likelihood Ratio

1) Project testing samples with P_A and P_G:

$$\begin{aligned} a_i &= P_A^T x_i \\ g_i &= P_G^T x_i \end{aligned}$$

2) Calculate the log-likelihood ratio of two samples to measure similarity:

$$r(x_1, x_2) = 2g_1^T g_2 - a_1^T a_1 - a_2^T a_2$$

TABLE 2: Efficient testing pipeline of Joint Bayesian algorithm.

Direct calculation requires computations of order $O(d^2)$, where d is the feature dimension. This can be prohibitively high in many practical scenarios that require comparing a query image with all of the reference images in a large database. Fortunately, the Joint Bayesian framework provides a natural mechanism for accelerating testing calculations considerably. This is ultimately possible because estimates of S_μ tend to be low rank, which then leads to low rank values for A and G, culminating in potentially orders of magnitude reductions in computational complexity.

We will now justify these claims in detail. To begin we have the following:

Lemma 2: The matrices A and G satisfy

$$\begin{aligned} A &= -P_A P_A^T, \quad \text{with} \quad \text{rank}[P_A] \leq \text{rank}[S_\mu] \\ G &= -P_G P_G^T, \quad \text{with} \quad \text{rank}[P_G] \leq \text{rank}[S_\mu] \end{aligned}$$

for some matrices P_A and P_G respectively.

The proof of this lemma is included in Section 8.3. Consequently, if S_μ happens to be low rank, then both A and G can be expressed using convenient low-rank factorizations to reduce the complexity of testing. Specifically, we plug the low-rank factorization into the log-likelihood ratio statistic (16) and end up with the modified decision function

$$r(x_1, x_2) = 2g_1^T g_2 - a_1^T a_1 - a_2^T a_2, \tag{17}$$

where $a_i = P_A^T x_i$, and $g_i = P_G^T x_i$ (for $i = 1, 2$) are low-dimensional features obtained by linear projection. Define $s = \text{rank}[S_\mu]$. Then the complexity of this linear projection is no more than $O(ds)$, and the complexity of the new decision function is at most $O(s)$. Therefore the overall complexity is $O(ds)$, which can be considerably smaller with respect to $O(d^2)$.

Importantly, the speedup will be even more significant in the task of face identification and face search, where we need to compute the log-likelihood ratio as a measurement of similarity between an input query sample and all reference samples in the database. In this situation, we can off-line pre-compute the low-dimension features for all samples in the reference database. Given a query sample, we only need to on-line compute its low-dimension feature once and then the new low-cost decision function (similarity metric) from (17) across the entire reference database. This can lead to multiple orders of magnitude speedup if there are many samples in the reference dataset. For example, if $d = 2000$, $s = 200$, and $N = 1,000,000$,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2533383, IEEE Transactions on Pattern Analysis and Machine Intelligence

its takes 2 hours for the direct implementation to go through the entire database, but it only takes 7 seconds for the efficient testing, over one thousand times speedup. A single end-to-end verification test on a pair of images can be done in milliseconds using standard hardware. The efficient testing pipeline of the Joint Bayesian is shown in Table 2.

But of course these efficiencies hinge on S_μ actually being low rank. Consequently, for this overall line of reasoning to have merit it is essential that we quantify why this property is likely to hold. For this purpose we introduce some additional notation. Let $\mathbb{X}$ denote the $d \times k$ matrix of all image features from all subjects, with the j-th column x_j representing the feature vector of image j. Also, define Φ to be the $n \times k$ matrix with i-th row given by all zeros except a vector of m_i ones starting at element index $e_i = \sum_{r=1}^{i-1} m_r + 1$. We may then express the relationship between all latent and observed variables using $\mathbb{X} = \mathbb{E} + \mathbb{M}\Phi$. We now have the following:

Lemma 3: With $E[\mathbf{h}_i \mathbf{h}_i^T] = E[\mathbf{h}_i]E[\mathbf{h}_i]^T$, the iterations from Table 1 are guaranteed to reduce (or leave unchanged once a fixed point is reached) the minimization problem

$$\begin{aligned} \min_{\mathbb{E},\mathbb{M}} \quad & n \log |S_\mu| + k \log |S_\varepsilon| \\ \text{s.t.} \quad & \mathbb{X} = \mathbb{E} + \mathbb{M}\Phi, \\ & S_\mu = \tfrac{1}{n}\mathbb{M}\mathbb{M}^T, \;\; S_\varepsilon = \tfrac{1}{k}\mathbb{E}\mathbb{E}^T. \end{aligned} \tag{18}$$

The proof can be found in the supplementary material from [21]. As discussed previously (and supported by a vast battery of empirical tests), the approximation $E[\mathbf{h}_i \mathbf{h}_i^T] \approx E[\mathbf{h}_i]E[\mathbf{h}_i]^T$ does not significantly affect estimation results. Hence the Joint Bayesian estimator can be well-characterized by a linearly constrained optimization problem, with log-det penalties on the respective covariances S_μ and S_ε. Such a penalty function favors low-rank solutions given that it is a concave, nondecreasing function of the embedded singular values [21]. However, even with the exact computation for $E[\mathbf{h}_i \mathbf{h}_i^T]$, it can be shown that Joint Bayesian still includes a concave, nondecreasing penalty function on the singular values of S_μ (albeit the constraint surface is no longer linear), so regardless of any approximation singular values of S_μ are favored to be shrunk to zero where possible.

To summarize then, the EM algorithm adopted by Joint Bayesian is likely to produce low-rank estimates for S_μ, which subsequently leads to low-rank estimates for A and G, which then ultimately facilitates extremely efficient testing via simple low-dimensional projections.

4.3 Invariance to Invertible Linear Transforms

We conclude this section by formalizing how the proposed Joint Bayesian training and testing pipelines, as implemented via Table 2 and Table 1 respectively, display a desirable form of feature invariance.

Lemma 4: Let W denote an arbitrary invertible linear transform of the d-dimensional face features, and let $W_i = \text{diag}(W, \ldots, W)$ denote a block diagonal matrix with $m_i + 1$ blocks. Then we have the following:

1) *Training invariance*: If S_μ and S_ε represent a global minimum of (6), then $WS_\mu W^T$ and $WS_\varepsilon W^T$ are an optimal solution to

$$\begin{aligned} \min_{S_\mu, S_\varepsilon} \quad & -\textstyle\sum_i \log P(W_i \mathbf{x}_i | S_\mu, S_\varepsilon) \\ \text{s.t.} \quad & S_\mu \in H^+, \;\; S_\varepsilon \in H^+. \end{aligned} \tag{19}$$

2) *Testing invariance*: The likelihood ratio statistic from (4) satisfies $r(x_1, x_2) = r(Wx_1, Wx_2)$.

The proof of this lemma is included in Section 8.4. This result, which is equally valid with or without the approximation $E[\mathbf{h}_i \mathbf{h}_i^T] = E[\mathbf{h}_i]E[\mathbf{h}_i]^T$, then implies that we need not be concerned about linear transformations of the features, nor with motivating what the optimal feature representation actually is. It is worth noting that these desirable invariance properties are quite unlike other sparse or low-rank models that incorporate, for example, convex penalties such as the ℓ_1 norm or the nuclear norm. With these penalties an invertible linear transform would lead to an entirely different decision rule and therefore different classification results. Hence significant manual involvement may be required to determine the optimal W.

