Coordinating Principal Component Analyzers

J.J. Verbeek and N. Vlassis and B. Kröse

Informatics Institute, University of Amsterdam
Kruislaan 403, 1098 SJ Amsterdam, The Netherlands

Abstract. Mixtures of Principal Component Analyzers can be used to model high dimensional data that lie on or near a low dimensional manifold. By linearly mapping the PCA subspaces to one global low dimensional space, we obtain a 'global' low dimensional coordinate system for the data. As shown by Roweis et al., ensuring consistent global low-dimensional coordinates for the data can be expressed as a penalized likelihood optimization problem. We show that a restricted form of the Mixtures of Probabilistic PCA model allows for a more efficient algorithm. Experimental results are provided to illustrate the viability method.

1 Introduction

With increasing sensor capabilities, powerful feature extraction methods are becoming increasingly important. Consider a robot sensing its environment with a camera yielding a stream of 100×100 pixel images. The observations made by the robot often have a much lower intrinsic dimension than the 10.000 dimensions the pixels provide. If we assume a fixed environment, and a robot that can rotate around its axis and translate through a room, the intrinsic dimension is only three. Linear feature extraction techniques are able to do a fair compression of the signal by mapping it to a much lower dimensional space. However, only in very few special cases the manifold on which the signal is generated is a linear subspace of the sensor space. This clearly limits the use of linear techniques and suggests to use non-linear feature extraction techniques.

Mixtures of Factor Analyzers (MFA) [2] can be used to model such non-linear data manifolds. This model provides local linear mappings between local latent spaces and the data space. However, the local latent spaces are not compatible with each other, i.e. the coordinate systems of neighboring factor analyzers might be completely differently oriented. Hence, if we move through the data space from one factor analyzer to the next we cannot predict the latent coordinate of a data point on the one factor analyzer from the latent coordinate on the other factor analyzer.

Recently, a model was proposed that integrates the local linear models into a global latent space, allowing for mapping back and forth between the global latent and the data-space [5]. The idea is that there is a linear map for each factor analyzer between the data-space and the global latent space. The model, which is fitted by maximizing penalized log-likelihood with an algorithm closely

related to the Expectation-Maximization (EM) algorithm [4], is discussed in the next section. In Section 3, we show how we can reduce the number of parameters to be estimated and simplify the algorithm of [5]. These simplifications remove the iterative procedure from the M-step and remove all matrix inversions from the algorithm. The price we pay is that the covariance matrices of the Gaussian densities we use are more restricted, as discussed in the same section. Experimental results are given in Section 4. A discussion and conclusions are provided in Section 5.

2 The density model

To model the data density in the high dimensional space we use mixtures of a restricted type of Gaussian densities. The mixture is formed as a weighted sum of its component densities, which are indexed by s. The mixing weight and mean of each component are given by respectively p_s and $\boldsymbol{\mu}_s$. The covariance matrices of the Gaussian densities are constrained to be of the form:

$$\mathbf{C} = \sigma^2(\mathbf{I}_D + \rho\boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top), \qquad \boldsymbol{\Lambda}^\top\boldsymbol{\Lambda} = \mathbf{I}_d, \qquad \rho > 0 \tag{1}$$

where D and d are respectively the dimension of the high-dimensional/data-space and the low-dimensional/latent space. We use $\mathbf{I}_d$ to denote the d-dimensional identity matrix. The d columns of $\boldsymbol{\Lambda}$, in factor analysis known as the *loading matrix*, are D-dimensional vectors spanning the local PCA. Directions within the PCA subspace have variance $\sigma^2(1+\rho)$, other directions have σ^2 variance. This is as the Mixture of Probabilistic Principal Component Analyzers (MPPCA) model [7], with the difference that here we do not only have isotropic noise outside the subspaces but also isotropic variance inside the subspaces. We use this density model to allow for convenient solutions later. In [9] we provide a Generalized EM algorithm to find maximum likelihood solutions for this model.

