

Chapter 18

Models for style and identity

In this chapter we discuss a family of models that explain observed data in terms of several underlying causes. These causes can be divided into three types: the identity of the object, the style in which it is observed, and the remaining variation.

To motivate these models, consider *face recognition*. For a facial image, the identity of the face (i.e., whose face it is) obviously influences the observed data. However, the style in which the face is viewed is also important. The pose, expression, and illumination are all style elements that might be modeled. Unfortunately, many other things also contribute to the final observed data: the person may have applied cosmetics, put on glasses, grown a beard, or dyed their hair. These myriad contributory elements are too difficult to model, and so are explained with a generic noise term.

In face recognition tasks, our goal is to infer whether the identities of face images are the same or different. For example, in *face verification*, we aim to infer a binary variable $w \in \{0, 1\}$, where $w = 0$ indicates that the identities differ and $w = 1$ indicates that they are the same. This task is extremely challenging when there are large changes in pose, illumination, or expression; the change in the image due to style may dwarf the change due to identity (figure 18.1).

The models in this chapter are generative, so the focus is on building separate density models over the observed image data \mathbf{x} for the cases where the faces do and don't have the same identity. They are all *subspace models* and describe data as a linear combination of basis vectors. We have previously encountered several models of this type, including factor analysis (section 7.6) and PPCA (section 17.5.1). The models in this chapter are most closely related to factor analysis, so we will start by reviewing this.

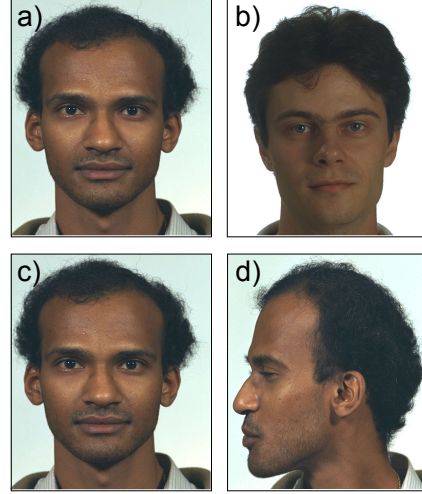
Factor analysis

Recall that the factor analysis model explained the i^{th} data example \mathbf{x}_i as

$$\mathbf{x}_i = \boldsymbol{\mu} + \boldsymbol{\Phi} \mathbf{h}_i + \boldsymbol{\epsilon}_i, \quad (18.1)$$

where $\boldsymbol{\mu}$ is the overall mean of the data. The matrix $\boldsymbol{\Phi} = [\phi_1, \phi_2, \dots, \phi_K]$ contains K factors in its columns. Each factor can be thought of as a basis vector in a

Figure 18.1 Face recognition. In face recognition the goal is to draw inferences about the identities of face images. This is difficult because the style in which the picture was taken can have a more drastic effect on the observed data than the identities themselves. For example, the images in a-b) are more similar to one another by most measures than the images in c-d) because the style (pose) has changed in the latter case. Nonetheless, the identities in a-b) are different but the identities in c-d) are the same. We must build models that tease apart the contributions of identity and style to make accurate inferences about whether the identities match.



high-dimensional space, and so together they define a subspace. The K elements of the hidden variable \mathbf{h}_i weight the K factors to explain the observed deviations of the data from the mean. Remaining differences that cannot be explained in this way are ascribed to additive noise ϵ_i which is normally distributed with diagonal covariance Σ .

In probabilistic terms, we write

$$\begin{aligned} Pr(\mathbf{x}_i|\mathbf{h}_i) &= \text{Norm}_{\mathbf{x}_i}[\boldsymbol{\mu} + \Phi\mathbf{h}_i, \Sigma] \\ Pr(\mathbf{h}_i) &= \text{Norm}_{\mathbf{h}_i}[\mathbf{0}, \mathbf{I}], \end{aligned} \quad (18.2)$$

where we have also defined a suitable prior over the hidden variable \mathbf{h}_i . Ancestral sampling from this model is illustrated in figure 18.2.

We can compute the likelihood of observing a new data example by marginalizing over the hidden variable, to get a final probability model

$$\begin{aligned} Pr(\mathbf{x}_i) &= \int Pr(\mathbf{x}_i, \mathbf{h}_i) d\mathbf{h}_i = \int Pr(\mathbf{x}_i|\mathbf{h}_i)Pr(\mathbf{h}_i) d\mathbf{h}_i \\ &= \text{Norm}_{\mathbf{x}_i}[\boldsymbol{\mu}, \Phi\Phi^T + \Sigma]. \end{aligned} \quad (18.3)$$

To learn this model from training data $\{\mathbf{x}_i\}_{i=1}^I$, we use the expectation maximization algorithm. In the E-step, we compute the posterior distribution $Pr(\mathbf{h}_i|\mathbf{x}_i)$ over each hidden variable \mathbf{h}_i ,

$$Pr(\mathbf{h}_i|\mathbf{x}_i) = \text{Norm}_{\mathbf{h}_i}[(\Phi^T \Sigma^{-1} \Phi + \mathbf{I})^{-1} \Phi^T \Sigma^{-1} (\mathbf{x}_i - \boldsymbol{\mu}), (\Phi^T \Sigma^{-1} \Phi + \mathbf{I})^{-1}]. \quad (18.4)$$

In the M-step we update the parameters as

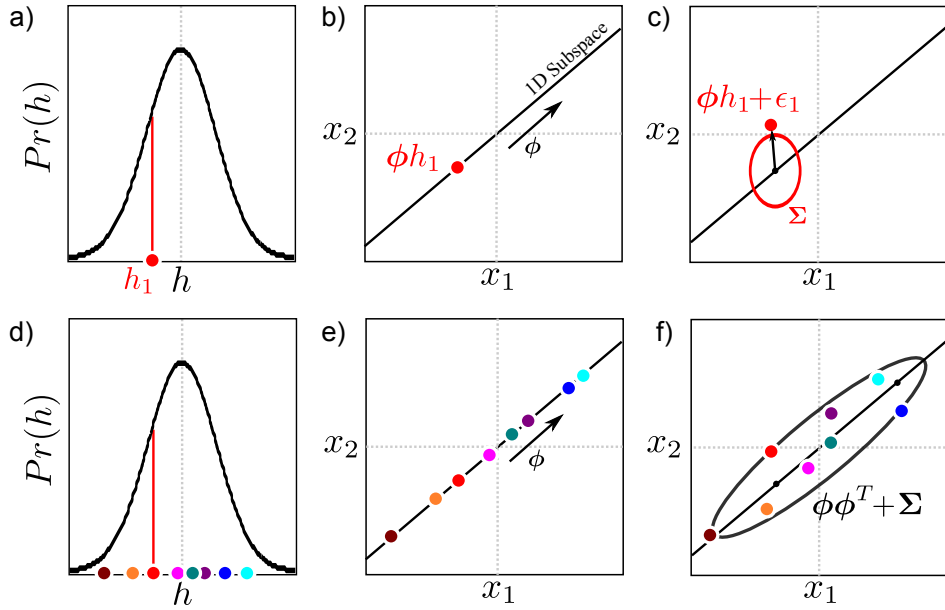


Figure 18.2 Ancestral sampling from factor analyzer. In both this figure and other subsequent figures in this chapter, we assume that the mean μ is zero. a) To generate from a factor analyzer defined over 2D data we first choose the hidden variable \mathbf{h}_i from the normally distributed prior. Here h_i is a 1D variable and a small negative value is drawn. b) For each case we weight the factors Φ by the hidden variable. This generates a point on the subspace (here a 1D subspace indicated by black line). c) Then we add the noise term ϵ_i , which is normally distributed with covariance Σ . Finally, we would add a mean term μ (not shown). d-f) This process is repeated many times. The final distribution of the data in (f) is a normal distribution that is oriented along the subspace. Deviations from this subspace are due to the noise term. The final covariance is $\Phi\Phi^T + \Sigma$.

$$\begin{aligned}
 \hat{\mu} &= \frac{\sum_{i=1}^I \mathbf{x}_i}{I} \\
 \hat{\Phi} &= \left(\sum_{i=1}^I (\mathbf{x}_i - \hat{\mu}) \mathbf{E}[\mathbf{h}_i]^T \right) \left(\sum_{i=1}^I \mathbf{E}[\mathbf{h}_i \mathbf{h}_i^T] \right)^{-1} \\
 \hat{\Sigma} &= \frac{1}{I} \sum_{i=1}^I \text{diag} \left[(\mathbf{x}_i - \hat{\mu})(\mathbf{x}_i - \hat{\mu})^T - \hat{\Phi} \mathbf{E}[\mathbf{h}_i] (\mathbf{x}_i - \hat{\mu})^T \right], \quad (18.5)
 \end{aligned}$$

where the expectations $E[\mathbf{h}_i]$ and $E[\mathbf{h}_i \mathbf{h}_i^T]$ over the hidden variable are extracted from the posterior distribution computed in the E-step. More details about factor analysis can be found in section 7.6.

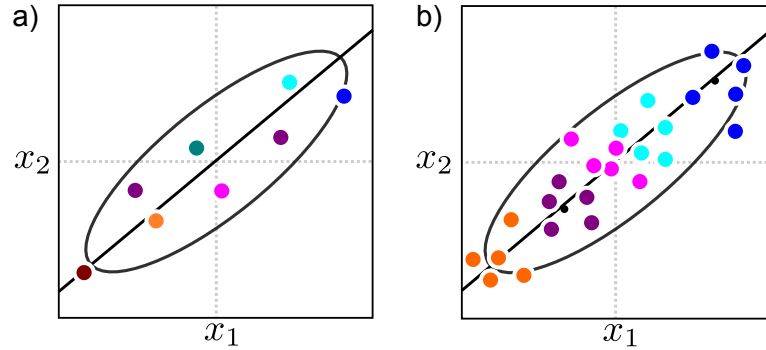


Figure 18.3 Subspace model vs. subspace identity model. a) The subspace model generates data that are roughly aligned along a subspace (here a 1D subspace defined by the black line) as illustrated in figure 18.2. b) In the identity subspace model, the overall data distribution is the same, but there is additional structure: points that belong to the same identity (same color) are generated in the same region of space.

18.1 Subspace identity model

The factor analysis model provides a good description of the pixel data \mathbf{x} in frontal face images: they really do lie close to a linear subspace (see figure 7.22). However, this description of the data does not account for identity. For images which have the same style (e.g., pose, lighting), we expect faces which have the same identity to lie in a similar part of the space (figure 18.3), but there is no mechanism to accomplish this in the original model.

We now extend the factor analysis model to take account of data examples which are known to have the same identity and show how to exploit this to make inferences about the identity of new data examples. We adopt the notation \mathbf{x}_{ij} to denote the j^{th} of J observed data examples from the i^{th} of I identities (individuals). In real-world data sets it is unlikely that we will have exactly J examples for every individual and the models we present do not require this, but this assumption simplifies the notation.

The generative explanation for the observed data \mathbf{x}_{ij} is now

$$\mathbf{x}_{ij} = \boldsymbol{\mu} + \boldsymbol{\Phi} \mathbf{h}_i + \boldsymbol{\epsilon}_{ij}, \quad (18.6)$$

where all of the terms have the same interpretations as before. The key difference is that now all of the J data examples from the same individual are formed by taking the *same* linear combination \mathbf{h}_i of the basis functions $\phi_1 \dots \phi_K$. However, a different noise term is added for each data example, and this explains the differences between the J face images of a given individual. We can write this in probabilistic form as

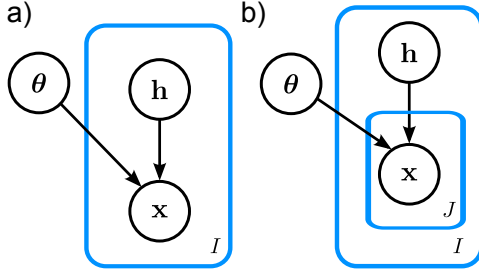


Figure 18.4 Graphical models for subspace model and subspace identity model. a) In the subspace model (factor analysis) there is one data example \mathbf{x}_i per hidden variable \mathbf{h}_i and some other parameters $\theta = \{\mu, \Phi, \Sigma\}$ that describe the subspace. b) In the subspace identity model, there are J data examples \mathbf{x}_{ij} per hidden variable \mathbf{h}_i and all of these J examples have the same identity.

$$\begin{aligned} Pr(\mathbf{h}_i) &= \text{Norm}_{\mathbf{h}_i}[\mathbf{0}, \mathbf{I}] \\ Pr(\mathbf{x}_{ij}|\mathbf{h}_i) &= \text{Norm}_{\mathbf{x}_{ij}}[\mu + \Phi \mathbf{h}_i, \Sigma], \end{aligned} \quad (18.7)$$

where as before we have defined a prior over the hidden variables. The graphical models for both factor analysis and the subspace identity model are illustrated in figure 18.4. Figure 18.5 illustrates ancestral sampling from the subspace identity model; as desired this produces data points that lie close together when the identity is the same.

