Aggregation of probabilistic PCA mixtures with a variational-Bayes technique over parameters

Pierrick Bruneau, Marc Gelgon, Fabien Picarougne

Aggregation of Probabilistic PCA mixtures with a variational-Bayes technique over parameters

Pierrick Bruneau[1,2], Marc Gelgon[1,2], Fabien Picarougne[2]
(1) INRIA Atlas - (2) LINA (UMR CNRS 6241) - Nantes university, France.
firstname.lastname@univ-nantes.fr

Abstract

This paper proposes a solution to the problem of aggregating versatile probabilistic models, namely mixtures of probabilistic principal component analyzers. These models are a powerful generative form for capturing high-dimensional, non Gaussian, data. They simultaneously perform mixture adjustment and dimensionality reduction. We demonstrate how such models may be advantageously aggregated by accessing mixture parameters only, rather than original data. Aggregation is carried out through Bayesian estimation with a specific prior and an original variational scheme. Experimental results illustrate the effectiveness of the proposal.

1. Introduction

This paper proposes an effective and original solution to the problem of aggregating versatile probabilistic models, namely mixtures of probabilistic principal component analyzers (MPPCA hereafter [8]). The contribution of the paper is a demonstration of how this aggregation can be conducted by accessing only the parameters of the models to merge, rather than the original data.

Probabilistic PCA (PPCA) is a dimensionality reduction technique that extends standard PCA with the following advantages : a) since a probabilistic model is fit to the data, Bayesian inference may be applied, in particular to determine the appropriate model complexity; b) mixtures of such PPCA components may be built and estimated, to capture high dimensional data sets supported by non linear manifolds. Let us emphasize that Bayesian MPPCAs enable the number of parameters to grow only as required by intrinsic data complexity requires. This is typically much lower than the number of parameters of Gaussian mixture model with high-dimensional covariance matrices. As a result, MPPCA models have better resilience to the curse of dimensionality.

This paper deals with the aggregation of such models, providing a central tool to growing needs for performing pattern recognition on distributed data sources, motivated by infrastructures such as peer-to-peer, grids or sensor networks. Our scheme focuses on statistical learning of global data models through the aggregation of a set of local, parametric models, and its main features are as follows:

- motivated by low computational or network-load cost, or for protecting confidentiality of individual data entries, our proposal is designed so that only model parameters need be accessed to aggregate mixtures of PPCA. In other words, the scheme operates on the components of the mixtures to aggregate, rather than original data.
- aggregation of a set of MMPCA models consists in their addition, followed by a "compression" phase that seeks an optimal combination of mixture components. A central issue to mixture aggregation is the determination of the number of components. We formulate it as Bayesian estimation and show how EM-like variational inference can address it. While this generalizes recent work [4] from GMMs to MPPCA, iterative update equations have to be largely reconsidered.

Section 2 recalls Probabilistic PCA and its extension to mixtures of PPCA; we then sketch an associated variational-Bayes estimation. Section 3 first discloses how mixtures of PPCA may be extended to handle components and presents a novel estimation scheme for this model. Section 4 provides experimental results.

2. Mixtures of Probabilistic Principal Components Analysers (MPPCA)

2.1 A probabilistic view to the PCA

Principal Component Analysis is a popular, baseline technique for dimensionality reduction. Given a d-

dimensional data set, the principal subspace is generally obtained by diagonalizing the sample covariance, i.e. by seeking an eigendecomposition of this $d \times d$ matrix. Tipping [9] proposed an alternative, probabilistic framework to PCA, based on the assumption that every data item y is generated by transforming a zero mean unit variance q-dimensional variable x ($q < d$) with additive isotropic noise.