5 Connections with PLDA

Probabilistic linear discriminant analysis (PLDA) [8] is a widely used technique for face verification and other computer vision tasks. Although derived from the perspective of factor analysis models, it shares many commonalities with the Joint Bayesian framework described herein. Consequently, this section outlines similarities and differences that practically affect behavior both in the context of face verification and beyond.

5.1 Cost Function Comparisons

Joint Bayesian and PLDA training both involve minimizing the negative log-likelihood of the training samples under Gaussian distributional assumptions and with independence across subjects. Specifically, we must solve

$$\min_{\Theta \in C} -\sum_i \log P(\mathbf{x}_i | \Theta), \tag{20}$$

where as before, $\mathbf{x}_i$ denotes the stacked vector of all training exemplars from subject i, and Θ agglomerates all model parameters, and C is a potential constraint set. For the Joint Bayesian model, $\Theta = \{S_\varepsilon, S_\mu\}$, and $P(\mathbf{x}_i | \Theta)$ is given by (8).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2533383, IEEE Transactions on Pattern Analysis and Machine Intelligence

In contrast, the PLDA model is parameterized as follows. The j-th sample from subject i is expressed as

$$x_{ij} = Bz_i + Rw_{ij} + \nu_{ij}, \tag{21}$$

where

$$P(z_i) = N(0, I), P(w_{ij}) = N(0, I), P(\nu_{ij}) = N(0, S_\nu).$$

Here B and R are low-rank matrices while S_ν is a diagonal covariance. Per these specifications, the input signal x_{ij} is considered to consist of two parts: an identity component Bz_i, and intra-personal variations $Rw_{ij} + \nu_{ij}$. Given the above, x_{ij} and $\mathbf{x}_i$ are also Gaussians given by

$$P(x_{ij}|B, R, S_\nu) = N(0, BB^T + RR^T + S_\nu) \tag{22}$$

and

$$P(\mathbf{x}_i|B, R, S_\nu) = N(0, Q\Psi\Psi^T Q^T + \Sigma_\nu). \tag{23}$$

respectively, where $\Sigma_\nu = \text{diag}(S_\nu, \ldots, S_\nu)$ and $\Psi = \text{diag}(B, R, \ldots, R)$. While technically Q, Ψ, and Σ_ν all depend on i, we omit this subscript to simplify notation.

The likelihood function (23) is closely connected with the Joint Bayesian counterpart from (8). In particular, after a few linear algebra manipulations, it is easily shown that when $S_\mu = BB^T$ and $S_\varepsilon = RR^T + S_\nu$, then $Q\Psi\Psi^T Q^T + \Sigma_\nu = Q\Sigma_h Q^T$, and hence the overall likelihoods and underlying cost functions become exactly equivalent. Consequently, in this respect PLDA can be formally viewed as applying a more constrained parameterization to the same Joint Bayesian cost function,[1] and for training purposes both rely on the EM algorithm. At the testing phase this relationship is also equivalently maintained. We will now examine previously unexplored reasons for why the looser parameterization and associated EM algorithm adopted by Joint Bayesian may be preferable for many applications such as face verification. Later in Section 7 we will present complementary empirical evidence for these claims.

5.2 The Effects of Different Parameterizations

The PLDA model requires that we choose the dimensionality, or number of columns, in B and R. Consequently, additional user involvement may be required for optimizing these factors. In contrast, the Joint Bayesian model implicitly assumes that S_μ and S_ε, and therefore equivalently B and R, are of unrestricted and unknown rank. The natural mechanism for favoring lower rank estimates, to the extent allowed by the data as discussed in Section 4.2, then effectively allows the Joint Bayesian model to learn the appropriate dimensionality without user tuning. Therefore, unless we somehow know the true intrinsic dimensionality a priori, the less-constrained parameterization used by Joint Bayesian may be advantageous.

Of course we could always consider running the PLDA algorithm and its associated EM algorithm with B and R initialized to be full rank. In this context, we might expect PLDA and its now equivalent log-likelihood function to inherit the same rank minimization properties from Section 4.2 as Joint Bayesian leading to similar performance and efficiencies. However, even when B and R are full rank, there remain important differences in the underlying EM algorithms that minimize the Joint Bayesian and PLDA objectives as we will now describe.

First, we note that if R is full rank in the PLDA model, it follows that S_ν is then formally unidentifiable in the strict statistical sense, meaning that multiple parameterizations lead to an equivalent likelihood model and associated intra-personal variation component. Hence we will consider the behavior of PLDA in the limit as $S_\nu \to 0$, equivalent to the assumption employed by Joint Bayesian. In this regime, Joint Bayesian and PLDA cost functions are essentially identical, with $S_\mu = BB^T$ and $S_\varepsilon = RR^T$ both full rank by assumption at initialization. However, while the cost functions are the same, the partitioning into hidden data, as required by the associated EM algorithms, are not. In particular, the hidden data of PLDA, denoted as $\mathbf{y}_i = [z_i; w_{i1}; \cdots; w_{iJ}]$ for the i-th subject, is related to that of Joint Bayesian via $\mathbf{h}_i = \Psi\mathbf{y}_i, \forall i$. This leads to different upper bounds for minimizing the negative log-likelihood functions at each iteration of PLDA.

To see this, we carefully revisit the E and M steps as derived in [8] for PLDA in the limit as $S_\nu \to 0$. The relationship between the latent variables $\mathbf{y}_i$ and the observations $\mathbf{x}_i$ for the i-th subject is defined by

$$\mathbf{x}_i = A\mathbf{y}_i$$

$$A = \begin{bmatrix} B & R & \mathbf{0} & \cdots & \mathbf{0} \\ B & \mathbf{0} & R & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ B & \mathbf{0} & \mathbf{0} & \cdots & R \end{bmatrix},$$

and we again omit a subscript i on A here for simplicity. For the E-step, the required moments are

$$E[\mathbf{y}_i] = A^T(AA^T)^{-1}\mathbf{x}_i \tag{24}$$

$$E[\mathbf{y}_i\mathbf{y}_i^T] = \mathbf{I} - A^T(AA^T)^{-1}A + E[\mathbf{y}_i]E[\mathbf{y}_i]^T, \tag{25}$$

where the expectation is with respect to $P(\mathbf{y}_i|\mathbf{x}_i, \Theta)$. These are related to those of Joint Bayesian via

$$E[\mathbf{h}_i] = \Psi E[\mathbf{y}_i] \text{ and } E[\mathbf{y}_i\mathbf{y}_i^T] = \Psi E[\mathbf{h}_i\mathbf{h}_i^T]\Psi^T, \tag{26}$$

and equivalent to those in [8] after application of the matrix inversion lemma. For the M-step, PLDA must

1. Note that we are assuming the data have zero-mean after a standard centering operation, so the extra mean factor from [8] can be omitted here.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2533383, IEEE Transactions on Pattern Analysis and Machine Intelligence

update the model parameters to find some new $\tilde{B}$ and $\tilde{R}$ that maximize

$$E\left[\sum_i \log P(\mathbf{y}_i, \mathbf{x}_i|\tilde{B}, \tilde{R})\right] \equiv \sum_i E\left[\log P(\mathbf{x}_i|\mathbf{y}_i, \tilde{B}, \tilde{R})\right]. \tag{27}$$