One way to think of this is that we have decomposed the variance in the model into two parts. The *between-individual* variation explains the differences between data due to different identities and the *within-individual* variation explains the differences between data examples due to all other factors. The data density for a single data point remains

$$Pr(\mathbf{x}_{ij}) = \text{Norm}_{\mathbf{x}_{ij}}[\mu, \Phi \Phi^T + \Sigma]. \quad (18.8)$$

However, the two components of the variance now have clear interpretations. The term $\Phi \Phi^T$ corresponds to the between-individual variation, and the term Σ is the within-individual variation.

18.1.1 Learning

Before we consider how to use this model to draw inferences about identity in face recognition tasks, we will briefly discuss how to learn the parameters $\theta = \{\mu, \Phi, \Sigma\}$. As for the factor analysis model, we exploit the EM algorithm to iteratively increase a bound on the log likelihood. In the E-step we compute the posterior probability distribution over each of the hidden variables \mathbf{h}_i given all of the data $\mathbf{x}_{i\bullet} = \{\mathbf{x}_{ij}\}_{j=1}^J$ associated with that particular identity,

Algorithm 18.1

Problem 18.1

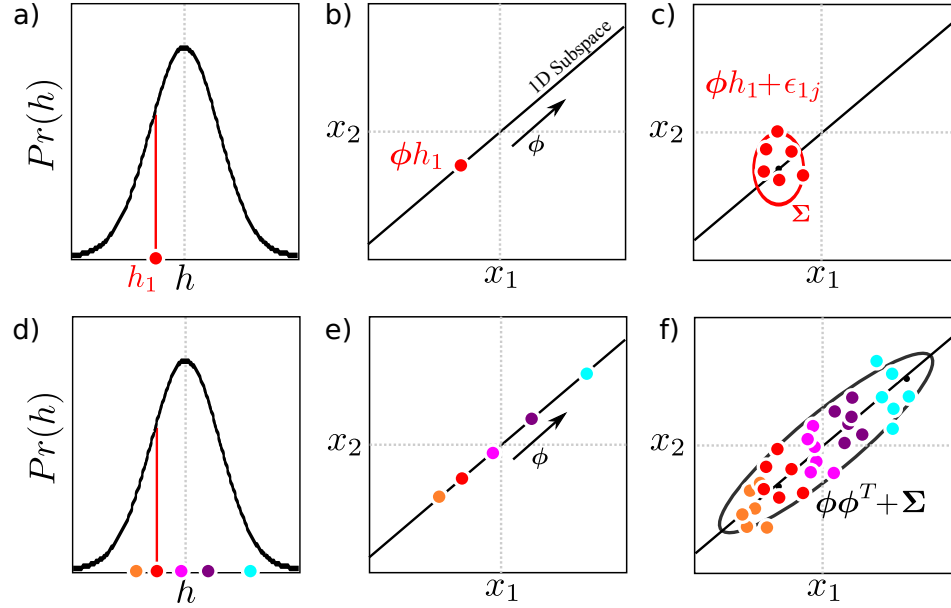


Figure 18.5 Ancestral sampling from identity subspace model. a) To generate from this model we first choose the hidden variable \mathbf{h}_i from the normally distributed prior. Here h_i is a 1D variable and a small negative number is drawn. b) We weight the factors Φ by the hidden variable. This generates a point on the subspace. c) Then we add different noise terms $\{\epsilon_{ij}\}_{j=1}^J$ to create each of the J examples $\{\mathbf{x}_{ij}\}_{j=1}^J$. In each case, the noise is normally distributed with covariance Σ . Finally, we would add a mean term μ (not shown). d-f) This process is repeated several times. The final distribution of the data in f) is a normal distribution with covariance $\Phi\Phi^T + \Sigma$. However, it is structured so that points with the same hidden variable (identity) are close to one another.

$$\begin{aligned}
 Pr(\mathbf{h}_i|\mathbf{x}_{i\bullet}) &= \frac{\prod_{j=1}^J Pr(\mathbf{x}_{ij}|\mathbf{h}_i)Pr(\mathbf{h}_i)}{\int \prod_{j=1}^J Pr(\mathbf{x}_{ij}|\mathbf{h}_i)Pr(\mathbf{h}_i) d\mathbf{h}_i} \\
 &= \text{Norm}_{\mathbf{h}_i} \left[\left(J\Phi^T \Sigma^{-1} \Phi + \mathbf{I} \right)^{-1} \Phi^T \Sigma^{-1} \sum_{j=1}^J (\mathbf{x}_{ij} - \mu), \left(J\Phi^T \Sigma^{-1} \Phi + \mathbf{I} \right)^{-1} \right].
 \end{aligned} \tag{18.9}$$

From this we extract the moments that will be needed in the M-step:

$$\begin{aligned}
 E[\mathbf{h}_i] &= (J\Phi^T \Sigma^{-1} \Phi + \mathbf{I})^{-1} \Phi^T \Sigma^{-1} \sum_{j=1}^J (\mathbf{x}_{ij} - \mu) \\
 E[\mathbf{h}_i \mathbf{h}_i^T] &= (J\Phi^T \Sigma^{-1} \Phi + \mathbf{I})^{-1} + E[\mathbf{h}_i] E[\mathbf{h}_i]^T.
 \end{aligned} \tag{18.10}$$

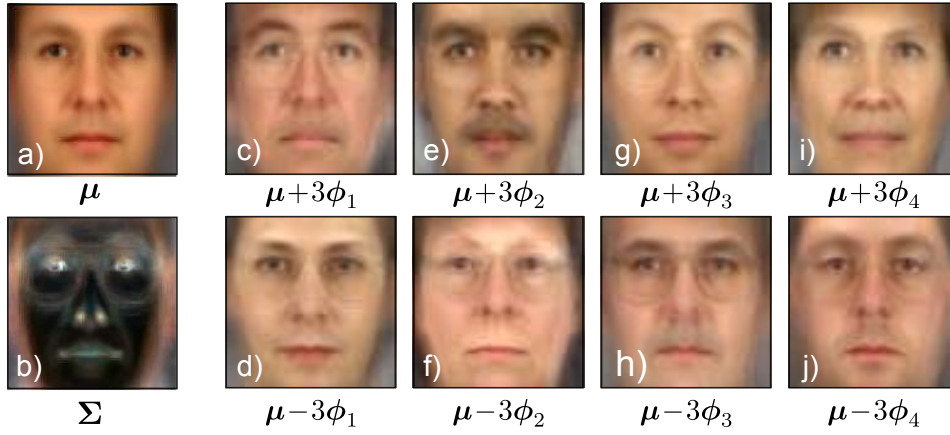


Figure 18.6 Subspace identity model parameters. These parameters were learned from $J = 3$ images of $I = 195$ individuals from the XM2VTS data set. a) Estimated mean μ . b) Estimated covariance, Σ . c-l) Four of 32 subspace directions explored by adding and subtracting multiples of each dimension to the mean.

In the M-step we update the parameters using the relations

Problem 18.2

$$\begin{aligned}
 \hat{\mu} &= \frac{\sum_{i=1}^I \sum_{j=1}^J \mathbf{x}_{ij}}{IJ} \\
 \hat{\Phi} &= \left(\sum_{i=1}^I \sum_{j=1}^J (\mathbf{x}_{ij} - \hat{\mu}) \mathbf{E}[\mathbf{h}_i]^T \right) \left(\sum_{i=1}^I J \mathbf{E}[\mathbf{h}_i \mathbf{h}_i^T] \right)^{-1} \\
 \Sigma &= \frac{1}{IJ} \sum_{i=1}^I \sum_{j=1}^J \text{diag} \left[(\mathbf{x}_{ij} - \hat{\mu})(\mathbf{x}_{ij} - \hat{\mu})^T - \hat{\Phi} \mathbf{E}[\mathbf{h}_i] (\mathbf{x}_{ij} - \hat{\mu})^T \right],
 \end{aligned} \tag{18.11}$$

which were generated by taking the derivative of the EM bound with respect to the relevant quantities, equating the results to zero, and re-arranging. We alternate the E- and M-steps until the log-likelihood of the data no longer increases.

Figure 18.6 shows parameters learned from 70×70 pixel face images from the XM2VTS database. A model with a $K=32$ dimensional hidden space was learned with 195 identities and 3 images per person. The subspace directions capture major changes that correlate with identity. For example, ethnicity and gender are clearly represented. The noise describes whatever remains. It is most prominent around high-contrast features such as the eyes.

In figure 18.7 we decompose pairs of matching images into their identity and noise components. To accomplish this, we compute the MAP hidden variable $\hat{\mathbf{h}}_i$. The posterior over \mathbf{h} is normal (equation 18.9) and so the MAP estimate is simply the mean of this normal. We can then visualize the identity component

Problem 18.3

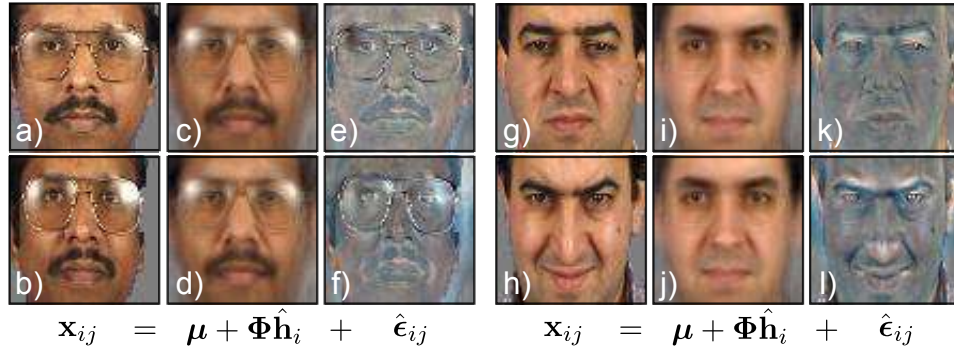


Figure 18.7 Fitting the subspace identity model to new data. a-b) Original images \mathbf{x}_{i1} and \mathbf{x}_{i2} . These faces can be decomposed into the sum of c-d) an identity component and e-f) a within-individual noise component. To decompose the image in this way, we computed the MAP estimate $\hat{\mathbf{h}}_i$ of the hidden variable and set the identity component to be $\boldsymbol{\mu} + \boldsymbol{\Phi} \hat{\mathbf{h}}_i$. The noise comprises whatever cannot be explained by the identity. g-l) A second example.

$\boldsymbol{\mu} + \boldsymbol{\Phi} \hat{\mathbf{h}}_i$ which is the same for each image of the same individual and looks like a prototypical view of that person. We can also visualize the within-individual noise $\hat{\boldsymbol{\epsilon}}_{ij} = \mathbf{x}_{ij} - \boldsymbol{\mu} - \boldsymbol{\Phi} \hat{\mathbf{h}}_i$, which explains how each image of the same person differs.

18.1.2 Inference

We will now discuss how to exploit the model to make inferences about new faces that were not part of the training data set. In face verification problems, we observe two data examples \mathbf{x}_1 and \mathbf{x}_2 and wish to infer the state of the world $w \in \{0, 1\}$ where $w = 0$ denotes the case where the data examples have different identities and $w = 1$ denotes the case where the data examples have the same identity.

This is a generative model, and so we calculate the posterior $Pr(w|\mathbf{x}_1, \mathbf{x}_2)$ over the world state using Bayes' rule

$$Pr(w = 1|\mathbf{x}_1, \mathbf{x}_2) = \frac{Pr(\mathbf{x}_1, \mathbf{x}_2|w = 1)Pr(w = 1)}{\sum_{n=0}^1 Pr(\mathbf{x}_1, \mathbf{x}_2|w = n)Pr(w = n)}. \quad (18.12)$$

To compute this we need the prior probabilities $Pr(w = 0)$ and $Pr(w = 1)$ of the data examples having different identities or the same identity. In the absence of any other information we might set these both to 0.5. We also need expressions for the likelihoods $Pr(\mathbf{x}_1, \mathbf{x}_2|w = 0)$ and $Pr(\mathbf{x}_1, \mathbf{x}_2|w = 1)$.