$$y = \Lambda x + \mu + \epsilon \quad (1)$$

Let us define the associated probability density functions (*pdf*) :

$$p(y|x) = \mathcal{N}(\Lambda x + \mu, \sigma^2 I_d) \quad (2)$$

$$p(x) = \mathcal{N}(0, I_q) \quad (3)$$

$$p(\epsilon) = \mathcal{N}(0, \sigma^2 I_d) \quad (4)$$

Results for linear Gaussian models [3] provide the following marginal distribution for y :

$$p(y) = \mathcal{N}(\mu, \Lambda\Lambda^T + \tau^{-1} I_d) \quad (5)$$

where $\tau = \sigma^{-2}$. Λ is a $d \times q$ matrix, usually known as the *factor* matrix. Later we denote $C = \Lambda\Lambda^T + \tau^{-1} I_d$ for concision. ML estimates for Λ were proven to span the principal subspace of the data sample [9]. This estimate has no closed-form solution, but may be obtained through an iterative scheme. More precisely, update formulas can be derived for each parameter by differentiation, leading to an EM-like algorithm [9].

The ML solution obtained for the PPCA model is up to an arbitrary rotation matrix. Still, this matrix can be recovered by diagonalizing $\Lambda_{ML}^T \Lambda_{ML}$ [9], with limited computational overhead as this matrix is $q \times q$. Post-multiplying Λ_{ML} by this rotation matrix allows us to obtain the scaled eigenvectors for our subspace, ordered by decreasing magnitude.

2.2 Handling a mixture of PPCA

The framework presented in paragraph 2.1 is naturally extended by introducing a latent variable z indicating the membership of a data item to a PPCA model (called component hereafter). A set of weights $\{\omega_k\}$ is associated with K components to describe the relative importance of components. z is a binary one-of-K variable, meaning that if any item y belongs to the component k, then $z_k = 1$ and $z_j = 0, \forall j \neq k$. Thus, a multimodal density is fitted on the data set, and each component of the mixture density determines its principal subspace. For a data item, the associated *pdf* is:

$$p(z) = \prod_k^K {\omega_k}^{z_k} \quad p(y|z) = \prod_k^K \mathcal{N}(y|\mu_k, C_k)^{z_k} \quad (6)$$

$$p(y) = \sum_k^K \omega_k \mathcal{N}(y|\mu_k, C_k) \quad (7)$$

Consequently, a data set $\mathbf{y} = \{y_1, \ldots, y_N\}$ has a likelihood function defined as follows :

$$p(\mathbf{y}) = \prod_n^N \sum_k^K \omega_k \mathcal{N}(y_n|\mu_k, C_k) \quad (8)$$

An iterative scheme can be obtained by differentiation, by analogy to the one-component case. ML estimation was addressed in [9], leaving open the target number of components K and the number of factors in each component q. Besides, like most ML approaches, it suffers from local minima and degeneracies issues ([3], ch.9,11). Closely related to our proposal, a variational-Bayes scheme was proposed for the single component PPCA [2] and the mixture of Factor Analysers [1, 6] . Factor Analysers and PPCA are very closely related models, but with sensibly different properties, discussed in [8]. Briefly stated, the design of variational algorithms enables them to overcome the ML defects indicated above. Using a Bayesian integration of the problem with proper uninformative priors, a tradeoff between desired properties (here, having a low number of components and factors per component) and the data likelihood is performed. This integrand is intractable, and an approximate solution is inferred by lower bounding using variational distributions. Update formulas are obtained from functional calculus. This leads to an algorithm which general form is akin to the commun ML EM-like algorithm, but which optimizes against distributions instead of parameter values.

Uninformative priors and suitable initialization strategies are employed to perform the automatic choice of K and q. Beal [1] proposes a birth and death strategy to address the automatic choice of K. This same problem was solved using a Dirichlet prior which favors a minimal number of effective components in [3], ch.11. q is found using Automatic Relevance Determination (ARD) [7]. More specifically, each factor has a zero-mean, normal prior, with a Gamma prior defined over its precision. If a factor plays no role in explaining data, the precision of the normal posterior will be driven to 0 by the Gamma posterior. This factor can then be discarded.

