Abstract—Despite the apparent spatio-temporal decomposition given by (Probabilistic) Principal Component Analysis ((P)PCA), there is in fact no temporal coupling built into these models. Here we augment PPCA with a temporal model in the latent space by coupling the latent variables in time with an autoregressive model and show that the new model may be viewed as a generalisation of PPCA. We present an algorithm which utilises both expectation maximisation and a forward-backward algorithm to infer the values of the model parameters and demonstrate that it is able to make good estimates of the parameter values for synthetic data. We show that the additional temporal information is advantageous when imputing values for missing observations when compared with two non-temporal PPCA methods, both against synthetic data and real UK industrial production output data.

I. INTRODUCTION

Principal Component Analysis (PCA, e.g., [1]) provides a sum of products decomposition of data, in which a multivariate observation is represented as a weighted sum of orthonormal principal components. Thus if \( t_n \) is the \( n \)th observation, a \( d \)-dimensional vector, the PCA decomposition is

\[
t_n = \sum_k x_{k,n} w_k
\]

where the \( w_k \) are the orthonormal principal components and \( x_{k,n} \) are corresponding weights. Frequently the observations are temporally ordered, so that PCA yields a decomposition into spatial variables, the principal components \( w_k \), and associated temporal variables, \( x_{k,n} \). The temporal variables may be regarded as sources which are mixed by the principal components to form the observations.

Although the classical formulation of PCA does not explicitly model noise on the observations, Tipping and Bishop [2] have formulated a probabilistic version of PCA (PPCA) in which the \( x_{k,n} \) are regarded as latent variables:

\[
t_n = Wx_n + e_n
\]

where \( W \) is a matrix of (orthogonal, but not necessarily orthonormal) components that mix Gaussian distributed latent variables, \( x_{k,n} \sim \mathcal{N}(0,1) \) for each \( n \), to form observations which are contaminated by Gaussian noise with covariance \( \sigma^2 I \). An Expectation-Maximisation (EM) algorithm to learn the latent variables is available, but Tipping and Bishop show that the first \( q \) columns of \( W \) span the same space as the first \( q \) conventional principal components.

Despite the apparent spatio-temporal decomposition given by (Probabilistic) Principal Component Analysis ((P)PCA) we note that there is in fact no temporal coupling built into these models: shuffling the order of observations makes no difference to the decomposition other than shuffling the order of the latent variables. Here we augment PPCA with a temporal model in the latent space. Specifically we couple the latent variables in time with an autoregressive (AR) model with coefficients \( \theta_{k,i} \):

\[
x_{k,n} = \sum_{i=1}^{p} \theta_{k,i} x_{k,n-i} + \epsilon_{k,n}
\]

where the innovations sequence is Gaussian distributed, \( \epsilon_{k,n} \sim \mathcal{N}(0, \phi^2) \). The Gaussian innovations sequence means that, like the PPCA model, each source itself is Gaussian distributed, so that our model may be viewed as a generalisation of (P)PCA.

We present an EM algorithm for inferring the parameters of the PPCA-AR model, whose Markov properties allow an efficient Baum-Welch-like algorithm to be derived. The additional temporal structure of the PPCA-AR model means temporal information may be advantageously used to impute values for missing observations, improving PCA-based methods such as [3] and we demonstrate its efficacy on industrial production output data from the UK Office for National Statistics.

In this paper we start, in section I-A, by reviewing PPCA. In section II we construct the PPCA-AR model as a linear dynamical system and describe an EM algorithm for determining estimates of the values of the model’s latent variables and parameters. Results are demonstrated for synthetic data and compared with PPCA. In section III we show how PPCA and PPCA-AR may be used to predict or reconstruct missing values and demonstrate this process on synthetic data. Finally, in section IV, we compare PPCA-AR with PPCA by applying them to both complete and artificially incomplete real data.