This is then equivalent to minimizing

$$\begin{aligned} &\sum_i E\left[\|\mathbf{x}_i - \tilde{A}\mathbf{y}_i\|_2^2\right] \\ &= \sum_i \text{trace}(\mathbf{x}_i\mathbf{x}_i^T - 2\mathbf{x}_i E[\mathbf{y}_i]^T\tilde{A}^T + \tilde{A}E[\mathbf{y}_i\mathbf{y}_i^T]\tilde{A}^T) \\ &= \sum_i \text{trace}(\mathbf{x}_i - \tilde{A}E[\mathbf{y}_i])(\mathbf{x}_i - \tilde{A}E[\mathbf{y}_i])^T) \\ &\qquad + n\text{trace}(\tilde{A}(\mathbf{I} - A^T(AA^T)^{-1}A)\tilde{A}^T) \end{aligned} \tag{28}$$

over $\tilde{A}$ defined analogously to A but using $\tilde{B}$ and $\tilde{R}$. It is easily shown that both $(\mathbf{x}_i - \tilde{A}E[\mathbf{y}_i])(\mathbf{x}_i - \tilde{A}E[\mathbf{y}_i])^T$ and $\tilde{A}(\mathbf{I} - A^T(AA^T)^{-1}A)\tilde{A}^T$ are positive semi-definite matrices. Therefore, the value of (28) is greater than or equal to zero for any $\tilde{A}$. However, when $\tilde{A} = A$,

$$\mathbf{x}_i - AE[\mathbf{y}_i] = \mathbf{x}_i - AA^T(AA^T)^{-1}\mathbf{x}_i = 0$$

and

$$A(\mathbf{I} - A^T(AA^T)^{-1}A)A^T = 0,$$

where the second equality follows from the fact that $\mathbf{I} - A^T(AA^T)^{-1}A$ is a projection operator to the orthogonal complement of range$[A^T]$. Therefore, both the first and second term in (28) are minimized by $\tilde{A} = A$. Moreover, it can be shown that this minimizer is unique when there are a sufficient number of training examples in general position. Hence $\{\tilde{B} = B, \tilde{R} = R\}$ uniquely optimizes the M-step for all B and R, and the EM-algorithm stalls upon a single iteration.

Note that there exists a pervasive perception that the EM algorithm is guaranteed to converge to a local optimum of the log-likelihood function, which seems to contradict the claim here of zero convergence away from any starting point. However, the paradox is resolved once we understand that EM is only actually guaranteed to produce a series of iterates satisfying

$$\log p(x|\theta_t) \leq \log p(x|\theta_{t+1}) \tag{29}$$

for any general likelihood function $p(x|\theta)$. Strict convergence to local minima (or stationary points) requires further assumptions [27]. In particular, the conditions for Zangwill's Global Convergence Theorem must be satisfied [28]. These conditions require, among other things, that

$$\log p(x|\theta_t) < \log p(x|\theta_{t+1}) \tag{30}$$

for *all θ_t which are not local minima*. This then explains why, with $S_\nu \to 0$, the PLDA algorithm becomes immediately stuck after a single iteration even though the gradient of the log-likelihood is far from zero. Additionally, even with $S_\nu > 0$, if S_ν is small the convergence rate becomes prohibitively slow.

To summarize then, for face verification (as well as many other applications) it may be valuable to allow the data to determine the appropriate dimensionality of the approximate low-dimensional subspaces relevant for maximal discrimination, as opposed to requiring heuristic manual selections. This then favors the looser parameterization of Joint Bayesian, which subsequently requires the Joint Bayesian variant of EM to better ensure convergence to meaningful solutions. Note that this previously unexamined distinction between parameterizations and attendant EM algorithms, and the convergence issues that result, likely has wide-ranging implications in numerous other applications of Bayesian-inspired factor analysis models.

6 Relationships with Other Works

In this section, we discuss the connections between our joint Bayesian formulation and two other types of leading supervised methods which are widely used in face recognition.

6.1 Connection with metric learning

Metric learning [9], [10], [11], [12], [29] applied to face recognition has recently attracted significant attention. The goal of metric learning is to find a new metric under which two classes are more separable. One important example involves learning a Mahalanobis distance

$$(x_1 - x_2)^T M (x_1 - x_2), \tag{31}$$

where M is a positive definite matrix. However, this strategy shares the same drawback of the conventional Bayesian faces. Both first project the joint representation to a lower dimension using the transform $[\mathbf{I}, -\mathbf{I}]$. As already discussed, this transformation may reduce the separability and degrade the accuracy.

In contrast, the proposed joint formulation metric from (4) faces no analogous restriction. To clarify this picture, we reformulate (4) as

$$(x_1 - x_2)^T A (x_1 - x_2) + 2x_1^T(A - G)x_2. \tag{32}$$

Comparing (31) and (32), we observe that the joint formulation provides additional freedom for constructing the discriminant surface. Consequently, the new metric could be viewed as a more general distance which better preserves the separability.

Ultimately, there are two components in the proposed metric from (4): the cross inner term $x_1^T G x_2$ and two norm terms $x_1^T A x_1$ and $x_2^T A x_2$. To investigate the differing roles these terms may occupy, we performed a preliminary experiment under five conditions with changing conditions for A and G: a) use the original A and G as estimated by Joint Bayesian; b) set $A \to 0$; c) set $G \to 0$; d) set $A \to G$; e) set $G \to A$. The experimental design and resulting classification percentages are shown in Table

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2533383, IEEE Transactions on Pattern Analysis and Machine Intelligence

Experiments	Accuracy
A and G	87.5%
$A \to 0$	84.93%
$G \to 0$	55.63%
$A \to G$	83.73%
$G \to A$	84.85%

TABLE 3: Roles of A and G in the log-likelihood ratio metric. Both training and testing are conducted on LFW data following the unrestricted protocol. SIFT features are used followed by dimensionality reduction to $d = 200$ via PCA. Readers can refer to Section 7 for more detailed information and testing of Joint Bayesian.

3. Rigorous empirical validation of Joint Bayesian is deferred to Section 7.

Not surprisingly, most of discriminative information lies with the inner product term ${x_1}^T G x_2$; however, the norms ${x_1}^T A x_1$ and ${x_2}^T A x_2$ also play significant roles. They serve as image-specific adjustments to the decision boundary.

In summary, a variety of different algorithms have been designed for learning discriminative Mahalanobis distances, but relatively few investigate more general forms such as that produced by Joint Bayesian. Other recent research explores a useful metric based on cosine similarity by discriminative learning [30]. As future work we plan to consider learning a log-likelihood ratio metric in a similar discriminative fashion.

6.2 Connection with Reference Based Methods

Reference-based methods [13], [14], [15], [16] gauge a face by its similarities to a set of reference faces. For example, in [13], each reference is represented by a SVM classifier which is trained from multiple images of the same person, and the SVM score is used as the similarity.