We will first consider the likelihood $Pr(\mathbf{x}_1, \mathbf{x}_2|w = 0)$ when the two data points have different identities. Here, each image is explained by a different hidden variable and so the generative equation looks like

$$\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\Phi} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Phi} \end{bmatrix} \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \end{bmatrix} + \begin{bmatrix} \boldsymbol{\epsilon}_1 \\ \boldsymbol{\epsilon}_2 \end{bmatrix}, \quad (18.13)$$

We note that this has the form of a factor analyzer:

$$\mathbf{x}' = \boldsymbol{\mu}' + \boldsymbol{\Phi}' \mathbf{h}' + \boldsymbol{\epsilon}'. \quad (18.14)$$

We can re-express this in probabilistic terms as

$$\begin{aligned} Pr(\mathbf{x}'|\mathbf{h}') &= \text{Norm}_{\mathbf{x}'}[\boldsymbol{\mu}' + \boldsymbol{\Phi}' \mathbf{h}', \boldsymbol{\Sigma}'] \\ Pr(\mathbf{h}') &= \text{Norm}_{\mathbf{h}'}[\mathbf{0}, \mathbf{I}], \end{aligned} \quad (18.15)$$

where $\boldsymbol{\Sigma}'$ is defined as

$$\boldsymbol{\Sigma}' = \begin{bmatrix} \boldsymbol{\Sigma} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma} \end{bmatrix}. \quad (18.16)$$

We can now compute the likelihood $Pr(\mathbf{x}_1, \mathbf{x}_2|w = 0)$ by writing the joint likelihood of the compound variables \mathbf{x}' and \mathbf{h}' and marginalizing over \mathbf{h}' , so that

$$\begin{aligned} Pr(\mathbf{x}_1, \mathbf{x}_2|w = 0) &= \int Pr(\mathbf{x}'|\mathbf{h}') Pr(\mathbf{h}') d\mathbf{h}' \\ &= \text{Norm}_{\mathbf{x}'}[\boldsymbol{\mu}', \boldsymbol{\Phi}' \boldsymbol{\Phi}'^T + \boldsymbol{\Sigma}'], \end{aligned} \quad (18.17)$$

where we have used the standard factor analysis result for the integration.

For the case where the faces match ($w = 1$), we know that both data examples must have been created from the same hidden variable. To compute the likelihood $Pr(\mathbf{x}_1, \mathbf{x}_2|w = 1)$, we write the compound generative equation

$$\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\Phi} \\ \boldsymbol{\Phi} \end{bmatrix} \mathbf{h}_{12} + \begin{bmatrix} \boldsymbol{\epsilon}_1 \\ \boldsymbol{\epsilon}_2 \end{bmatrix}, \quad (18.18)$$

which we notice also has the form of a standard factor analyzer (equation 18.14) and so we can compute the likelihood using the same method.

One way to think about this process is that we are comparing the likelihood for two different models of the data (figure 18.8a). However, it should be noted that the model that categorizes the faces as different ($w = 0$) has two variables (\mathbf{h}_1 and \mathbf{h}_2), whereas the model that categorizes the faces as the same ($w = 1$) has only one (\mathbf{h}_{12}). One might expect then that the model with more variables would always provide a superior explanation of the data. In fact this does not happen here, because we marginalized these variables out of the likelihoods, and so the final expressions do not include these hidden variables. This is an example of *Bayesian model selection*: it is valid to compare models with different numbers of parameters as long as they are marginalized out of the final solution.

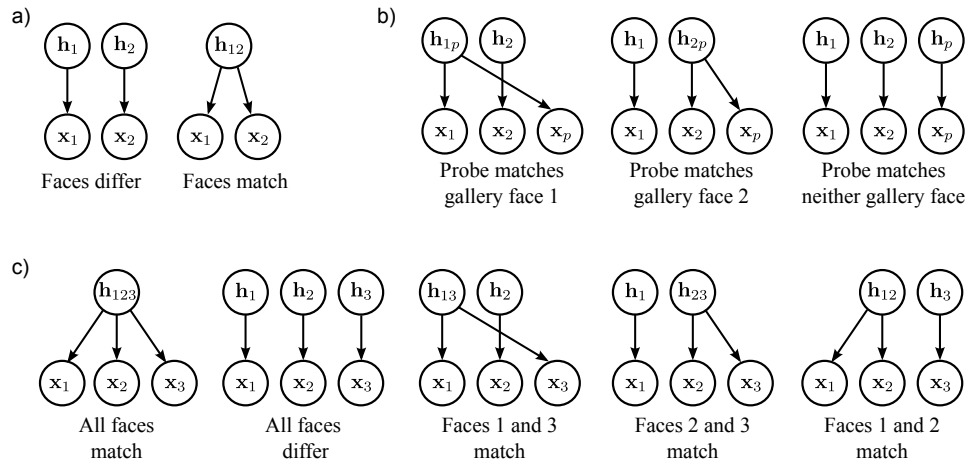


Figure 18.8 Inference as model comparison. a) Verification task. Given two faces x_1 and x_2 we must decide whether (i) they belong to different people and hence have separate hidden variables h_1, h_2 or (ii) they belong to the same person and hence share a single hidden variable h_{12} . These two hypotheses are illustrated as the two graphical models. b) Open set identification task. We are given a library $\{x_i\}_{i=1}^I$ of faces that belong to different people and a probe face x_p . In this case (where $I = 2$), we must decide whether the probe matches (i) gallery face 1, (ii) gallery face 2, or (iii) none of the gallery faces. In closed set identification, we simply omit the latter model. c) Clustering task. Given three faces x_1, x_2 , and x_3 we must decide whether (i) they are all from the same person, (ii) all from different people, or (iii-v) two of the three match.

18.1.3 Inference in other recognition tasks

Face verification is only one of several possible face recognition problems. Others include:

- *Closed set identification*: find which one of N gallery faces matches a given probe face.
- *Open set identification*: choose one of N gallery faces that matches a probe or identify that there is no match in the gallery.
- *Clustering*: given N faces, find how many different people are present and which face belongs to which person.

Problem 18.4

All of these models can be thought of in terms of model comparison (figure 18.8). For example, consider a clustering task in which we have three faces x_1, x_2, x_3 and wish to know if (i) there are three different identities, or (ii) all of the images belong to the same person, or (iii) two images belong to the same person and the third belongs to someone different (distinguishing between the three different ways that

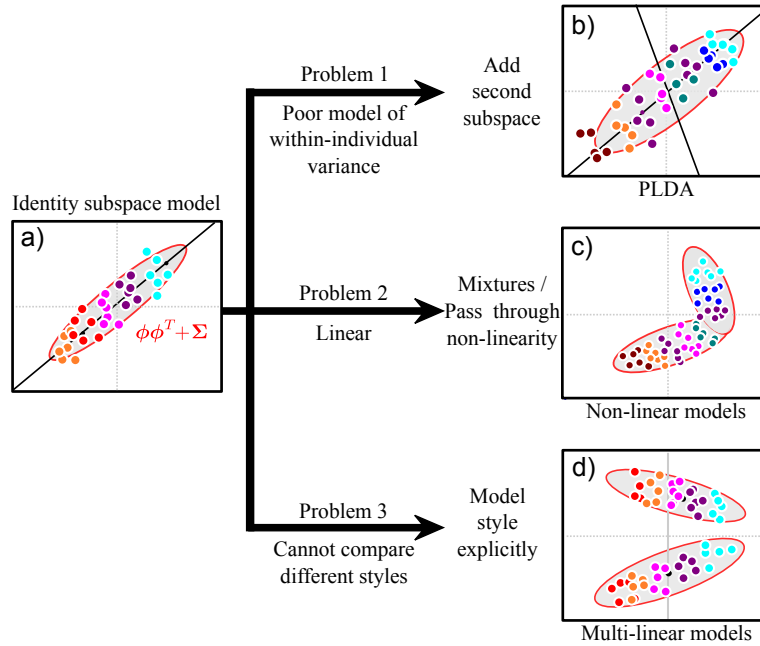


Figure 18.9 Identity models. a) There are three limitations to the subspace identity model. b) First, it has an impoverished model of the within-individual noise. To remedy this we develop probabilistic linear discriminant analysis. c) Second, it is linear and can only describe the distribution of faces as a normal distribution. Hence, we develop nonlinear models based on mixtures and kernels. d) Third, it does not work well when there are large style changes. To cope with this, we introduce multi-linear models.

this can happen). The world can take five states $w \in \{1, 2, 3, 4, 5\}$ corresponding to these five situations, and each is explained by a different compound generative equation. For example, if the first two images are the same person, but the third is different we would write

$$\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu} \\ \boldsymbol{\mu} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\Phi} & \mathbf{0} \\ \boldsymbol{\Phi} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Phi} \end{bmatrix} \begin{bmatrix} \mathbf{h}_{12} \\ \mathbf{h}_3 \end{bmatrix} + \begin{bmatrix} \boldsymbol{\epsilon}_1 \\ \boldsymbol{\epsilon}_2 \\ \boldsymbol{\epsilon}_3 \end{bmatrix}, \quad (18.19)$$

which again has the form of a factor analyzer and so we can compute the likelihood using the method described earlier. We compare the likelihood for this model to the likelihoods for the other models using Bayes' rule with suitable priors.

Problem 18.5

18.1.4 Limitations of identity subspace model

The subspace identity model has three main limitations (figure 18.9).

1. The model of within-individual covariance (diagonal) is inadequate.
2. It is a linear model and cannot model non-Gaussian densities.
3. It cannot model large changes in style (e.g., frontal vs. profile faces).

Problem 18.6
Problem 18.7

We tackle these problems by introducing probabilistic linear discriminant analysis (section 18.2), nonlinear identity models (section 18.3), and multi-linear models (sections 18.4-18.5), respectively.

18.2 Probabilistic linear discriminant analysis

The subspace identity model explains the data as the sum of a component due to the identity and an additive noise term. However, the noise term is rather simple: it describes the within-individual variation as a normal distribution with diagonal covariance. The estimated noise components that we visualized in figure 18.7 contain considerable structure, which suggests that modeling the within-individual variation at each pixel as independent is insufficient.

Probabilistic linear discriminant analysis (PLDA) uses a more sophisticated model for the within-individual variation. This model adds a new term to the generative equation that describes the within-individual variation as also lying on a subspace determined by a second factor matrix Ψ . The j^{th} image \mathbf{x}_{ij} of the i^{th} individual is now described as

$$\mathbf{x}_{ij} = \boldsymbol{\mu} + \Phi \mathbf{h}_i + \Psi \mathbf{s}_{ij} + \boldsymbol{\epsilon}, \quad (18.20)$$

where \mathbf{s}_{ij} is a hidden variable that represents the *style* of this face: it describes systematic contributions to the image from uncontrolled viewing parameters. Notice that it differs for each instance j , and so it tells us nothing about identity.

The columns of Φ describe the space of between-individual variation and \mathbf{h}_i determines a point in this space. The columns of Ψ describe the space of within-individual variation, and \mathbf{s}_{ij} determines a point in this space. A given face is now modeled as the sum of a term $\boldsymbol{\mu} + \Phi \mathbf{h}_i$ that derives from the identity of the individual, a term $\Psi \mathbf{s}_{ij}$ that models the style of this particular image, and a noise term $\boldsymbol{\epsilon}_{ij}$ that explains any remaining variation (figure 18.10).

Once again, we can write the model in probabilistic terms:

$$\begin{aligned} Pr(\mathbf{h}_i) &= \text{Norm}_{\mathbf{h}_i}[\mathbf{0}, \mathbf{I}] \\ Pr(\mathbf{s}_{ij}) &= \text{Norm}_{\mathbf{s}_{ij}}[\mathbf{0}, \mathbf{I}] \\ Pr(\mathbf{x}_{ij} | \mathbf{h}_i, \mathbf{s}_{ij}) &= \text{Norm}_{\mathbf{x}_{ij}}[\boldsymbol{\mu} + \Phi \mathbf{h}_i + \Psi \mathbf{s}_{ij}, \Sigma], \end{aligned} \quad (18.21)$$

where now we have defined priors over both hidden variables.