A. PPCA

We start by reviewing PPCA, as formulated by Tipping and Bishop [2]. Each of the \( N \) observations is represented by a single, \( d \)-dimensional vector, \( t_n \), which are assumed to have zero mean. Tipping and Bishop construct PPCA as a latent variable model (of order \( q \)) of the form (2) where \( W \) is the \( d \times q \) component mixture matrix, \( x_n \) is the \( q \)-dimensional vector of latent variables, \( e_n \) is the vector of
Gaussian observation noise whose elements are distributed as \(\mathcal{N}(0, \sigma^2 I)\), and \(I\) is the identity matrix. The scale of \(W\) or \(x_n\) must be fixed, so \(x_n\) is defined to be Gaussian with covariance \(I\); \(x_n \sim \mathcal{N}(0, I)\). This formulation leads to the Gaussian probabilities

\[
p(t_n | x_n) = \mathcal{N}(t_n | Wx_n, \sigma^2 I) \tag{4}
\]

\[
p(t_n) = \mathcal{N}(t_n | 0, WW^T + \sigma^2 I) \tag{5}
\]

Tipping and Bishop derive an EM algorithm for PPCA, as follows:

**E step**

\[
M = W^T W + \sigma^2 I \tag{6}
\]

\[
\langle x_n | t_n \rangle = M^{-1} W^T t_n \tag{7}
\]

\[
\langle x_n x_n^T | t_n \rangle = \sigma^2 M^{-1} + \langle x_n | t_n \rangle \langle x_n | t_n \rangle^T \tag{8}
\]

**M step**

\[
W_{mm} = \left[ \sum_{n=1}^{N} t_n \langle x_n | t_n \rangle^T \right] \left[ \sum_{n=1}^{N} \langle x_n x_n^T | t_n \rangle \right]^{-1} \tag{9}
\]

\[
\sigma_{mm}^2 = \frac{1}{Nd} \sum_{n=1}^{N} ||t_n||^2 - 2\langle x_n | t_n \rangle^T W_{mm} t_n
+ \text{Tr}(W_{mm} \langle x_n x_n^T | t_n \rangle W_{mm}^T) \tag{10}
\]

where \(\text{Tr}\) denotes the trace, \(\langle x_n | t_n \rangle\) is the expected value and \(\langle x_n x_n^T | t_n \rangle\) the covariance of \(x_n\) conditioned on \(t_n\).

Although the latent variables and parameters may be learned using an EM algorithm, [2] also usefully provides an algebraic solution. If \(S\) is the sample covariance matrix of all \(N\) observations, i.e.

\[
S = \frac{1}{N} \sum_{n=1}^{N} t_n t_n^T \tag{11}
\]

then they show that the best fit (in the mean squared error sense) for a given model order is achieved when \(W\) is composed of the \(q\) principal eigenvectors of \(S\), i.e. those with the largest associated eigenvalues. If \(U_q\) is the matrix composed of the \(q\) principal eigenvectors (the usual PCA principal components), by convention orthonormal, and \(\Lambda_q\) is the diagonal matrix containing their associated eigenvalues, \(\lambda_1, \ldots, \lambda_q\), then

\[
\sigma_{mm}^2 = \frac{1}{d - q} \sum_{i=q+1}^{d} \lambda_i \tag{12}
\]

\[
W_{mm} = U_q (\Lambda_q - \sigma_{mm}^2 I)^{\frac{1}{2}} \tag{13}
\]

is the maximum likelihood solution, where \(\sigma^2\) captures the amount of variance lost when the original data is projected onto \(U_q\). The optimal reconstruction of the data, \(t_n\), of observation \(t_n\) is given by

\[
\hat{t}_n = W (W^T W)^{-1} (W^T W + \sigma^2 I) \langle x_n | t_n \rangle. \tag{14}
\]

For our new model, described in section II, we impose structure on the latent variables that removes the \(\mathcal{N}(0, I)\) constraint on \(x_n\), so we need a way of controlling the scale ambiguity between \(W\) and \(x_n\). It is easily seen from (13) that \(W\) is orthogonal but not orthonormal as \(W^T W = \Lambda_q - \sigma^2 I\), which is a diagonal matrix, but not likely to be the identity. We may reorder the model to make it orthonormal. Let \(D\) be the diagonal matrix \((\Lambda_q - \sigma^2 I)^{\frac{1}{2}}\); then, from (14)

\[
\hat{t}_n = U_q (D + D^{-1} \sigma^2 I) \langle x_n | t_n \rangle \tag{15}
\]

We may define \(W = U_q\) (now orthonormal) and combine the diagonal matrix \(D + D^{-1} \sigma^2 I\) with the \(\langle x_n | t_n \rangle\) term, giving a model where the scale of \(W\) is fixed and scale information resides in the \(x_n\).