From a Bayesian viewpoint, if we model each reference as a Gaussian with mean μ_i and a common covariance, then the similarity from a face x to each reference is the conditional likelihood $P(x\,|\mu_i)$. Given n references, x can be represented as $[P(x\,|\mu_1), ..., P(x\,|\mu_n)]$. With this reference-based representation, we can define the similarity between two faces $\{x_1, x_2\}$ as the log-likelihood ratio

$$\log\left(\frac{\frac{1}{n}\sum_{i=1}^{n} P(x_1|\mu_i)\,P(x_2|\mu_i)}{\left(\frac{1}{n}\sum_{i=1}^{n} P(x_1|\mu_i)\right)\left(\frac{1}{n}\sum_{i=1}^{n} P(x_2|\mu_i)\right)}\right). \quad (33)$$

If we consider that the references are infinite and independently sampled from a distribution $P(\mu)$, the above equation can be rewritten as

$$\log\left(\frac{\int P(x_1|\,\mu)\,P(x_2|\,\mu)\,P(\mu)d\mu}{\int P(x_1|\,\mu)\,P(\mu)d\mu \int P(x_2|\,\mu)\,P(\mu)d\mu}\right). \quad (34)$$

Now furthermore assume that

$$\begin{aligned} P(\mu) &= N(0, S_\mu) \quad (35)\\ P(x|\mu) &= N(\mu, S_\varepsilon) \end{aligned}$$

Then we have

$$\int P(x|\mu)P(\mu)d\mu = P(x) = N(0, S_\mu + S_\varepsilon)$$

$$\int P(x_1|\,\mu)\,P(x_2|\,\mu)\,P(\mu)d\mu = P(x_1, x_2) = N(0, \Sigma)$$

$$\Sigma = \begin{pmatrix} S_\mu + S_\varepsilon & S_\mu \\ S_\mu & S_\mu + S_\varepsilon \end{pmatrix}. \quad (36)$$

It is now obvious from (36) that the numerator of (34) is equal to $\log P(x_1, x_2|H_I)$ and the denominator of (34) is equal to $\log P(x_1, x_2|H_E)$. Therefore the two metrics are equivalent, and Joint Bayesian can be considered as a kind of probabilistic reference-based method, with infinite Gaussian references.

7 Experiments

In this section, we compare our Joint Bayesian approach with conventional Bayesian faces and other competitive supervised methods on the four benchmarks LFW [18], WDRef [17], Multi-PIE [19], and YouTube Faces [20] datasets. We use LBP features [31] and high-dimensional LBP features [32] in all experiments.

LBP feature. We first normalize the image to 100×100 pixels using an affine transformation calculated based on 5 landmarks (two eyes, noise and two mouth tips). Then the image is divided into 10×10 non-overlapped cells, and each cell within the image is mapped to a vector using LBP descriptors. All descriptors are concatenated to form the final feature.

High-dimensional LBP feature. We first rectify images using a similarity transformation based on five landmarks (two eyes, nose, and mouth corners). Then we extract patches centered around 27 landmarks in 5 scales. The side lengths of the image in each scale are 300, 212, 150, 106, and 75. The patch size is fixed to 40×40 in all scales. We divide each patch into 4×4 non-overlapped cells. Each cell is then mapped to a vector using an LBP descriptor. Next all descriptors are concatenated to form the final feature.

The feature dimensions of LBP and high-dimensional LBP are 5,900 and 127,440 respectively. We apply PCA to reduce the dimensionality of the raw feature to a feasible range for subsequent supervised learning.

7.1 Comparison with other Bayesian face methods

In the first experiment, we compare conventional Bayesian face recognition [1], Wang and Tang's unified subspace work [3], the naive formulation discussed in Section 2.1, and our Joint Bayesian algorithm. The first two methods are based on the difference between a given face pair while the last two model the full joint distribution. All algorithms are tested on two datasets: LFW and WDRef. When testing with LFW,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2533383, IEEE Transactions on Pattern Analysis and Machine Intelligence

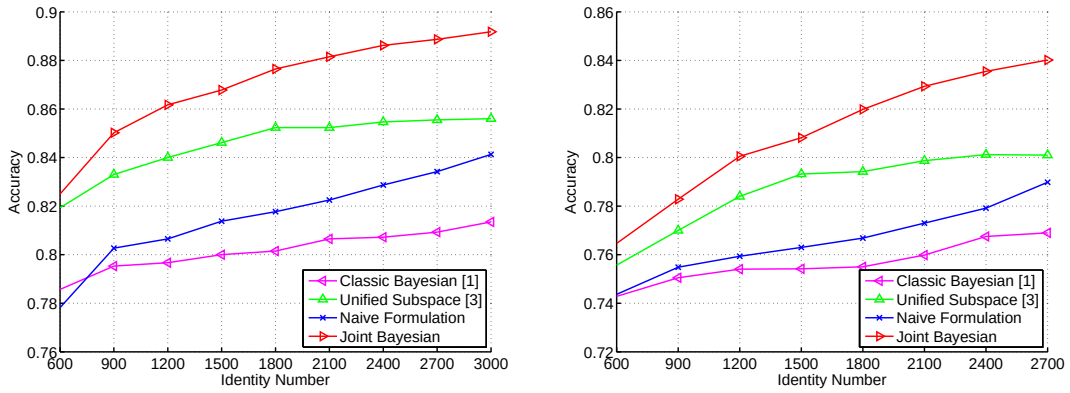

Fig. 2: Comparison with other Bayesian face related works. The joint Bayesian method is consistently better other a wide range of different training data sizes and on two databases: LFW(left) and WDRef(right).

all identities in WDRef are used for the training. We vary the number of identities n in the training data from 600 to 3000 to examine performance with respect to training data size. When testing with WDRef, we split the data into two mutually exclusive parts: 300 different subjects are used for testing, the others are for training. Similar to the standard LFW protocol, the test images are divided into 10 cross-validation sets and each set contains 300 intra-personal and extra-personal pairs. We use LBP features and reduce the feature dimension by PCA to the algorithm-dependent dimension that performed best ($d = 2000$ for joint Bayesian and unified subspace, $d = 400$ for Bayesian faces and naive formulation methods).

As shown in Fig. 2, by enforcing an appropriate prior on the face representation, our proposed joint Bayesian method performs substantially better on various training data sizes. The unified subspace algorithm is the next best by taking advantage of subspace selection over face differences, i.e. retaining the identity component and excluding the intra-person variation component and noise. We also note that when the training data size is small, the naive formulation displays the worst performance, the reason being that it must estimate more parameters in higher dimension using only sample averages. However, as training data increases, the performance of conventional Bayesian and unified subspace method (which both rely only on the difference of face pair) gradually begin to saturate. In contrast, the performance of the naive joint formulation keeps increasing as the training data increases. Its performance surpasses that of the conventional Bayesian method and is approaching that of the unified subspace method. The trend of the joint Bayesian method shares the same pattern as the naive joint formulation but at a higher discrimination level demonstrating its overall efficacy.

7.2 Comparison with LDA and PLDA

In this experiment, we compare Joint Bayesian with two subspace learning methods, LDA [22] and PLDA [8]. We use the same experimental setting as described in section 7.1. LDA is based on our own implementation and PLDA is from the authors' implementation [8]. In the experiments, the original LBP features are reduced to the best dimension by PCA for each method (2000 for all methods).