The same model can be rephrased using hidden variables $\mathbf{z}$, which we use to denote 'internal' coordinates of the subspaces. We scale the coordinates $\mathbf{z}$ such that: $p(\mathbf{z} \mid s) = \mathcal{N}(\mathbf{z}; 0, \mathbf{I}_d)$. The internal coordinates allow us to clearly express the link to the global latent space, for which we denote coordinates with $\mathbf{g}$. All mixture components have their own linear mapping to the global space, given by a translation $\boldsymbol{\kappa}$ and a matrix $\mathbf{A}$, i.e. $p(\mathbf{g} \mid \mathbf{z}, s) = \delta(\boldsymbol{\kappa}_s + \mathbf{A}_s\mathbf{z})$, where $\delta(\cdot)$ denotes the distribution with mass 1 at the argument. The generative model reads:

$$p(\mathbf{x} \mid \mathbf{z}, s) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_s + \sqrt{\rho_s}\sigma_s\boldsymbol{\Lambda}_s\mathbf{z}, \sigma_s^2\mathbf{I}_D), \tag{2}$$

$$p(\mathbf{x}) = \sum_s p_s\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_s, \sigma_s^2(\mathbf{I}_D + \rho_s\boldsymbol{\Lambda}_s\boldsymbol{\Lambda}_s^\top)), \quad p(\mathbf{g}) = \sum_s p_s\mathcal{N}(\mathbf{g}; \boldsymbol{\kappa}_s, \mathbf{A}_s\mathbf{A}_s^\top). \tag{3}$$

We put an extra constraint on the projection matrices:

$$\mathbf{A}_s = \alpha_s\sigma_s\sqrt{\rho_s}\mathbf{R}_s, \qquad \mathbf{R}_s^\top\mathbf{R}_s = \mathbf{I}_d, \qquad \alpha_s > 0, \tag{4}$$

hence $\mathbf{R}_s$ implements only rotations plus reflections.

Note that the model assumes that locally there is a linear correspondence between the data space and the latent space. It follows that the densities $p(\mathbf{g} \mid \mathbf{x}, s)$ and $p(\mathbf{x} \mid \mathbf{g}, s)$ are Gaussian and hence both $p(\mathbf{g} \mid \mathbf{x})$ and $p(\mathbf{x} \mid \mathbf{g})$ are mixtures of Gaussian densities. In the next section we discuss how this density model allows for an efficient learning scheme, as compared to the expressive but expensive MFA model proposed in [5].

3 The learning algorithm

The goal is, given observable data $\{\mathbf{x}_n\}$, to find a good density model in the data-space *and* mappings $\{\mathbf{A}_s, \boldsymbol{\kappa}_s\}$ that give rise to 'consistent' estimates for the hidden $\{\mathbf{g}_n\}$. With consistent we mean that if a point $\mathbf{x}$ in the data-space is well modeled by two PCA's, then the corresponding estimates for its latent coordinate $\mathbf{g}$ should be close to each other, i.e. the subspaces should 'agree' on the corresponding $\mathbf{g}$.

Objective Function: To measure the level of agreement, one can consider for all data points how uni-modal the distribution $p(\mathbf{g} \mid \mathbf{x})$ is. This idea was also used in [10]. There the goal was to find a global linear low-dimensional projection of supervised data, that preserves the manifold structure of the data. In [5] it is shown how the double objective of likelihood and uni-modality can be implemented as a penalized log-likelihood optimization problem. Let $Q(\mathbf{g} \mid \mathbf{x}_n) = \mathcal{N}(\mathbf{g}; \mathbf{g}_n, \boldsymbol{\Sigma}_n)$ a Gaussian approximation of the mixture $p(\mathbf{g} \mid \mathbf{x}_n)$ and $Q(s \mid \mathbf{x}_n) = q_{ns}$. We define:

$$Q(\mathbf{g}, s \mid \mathbf{x}_n) = Q(s \mid \mathbf{x}_n) Q(\mathbf{g} \mid \mathbf{x}_n). \tag{5}$$

As a measure of uni-modality we can use a sum of Kullback-Leibler divergences:

$$\sum_{ns} \int \mathrm{d}\mathbf{g}\, Q(\mathbf{g}, s \mid \mathbf{x}_n) \log \left[\frac{Q(\mathbf{g}, s \mid \mathbf{x}_n)}{p(\mathbf{g}, s \mid \mathbf{x}_n)} \right] = \sum_n D_{KL}(q_{ns} \parallel p_{ns}) + \sum_{ns} q_{ns} \mathcal{D}_{ns}$$

where $\mathcal{D}_{ns} = D_{KL}(Q(\mathbf{g} \mid \mathbf{x}_n) \parallel p(\mathbf{g} \mid \mathbf{x}_n, s))$ and $p_{ns} = p(s \mid \mathbf{x}_n)$. The total objective function, combining log-likelihood and the penalty term, then becomes:

$$\boldsymbol{\Phi} = \sum_n \log p(\mathbf{x}_n) - D_{KL}(\{q_{ns}\} \parallel \{p_{ns}\}) - \sum_s q_{ns} \mathcal{D}_{ns} \tag{6}$$

$$= \sum_{ns} \int \mathrm{d}\mathbf{g}\, Q(\mathbf{g}, s \mid \mathbf{x}_n) \big[-\log Q(\mathbf{g}, s \mid \mathbf{x}_n) + \log p(\mathbf{x}_n, \mathbf{g}, s) \big]. \tag{7}$$