**II. PPCA WITH AUTOREGRESSIVE MODEL**

Standard (P)PCA assumes that there is no meaningful ordering to the observations. Randomly reordering them makes no difference to the model, other than reordering the latent variables. However, in many cases there is a temporal structure to the observations and we can augment the model to take it into account. If we think of PPCA as a mixture of sources, then we are assuming that the sources, i.e. the latent variables, are temporally ordered. This new model, PPCA-AR, models each latent variable dimension independently with an autoregressive (AR) model. If \(X\) is the matrix of ordered latent variables, \((x_1, \ldots, x_N)\), then we apply independent AR models to each row of \(X\). Note that we must remove the \(\mathcal{N}(0, I)\) constraint on \(x_n\).

Let us consider first the AR model for a single row of \(X\), \(x\). AR assumes that the \(N\) individual observations in \(x\) are related by a model (of order \(p\)) of the form

\[
x_n = \sum_{i=1}^{p} \theta_i x_{n-i} + \epsilon_n \tag{16}
\]

where \(\theta = (\theta_1, \ldots, \theta_p)^T\) is a vector of scalar parameters that are specific to this row \(x\), \(\epsilon_n\) is the excitation noise associated with this \(x_n\) and \(n > p\). Each \(\epsilon_n\) is assumed to be drawn from an independent Gaussian distribution \(\mathcal{N}(0, \phi^2)\). By defining \(\hat{x}_n\) and \(\hat{\epsilon}_n\) as the lag vectors

\[
\hat{x}_n = \begin{pmatrix} x_{n-p+1} \\ x_{n-p+2} \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix} \quad \text{and} \quad \hat{\epsilon}_n = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ \epsilon_n \end{pmatrix} \tag{17}
\]

we can represent this autoregressive model as

\[
\hat{x}_n = \Theta \hat{x}_{n-1} + \hat{\epsilon}_n \tag{18}
\]

so that

\[
p(\hat{x}_n | \hat{x}_{n-1}) = \mathcal{N}(\hat{x}_n | \Theta \hat{x}_{n-1}, \Phi) \tag{19}
\]
with $\Theta$ the $p \times p$ matrix

$$
\Theta = \begin{pmatrix}
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
\theta_p & \theta_{p-1} & \theta_2 & \theta_1 \\
\end{pmatrix}
$$
(20)

and $\Phi$ the $p \times p$ matrix with $\Phi_{pp} = \phi^2$ and zeros elsewhere. We expect the first $p - 1$ elements on the main diagonal of $\Phi$ to be zero because the first $p$ elements of $x_n$ are exactly determined by $\Theta \tilde{x}_{n-1}$ and hence they have no variance.

Depending on the nature of the data, the rows of $X$ may best be represented by AR models of different orders, so we choose not to represent them all together in a single, multi-dimensional AR model. Instead, by stacking the $\tilde{x}_n$ and $\tilde{e}_n$ for each row of $X$, and constructing $\Theta$ and $\Phi$ as block-diagonal matrices of each row’s individual $\Theta$ and $\Phi$ matrices respectively, we may write the AR equations for all rows of $X$ in the form of equation (18).

A. PPCA-AR as a Linear Dynamical System

Combining (18) with a straightforward adjustment to the PPCA latent variable equation in (2), we arrive at two equations that define the PPCA-AR model as a linear dynamical system:

$$
\tilde{x}_n = \Theta \tilde{x}_{n-1} + \tilde{e}_n \\
t_n = \tilde{W} \tilde{x}_n + e_n
$$
(21) (22)

where $\tilde{W} = WQ$ and $Q$ is the matrix of ones and zeros that extracts $x_n$ from $\tilde{x}_n$. From (19) and an appropriately adjusted (4) we get the Gaussian probabilities

$$
p(\tilde{x}_n|\tilde{x}_{n-1}) = N(\tilde{x}_n|\Theta \tilde{x}_{n-1}, \Phi) \\
p(t_n|x_n) = N(t_n|\tilde{W} \tilde{x}_n, \sigma^2 I)
$$
(23) (24)

which makes clear the recursive dependency between the probabilities for each $\tilde{x}_n$ and the similarity with the Kalman filter, which is also a linear Gaussian state space model.