We also study the verification accuracy while varying the sub-space dimensions. For PLDA, the sub-space dimensions are determined by the user-defined column number of the matrices F and G in (21). For Joint Bayesian, the sub-space dimension is determined automatically by initializing the algorithm with full-rank covariance matrices. However, we can artificially truncate the singular values of P_A and P_G in (17) for the purpose of comparing with PLDA and LDA. Of course such truncation has no effect once we reach the range of zero-valued singular values learned by the Joint Bayesian with its preference for low-rank representations. However, it is important to emphasize that PLDA and LDA have a distinct advantage by explicitly learning a new subspace for each dimension. In contrast, for Joint Bayesian we have learned only a single model with mostly zero valued (or nearly so) singular values in the learned P_A and P_G, which then automatically determines the intrinsic dimensionality.

As shown in Fig. 3, if the sub-space dimensionality is too high or too low, the results of both LDA and PLDA will degrade. In the case of PLDA, this demonstrates that when the subspace dimensionality is too low, the model is underfitting, and when it is too high, the EM algorithm is no longer well-posed for reasons described in Section 5. LDA also displays a similar sensitivity and fails to fully exploit higher dimensional features. On the other hand, with Joint Bayesian once the dimensionality is sufficiently high such that we are no longer artificially truncating significant singular values, the performance remains constant. Overall, Joint Bayesian is independent of any such manual tuning in a pracitcal setting and operates naturally to learn appropriate discriminative

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2533383, IEEE Transactions on Pattern Analysis and Machine Intelligence

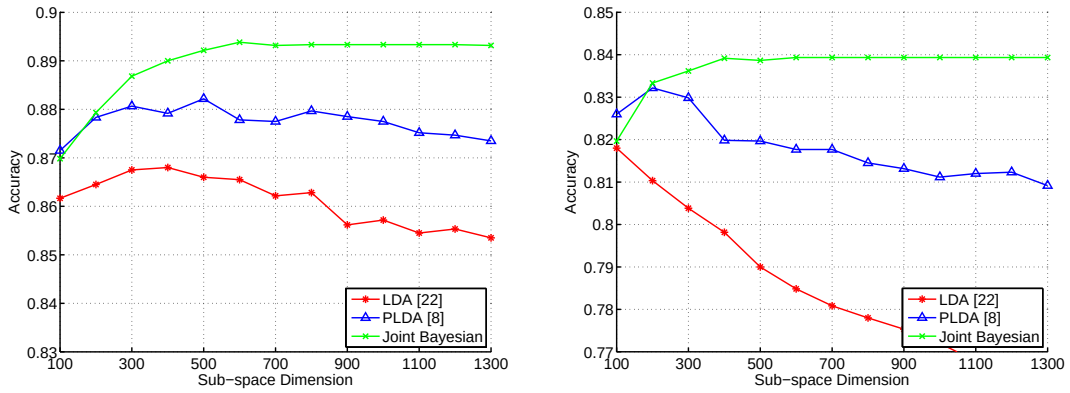

Fig. 3: Comparison with LDA and PLDA on LFW(left) and WDRef(right).

Algorithms	LBP	High-dim LBP
Bayesian Face [1]	82.65%	89.97%
LDA [22]	84.20%	91.37%
PLDA [8]	84.67%	92.38%
Joint Bayesian	86.47%	92.95%

TABLE 4: Comparison over Multi-PIE dataset. Proposed method achieves the best result with both LBP and High-dim LBP features.

subspaces. This property makes it very convenient for deployment.

7.3 Comparison on Multi-PIE dataset

We now further evaluate the generalization ability of our method. All algorithms are evaluated on the Multi-PIE dataset and trained on WDRef. Unlike the previous LFW and WDRef datasets, Multi-PIE data is collected in a controlled laboratory environment and displays considerable differences from image data collected from the Internet such as LFW and WDRef.

We follow the settings in [32] and [33] which are similar to the LFW protocol. We randomly select 49 test identities. Each identity contains 7 different pose categories and 4 illumination conditions. The pose categories range from -60% to +60% and the four illumination conditions are no-flash, left-flash, right-flash and left-right-flash. Then we randomly select 3000 intra-personal and 3000 extra-personal pairs that are placed in 10 folders for cross-validation evaluation. As shown in Table 4, we achieve 92.95% which is above the other supervised learning methods.

7.4 Comparison on YouTube Faces dataset

In this section, we validate the proposed Joint Bayesian method on the task of video-level verification. Joint Bayesian measures the similarity of two images. To extend Joint Bayesian to measuring the similarity of two videos, we use the maximum similarity of all image pairs between the two videos as the video-level similarity.

The YouTube Faces dataset [20] is a common benchmark for evaluating video-level face verification. It contains 3,425 videos of 1,595 different people. On average, a face track from a video clip consists of 181.3 frames of faces. We report comparative results under two settings: a) the Youtube dataset is split for training and testing consistent with previous works [20][34][35]; b) the WDRef dataset is used for training while the Youtube dataset is reserved for testing.

When training on the YouTube dataset, we follow the unrestricted protocol in [20]. We divide 5000 video pairs into 10 splits, with each split containing 250 intra-personal pairs and 250 inter-personal pairs. The subject identity labels are accessible during training. We randomly selected 10 frames from each training video to construct the training set. On average there are around 30k frames of 1,400 persons for training in each round. When training on the WDRef dataset, we follow the same setting as described in section 7.1.

During testing, given a pair of videos, we extract high-dimensional LBP features for all frames of the two videos and compute the pairwise similarities between the frames of the two videos. Finally, we keep the maximum similarity. It is worth mentioning that by applying the efficient testing methods in Section 4.2, all image-level similarities can be computed in around **one millisecond**, indicating that our method is applicable to real-time video-level verification on both PC and mobile devices.

Table 5 shows the results of our method along with LDA and PLDA under the aforementioned two settings using our high-dimensional LBP features for consistency, as well as the results of recent state-of-the-art algorithms. Joint Bayesian achieves 84.40% when training on Youtube, and 87.12% when training on WDref, surpassing all other methods. Moreover, we emphasize that although PLDA performs second best, this requires both tuning of the subspace dimension via cross-validation and an extremely large training cost for each trial dimension without the tailored analytical simplifications described in Section 4. Considering the simplicity of the extension from image-level similarity to video-level similarity, these results provide additional compelling evidence

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2533383, IEEE Transactions on Pattern Analysis and Machine Intelligence

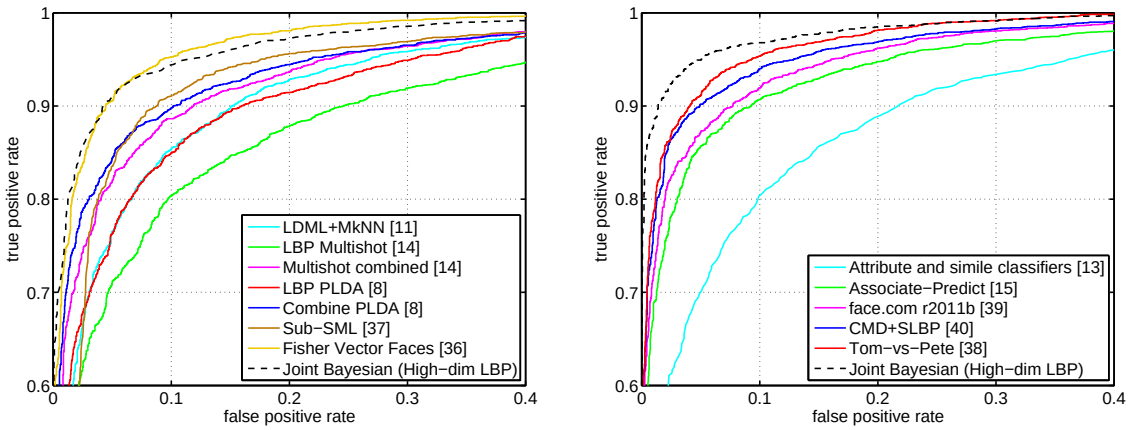

Fig. 4: Comparison with state-of-the-art methods on LFW. Without outside training data(left) and with outside training data(right). Note that Joint Bayesian automatically determines the subspace dimension and here we artificially truncate singular values to produce lower-dimensional models merely for purposes of comparison.