18.2.1 Learning

Algorithm 18.2

In the E-step, we collect together all the J observations $\{\mathbf{x}_{ij}\}_{j=1}^J$ associated with

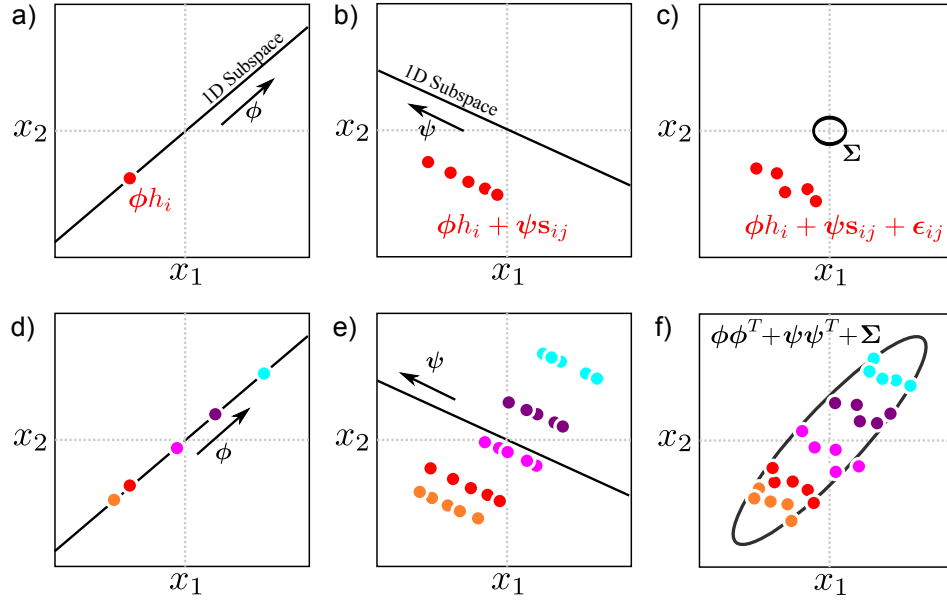


Figure 18.10 Ancestral sampling from PLDA model. a) We sample a hidden variable \mathbf{h}_i from the identity prior and use this to weight the between-individual factors Φ . b) We sample J hidden variables $\{\mathbf{s}_{ij}\}_{j=1}^J$ from the style prior and use these to weight the within-individual factors Ψ . c) Finally, we add normal noise with diagonal covariance, Σ . d-f) This process is repeated for several individuals. Notice that the clusters associated with each identity in f) are now oriented (compare to figure 18.5f); we have constructed a more sophisticated model of within-individual variation.

the same identity to form the compound system

$$\begin{bmatrix} \mathbf{x}_{i1} \\ \mathbf{x}_{i2} \\ \vdots \\ \mathbf{x}_{iJ} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu} \\ \vdots \\ \boldsymbol{\mu} \end{bmatrix} + \begin{bmatrix} \Phi & \Psi & \mathbf{0} & \dots & \mathbf{0} \\ \Phi & \mathbf{0} & \Psi & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \Phi & \mathbf{0} & \mathbf{0} & \dots & \Psi \end{bmatrix} \begin{bmatrix} \mathbf{h}_i \\ \mathbf{s}_{i1} \\ \mathbf{s}_{i2} \\ \vdots \\ \mathbf{s}_{iJ} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\epsilon}_{i1} \\ \boldsymbol{\epsilon}_{i2} \\ \vdots \\ \boldsymbol{\epsilon}_{iJ} \end{bmatrix}, \quad (18.22)$$

which takes the form of the original subspace identity model $\mathbf{x}'_i = \boldsymbol{\mu}' + \Phi' \mathbf{h}'_i + \boldsymbol{\epsilon}'$. We can hence compute the joint posterior probability distribution over all of the hidden variables in \mathbf{h}' using equation 18.9.

In the M-step we write a compound generative equation for each image,

$$\mathbf{x}_{ij} = \boldsymbol{\mu} + [\Phi \quad \Psi] \begin{bmatrix} \mathbf{h}_i \\ \mathbf{s}_{ij} \end{bmatrix} + \boldsymbol{\epsilon}_{ij}. \quad (18.23)$$

On noting that this has the form $\mathbf{x}_{ij} = \boldsymbol{\mu} + \Phi'' \mathbf{h}''_{ij} + \boldsymbol{\epsilon}_{ij}$ of the standard factor analysis model, we can solve for the unknown parameters using equations 18.5.

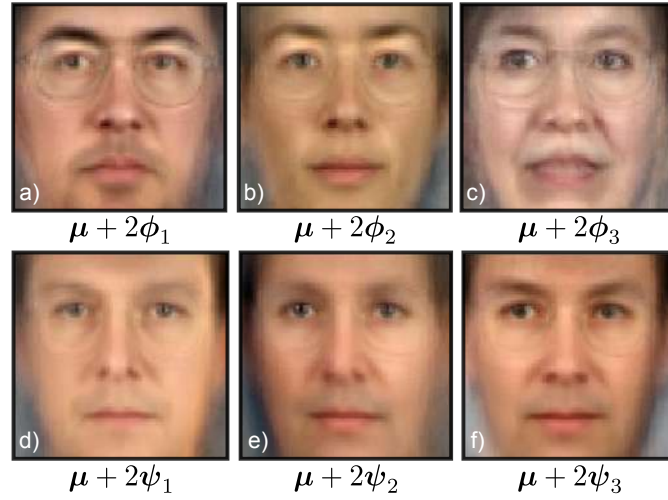


Figure 18.11 PLDA model. a-c) As we move around in the between-individual subspace Φ the images look like different people. d-f) As we move around in the within-individual subspace Ψ the images look like the same person viewed in slightly different poses and under different illuminations. The PLDA model has successfully separated out contributions that correlate with identity from those that don't. Adapted from Li *et al.* (2011). ©2012 IEEE.

The computations require the expectations $E[\mathbf{h}_{ij}'']$ and $E[\mathbf{h}_{ij}''\mathbf{h}_{ij}'^T]$, and these can be extracted from the posterior computed in the E-step.

Figure 18.11 shows parameters learned from $J = 3$ examples each of $I = 195$ people from the XM2VTS database with 16 between-individual basis functions in Φ and 16 within-individual basis functions in Ψ . The figure demonstrates that the model has distinguished these two components.

18.2.2 Inference

As for the subspace identity model, we perform inference by comparing the likelihoods of models using Bayes's rule. For example, in the verification task we compare models that explain the two data examples \mathbf{x}_1 and \mathbf{x}_2 as having either their own identities \mathbf{h}_1 and \mathbf{h}_2 or sharing a single identity \mathbf{h}_{12} . When the identities are different ($w=0$), the data are generated as

$$\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu} \end{bmatrix} + \begin{bmatrix} \Phi & \mathbf{0} & \Psi & \mathbf{0} \\ \mathbf{0} & \Phi & \mathbf{0} & \Psi \end{bmatrix} \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix} + \begin{bmatrix} \boldsymbol{\epsilon}_1 \\ \boldsymbol{\epsilon}_2 \end{bmatrix}. \quad (18.24)$$

When the identities are the same ($w=1$), the data are generated as

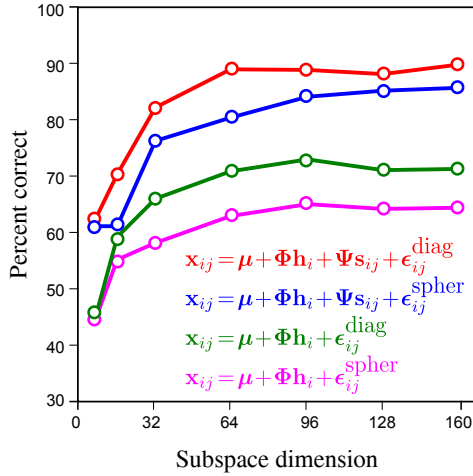


Figure 18.12 Face recognition results. The models were trained using three 70×70 RGB images each from 195 people from the XM2VTS database and tested using two images each from 100 different people. A gallery was formed from one image of each of the test individuals. For each of the remaining 100 test images, the system had to identify the match in the gallery. Plots show % correct performance as a function of subspace dimensionality (number of columns in Φ and Ψ). Results show that as the noise model becomes more complex (adding within-individual subspace, using diagonal rather than spherical additive noise) the results improve systematically.

$$\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu} \end{bmatrix} + \begin{bmatrix} \Phi & \Psi & \mathbf{0} \\ \Phi & \mathbf{0} & \Psi \end{bmatrix} \begin{bmatrix} \mathbf{h}_{12} \\ \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix} + \begin{bmatrix} \boldsymbol{\epsilon}_1 \\ \boldsymbol{\epsilon}_2 \end{bmatrix}. \quad (18.25)$$

Both of these formulae have the same form $\mathbf{x}' = \boldsymbol{\mu}' + \Phi' \mathbf{h}' + \boldsymbol{\epsilon}'$ as the original factor analysis model, and so the likelihood of the data \mathbf{x}' after marginalizing out the hidden variables \mathbf{h}' is given by

$$Pr(\mathbf{x}') = \text{Norm}_{\mathbf{x}'}[\boldsymbol{\mu}', \Phi' \Phi'^T + \Sigma'], \quad (18.26)$$

where the particular choice of Φ' comes from equations 18.24 or 18.25, respectively. Other inference tasks concerning identity such as closed set recognition and clustering can be formulated in a similar way; we associate one value of the discrete world variable $w = \{1, \dots, K\}$ with each possible configuration of identities, construct a generative model for each and compare the likelihoods via Bayes' rule.

Figure 18.12 compares closed-set identification performance for several models as a function of the subspace size (for the PLDA models, the size of Ψ and Φ were always the same). The results are not state-of-the-art: the images were not properly pre-processed, and anyway, this data set is considered relatively unchallenging. Nonetheless, the pattern of results nicely demonstrates an important point. The %-correct classification improves as we increase the model's ability to describe within-individual noise: building more complex models is worth the time and effort!

18.3 Non-linear identity models

The models discussed so far describe the between-individual and within-individual variance by means of linear models and produce final densities that are normally

distributed. However, there is no particular reason to believe that the distribution of faces is normal. We now briefly discuss two methods to generalize the preceding models to the nonlinear case.

The first approach is to note that since the identity subspace model and PLDA are both valid probabilistic models, we can easily describe a more complex distribution in terms of mixtures of these elements. For example, a mixture of PLDAs model (figure 18.13) can be written as

$$\begin{aligned} Pr(c_i) &= \text{Cat}_{c_i}[\boldsymbol{\lambda}] \\ Pr(\mathbf{h}_i) &= \text{Norm}_{\mathbf{h}_i}[\mathbf{0}, \mathbf{I}] \\ Pr(\mathbf{s}_{ij}) &= \text{Norm}_{\mathbf{s}_{ij}}[\mathbf{0}, \mathbf{I}] \\ Pr(\mathbf{x}_{ij}|c_i, \mathbf{h}_i, \mathbf{s}_{ij}) &= \text{Norm}_{\mathbf{x}_{ij}}[\boldsymbol{\mu}_{c_i} + \boldsymbol{\Phi}_{c_i}\mathbf{h}_i + \boldsymbol{\Psi}_{c_i}\mathbf{s}_{ij}, \boldsymbol{\Sigma}_{c_i}], \end{aligned} \quad (18.27)$$

where $c_i \in [1 \dots C]$ is a hidden variable that determines to which of the c clusters the data belong. Each cluster has different parameters, so the full model is nonlinear. To learn this model, we embed the existing learning algorithm inside a second EM loop that associates each identity with a cluster. In inference, we assume that faces must belong to the same cluster if they match.