The initial latent variable is assumed to be Gaussian distributed:

$$
p(\tilde{x}_1) = N(\tilde{x}_1|m_0, V_0)
$$
(25)

Ignoring the problem of how to determine the best values for the model orders $p$ and $q$, the set of all the parameters of this model, $\Omega = \{\Theta, \Phi, W, \sigma^2, m_0, V_0\}$, may be estimated by maximum likelihood using an EM algorithm [4], as described in the following sections.

1) E step: Using the forward-backward algorithm [5] (for a tutorial see [6]), the E step is calculated in two parts: after the forward sweep through the latent variables we arrive at, for each $n$, $p(\tilde{x}_n|t_1, ..., t_n) = N(\tilde{x}_n|m_n, V_n)$, where

$$
P_0 = V_0 \\
m_1 = m_0 + K_1(t_1 - \tilde{W}m_0)
$$
(26) (27)

and

$$
P_{n-1} = \Theta V_{n-1} \Theta^T + \Phi \\
K_n = P_{n-1}(W P_{n-1} \tilde{W}^T + \sigma^2 I)^{-1} \\
m_n = \Theta m_{n-1} + K_n(t_n - \tilde{W} \Theta m_{n-1}) \\
V_n = (I - K_n \tilde{W}) P_{n-1}.
$$
(28) (29) (30) (31)

If $T$ is the matrix, $(t_1, ..., t_N)$, of ordered observations, then after both the forward and backward sweeps have been completed we obtain the posterior distribution for each latent variable $p(\tilde{x}_n|T) = N(\tilde{x}_n|m_n, V_n)$ where

$$
m_N = m_N \\
V_N = V_N
$$
(32) (33)

and

$$
J_n = V_n \Theta^T P_{n-1}^{-1} \\
m_n = m_n + J_n(m_{n+1} - \Theta m_n) \\
V_n = V_n + J_n(V_{n+1} - P_n) J_n^T
$$
(34) (35) (36)

For the maximisation step the following expectations are required:

$$
\langle \tilde{x}_n|T \rangle = m_n \\
\langle \tilde{x}_n \tilde{x}_m^T|T \rangle = J_{n-m} V_{n-m} m_{n-m} \\
\langle \tilde{x}_n^T|T \rangle = V_n + m_n m_n^T
$$
(37) (38) (39)

2) M step: From (25), (23) and (24) the complete data log likelihood to be maximised is

$$
p(T, X|\Omega) = -\frac{1}{2} \log |V_0| - \frac{1}{2} (\tilde{x}_1 - m_0)^T V_0^{-1} (\tilde{x}_1 - m_0) \\
- \frac{1}{2} N \log |\Phi| - \frac{1}{2} N \log |\sigma^2 I|
$$

$$
- \frac{1}{2} \sum_{n=2}^N (\tilde{x}_n - \Theta \tilde{x}_{n-1})^T \Phi^{-1} (\tilde{x}_n - \Theta \tilde{x}_{n-1}) \\
- \frac{1}{2} \sum_{n=1}^N (t_n - \tilde{W} \tilde{x}_n)^T (\sigma^2 I)^{-1} (t_n - \tilde{W} \tilde{x}_n).
$$
(40)

By differentiating this with respect to each of the model parameters in $\Omega$ and setting the results to zero, we obtain update expressions as follows (the expectation dependencies on $T$ have been omitted to aid readability):

$$
(m_0)_{\text{new}} = \langle \tilde{x}_1 \rangle
$$
(41)

$$
(V_0)_{\text{new}} = \langle \tilde{x}_1 \tilde{x}_1^T \rangle + \langle \tilde{x}_1 \rangle \langle \tilde{x}_1 \rangle^T
$$
(42)

$$
\Theta_{\text{new}} = \left[ \sum_{n=2}^N \langle \tilde{x}_n \tilde{x}_n^T \rangle \right]^{-1} \left[ \sum_{n=2}^N \langle \tilde{x}_n \tilde{x}_{n-1}^T \rangle \right]
$$
(43)