Algorithms	Accuracy
MBGS+LBP [20]	76.40%
STFRD+PMML [34]	79.48%
APEM (fusion) [35]	79.06%
LDA (Train on Youtube) [22]	82.20%
PLDA (Train on Youtube) [8]	83.60%
Joint Bayesian (Train on Youtube)	84.40%
LDA (Train on WDRef) [22]	85.06%
PLDA (Train on WDRef) [8]	86.54%
Joint Bayesian (Train on WDRef)	87.12%

TABLE 5: Performance comparison on YouTube Faces Dataset. Joint Bayesian significantly outperforms existing state-of-the-art algorithms, as well as LDA and PLDA carefully tuned with high-dimensional LBP features.

for the efficacy of the proposed Joint Bayesian method.

7.5 Comparison with LFW state-of-the-art

This section presents the best performance of Joint Bayesian on the LFW dataset, along with existing state-of-the-art methods for comparison, under two settings: supervised learning without and with outside training data.

Without outside training data. In order to fairly compare with other approaches published on the LFW website [8], [11], [14], [36], [37], we follow the LFW unrestricted protocol, using only LFW for training.

With outside training data. Our purposes here are twofold: 1) verify the generalization ability from one dataset to another dataset; 2) see what improvement is possible using limited outside training data. We follow the standard restricted protocol in LFW using the high-dimensional LBP features and compare with other top-performing algorithms that use outside training data [13], [15], [38], [39], [40].

We achieve **93.18%** under the LFW unrestricted protocol (known identity information) without outside training data. At the time of submission, this represented the top performing algorithm; however, very recently two new hand-crafted features lead to a bit higher performance(see for example [42]). Although feature development is not our focus here, it is of course likely that our Joint Bayesian pipeline could equally benefit from such features as well. Note that without labeled outside training data, DNN-based features are not competitive under this protocol. Using WDRef as outside training data, we achieve **95.17%**. As shown in Fig. 4, Joint Bayesian achieves state-of-the-art performance under both settings. Additionally, we are able to push the accuracy even further to **96.33%** (approaching human performance) if transfer learning [21] is applied to alleviate the distributional differences between the outside training data (WDRef) and the testing data (LFW). At the time of original submission, this represented the highest published result on the challenging LFW data satisfying all the requirements for the unrestricted protocol, and yet this accuracy is nonetheless still achievable with an extremely fast and scalable algorithm.

Since the time of our original submission, a far more complex set of face verification features have been derived using a deep learning architecture [41]. When these features are then substituted into our Joint Bayesian framework, performance can be further improved to **99.15%**, which to our knowledge represents the first method to break the 99% barrier and essentially saturate the LFW benchmark under the unrestricted protocol. This is further compelling evidence that Joint Bayesian represents a useful generic tool for incorporation in face verification pipelines irrespective of the specific features that are used. Moreover, it especially speaks to the continued relevance of this methodology even in the emerging era of deep learning.

8 Proofs and Derivations

8.1 Conditional distribution $P(h|x,\Theta)$

We omit the subject-indicator subscript i here for convenience, and then introduce an auxiliary variable

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2533383, IEEE Transactions on Pattern Analysis and Machine Intelligence

$\mathbf{z}$ defined as

$$\mathbf{z} = \begin{bmatrix} \mathbf{h} \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{I} \\ Q \end{bmatrix} \mathbf{h}.$$

The distribution of $\mathbf{h}$ is $N(0, \Sigma_h)$, hence any linear transformation such as $\mathbf{z}$ is also Gaussian, in this case given by

$$P(\mathbf{z}|\Theta) = N(0, \Sigma_z)$$

$$\Sigma_z = \begin{bmatrix} \Sigma_h & \Sigma_h Q^T \\ Q\Sigma_h & Q\Sigma_h Q^T \end{bmatrix}.$$

Given the joint Gaussian distribution, any conditional distribution is likewise a Gaussian and can be easily derived using standard identites producing

$$P(\mathbf{h}|\mathbf{x}, \Theta_t) = N(\overline{\mu}, \overline{\Sigma})$$

where

$$\begin{aligned} \overline{\mu} &= \Sigma_h Q^T (Q\Sigma_h Q^T)^{-1} \mathbf{x} \\ \overline{\Sigma} &= \Sigma_h - \Sigma_h Q^T Q \Sigma_h Q^T Q \Sigma_h. \end{aligned}$$

From the mean and covariance, we can easily derive the first- and second- order moments of the conditional distribution in Equation (11).

8.2 Proof of Lemma 1

If each subject has the same number of samples, then the log-likelihood function

$$-\sum_i \log P(\mathbf{x}_i|S_\mu, S_\varepsilon)$$

can be simplified to

$$\begin{aligned} &n(m-1)\log|S_\varepsilon| + n\log|mS_\mu + S_\varepsilon| \\ &+ \sum_i \sum_j \text{trace}(S_\varepsilon (x_{ij} - \bar{x}_i)(x_{ij} - \bar{x}_i)^T) \\ &+ \sum_i \text{trace}(m(mS_\mu + S_\varepsilon)\bar{x}_i \bar{x}_i^T), \end{aligned} \tag{37}$$

where n is the number of subjects and m is the number of samples per subject, and $\bar{x}_i$ is the sample mean of the i^{th} subject. This result follows using the identity $|\Sigma_x| = |mS_\mu + S_\varepsilon||S_\varepsilon|^{m-1}$. We can then obtain an unconstrained closed-form solution by taking derivatives of (37) with respect to S_μ and S_ε and equating to zero which gives

$$\begin{aligned} S_\varepsilon &= \frac{1}{n}\frac{1}{m-1}\sum_i \sum_j (x_{ij} - \bar{x}_i)(x_{ij} - \bar{x}_i)^T \\ S_\mu &= \frac{1}{n}\sum_i \bar{x}_i \bar{x}_i^T - \frac{1}{m} S_\varepsilon. \end{aligned}$$

As an unconstrained optimal solution, it is clear that S_μ will not necessarily satisfy the positive semi-definite requirement of a covariance matrix. However, in the limit as $m \to \infty$, the term $\frac{1}{m}S_\varepsilon$ will converge to zero. The limiting solutions

$$\begin{aligned} S_\varepsilon &= \frac{1}{mn}\sum_i \sum_j (x_{ij} - \bar{x}_i)(x_{ij} - \bar{x}_i)^T \\ S_\mu &= \frac{1}{n}\sum_i \bar{x}_i \bar{x}_i^T \end{aligned}$$

now satisfy the necessary positive semi-definite constraint and are also symmetric as required. As these unconstrained solutions now satisfy the constraints and are globally optimal, they must then now represent globally optimal constrained solutions.