A second approach is based on the Gaussian process latent variable model (see section 17.8). The idea is to induce a complex density by passing the hidden variable through a nonlinear function $\mathbf{f}[\bullet]$ before using the result to weight the basis functions. For example, the generalization of the subspace identity model to the nonlinear case can be written as

$$\begin{aligned} Pr(\mathbf{h}_i) &= \text{Norm}_{\mathbf{h}_i}[\mathbf{0}, \mathbf{I}] \\ Pr(\mathbf{x}_{ij}|\mathbf{h}_i, \boldsymbol{\mu}, \boldsymbol{\Phi}, \boldsymbol{\Sigma}) &= \text{Norm}_{\mathbf{x}_{ij}}[\boldsymbol{\mu} + \boldsymbol{\Phi}\mathbf{f}[\mathbf{h}_i], \boldsymbol{\Sigma}]. \end{aligned} \quad (18.28)$$

Although this model is conceptually simple, it is harder to work with in practice: it is no longer possible to marginalize over the hidden variables. However, the model is still linear with respect to the factor matrix $\boldsymbol{\Phi}$, and it is possible to marginalize over this and the mean $\boldsymbol{\mu}$, giving a likelihood term of the form

$$Pr(\mathbf{x}_{ij}|\mathbf{h}_i, \boldsymbol{\Sigma}) = \int \int \text{Norm}_{\mathbf{x}_{ij}}[\boldsymbol{\mu} + \boldsymbol{\Phi}\mathbf{f}[\mathbf{h}_i], \boldsymbol{\Sigma}] d\boldsymbol{\mu} d\boldsymbol{\Phi}. \quad (18.29)$$

This model can be expressed in terms of inner products of the transformed hidden variables $\mathbf{f}[\mathbf{h}]$ and so is amenable to kernelization. Unfortunately, because we cannot marginalize over \mathbf{h}_i , it is no longer possible exactly to compare model likelihoods directly in the inference stage. However, in practice there are ways to approximate this process.

18.4 Asymmetric bilinear models

The models that we have discussed so far are sufficient if the within-individual variation is small. However, there are other situations where the style of the data

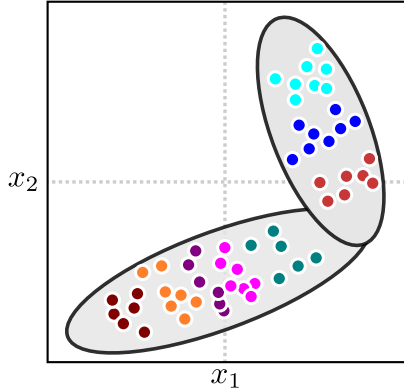


Figure 18.13 Mixture models. One way to create more complex models is to use a mixture of subspace identity models, or a mixture of PLDAs. A discrete variable is associated with each data point that indicates to which mixture component it belongs. If two faces belong to the same person, this must be the same; every image of the same person is associated with one mixture component. The within-individual model may also vary between components. Consequently, the within-individual variation may differ depending on the identity of the face.

may change considerably. For example, consider the problem of face recognition when some of the faces are frontal and others profile. Unfortunately, any given frontal face has more in common visually with other non-matching frontal faces than it does with a matching profile face.

Motivated by this problem, we now develop the *asymmetric bilinear model* for modeling identity and style: as before, we treat the identity \mathbf{h}_i as continuous, but now we treat the style $s \in \{1 \dots S\}$ as discrete taking one of S possible values. For example, in the cross-pose face recognition example, $s = 0$ might indicate a frontal face, and $s = 1$ might indicate a profile face. The model is hence asymmetric as it treats identity and style differently. The expression of the identity depends on the style category so that the same identity may produce completely different data in different styles.

We adopt the notation \mathbf{x}_{ijs} to denote the j^{th} of J examples of the i^{th} of I identities in the s^{th} of S styles. The data are generated as

$$\mathbf{x}_{ijs} = \boldsymbol{\mu}_s + \boldsymbol{\Phi}_s \mathbf{h}_i + \boldsymbol{\epsilon}_{ijs}, \quad (18.30)$$

where $\boldsymbol{\mu}_s$ is a mean vector associated with the s^{th} style, $\boldsymbol{\Phi}_s$ contains basis functions associated with the s^{th} style, and $\boldsymbol{\epsilon}_{ijs}$ is additive normal noise with a covariance $\boldsymbol{\Sigma}_s$ that also depends on the style. When the noise covariances are spherical, this model is a probabilistic form of *canonical correlation analysis*. When the noise is diagonal, it is known as *tied factor analysis*. We will use the generic term *asymmetric bilinear model* to cover both situations.

Equation 18.30 is easy to parse; for a given individual, the identity \mathbf{h}_i is constant. The data are explained as a weighted linear sum of basis functions, where the weights determine the identity. However, the basis functions (and other aspects of the model) are now contingent on the style.

We can write the model in probabilistic terms as

$$\begin{aligned} Pr(s) &= \text{Cat}_s[\boldsymbol{\lambda}] \\ Pr(\mathbf{h}_i) &= \text{Norm}_{\mathbf{h}_i}[\mathbf{0}, \mathbf{I}] \\ Pr(\mathbf{x}_{ijs} | \mathbf{h}_i, s) &= \text{Norm}_{\mathbf{x}_{ijs}}[\boldsymbol{\mu}_s + \boldsymbol{\Phi}_s \mathbf{h}_i, \boldsymbol{\Sigma}_s], \end{aligned} \quad (18.31)$$

Problem 18.8

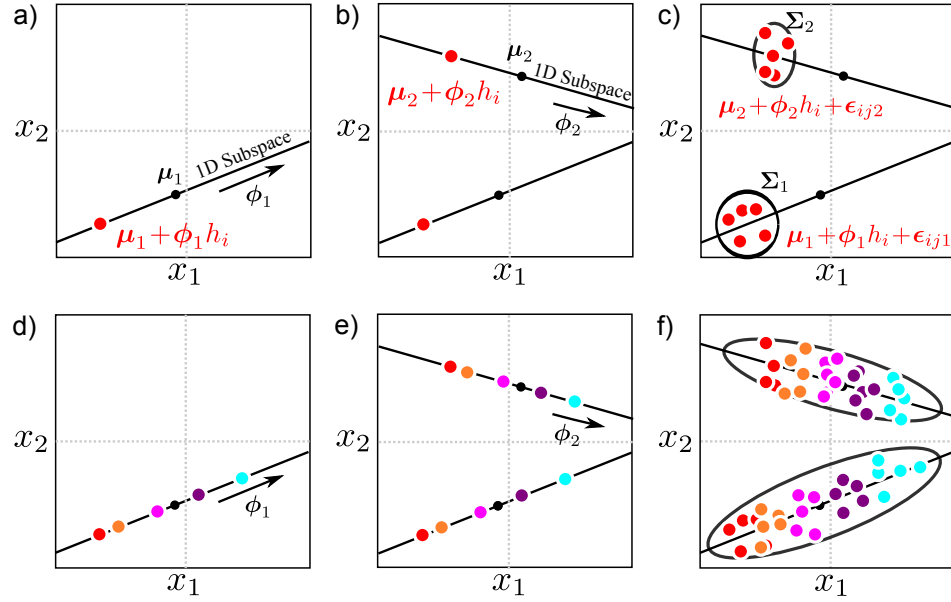


Figure 18.14 Asymmetric bilinear model with two styles. a) We draw a hidden variable \mathbf{h}_i from the prior and use this to weight basis functions Φ_1 (one basis function ϕ_1 shown). The result is added to the mean μ_1 . b) We use the same value of \mathbf{h}_i to weight a second set of basis functions Φ_2 and add the result to μ_2 . c) We add normally distributed noise with a diagonal covariance Σ_s that depends on the style. d-f) When we repeat this procedure, it produces one normal distribution per style. The data within each style are structured so that nearby points have the same identity (color) and identities that are close in one cluster are also close in the other cluster.

where λ contains parameters that determine the probability of observing data in each style. Figure 18.14 demonstrates ancestral sampling from this model. If we marginalize over the identity parameter \mathbf{h} and the style parameter s , the overall data distribution (without regard to the structure of the style clusters) is a mixture of factor analyzers,

$$Pr(\mathbf{x}) = \sum_{s=1}^S \lambda_s \text{Norm}_{\mathbf{x}}[\mu_s, \Phi_s \Phi_s^T + \Sigma_s]. \quad (18.32)$$

18.4.1 Learning

Algorithm 18.3

For simplicity, we will assume that the styles of each training example are known and so it is also trivial to estimate the categorical parameters λ . As for the previous models in this chapter, we employ the EM algorithm.

In the E-step, we compute a posterior distribution over the hidden variable \mathbf{h}_i that represents the identity, using all of the training data for that individual regardless of the style. Employing Bayes' rule we have

$$Pr(\mathbf{h}_i | \mathbf{x}_{i\bullet\bullet}) = \frac{\prod_{j=1}^J \prod_{s=1}^S Pr(\mathbf{x}_{ijs} | \mathbf{h}_i) Pr(\mathbf{h}_i)}{\int \prod_{j=1}^J \prod_{s=1}^S Pr(\mathbf{x}_{ijs} | \mathbf{h}_i) Pr(\mathbf{h}_i) d\mathbf{h}_i}, \quad (18.33)$$

where $\mathbf{x}_{i\bullet\bullet} = \{\mathbf{x}_{ijs}\}_{j,s=1}^{J,S}$ denotes all the data associated with the i^{th} individual.

One way to compute this is to write a compound generative equation for $\mathbf{x}_{i\bullet\bullet}$. For example, with $J = 2$ images at each of $S = 2$ styles we would have

$$\begin{bmatrix} \mathbf{x}_{i11} \\ \mathbf{x}_{i12} \\ \mathbf{x}_{i21} \\ \mathbf{x}_{i22} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \\ \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix} + \begin{bmatrix} \boldsymbol{\Phi}_1 \\ \boldsymbol{\Phi}_2 \\ \boldsymbol{\Phi}_1 \\ \boldsymbol{\Phi}_2 \end{bmatrix} \mathbf{h}_i + \begin{bmatrix} \boldsymbol{\epsilon}_{11} \\ \boldsymbol{\epsilon}_{12} \\ \boldsymbol{\epsilon}_{21} \\ \boldsymbol{\epsilon}_{22} \end{bmatrix}, \quad (18.34)$$

which has the same form as the identity subspace model, $\mathbf{x}'_{ij} = \boldsymbol{\mu}' + \boldsymbol{\Phi}'\mathbf{h}_i + \boldsymbol{\epsilon}'_{ij}$. We can hence compute the posterior distribution using equation 18.9 and extract the expected values needed for the M-step using equation 18.10.

In the M-step we update the parameters $\boldsymbol{\theta}_s = \{\boldsymbol{\mu}_s, \boldsymbol{\Phi}_s, \boldsymbol{\Sigma}_s\}$ for each style separately using all of the relevant data. This gives the updates

$$\begin{aligned} \hat{\boldsymbol{\mu}}_s &= \frac{\sum_{i=1}^I \sum_{j=1}^J \mathbf{x}_{ijs}}{IJ} \\ \hat{\boldsymbol{\Phi}}_s &= \left(\sum_{i=1}^I \sum_{j=1}^J (\mathbf{x}_{ijs} - \hat{\boldsymbol{\mu}}_s) \mathbf{E}[\mathbf{h}_i]^T \right) \left(J \sum_{i=1}^I \mathbf{E}[\mathbf{h}_i \mathbf{h}_i^T] \right)^{-1} \\ \hat{\boldsymbol{\Sigma}}_s &= \frac{1}{IJ} \sum_{i=1}^I \sum_{j=1}^J \text{diag} \left[(\mathbf{x}_{ijs} - \hat{\boldsymbol{\mu}}_s)^T (\mathbf{x}_{ijs} - \hat{\boldsymbol{\mu}}_s) - \hat{\boldsymbol{\Phi}}_s \mathbf{E}[\mathbf{h}_i] \mathbf{x}_{ijs}^T \right]. \end{aligned} \quad (18.35)$$

As usual, we iterate these two steps until the system converges and the log likelihood ceases to improve. Figure 18.15 shows examples of the learned parameters for a data set that includes faces at two poses.

18.4.2 Inference

There are a number of possible forms of inference in this model. These include:

1. Given \mathbf{x} , infer the style $s \in \{1, \dots, S\}$.
2. Given \mathbf{x} , infer the parameterized identity \mathbf{h} .
3. Given \mathbf{x}_1 and \mathbf{x}_2 , infer whether they have the same identity or not.
4. Given \mathbf{x}_1 in style s_1 , translate the style to s_2 to create $\hat{\mathbf{x}}_2$.

We will consider each in turn.