$$
\Phi_{\text{new}} = \frac{1}{N-1} \sum_{n=2}^N [\langle \tilde{x}_n \tilde{x}_n^T \rangle - \Theta_{\text{new}} \langle \tilde{x}_{n-1} \tilde{x}_{n-1}^T \rangle] \\
- \langle \tilde{x}_n \tilde{x}_{n-1}^T \rangle \Theta_{\text{new}}^T - \Theta_{\text{new}} \langle \tilde{x}_{n-1} \tilde{x}_{n-1}^T \rangle \Theta_{\text{new}}^T
$$
(44)

$$
W_{\text{new}} = \left[ \sum_{n=1}^N t_n \langle \tilde{x}_n \rangle^T Q \right] \left[ \sum_{n=1}^N Q \langle \tilde{x}_n \tilde{x}_n^T \rangle Q \right]^{-1}
$$
(45)
\[ \sigma^2_{\text{new}} = \frac{1}{Nd} \sum_{n=1}^{N} \text{Tr} \left[ t_n t_n^T - \tilde{W}_{\text{new}} \langle \tilde{x}_n \rangle t_n^T \right. \\
\left. - \langle \tilde{x}_n \rangle \tilde{W}_{\text{new}}^T (\tilde{W}_{\text{new}}^{\top} + \tilde{W}_{\text{new}} \langle \tilde{x}_n \rangle \tilde{W}_{\text{new}}^{\top}) \right] \]  

(46)

Notice that, as may be expected, the expressions for the expectations in (37)-(39) differ considerably from those for PPCA in (6)-(8), but the definitions for \( W_{\text{new}} \) and \( \sigma^2_{\text{new}} \) ((45)-(46) and (9)-(10)) show a close similarity.

**B. Enforcing constraints**

The update expressions for \( \Theta \) and \( \Phi \) in (43) and (44) do not enforce the required restrictions on the structure of these matrices (see (20) and after). Here, by considering each row of \( X \) separately, we show how these constraints may be respected in the solution. In this section, instead of introducing a new subscript, the matrices and vectors are those that pertain only to a single row.

The matrix \( \Theta \) may be written as a function of vector \( \theta \) using the Kronecker product:

\[ \Theta = Z(\theta \otimes I_p) + Y \]  

(47)

where \( Z \) and \( Y \) are matrices containing only ones and zeros and \( I_p \) is the \( p \times p \) identity matrix. This allows the log likelihood (40) to be rewritten as a function of \( \theta \) rather than \( \Theta \) and the differentiation to be performed accordingly. The result is

\[ \theta_{\text{new}} = 2(FF^T)^{-1} \left[ \sum_{n=2}^{N} (\tilde{x}_n - \tilde{x}_{n-1} Y^T) \Phi^{-1} Z(I_p \otimes \tilde{x}_{n-1}) \right]^T \]  

(48)

where \( F = \sum_{n=2}^{N} (I_p \otimes \tilde{x}_{n-1}) Y \Phi^{-1} Z(I_p \otimes \tilde{x}_{n-1}) \). \( \Theta \) may now be constructed from (47).

For the excitation noise variance, the log likelihood for one row of \( X \) may be rewritten as a function of \( \phi^2 \) rather than \( \Phi \); differentiation with respect to \( \phi^2 \) yields

\[ \phi^2_{\text{new}} = \frac{1}{N-1} \sum_{n=2}^{N} \left( \langle x_n^2 \rangle - 2 \langle x_n \tilde{x}_{n-1} \rangle \theta \right. \]
\[ \left. + \theta^T \langle \tilde{x}_{n-1} \tilde{x}_{n-1} \rangle \theta \right) \]  

(49)

which may then be inserted into \( \Phi \).

In this combined model we have removed the \( \mathcal{N}(0,1) \) prior on each element of \( X \). As remarked above, this prior constrains the size of the latent variables and thus fixes the scale of \( W \). To resolve the scale ambiguity between \( W \) and \( X \) we restrict \( W \) to be orthonormal, i.e. we constrain \( W \) so that \( W^T W = I \). Using Lagrange multipliers in the maximisation of the log likelihood with respect to \( W \) we obtain a new update statement for \( W \):

\[ W_{\text{new}} = B (B^T B)^{-\frac{1}{2}} \]  

(50)

where \( B = \sum_{n=1}^{N} t_n \langle \tilde{x}_n \rangle^T Q^T \). Note that this does not take us away from a generalisation of standard PPCA as we have previously noted that PPCA can be reformulated with an orthonormal \( W \); see (15).