8.3 Proof of Lemma 2

From Equation (5) and the standard formulae for the inverse of a block partitioned matrix, it is straightforward to show that

$$F + G = \left(U - S_\mu U^{-1} S_\mu)^{-1}\right)^{-1} \tag{38}$$

where $U = S_\mu + S_\varepsilon$. It follows then that

$$A = U^{-1} - (U - S_\mu U^{-1} S_\mu)^{-1}.$$

We may then apply the Woodbury matrix inversion identity and arrive at

$$A = -U^{-1} S_\mu (U - S_\mu U^{-1} S_\mu)^{-1} S_\mu U^{-1}. \tag{39}$$

From the Shur Complement Lemma, $U - S_\mu U^{-1} S_\mu \succeq 0$. Consequently, the matrix A is a negative semi-definite symmetric matrix and hence can be expressed as $A = -P_A P_A^T$ for some matrix P_A.

For the matrix G, also from (5), it follows that

$$\begin{bmatrix} F+G & G \\ G & F+G \end{bmatrix} \begin{bmatrix} S_\mu + S_\varepsilon & S_\mu \\ S_\mu & S_\mu + S_\varepsilon \end{bmatrix} = \mathbf{I}, \tag{40}$$

which can easily be solved to find that

$$G = -(2S_\mu + S_\varepsilon)^{-1} S_\mu S_\varepsilon^{-1}. \tag{41}$$

We then define V such that $S_\mu = VV^T$, which is always possible since S_μ is a covariance. Again, we use the Woodbury matrix inversion identity to reexpress G as

$$G = -S_\varepsilon^{-1} V (I + 2V^T S_\varepsilon^{-1} V)^{-1} V^T S_\varepsilon^{-1} \tag{42}$$

from which it is immediately apparent that G is a negative semi-definite symmetric matrix, and hence $G = -P_G P_G^T$ for some P_G.

Finally, regarding matrix rank, in (39) and (41) the matrices A and G include S_μ in product form. Given that $\text{rank}[XY] \leq \text{rank}[Y]$, it immediately follows that

$$\text{rank}[A] = \text{rank}[P_A] \leq \text{rank}[S_\mu]$$

$$\text{rank}[G] = \text{rank}[P_G] \leq \text{rank}[S_\mu].$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2533383, IEEE Transactions on Pattern Analysis and Machine Intelligence

8.4 Proof of Lemma 4

We first consider Equation (18) and let $\mathbb{E}_*$ and $\mathbb{M}_*$ denote an optimal solution. With an invertible linear transform the constraint becomes $W\mathbb{X} = \mathbb{E} + \mathbb{M}\Phi$ or equivalently $\mathbb{X} = W^{-1}\mathbb{E} + W^{-1}\mathbb{M}\Phi$. Defining $\tilde{\mathbb{E}} = W^{-1}\mathbb{E}$ and $\tilde{\mathbb{M}} = W^{-1}\mathbb{M}$, (18) becomes

$$\begin{aligned} \min_{\tilde{\mathbb{E}},\tilde{\mathbb{M}}} \quad & n\log|S_\mu| + k\log|S_\varepsilon| \\ \text{s.t.} \quad & \mathbb{X} = \tilde{\mathbb{E}} + \tilde{\mathbb{M}}\Phi, \\ & S_\mu = \tfrac{1}{n}W\tilde{\mathbb{M}}\tilde{\mathbb{M}}^T W^T, \;\; S_\varepsilon = \tfrac{1}{k}W\tilde{\mathbb{E}}\tilde{\mathbb{E}}^T W^T. \end{aligned}$$

However, since $\log|AB| = \log|A| + \log|B|$ for square matrices A and B, this is equivalent to solving

$$\begin{aligned} \min_{\tilde{\mathbb{E}},\tilde{\mathbb{M}}} \quad & n\log\left|\tilde{\mathbb{M}}\tilde{\mathbb{M}}^T\right| + k\log\left|\tilde{\mathbb{E}}\tilde{\mathbb{E}}^T\right| \\ \text{s.t.} \quad & \mathbb{X} = \tilde{\mathbb{E}} + \tilde{\mathbb{M}}\Phi, \\ & S_\mu = \tfrac{1}{n}W\tilde{\mathbb{M}}\tilde{\mathbb{M}}^T W^T, \;\; S_\varepsilon = \tfrac{1}{k}W\tilde{\mathbb{E}}\tilde{\mathbb{E}}^T W^T, \end{aligned}$$

which is obviously minimized when $\tilde{\mathbb{E}} = \mathbb{E}_*$ and $\tilde{\mathbb{M}} = \mathbb{M}_*$, directly leading to the stated result. With some additional effort, a similar result can be shown when we do not adopt the assumption $E[\mathbf{h}_i\mathbf{h}_i^T] = E[\mathbf{h}_i]E[\mathbf{h}_i]^T$; however, we omit the details here for brevity.

Regarding the second invariance, by plugging the new optimal covariances into (5), we arrive at

$$\begin{aligned} \tilde{A} &= (W^T)^{-1}AW^{-1} \\ \tilde{G} &= (W^T)^{-1}GW^{-1} \end{aligned}$$

By plugging the above equations into the log-likelihood ratio test, we get

$$\begin{aligned} r(\tilde{x}_1, \tilde{x}_2) &= \tilde{x}_1^T\tilde{A}\tilde{x}_1 + \tilde{x}_2^T\tilde{A}\tilde{x}_2 - 2\tilde{x}_1^T\tilde{G}\tilde{x}_2 \\ &= x_1^T A x_1 + x_2^T A x_2 - 2x_1^T G x_2 \\ &= r(x_1, x_2) \end{aligned}$$

9 Conclusion

In this paper we have revisited the classical Bayesian face recognition algorithm and proposed a joint formulation in the same probabilistic framework. The resulting modifications retain much of the practicality and scalability of the original, while enhancing performance above existing state-of-the-art face verification algorithms on a wide battery of tests even without a tuning parameters for determining the latent dimension. Ultimately, this demonstrates that given modern, low-level features and training data of moderate size, a strikingly simple algorithm can prove to be highly competitive.

References

[1] B. Moghaddam, T. Jebara, and A. Pentland, "Bayesian face recognition," *Pattern Recognition*, vol. 33, pp. 1771–1782, 2000.

[2] P. J. Phillips, H. Moon, S. Rizvi, and P. J. Rauss, "The FERET Evaluation Methodology for Face-Recognition Algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 1090–1104, 2000.

[3] X. Wang and X. Tang, "A unified framework for subspace face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 1222–1228, 2004.

[4] X. Wang and X. Tang, "Subspace Analysis Using Random Mixture Models," in *Computer Vision and Pattern Recognition*, vol. 1, 2005, pp. 574–580.

[5] X. Wang and X. Tang, "Bayesian face recognition using Gabor features," in *ACM Multimedia Conference*, 2003, pp. 70–73.

[6] Z. Li and X. Tang, "Bayesian Face Recognition Using Support Vector Machine and Face Clustering," in *Computer Vision and Pattern Recognition*, vol. 2, 2004, pp. 374–380.