The objective corresponds to a constrained EM procedure, c.f. [8] where the same idea is used to derive a probabilistic version of Kohonen's Self-Organizing Map [3]. Our density model differs with that of [5] in two aspects: (i) we use an isotropic noise model outside the subspaces (as opposed to diagonal covariance matrix) and (ii) we use isotropic variance inside the subspace (as opposed to general Gaussian). Also, using our density model it turns out that to optimize $\boldsymbol{\Phi}$ with respect to $\boldsymbol{\Sigma}_n$, it should be of the form[1] $\boldsymbol{\Sigma}_n = \beta_n^{-1} \mathbf{I}_d$. Therefore we

[1] Once we realize that the matrices $\mathbf{V}_s$ in [5] are of the form $c\mathbf{I}_d$ with our density model, it can be seen easily by setting $\partial \boldsymbol{\Phi} / \partial \boldsymbol{\Sigma}_n = 0$ that $\boldsymbol{\Sigma}_n = \beta^{-1} \mathbf{I}_d$.

work with β_n from now on. Using our density model and $\mathbf{g}_{ns} = \mathbf{g}_n - \boldsymbol{\kappa}_\mathbf{s}$ and $\mathbf{x}_{ns} = \mathbf{x}_n - \boldsymbol{\mu}_\mathbf{s}$ we can write (7) as:

$$\boldsymbol{\Phi} = \sum_{ns} q_{ns} \Bigg[-\frac{d}{2}\log\beta_n - \log q_{ns} - \frac{e_{ns}}{2\sigma_s^2} - \frac{v_s}{2}\Big[d\beta_n^{-1} + \frac{\mathbf{g}_{ns}^\top \mathbf{g}_{ns}}{\rho_s + 1}\Big] \tag{8}$$

$$-D\log\sigma_s + \frac{d}{2}\log\frac{v_s}{\rho_s+1} + \log p_s\Bigg] + const. \qquad \text{with}$$

$$e_{ns} = \parallel \mathbf{x}_{ns} - \alpha_s^{-1}\boldsymbol{\Lambda}_s\mathbf{R}_s^\top \mathbf{g}_{ns} \parallel^2 \qquad \text{and} \qquad v_s = \frac{\rho_s + 1}{\sigma_s^2 \rho_s \alpha_s^2}, \tag{9}$$

where v_s is the inverse variance of $p(\mathbf{g} \mid s, \mathbf{x})$ and e_{ns} is the squared distance between $\mathbf{x}_n$ and $\mathbf{g}_n$ mapped into the data space by component s.

Optimization: To optimize $\boldsymbol{\Phi}$ we use an EM-style algorithm, a simplified version of the algorithm provided in [5]. The simplifications are: (i) the iterative process to solve for the $\boldsymbol{\Lambda}_s, \mathbf{A}_s$ is no longer needed; an exact update is possible and (ii) the algorithm no longer involves matrix inversions. The same manner of computation is used: in the E-step, we compute the uni-modal distributions $Q(s, \mathbf{g} \mid \mathbf{x}_n)$, parameterized by $\beta_n, \mathbf{g}_n$ and q_{ns}. Let $\langle \mathbf{g}_n \rangle_s = \mathrm{E}_{p(\mathbf{g}|\mathbf{x}_n, s)}[\mathbf{g}]$ denote the expected value of $\mathbf{g}$ given $\mathbf{x}_n$ and s. We use the following identities:

$$\langle \mathbf{g}_n \rangle_s = \boldsymbol{\kappa}_s + \mathbf{R}_s \boldsymbol{\Lambda}_s^\top \mathbf{x}_{ns} \alpha_s \rho_s / (\rho_s + 1), \tag{10}$$

$$\mathcal{D}_{ns} = \frac{v_s}{2}\big[d\beta_n^{-1} + \parallel \mathbf{g}_n - \langle \mathbf{g}_n \rangle_s \parallel^2 \big] + \frac{d}{2}[\log\beta_n - \log v_s]. \tag{11}$$