**C. Illustration: synthetic data**

In order to demonstrate the performance of PPCA-AR, we construct a number of datasets which we know conform to the model. The AR coefficients \( \theta \) are selected to ensure that the resulting data is stationary. This is achieved by randomly selecting \( p \) points within the unit circle in the complex plane. If a point has an imaginary part then its complex conjugate must be one of the other points in the set. These points are used as the roots of the auxiliary polynomial and \( \theta \) is the vector whose elements are the coefficients of that polynomial.

With these parameters decided, the synthetic data is constructed according to \( T = WX + E \), where \( W \) is a random \( d \times q \) matrix, each element of \( E \) is generated from a Gaussian distribution with mean 0 and variance \( \sigma^2 \) and each row of the \( q \times N \) matrix \( X \) is generated from an autoregressive model of order \( p \), using the appropriate \( \theta \) and \( \phi^2 \) for that row. Each row was created with length \( 2N \) and then the first \( N \) elements are discarded to avoid burn-in effects and ensure that the whole series conforms to the autoregression model.

With \( d = 3, N = 200, q = 2 \) and \( p = 5 \), datasets were generated with every combination of \( \sigma^2 \) and \( \phi^2 \) (the same value for both rows of \( X \)) in the set \( 0.001, 0.01, 0.1, 1, 10 \) and different \( \theta \) values. This was repeated 100 times, giving 2500 datasets in all. PPCA-AR was trained against each of these datasets. Figure 1 shows that there is a close correlation between the model’s estimates of \( \sigma^2 \) and the actual values used to generate the data, even where the actual \( \sigma^2 \) value is significantly smaller than the actual \( \phi^2 \) values. Figure 2 shows that the model’s estimates of the \( \phi^2 \) values for each row are, to some extent, correlated with the actual values, but they are heavily influenced by the observation noise and tend to be over-estimated.

![Figure 1: Comparison of estimated observation noise variance \( \sigma^2 \) with actual values for each of 2500 trials. Note that \( \sigma^2 \) is plotted on a logarithmic scale.](image)

For these same 2500 datasets the accuracy of the model’s reconstruction of the data was determined by calculating the mean squared error between the reconstruction and the original data. Figure 3 shows how the PPCA-AR accuracy
Figure 2: Comparison of estimated excitation noise variance $\phi^2$ with actual values for each of 2500 trials. Top and bottom panels correspond to the two rows of $X$. Gray dots are estimated values; solid black lines are actual values; grey dotted line is the actual observation noise variance, $\sigma^2$.

Figure 3: Comparison of PPCA-AR mean squared error of data reconstruction with the equivalent PPCA error. The PPCA-AR error is always higher than that for PPCA (the grey line indicates equality).

Figure 4: Comparison of PPCA-AR mean squared error of data reconstruction with the actual observation noise variance, $\sigma^2$. Note the logarithmic ordinate.

Figure 5: Comparison of estimated $\theta$ values (crosses) with actuals (circles) for the 5 rows of $X$.

Figure 6: Hinton diagrams comparing the actual (left) and estimated (right) $W$ matrices.

III. MISSING DATA

As mentioned above, the additional constraints imposed by the PPCA-AR model mean that it is unlikely to reconstruct the data as well as standard PPCA, but the extra structure it enforces on the data allows it to better predict missing values.

We start by describing two methods for dealing with missing data values in PPCA. This leads to two variants which will be referred to as PPCA-t and PPCA-g. We then describe
how PPCA-AR deals with missing data and compare the performance of each of these three models using synthetic data.