[7] S. J. D. Prince and J. H. Elder, "Probabilistic Linear Discriminant Analysis for Inferences About Identity," in *International Conference on Computer Vision*, 2007, pp. 1–8.

[8] P. Li, U. Mohammed, J. Elder, and S. Prince, "Probabilistic Models for Inference about Identity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 144–157, 2012.

[9] K. Q. Weinberger, J. Blitzer, and L. K. Saul, "Distance Metric Learning for Large Margin Nearest Neighbor Classification," *Journal of Machine Learning Research*, vol. 10, pp. 207–244, 2005.

[10] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *International Conference on Machine Learning*, 2007, pp. 209–216.

[11] M. Guillaumin, J. J. Verbeek, and C. Schmid, "Is that you? Metric learning approaches for face identification," in *International Conference on Computer Vision*, 2009, pp. 498–505.

[12] Y. Ying and P. Li, "Distance metric learning with eigenvalue optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 1–26, 2012.

[13] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, "Attribute and Simile Classifiers for Face Verification," in *International Conference on Computer Vision*, 2009, pp. 365–372.

[14] Y. Taigman, L. Wolf, and T. Hassner, "Multiple One-Shots for Utilizing Class Label Information," in *British Machine Vision Conference*, 2009.

[15] Q. Yin, X. Tang, and J. Sun, "An associate-predict model for face recognition," in *Computer Vision and Pattern Recognition*, 2011, pp. 497–504.

[16] C. Zhu, F. Wen, and J. Sun, "A rank-order distance based clustering algorithm for face tagging," in *Computer Vision and Pattern Recognition*, 2011, pp. 481–488.

[17] D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun, "Bayesian face revisited: A joint formulation," in *European Conference on Computer Vision*, 2012, vol. 7574, pp. 566–579.

[18] G. B. Huang, M. Ramesh, T. Berg, E. Learned-Miller, and A. Hanson, "Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments," *European Conference on Computer Vision*, 2008.

[19] R. Gross, I. Matthews, J. F. Cohn, T. Kanade, and S. Baker, "Multi-PIE," in *IEEE International Conference on Automatic Face and Gesture Recognition*, vol. 28, 2008, pp. 1–8.

[20] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *Computer Vision and Pattern Recognition*, 2011, pp. 529–534.

[21] X. Cao, D. Wipf, F. Wen, and G. Duan, "A practical transfer learning algorithm for face verification," in *International Conference on Computer Vision*, 2013.

[22] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 711–720, 1997.

[23] S. Ioffe, "Probabilistic Linear Discriminant Analysis," in *European Conference on Computer Vision*, 2006, pp. 531–542.

[24] J. Susskind, R. Memisevic, G. Hinton, and M. Pollefeys, "Modeling the joint density of two images under a variety of transformations," in *CVPR*, 2011, pp. 2793–2800.

[25] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," 1977.

[26] F. Leger, G. Yu, and G. Sapiro, "Efficient matrix completion with gaussian models," in *Acoustics Speech and Signal Processing*, 2011.

[27] C. Wu, "On the convergence properties of the em algorithm," in *Annals of Statistics*, 1983.

[28] D.G.Luenberger, "Linear and nonlinear programming," in *Addison Wesley, Reading, Massachusetts*, 1984.

[29] D. Ramanan and S. Baker, "Local distance functions: A taxonomy, new algorithms, and an evaluation," in *International Conference on Computer Vision*, 2009, pp. 301–308.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2533383, IEEE Transactions on Pattern Analysis and Machine Intelligence

[30] H. V. Nguyen and L. Bai, "Cosine Similarity Metric Learning for Face Verification," in *ACCV*, 2010, pp. 709–720.
[31] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 971–987, 2002.
[32] D. Chen, X. Cao, F. Wen, and J. Sun, "Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification," in *Computer Vision and Pattern Recognition*, 2013, pp. 3025–3032.
[33] Z. Cao, Q. Yin, X. Tang, and J. Sun, "Face recognition with learning-based descriptor," in *Computer Vision and Pattern Recognition*, 2010, pp. 2707–2714.
[34] Z. Cui, W. Li, D. Xu, S. Shan, and X. Chen, "Fusing robust face region descriptors via multiple metric learning for face recognition in the wild," in *Computer Vision and Pattern Recognition*, 2013, pp. 3554–3561.
[35] H. Li, G. Hua, Z. Lin, J. Brandt, and J. Yang, "Probabilistic elastic matching for pose variant face verification," in *Computer Vision and Pattern Recognition*, 2013, pp. 3499–3506.
[36] K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Fisher Vector Faces in the Wild," in *British Machine Vision Conference*, 2013.
[37] Q. Cao, Y. Ying, and P. Li, "Similarity Metric Learning for Face Recognition," in *ICCV*, 2013.
[38] T. Berg and P. Belhumeur, "Tom-vs-pete classifiers and identity-preserving alignment for face verification," in *British Machine Vision Conference*, 2012, pp. 129.1–129.11.
[39] Y. Taigman and L. Wolf, "Leveraging Billions of Faces to Overcome Performance Barriers in Unconstrained Face Recognition," 2011.
[40] C. Huang, S. Zhu, and K. Yu, "Large scale strongly supervised ensemble metric learning, with applications to face verification and retrieval," in *NEC Technical Report TR115*, 2011.
[41] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Technical report, arXiv:1406.4773*, 2014.
[42] X. Zhu, Z. Lei, J. Yan, D. Yi, and S. Li, "High-fidelity pose and expression normalization for face recognition in the wild," in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, June 2015, pp. 787–796.

Dong Chen received the B.S. and Ph.D. degree from the University of Science and Technology of China in 2010, 2015. He joined Microsoft Research in July 2015 as an associate researcher. His research interests include face detection and recognition. He is also interested in computer vision.

Xudong Cao received the B.S. degree from Tsinghua University in 2008. He joined Microsoft Research in December 2011 as an associate researcher. His current major research interests include face alignment and recognition. He is also interested in computer vision and machine learning.

David Wipf received the B.S. degree with highest honors from the University of Virginia, and the Ph.D. degree from UC San Diego, where he was an NSF IGERT Fellow. Later he was an NIH Postdoctoral Fellow at UC San Francisco. Since 2011 he has been with Microsoft Research in Beijing. His research interests include Bayesian learning techniques applied to signal/image processing and computer vision. He is the recipient of several awards including the 2012 Signal Processing Society Best Paper Award, the Biomag 2008 Young Investigator Award, and the 2006 NIPS Outstanding Paper Award.

Fang Wen recieved the Ph.D and M.S. degree in Pattern Recognition and Intelligent System and B.S degree in Automation from Tsinghua University in 2003, 1997, respectively. Now she is a lead researcher of visual computing group at Microsoft Research. Her research interests include computer vision, pattern recogntion and multimedia search.

Jian Sun is currently a principal researcher at Microsoft Research Asia. He got the B.S. degree, M.S. degree and Ph.D. degree from Xian Jiaotong University in 1997, 2000 and 2003. He joined Microsoft Research Asia in July, 2003. His current two major research interests are interactive computer vision (user interface + vision) and internet computer vision (large image collection + vision). He is also interested in stereo matching and computational photography. He has won the Best Paper Award at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2009.