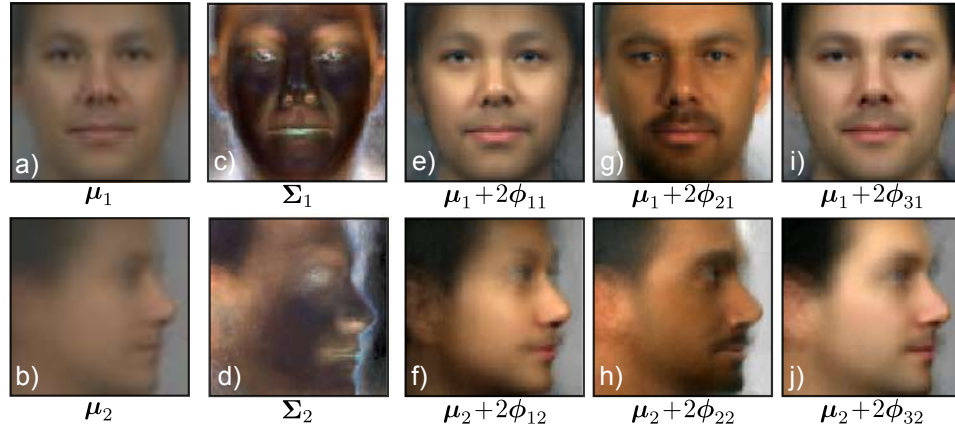


Figure 18.15 Learned parameters of asymmetric bilinear model with two styles (frontal and profile faces). This model was learned from one 70×70 image of each style in 200 individuals from the FERET data set. a,b) Mean vector for each style. c-d) Diagonal covariance for each style. e-f) Varying first basis function in each style (notation ϕ_{ks} denotes k^{th} basis function of s^{th} style). g-h) Varying second basis function in each style. i-l) Varying third basis function in each style. Manipulating the two sets of basis functions in the same way produces images that look like the same person, viewed in each of the styles. Adapted from Prince *et al.* (2008). ©2008 IEEE.

Inferring style

The likelihood of the data given style s but regardless of identity \mathbf{h} is

$$\begin{aligned}
 Pr(\mathbf{x}|s) &= \int Pr(\mathbf{x}|\mathbf{h}, s) Pr(\mathbf{h}) d\mathbf{h} \\
 &= \int \text{Norm}_{\mathbf{x}}[\boldsymbol{\mu}_s + \boldsymbol{\Phi}_s \mathbf{h}, \boldsymbol{\Sigma}_s] Pr(\mathbf{h}) d\mathbf{h} \\
 &= \text{Norm}_{\mathbf{x}}[\boldsymbol{\mu}_s, \boldsymbol{\Phi}_s \boldsymbol{\Phi}_s^T + \boldsymbol{\Sigma}_s].
 \end{aligned} \tag{18.36}$$

The posterior $Pr(s|\mathbf{x})$ over style s can be computed by combining this likelihood with the prior $Pr(s)$ using Bayes' rule. The prior for style s is given by

Problem 18.9

$$Pr(s) = \text{Cat}_s[\boldsymbol{\lambda}]. \tag{18.37}$$

Inferring identity

The likelihood of the data for a fixed identity \mathbf{h} but regardless of style s is

$$\begin{aligned}
Pr(\mathbf{x}|\mathbf{h}) &= \sum_{s=1}^S Pr(\mathbf{x}|\mathbf{h}, s) Pr(s) \\
&= \sum_{s=1}^S \text{Norm}_x[\boldsymbol{\mu}_s + \boldsymbol{\Phi}_s \mathbf{h}, \boldsymbol{\Sigma}_s] \lambda_s.
\end{aligned} \tag{18.38}$$

The posterior over identity can now be combined with the prior $Pr(\mathbf{h})$ using Bayes' rule and is given by

$$Pr(\mathbf{h}|\mathbf{x}) = \sum_{s=1}^S \lambda_s \text{Norm}_{\mathbf{h}_i}[(\boldsymbol{\Phi}_s^T \boldsymbol{\Sigma}_s^{-1} \boldsymbol{\Phi}_s + \mathbf{I})^{-1} \boldsymbol{\Phi}_s^T \boldsymbol{\Sigma}_s^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_s), (\boldsymbol{\Phi}_s^T \boldsymbol{\Sigma}_s^{-1} \boldsymbol{\Phi}_s + \mathbf{I})]. \tag{18.39}$$

Note that this posterior distribution is a mixture of Gaussians, with one component for each possible style.

Identity matching

Given two examples $\mathbf{x}_1, \mathbf{x}_2$, compute the posterior probability that they have the same identity, even though they may be viewed in different styles. We will initially assume the styles are known and are s_1 and s_2 , respectively. We first build the compound model

$$\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\mu}_{s_1} \\ \boldsymbol{\mu}_{s_2} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\Phi}_{s_1} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Phi}_{s_2} \end{bmatrix} \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_1 \end{bmatrix} + \begin{bmatrix} \boldsymbol{\epsilon}_1 \\ \boldsymbol{\epsilon}_2 \end{bmatrix}, \tag{18.40}$$

which represents the case where the identities differ ($w = 0$). We compute the likelihood by noting that this has the form $\mathbf{x}' = \boldsymbol{\mu}' + \boldsymbol{\Phi}' \mathbf{h}' + \boldsymbol{\epsilon}'$ of the original factor analyzer, and so we can write

$$Pr(\mathbf{x}'|w=0) = \text{Norm}_{\mathbf{x}'}[\boldsymbol{\mu}', \boldsymbol{\Phi}' \boldsymbol{\Phi}'^T + \boldsymbol{\Sigma}'], \tag{18.41}$$

where $\boldsymbol{\Sigma}'$ is a diagonal matrix containing the (diagonal) covariances of the elements of $\boldsymbol{\epsilon}'$ (as in equation 18.16).

Similarly, we can build a system for when the identities match ($w=1$)

$$\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\mu}_{s_1} \\ \boldsymbol{\mu}_{s_2} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\Phi}_{s_1} \\ \boldsymbol{\Phi}_{s_2} \end{bmatrix} \mathbf{h}_{12} + \begin{bmatrix} \boldsymbol{\epsilon}_1 \\ \boldsymbol{\epsilon}_2 \end{bmatrix}, \tag{18.42}$$

and compute its likelihood $Pr(\mathbf{x}'|w=1)$ in the same way. The posterior probability $Pr(w=1|\mathbf{x}')$ can then be computed using Bayes' rule.

If we do not know the styles, then each likelihood term will become a mixture of Gaussians where each component has the form of equation 18.41. There will be one component for every one of the S^2 combinations of the two styles. The mixing weights will be given by the probability of observing that combination so that $Pr(s_1 = m, s_2 = n) = \lambda_m \lambda_n$.



Figure 18.16 Style translation based on asymmetric bilinear model from figure 18.15. a) Original face in style 1 (frontal). b) Translated to style 2 (profile). c-d) A second example. Adapted from Prince *et al.* (2008). ©2008 IEEE.

Style translation

Finally, let us consider style translation. Given observed data \mathbf{x} in style s_1 translate to the style s_2 while maintaining the same identity. A simple way to get a point estimate of the translated styles is to first estimate the identity variable \mathbf{h} based on the observed image \mathbf{x}_{s_1} . To do this, we compute the posterior distribution

$$Pr(\mathbf{h}|\mathbf{x}, s_1) = \text{Norm}_{\mathbf{h}_i}[(\Phi_{s_1}^T \Sigma_{s_1}^{-1} \Phi_{s_1} + \mathbf{I})^{-1} \Phi_{s_1}^T \Sigma_{s_1}^{-1} (\mathbf{x}_i - \mu_{s_1}), (\Phi_{s_1}^T \Sigma_{s_1}^{-1} \Phi_{s_1} + \mathbf{I})]. \quad (18.43)$$

and then set \mathbf{h} to the MAP estimate

$$\begin{aligned} \hat{\mathbf{h}} &= \underset{\mathbf{h}}{\text{argmax}} [Pr(\mathbf{h}|\mathbf{x}, s_1)] \\ &= (\Phi_{s_1}^T \Sigma_{s_1}^{-1} \Phi_{s_1} + \mathbf{I})^{-1} \Phi_{s_1}^T \Sigma_{s_1}^{-1} (\mathbf{x}_i - \mu_{s_1}), \end{aligned} \quad (18.44)$$

which is just the mean of this distribution.

We then generate the image in the second style as

$$\mathbf{x}_{s_2} = \mu_{s_2} + \Phi_{s_2} \hat{\mathbf{h}}, \quad (18.45)$$

which is the original generative equation with the noise term omitted.

18.5 Symmetric bilinear and multilinear models

As the name suggests, symmetric bilinear models treat both style and identity equivalently. Both are continuous variables, and the model is linear in each. To write these models in a compact way, it is necessary to introduce tensor product notation. In this notation (see appendix C.3), the generative equation for the subspace identity model (equation 18.6) is written as

$$\mathbf{x}_{ij} = \mu + \Phi \times_2 \mathbf{h}_i + \epsilon_{ij}, \quad (18.46)$$

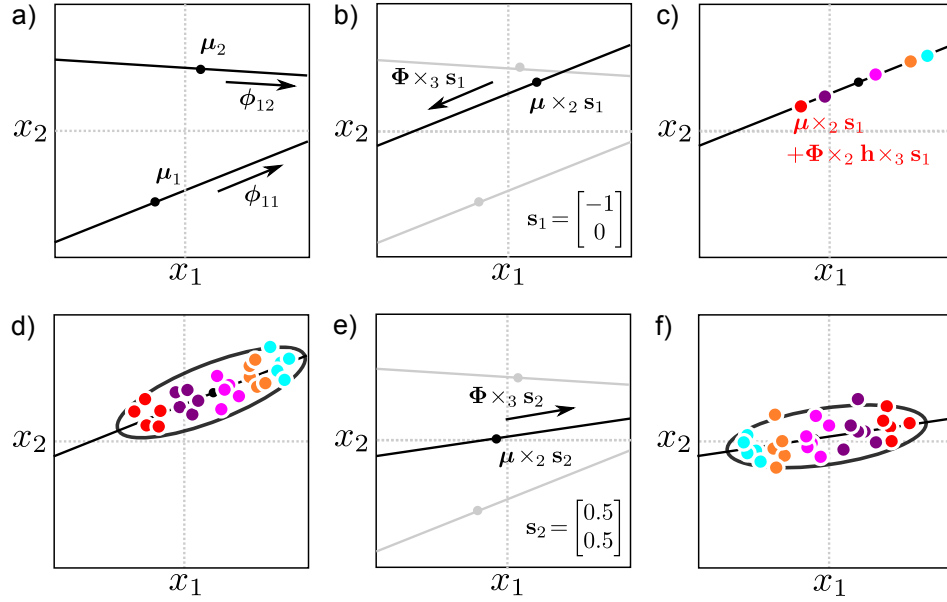


Figure 18.17 Ancestral sampling from symmetric bilinear model, with 1D identity and 2D style. a) In this model each style dimension consists of a subspace identity model with a 1D subspace. b) For a given style vector \mathbf{s}_1 , we weight these models to create a new subspace identity model. c) We then generate from this by weighting the factor by the hidden variable \mathbf{h} and d) adding noise to generate different instances of this identity in this style. e) A different weighting induced by the style vector \mathbf{s}_2 creates a different subspace identity model. f) Generation from the resulting subspace identity model.

where the notation $\Phi \times_2 \mathbf{h}_i$ means take the dot product of the second dimension of Φ with \mathbf{h}_i . Since Φ was originally a 2D matrix, this returns a vector.

In the symmetric bilinear model, the generative equation for the j^{th} example of the i^{th} identity in the k^{th} style is given by

$$\mathbf{x}_{ijk} = \boldsymbol{\mu} + \Phi \times_2 \mathbf{h}_i \times_3 \mathbf{s}_k + \boldsymbol{\epsilon}_{ijk}, \quad (18.47)$$

where \mathbf{h}_i is a 1D vector representing the identity, \mathbf{s}_k is a 1D vector representing the style and Φ is now a 3D tensor. In the expression $\Phi \times_2 \mathbf{h}_i \times_3 \mathbf{s}_k$ we take the dot product with two of these three dimensions, leaving a column vector as desired.