A. PPCA with missing data

PPCA may be used to estimate missing data values using an iterative algorithm similar to that demonstrated in [3]. For a given model order, \( W \) and \( \sigma^2 \) are initialised with random values and the missing data values set to zero (since the data is mean-centred this starts them off at the mean value for their particular dimension). The process then iterates through the following steps until convergence:

(a) Calculate the expected values \( \langle x_n | t_n \rangle \) and \( \langle x_n x_n^T | t_n \rangle \) from (6)-(8) using the current estimates for \( W \) and \( \sigma^2 \).
(b) Calculate new estimates for \( W \) and \( \sigma^2 \) from (9)-(10).
(c) Calculate the reconstruction, \( T_{R_t} \), of \( T \) from (14).
(d) Replace the missing values in \( T \) with the equivalent values in \( T_{R_t} \).

This method includes the current estimates for the missing data in the calculation of the latent variables and will be referred to as PPCA-t.

Where data is missing it would be better for the latent variables in \( X \) to be estimated only from those values in \( T \) that are not missing as these constitute all the real evidence we have for the model. Each observation, \( t_n \), may be considered to be split into an observed part, \( g_n \), and a missing part, \( h_n \). For each observation \( W \) is similarly split giving us

\[
 t_n = \begin{pmatrix} g_n \\ h_n \end{pmatrix} \text{ and } W = \begin{pmatrix} W_{g_n} \\ W_{h_n} \end{pmatrix}. \tag{51}
\]

In the Expectation step, (6)-(8), \( t_n \) is replaced by \( g_n \), and \( \langle x_n | t_n \rangle \) and \( \langle x_n x_n^T | t_n \rangle \) are replaced by \( \langle x_n | g_n \rangle \) and \( \langle x_n x_n^T | g_n \rangle \) respectively. The Maximisation step remains unchanged. This method will be referred to as PPCA-g.

From (5) the covariance of \( t_n \) is given by

\[
 WW^T + \sigma^2 I = \begin{pmatrix} W_{g_n} W_{g_n}^T + \sigma^2 I & W_{g_n} W_{h_n}^T \\ W_{h_n} W_{g_n}^T & W_{h_n} W_{h_n}^T + \sigma^2 I \end{pmatrix}. \tag{52}
\]

Hence [7, section 8.1.3] the covariance of missing data, \( h_n \), given the observed data, \( g_n \), is

\[
 C_{h_n} = (W_{h_n} W_{h_n}^T + \sigma^2 I)^{-1} W_{h_n} W_{g_n}^T. \tag{53}
\]

B. PPCA-AR with missing data

With PPCA-AR, as for PPCA-g, where data is missing we estimate the values for the \( \bar{x}_n \) vectors based only on the observed data in \( t_n \). This is achieved by replacing \( t_n \) with \( g_n \) (as defined above) in the Expectation step in expressions (26) - (39). Once the iterative process has converged, we can calculate the probability distribution for \( t_n \):

\[
p(t_n) = N(t_n | \bar{W} \bar{m}_n, \bar{W} \bar{A}_n \bar{W}^T + \sigma^2 I) \tag{54}
\]

where \( \bar{A}_n = \bar{V}_n + \bar{m}_n \bar{m}_n^T \). The covariance of \( t_n \) is given by

\[
 \bar{W} \bar{A}_n \bar{W}^T + \sigma^2 I = \begin{pmatrix} \bar{W}_{g_n} \bar{A}_n \bar{W}_{g_n}^T + \sigma^2 I & \bar{W}_{g_n} \bar{A}_n \bar{W}_{h_n}^T \\ \bar{W}_{h_n} \bar{A}_n \bar{W}_{g_n}^T & \bar{W}_{h_n} \bar{A}_n \bar{W}_{h_n}^T + \sigma^2 I \end{pmatrix}. \tag{55}
\]

and the variance for the \( i \)th missing data value in \( h_n \) is given by the \( i \)th element of the covariance matrix \( C_{h_n} \). Again, the calculation of the variances is not performed for each iteration, just once when the iterative process has converged.

C. Illustration: missing synthetic data

We illustrate the ability of PPCA and PPCA-AR to estimate missing values by constructing synthetic datasets using the method described in section II-C and then declaring individual values to be present or missing by defining a matrix, \( M \), of indicator variables, \( m_{i,j} \), each having the value 1 if the associated data value in \( t_{i,j} \) is present or 0 if it is missing.