The distributions Q can be found by iterating the fixed-point equations:

$$\beta_n = \sum_s q_{ns} v_s, \qquad \mathbf{g}_n = \beta_n^{-1} \sum_s q_{ns} v_s \langle \mathbf{g}_n \rangle_s, \qquad q_{ns} = \frac{p_{ns} \exp -\mathcal{D}_{ns}}{\sum_{s'} p_{ns'} \exp -\mathcal{D}_{ns'}},$$

where we used $p_{ns} = p(s \mid \mathbf{x}_n)$. In the M-step, we update the parameters of the mixture model. Using notation:

$$C_s = \sum_n q_{ns} \parallel \mathbf{g}_{ns} \parallel^2, \qquad E_s = \sum_n q_{ns} e_{ns}, \qquad G_s = d\sum_n q_{ns}\beta_n^{-1}, \tag{12}$$

the update equations are:

$$\boldsymbol{\kappa}_s = \frac{\sum_n q_{ns}\mathbf{g}_n}{\sum_n q_{ns}}, \qquad \boldsymbol{\mu}_s = \frac{\sum_n q_{ns}\mathbf{x}_n}{\sum_n q_{ns}}, \qquad \alpha_s = \frac{C_s + G_s}{\sum_n q_{ns}(\mathbf{g}_{ns}^\top \mathbf{R}_s \boldsymbol{\Lambda}_s^\top \mathbf{x}_{ns})},$$

$$\rho_s = \frac{D(C_s + G_s)}{d(\alpha_s^2 E_s + G_s)}, \quad \sigma_s^2 = \frac{E_s + \rho_s^{-1}\alpha_s^{-2}[C_s + (\rho_s+1)G_s]}{(D+d)\sum_n q_{ns}}, \quad p_s = \frac{\sum_n q_{ns}}{\sum_{ns'} q_{ns'}}.$$

Note that the above equations require E_s which in turn requires $\boldsymbol{\Lambda}_s\mathbf{R}_s^\top$ via equations (9) and (12). To find $\boldsymbol{\Lambda}_s\mathbf{R}_s^\top$ we have to minimize:

$$\sum_n q_{ns} e_{ns} = \sum_n q_{ns} \parallel \mathbf{x}_{ns} - \alpha_s^{-1}\boldsymbol{\Lambda}_s\mathbf{R}_s^\top \mathbf{g}_{ns} \parallel^2 = -\sum_n q_{ns}\mathbf{x}_{ns}^\top(\boldsymbol{\Lambda}_s\mathbf{R}_s^\top)\mathbf{g}_{ns} + const.$$

This problem is known as the 'weighted Procrustes rotation' [1]. Let

$$\mathbf{C} = [\sqrt{q_{1s}}\mathbf{x}_{1s} \cdots \sqrt{q_{ns}}\mathbf{x}_{ns}][\sqrt{q_{1s}}\mathbf{g}_{1s} \cdots \sqrt{q_{ns}}\mathbf{g}_{ns}]^\top, \quad \text{with SVD:} \quad \mathbf{C} = \mathbf{U}\mathbf{L}\boldsymbol{\Gamma}^\top,$$

where the $\mathbf{g}_{ns}$ have been padded with zeros to form D-dimensional vectors, then the optimal $\boldsymbol{\Lambda}_s\mathbf{R}_s^\top$ is given by the first d columns of $\mathbf{U}\boldsymbol{\Gamma}^\top$.

4 Experimental Illustration

To demonstrate the method, we captured 40×40 pixel gray valued images of a face with a camera. The face has two degrees of freedom, namely looking up-down and left-right. We learned a coordinated mixture model with 1000 images. We used a global PCA projection to 22 dimensions, preserving over 70% of the variance in the data set. We used a latent dimensionality of two and 20 mixture components. We initialized the coordinated mixture model by clamping the latent coordinates $\mathbf{g}_n$ at coordinates found by Isomap [6] and clamping the β_n at small values for the first 50 iterations. The q_{ns} were initialized uniformly random, and updated from the start. The obtained coordinated mixture model was used to map 1000 'test' images. For each test image $\mathbf{x}_n$ we approximated $p(\mathbf{g} \mid \mathbf{x}_n)$ with a single Gaussian $Q_n = \arg\min_Q D_{KL}(Q \parallel p(\mathbf{g} \mid \mathbf{x}_n))$ with a certain mean and standard deviation. In Figure 1 we show these means (location of circle) and standard deviations (radius). To illustrate the discovered parametrization further, two examples of linear traversal of the latent space are given.