In probabilistic form, we write

$$\begin{aligned} Pr(\mathbf{h}_i) &= \text{Norm}_{\mathbf{h}_i}[\mathbf{0}, \mathbf{I}] \\ Pr(\mathbf{s}_k) &= \text{Norm}_{\mathbf{s}_k}[\mathbf{0}, \mathbf{I}] \\ Pr(\mathbf{x}_{ijk} | \mathbf{h}_i, \mathbf{s}_k) &= \text{Norm}_{\mathbf{x}_{ijk}}[\boldsymbol{\mu} + \Phi \times_2 \mathbf{h}_i \times_3 \mathbf{s}_k, \Sigma]. \end{aligned} \quad (18.48)$$

For a fixed style vector \mathbf{s}_k this model is exactly a subspace identity model with

hidden variable \mathbf{h}_i . The choice of style determines the factors by weighting a set of basis functions to create them. It is also possible to make the mean vector depend on the style by using the model

$$\mathbf{x}_{ijk} = \boldsymbol{\mu} \times_2 \mathbf{s}_k + \boldsymbol{\Phi} \times_2 \mathbf{h}_i \times_3 \mathbf{s}_k + \boldsymbol{\epsilon}_{ijk}, \quad (18.49)$$

where $\boldsymbol{\mu}$ is now a matrix with basis functions in the columns that are weighted by the style \mathbf{s} . Ancestral sampling from this model is illustrated in figure 18.17.

It is instructive to compare the asymmetric and symmetric bilinear models. In the asymmetric bilinear model, there were a discrete number of styles each of which generated data that individually looked like a subspace identity model, but the model induced a relationship between the position of an identity in one style cluster and in another. In the symmetric bilinear model, there is a continuous family of styles, that produces a continuous family of subspace identity models. Again the model induces a relationship between the position of an identity in each.

Up to this point, we have described the model as a subspace identity model for fixed style. The model is symmetric, and so it is possible to reverse the roles of the variables. For a fixed identity, the model looks like a subspace model where the basis functions are weighted by the variable \mathbf{s}_k . In other words, the model is linear in both sets of hidden variables when the other is fixed. It is *not*, however, simultaneously linear in both \mathbf{h} and \mathbf{s} together. These variables have a nonlinear interaction, and overall the model is nonlinear.

18.5.1 Learning

Unfortunately, is not possible to compute the likelihood of the bilinear model in closed form; we cannot simultaneously marginalize over both sets of hidden variables and compute

$$Pr(\mathbf{x}_{ijk}|\boldsymbol{\theta}) = \iint Pr(\mathbf{x}_{ijk}, \mathbf{h}_i, \mathbf{s}_k|\boldsymbol{\theta}) d\mathbf{h}_i d\mathbf{s}_k, \quad (18.50)$$

where $\boldsymbol{\theta} = \{\boldsymbol{\mu}, \boldsymbol{\Phi}, \boldsymbol{\Sigma}\}$ represents all of the unknown parameters. The usual approach for learning models with hidden variables is to use the EM algorithm, but this is no longer suitable because we cannot compute the joint posterior distribution over the hidden variables $Pr(\mathbf{h}_i, \mathbf{s}_k|\{\mathbf{x}_{ijk}\}_{j=1}^J)$ in closed form either.

For the special case of spherical additive noise $\boldsymbol{\Sigma} = \sigma^2 \mathbf{I}$, and complete data (where we see J examples of each of the I individuals in each of K styles), it is possible to solve for the parameters in closed form using a method similar to that used for PPCA (section 17.5.1). This technique relies on the *N-mode singular value decomposition*, which is the generalization of the SVD to higher dimensions.

For models with diagonal noise, we can approximate by maximizing over one of the hidden variables rather than marginalizing over them. For example, if we maximize over the style parameters so that



Figure 18.18 Translation of styles using symmetric bilinear model. a) We learn the model from a set of images, where the styles (rows) and identities (columns) are known. Then we are given a new image which has a previously unseen identity and style. b) The symmetric bilinear model can estimate the identity parameters and simulate the image in new styles, or c) estimate the style parameters and simulate new identities. In both cases, the simulated results are close to the ground truth. Adapted from Tenenbaum & Freeman (2000). ©2000 MIT Press.

$$\theta = \operatorname{argmax}_{\theta} \left[\sum_{k=1}^K \max_{\mathbf{s}_k} \left[\sum_{i=1}^I \sum_{j=1}^J \log \left[\int Pr(\mathbf{x}_{ijk}, \mathbf{h}_i, \mathbf{s}_k | \theta) d\mathbf{h}_i \right] \right] \right], \quad (18.51)$$

then the remaining model is linear in the hidden variables \mathbf{h}_i . It would hence be possible to apply an alternating approach in which we first fix the styles and learn the parameters with the EM algorithm and then fix the parameters and update the style parameters using optimization.

18.5.2 Inference

Various forms of inference are possible, including all of those discussed for the asymmetric bilinear model. We can, for example, make decisions about whether identities match by comparing different compound models. It is not possible to marginalize over both the identity and style variables in these models, and so we maximize over the style variable in a similar manner to the learning procedure. Similarly, we can translate from one style to another by estimating the identity

variable \mathbf{h} (and the current style variable \mathbf{s} if unknown) from the observed data. We then use the generative equation with a different style vector \mathbf{s} to simulate a new example in a different style.

The symmetric bilinear model has a continuous parameterization of style, and so it is also possible to perform a new translation task: given an example whose identity we have not previously seen *and* whose style we have not previously seen, we can translate either its style or identity as required. We first compute the current identity and style which can be done using a nonlinear optimization approach,

$$\hat{\mathbf{h}}, \hat{\mathbf{s}} = \operatorname{argmax}_{\mathbf{h}, \mathbf{s}} [Pr(\mathbf{x}|\boldsymbol{\theta}, \mathbf{h}, \mathbf{s})Pr(\mathbf{h})Pr(\mathbf{s})]. \quad (18.52)$$

Then we simulate new examples using the generative equation, modifying the style \mathbf{s} or identity \mathbf{h} as required. An example of this is shown in figure 18.18.

18.5.3 Multi-linear models

The symmetric bilinear model can be extended to create *multi-linear* or *multi-factor* models. For example, we might describe our data as depending on three hidden variables, \mathbf{h}, \mathbf{s} and \mathbf{t} , so the generative equation becomes

$$\mathbf{x}_{ijkl} = \boldsymbol{\mu} + \boldsymbol{\Phi} \times_2 \mathbf{h}_i \times_3 \mathbf{s}_k \times_4 \mathbf{t}_l + \boldsymbol{\epsilon}_{ijkl}, \quad (18.53)$$

and now the tensor $\boldsymbol{\Phi}$ becomes four-dimensional. As in the symmetric bilinear model, it is not possible to marginalize over all of the hidden variables in closed form, and this constrains the possible methods for learning and inference.

18.6 Applications

We have illustrated many of the models in this chapter with examples from face recognition. In this section, we will describe face recognition in more detail and talk about some of the practicalities of building a recognition system. Subsequently, we will discuss an application in which a visual texture is compactly represented as a multi-linear model. Finally, we will describe a nonlinear version of the multilinear model that can be used to synthesize animation data.

18.6.1 Face recognition

To provide a more concrete idea of how well these algorithms work in practice, we will discuss a recent application in detail. Li *et al.* (2011) present a recognition system based on probabilistic linear discriminant analysis.

Eight keypoints on each face were identified, and the images were registered using a piecewise affine warp. The final image size was 400×400 . Feature vectors were extracted from the area of the image around each keypoint. The feature

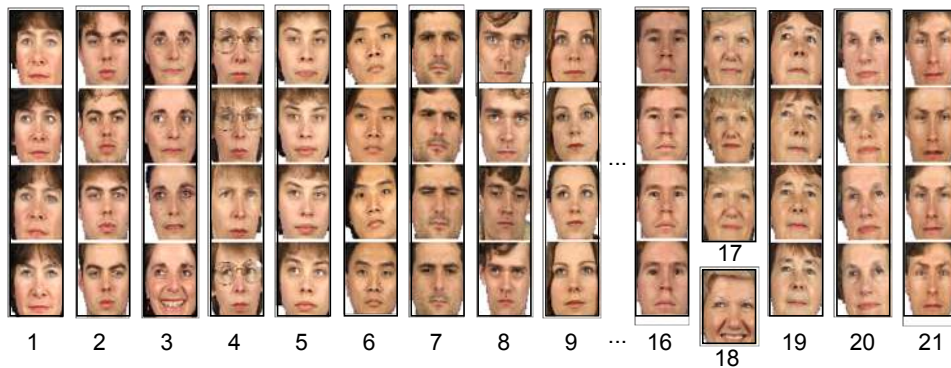


Figure 18.19 Face clustering results from Li *et al.* (2011). The algorithm was presented with a set of 80 faces consisting of with 4 pictures each of 20 people and clustered these almost perfectly; it correctly found 19 of these 20 groups of images but erroneously divided the data from one individual into two separate clusters. The algorithm works well for these frontal faces despite changes in expression (e.g., cluster 3), changes in hairstyle (e.g., cluster 9) and the addition or removal of glasses (e.g., cluster 4). Adapted from Li *et al.* (2011). ©2011 IEEE.

vectors consisted of image gradients at eight orientations and three scales at points in a 6×6 grid centered on the keypoint. A separate recognition model was built for each keypoint and these were treated as independent in the final recognition decision.

The system was trained using only the first 195 individuals from the XM2VTS database and signal and noise subspaces of size 64. In testing, the algorithm was presented with 80 images taken from the remaining 100 individuals in the database and was required to cluster them into groups according to identity. There may be 80 images of the same person or 80 images of different people or any permutation between these extremes.

In principle it is possible to calculate the likelihood for each possible clustering of the data. Unfortunately, in practice there are far too many possible configurations. Hence, Li *et al.* (2011) adopted a greedy agglomerative strategy. They started with the hypothesis that there are 80 different individuals. They considered merging all pairs of individuals and chose the combination that increased the likelihood the most. They repeated this process until the likelihood could not be improved. Example clustering results are illustrated in figure 18.19 and are typical of state-of-the-art recognition algorithms; they cope relatively easily with frontal faces under controlled lighting conditions.

However, for more natural images the same algorithms struggle even with more sophisticated preprocessing. For example, Li *et al.* (2011) applied the PLDA model to face verification in the ‘Labeled Faces in the Wild’ dataset (Huang *et al.* 2007b), which contains images of famous people collected from the internet, and obtained an equal-error rate of approximately 10%. This is typical of the state of the art

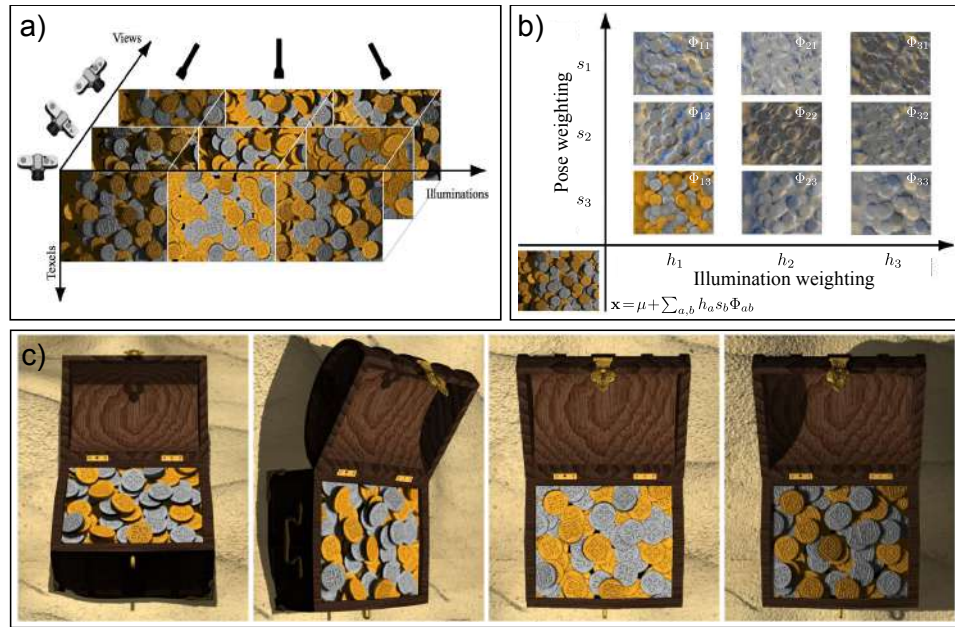


Figure 18.20 Tensor textures. a) The training set consists of renderings of a set of coins viewed from several different directions and with several different lighting directions. b) A new texture (bottom left corner) is computed as a weighted sum of the learned basis functions stored in the 3D tensor Φ . c) Several frames of a video sequence in which the texture is synthesized appropriately from the model. Adapted from Vasilescu & Terzopoulos (2004). ©2003 ACM.

at the time of writing and is much worse than for faces captured under controlled conditions.