A new dataset was generated with parameters \( d = 30 \), \( N = 500 \), \( q = 20 \), \( p = 5 \), \( \sigma^2 = 0.04 \) and \( \phi^2 = 0.01 \). Data was made missing by creating holes centred on observations chosen uniformly at random; hole lengths were drawn from a Poisson distribution with mean 3. Holes were added to each row of the data independently to ensure that each row was missing approximately the same proportion of values. Datasets were generated with 0% to 90% missing data at 5% intervals. The holes were cumulative, meaning, for example, that the 20% missing dataset was based on the 15% missing dataset with 5% more holes added.

The three methods described in this paper, PPCA-t, PPCA-g and PPCA-AR, were trained against each dataset until convergence. PPCA-g displayed a tendency to oscillate, which was prevented by smoothing the update of \( W \) by setting it equal to the linear combination \( 0.75 W_{old} + 0.25 W_{new} \). This does not impact the convergence of the EM algorithm [8]. The methods were compared by measuring the mean squared error of the reconstruction of the observed values and missing values separately. Figure 7 shows these errors
plotted against the proportions of missing data. Since Tipping and Bishop show that PPCA gives the optimal reconstruction of the observations (in the mean squared error sense), it is not surprising that this new model, which enforces additional constraints, performs less well in this respect. However, these additional constraints provide more information about the underlying nature of the observations and the new model is consistently better at estimating the values of missing data.

IV. RESULTS: REAL DATA

The UK Office of National Statistics [9] publishes economic and socio-economic time series data relating to the UK. The time series release “Detailed Index of Production”, table “A1: Output of the production industries : seasonally adjusted” as at 9th April 2009 contains monthly production figures from January 1968 to February 2009 for 24 different industries, scaled so that 2003 figures equal 100 units. The 22nd dimension, series “CKZO: IOP: CA_1: Extraction of oil and gas: CVMSA”, was omitted as figures only existed from January 1976. All values were divided by 100 to ensure that the maximum variance for any dimension was less than 10. The result is a dataset with \( d = 23 \) and \( N = 494 \).

Gaps were made in the data using the same method as for the synthetic data. Arbitrary values of \( p = 5 \) and \( q = 10 \) were selected for PPCA-AR and \( q = 10 \) for PPCA-t and PPCA-g. The three methods described in this paper, PPCA-t, PPCA-g and PPCA-AR, were run against each dataset until convergence. The results, shown in figure 8, show that PPCA-AR is significantly better at predicting the missing data. It is instructive to compare the reconstructions made by PPCA-AR and PPCA. Figure 9 shows the reconstructions for dimension 22 (that with the largest variance) and dimension 3 (that with the smallest variance) with 50% of the values missing. Regarding dimension 22, the horizontal line which PPCA-t and PPCA-g seem to favour where data is missing corresponds to the mean value of the observed data for this dimension.

Figure 10 shows how the variances associated with the reconstructions differ between PPCA-AR and PPCA. As well as calculating more accurate estimates for the missing values, PPCA-AR is much more confident in those estimates.

V. CONCLUSIONS

Although PPCA, for a particular model order, gives the optimal mean squared reconstruction of data, the PPCA-AR model, which accounts for additional temporal structure, provides better reconstruction of missing values.

Here we have not addressed the problem of determining the model order, namely \( p \) for each latent variable and \( q \). Automatic relevance determination (ARD) [10] is effective for model order estimation of \( p \), even in the presence of leptokurtic noise [11], and \( q \) [12]. Current work is extending PPCA-AR for Student-t distributed latent variables in a variational Bayesian framework.
Figure 9: Index of Production Data. Reconstruction of missing data by PPCA-g (light-blue), PPCA-t (dark blue) and PPCA-AR (red) for those dimensions, 22 and 3, that have the greatest and smallest variances respectively. Vertical grey stripes indicate where values are missing and the black line shows the actual values. The bar chart (bottom) shows the total number of values missing across all dimensions.

Figure 10: Index of Production Data, dimension 1. Comparison of 1 standard deviation error bars for missing data by PPCA-g (light-blue), PPCA-t (dark blue) and PPCA-AR (red), with the actual values shown in black. Only times where data is missing for this dimension are shown (40% are missing).

REFERENCES