5 Conclusions and Discussion

We showed how a special case of the density model used in [5] leads to a more efficient algorithm to coordinate probabilistic local linear descriptions of a data manifold. The M-step can be computed at once, the iterative procedure to find solutions for a Riccati equation is no longer needed. Furthermore, the update equations do not involve matrix inversion anymore. However, still d singular values and vectors of a $D \times D$ matrix have to be found.

The application of this method to partially supervised data sets is an interesting possibility and a topic of future research. Another important issue, not addressed here, is that often when we collect data from a system with limited degrees of freedom we actually observe *sequences* of data. If we assume that the system can vary its state only in continuous manner, these sequences should correspond to paths on the manifold of observable data. This fact might be exploited to find a low dimensional embedding of the manifold. In [9] we report on promising results of experiments where we used this model to map omni-directional camera images, recorded through an office, to a 2d latent space (the location in the office), where the data was 'supervised' in the sense that 2d workfloor coordinates are known.

Acknowledgment: This research is supported by the Technology Foundation STW (project nr. AIF4997) applied science division of NWO and the technology program of the Dutch Ministry of Economic Affairs.

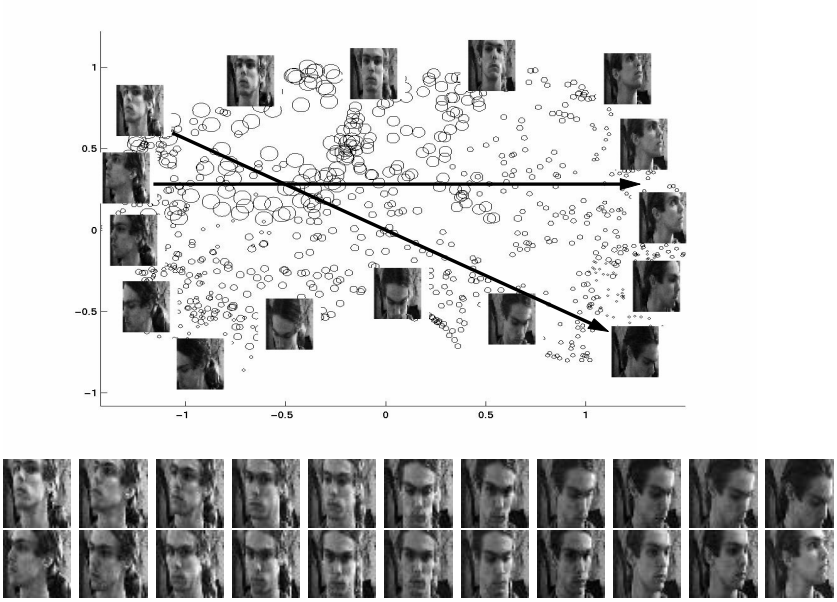

Fig. 1. Latent coordinates and linear trajectories in the latent space.

References

1. T.F. Cox and M.A.A. Cox. *Multidimensional Scaling*. Number 59 in Monographs on statistics and applied probability. Chapman & Hall, 1994.
2. Z. Ghahramani and G.E. Hinton. The EM Algorithm for Mixtures of Factor Analyzers. Technical Report CRG-TR-96-1, University of Toronto, Canada, 1996.
3. T. Kohonen. *Self-Organizing Maps*. Springer Series in Information Sciences. Springer-Verlag, Heidelberg, Germany, 2001.
4. R.M. Neal and G.E. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In M.I. Jordan, editor, *Learning in Graphical Models*, pages 355–368. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.
5. S.T. Roweis, L.K. Saul, and G.E. Hinton. Global coordination of local linear models. In T.G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*. MIT Press, 2002.
6. J.B. Tenenbaum, V. de Silva, and J.C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.
7. M.E. Tipping and C.M. Bishop. Mixtures of probabilistic principal component analysers. *Neural Computation*, 11(2):443–482, 1999.
8. J.J. Verbeek, N. Vlassis, and B. Kröse. The Generative Self-Organizing Map: A Probabilistic Generalization of Kohonen's SOM. Technical Report IAS-UVA-02-03, Informatics Institute, University of Amsterdam, The Netherlands, May 2002.
9. J.J. Verbeek, N. Vlassis, and B. Kröse. Procrustes Analysis to Coordinate Mixtures of Probabilistic Principal Component Analyzers. Technical report, Informatics Institute, University of Amsterdam, The Netherlands, February 2002.
10. N. Vlassis, Y. Motomura, and B. Kröse. Supervised dimension reduction of intrinsically low-dimensional data. *Neural Computation*, 14(1):191–215, January 2002.