18.6.2 Modeling texture

The interaction of light with a surface can be described by the bi-directional reflectance distribution function; essentially, this describes the outgoing light at each angle from the surface given incoming light at a particular angle to the surface. The bi-directional texture function (BTF) generalizes this model to also depend on the 2D position on the surface of the object. If we know the BTF, then we know how a textured surface will appear from every angle and under every lighting combination.

This function could be approximated by taking several thousand images of the surface viewed from different angles and under different lighting conditions. However, the resulting data are clearly highly redundant. Vasilescu & Terzopoulos (2004) described the BTF using a multi-linear model known as ‘TensorTextures’.

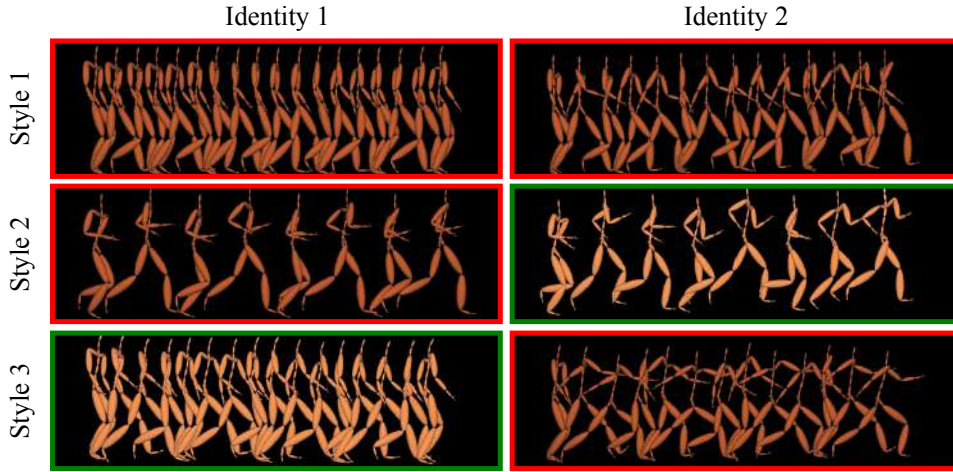


Figure 18.21 Multi-factor GPLVM applied to animation synthesis. A three-factor model was learned with factors for identity, style, and position in the gait sequence. The figures shows training data (red boxes) and synthesized data from the learned model (green boxes). In each case, it manages to simulate the style and identity well. Adapted from Wang *et al.* (2007).

It contained style factors that represent the lighting and viewing directions (figure 18.20a-b). Although both of these quantities are naturally 2D, they represented them as vectors of size 21 and 37, respectively, where each training example was transformed to one coordinate axis.

Figure 18.20c shows images generated from the TensorTextures model; the hidden variables associated in each style were chosen by linearly interpolating between the hidden variables of nearby training examples, and the image was synthesized without the addition of noise. It can be seen that the TensorTextures model has learned a compact representation of the appearance variation under changes in viewpoint and illumination, including complex effects due to self-occlusion, inter-reflection, and self-shadowing.

18.6.3 Animation synthesis

Wang *et al.* (2007) developed a multi-factor model that depended nonlinearly on the identity and style components. Their approach was based on the Gaussian process latent variable model; the style and identity factors were transformed through nonlinear functions before weighting the tensor Φ . In this type of model, the likelihood might be given by:

$$Pr(\mathbf{x}_{ijk} | \Sigma, \mathbf{h}_i, \mathbf{s}_k) = \iint \text{Norm}_{\mathbf{x}_{ijk}} [\boldsymbol{\mu} + \Phi \times_2 \mathbf{f}[\mathbf{h}_i] \times_3 \mathbf{g}[\mathbf{s}_k], \Sigma] d\Phi d\boldsymbol{\mu}, \quad (18.54)$$

where $\mathbf{f}[\bullet]$ and $\mathbf{g}[\bullet]$ are nonlinear functions that transform the identity and style parameters respectively. In practice, the tensor Φ can be marginalized out of the final likelihood computation along with the overall mean μ . This model can be expressed in terms of inner products of the identity and style parameters and can hence be kernelized. It is known as the *multi-factor Gaussian process latent variable model* or the *multi-factor GPLVM*.

Wang *et al.* (2007) used this model to describe human locomotion. A single pose was described as an 89-dimensional vector, which consisted of 43 joint angles, the corresponding 43 angular velocities and the global translational velocity. They built a model consisting of three factors; the identity of the individual, the gait of locomotion (walk, stride, or run), and the current state in the motion sequence. Each was represented as a 3D vector. They learned the model from human capture data using an RBF kernel.

Figure 18.21 shows the results of style translation in this model. The system can predict realistic body poses in styles that have not been observed in the training data. Since the system is generative, it can also be used to synthesize novel motion sequences for a given individual in which the gait varies over time.

Discussion

In this chapter, we have examined a number of models that describe image data as a function of style and content variables. During training, these variables are forced to take the same value for examples where we know the style or content are the same. We have demonstrated a number of different forms of inference including identity recognition and style translation.

Notes

Face recognition: For a readable introduction to face recognition consult Chellappa *et al.* (2010). For more details, consult the review paper by Zhao *et al.* (2003) or the edited book by Li & Jain (2005).

Subspace methods for face recognition: Turk & Pentland (2001) developed the *eigenfaces* method in which the pixel data were reduced in dimension by linearly projecting it onto a subspace corresponding to the principal components of the training data. The decision about whether two faces matched or not was based on the distance between these low dimensional representations. This approach quickly supplanted earlier techniques that had been based on measuring the relative distance between facial features (Brunelli & Poggio 1993).

The subsequent history of face recognition has been dominated by other *subspace methods*. Researchers have variously investigated the choice of basis functions (e.g., Bartlett *et al.* 1998; Belhumeur *et al.* 1997; He *et al.* 2005; Cai *et al.* 2007), analogous nonlinear techniques (Yang 2002), and the choice of distance metric (Perlibakas 2004). The relationship between different subspace models is discussed in (Wang & Tang 2004b). A review of subspace methods (without particular reference to face recognition) can be found in De La Torre (2011).

Linear discriminant analysis: A notable sub-category of these subspace methods consists of approaches based on linear discriminant analysis (LDA). The Fisherfaces algorithm (Belhumeur *et al.* 1997) projected face data to a space where the ratio of between-individual variation to within-individual variation was maximized. Fisherfaces is limited to directions in which at least some within-individual variance has been observed (the small-sample problem). The null-space LDA approach (Chen *et al.* 2000) exploited the signal in the remaining subspace. The Dual-Space LDA approach (Wang & Tang 2004a) combined these two sources of information.

Probabilistic approaches: The identity models in this chapter are probabilistic re-interpretations of earlier non-probabilistic techniques. For example, the subspace identity model is very similar to the eigenfaces algorithm (Turk & Pentland 2001) and probabilistic LDA is very similar to the Fisherfaces algorithm (Belhumeur *et al.* 1997). For more details about these probabilistic versions, consult Li *et al.* (2011) and Ioffe (2006) who presented a slightly different probabilistic LDA algorithm. There have also been many other probabilistic approaches to face recognition (Liu & Wechsler 1998; Moghaddam *et al.* 2000; Wang & Tang 2003; Zhou & Chellappa 2004).

Alignment and pose changes: An important part of most face recognition pipelines is to accurately identify facial features so that either (i) the face image can be aligned to a fixed template or (ii) the separate parts of the face can be treated independently (Wiskott *et al.* 1997; Moghaddam & Pentland 1997). Common methods to identify facial features include the use of active shape models (Edwards *et al.* 1998) or pictorial structures (Everingham *et al.* 2006; Li *et al.* 2010).

For larger pose changes, it may not be possible to warp the face accurately to a common template and explicit methods are required to compare the faces. These include fitting 3D morphable models to the images and then simulating a frontal image from a non-frontal one (Blaiz *et al.* 2005), predicting the face at one pose from another using statistical methods (Gross *et al.* 2002; Lucey & Chen 2006) or using the tied factor analysis model discussed in this chapter (Prince *et al.* 2008). A review of face recognition across large pose changes can be found in Zhang & Gao (2009).

Current work in face recognition: It is now considered that face recognition for frontal faces in constant lighting and with no pose or expression changes is almost solved. Earlier databases that have these characteristics (e.g., Messer *et al.* 1999; Phillips *et al.* 2000) have now been supplanted by test databases containing more variation (Huang *et al.* 2007b).

Several recent trends have emerged in face recognition. These include a resurgence of interest in discriminative models (e.g., Wolf *et al.* 2009; Taigman *et al.* 2009; Kumar *et al.* 2009), learning metrics to discriminate identity (e.g., Nowak & Jurie 2007; Ferencz *et al.* 2008; Guillaumin *et al.* 2009; Nguyen & Bai 2010), the use of sparse representations (e.g., Wright *et al.* 2009), and a strong interest in preprocessing techniques. In particular, many current methods are based on Gabor features (e.g., Wang & Tang 2003), local binary patterns (Ojala *et al.* 2002; Ahonen *et al.* 2004), three-patch local binary patterns (Wolf *et al.* 2009), or SIFT features (Lowe 2004). Some of the most successful methods combine or select several different preprocessing techniques (Li *et al.* 2011; Taigman *et al.* 2009; Pinto & Cox 2011).

Bilinear and multi-linear models: Bilinear models were introduced to computer vision by Tenenbaum & Freeman (2000) and multi-linear models were explored by Vasilescu & Terzopoulos (2002). Kernelized multi-linear models were discussed by Li *et al.* (2005) and Wang *et al.* (2007). An alternative approach to nonlinear multi-factor models was presented in Elgammal & Lee (2004). The most common use of bilinear and multi-linear models in computer vision has been for face recognition in situations where the capture conditions vary (Grimes *et al.* 2003; Lee *et al.* 2005; Cuzzolin 2006; Prince *et al.* 2008).

Problems

Problem 18.1 Prove that the posterior distribution over the hidden variable in the subspace identity model is as given in equation 18.9.

Problem 18.2 Show that the M-step updates for the subspace identity model are as given in equation 18.11.

Problem 18.3 Develop a closed form solution for learning the parameters $\{\mu, \Phi, \sigma^2\}$ of a subspace identity model where the noise is spherical:

$$Pr(\mathbf{x}_{ij}) = \text{Norm}_{\mathbf{x}_{ij}}[\mu, \Phi\Phi^T + \sigma^2\mathbf{I}].$$

Hint: Assume you have exactly $J = 2$ examples of each of the I training images and base your solution on probabilistic PCA.

Problem 18.4 In a face clustering problem, how many possible models of the data are there with 2,3,4,10, and 100 faces?

Problem 18.5 An alternative approach to face verification using the identity subspace model is to compute the probability of the observed data \mathbf{x}_1 and \mathbf{x}_2 under the models:

$$\begin{aligned} Pr(\mathbf{x}_1, \mathbf{x}_2 | w = 0) &= Pr(\mathbf{x}_1)Pr(\mathbf{x}_2) \\ Pr(\mathbf{x}_1, \mathbf{x}_2 | w = 1) &= Pr(\mathbf{x}_1)Pr(\mathbf{x}_2 | \mathbf{x}_1). \end{aligned}$$

Write down expressions for the marginal probability terms $Pr(\mathbf{x}_1)$, $Pr(\mathbf{x}_2)$ and the conditional probability $Pr(\mathbf{x}_2 | \mathbf{x}_1)$. How could you use these expressions to compute the posterior $Pr(w | \mathbf{x}_1, \mathbf{x}_2)$ over the world state?

Problem 18.6 Propose a version of the subspace identity model that is robust to outliers in the training data.

Problem 18.7 Moghaddam *et al.* (2000) took a different probabilistic approach to face verification. They took the difference $\mathbf{x}_\Delta = \mathbf{x}_2 - \mathbf{x}_1$ and modeled the likelihoods of this vector $Pr(\mathbf{x}_\Delta|w = 0)$ and $Pr(\mathbf{x}_\Delta|w = 1)$ when the two faces match or don't. Propose expressions for these likelihoods and discuss learning and inference in this model. Identify one possible disadvantage of this model.

Problem 18.8 Develop a model that combines the advantages of PLDA and the asymmetric bilinear model; it should be able to model the within-individual covariance with a subspace, but also be able to compare data between disparate styles. Discuss learning and inference in your model.

Problem 18.9 In the asymmetric bilinear model, how would you infer whether the style of two examples is the same or not, regardless of whether the images matched?