3. Aggregating MPPCA models from their parameters

Let us now consider our input to be an existing MPPCA model. This mixture might be redundant (i.e. overlapping or excessively numerous components, or over-complex factor matrices), for example if obtained from the aggregation of different sources of the same underlying signal - which is our target application. In this section, we show first how such an input can be seen as the

limit representation of a virtual data set. Then, we incorporate this representation in the algorithm sketched in section 2.2 in replacement of an ordinary data set. As a result, we obtain the low complexity model that best fits the data which would have been generated from the input mixture, without resorting to the data itself or any sampling scheme. Consider a sample originating from an arbitrary input PPCA mixture having L components. We denote this sample $\mathbf{y} = \{\hat{y}_l\}$, where $\hat{y}_l$ is the subset of items associated with the PPCA component l. The conditional likelihood $\mathcal{L}$ of this data with respect to an output mixture (indexed by k) may be expressed as:

$$\mathcal{L}(\mathbf{y}|\mathbf{z}) = \prod_{l}^{L}\prod_{k}^{K}\left[p(\hat{y}_l|\omega_k,\mu_k,C_k)\right]^{z_{lk}} \tag{9}$$

This formulation is made under the assumption that all data generated from a component in the input mixture will be assigned to the same component in the output mixture. This assumption generally holds as an aggregation task is mostly about regrouping input components optimally. Expanding the log of the previous expression provides :

$$\ln\mathcal{L}(\mathbf{y}|\mathbf{z}) = \ln\prod_{l}^{L}\prod_{k}^{K}\prod_{j}^{|\hat{y}_l|}\mathcal{N}(y_{lj}|\omega_k,\mu_k,C_k)^{z_{lk}} \tag{10}$$

$$\begin{aligned}\ln\mathcal{L}(\mathbf{y}|\mathbf{z}) &= \sum_{l}^{L}\sum_{k}^{K} z_{lk}\sum_{j}^{|\hat{y}_l|}\ln\mathcal{N}(y_{lj}|\omega_k,\mu_k,C_k)\\ &= \sum_{l}^{L}\sum_{k}^{K} z_{lk}\ln\mathcal{L}_{lk}\end{aligned}$$

We now perform an asymptotical approximation [10]:

$$\ln\mathcal{L}_{lk} = \sum_{j}^{|\hat{y}_l|}\ln\mathcal{N}(y_{lj}|\omega_k,\mu_k,C_k) \tag{11}$$

$$\simeq \sum_{j}^{N\omega_l}\ln\mathcal{N}(y_{lj}|\omega_k,\mu_k,C_k) \tag{12}$$

$$\simeq N\omega_l\left[-\mathrm{KL}(\mathcal{N}(\mu_l,C_l)\parallel\mathcal{N}(\mu_k,C_k) - \mathrm{H}(\mathcal{N}(\mu_l,C_l)\right] \tag{13}$$

where KL denotes the Kullback-Leibler divergence and H the entropy. Since these quantities have closed forms for Gaussians, we obtain :

$$\begin{aligned}\ln\mathcal{L}_{lk} &= N\omega_l[\frac{d}{2}\ln(2\pi) + \frac{1}{2}\det(\Lambda_k\Lambda_k^T + \tau_k^{-1}I_d)\\ &+ \frac{1}{2}\mathrm{Tr}((\Lambda_k\Lambda_k^T + \tau_k^{-1}I_d)^{-1}\\ &[\Lambda_l\Lambda_l^T + \tau_l^{-1} + (\mu_l-\mu_k)(\mu_l-\mu_k)^T)])]\end{aligned}$$

In the remainer of the paper, we discard the influence of τ_l^{-1}, as the ML value of this term embodies the smallest eigenvalues of the respective components. Since $\Lambda_l\Lambda_l^T = \sum_j \Lambda_l^{\cdot j}\Lambda_l^{\cdot jT}$, we may describe $\ln\mathcal{L}$ as the combined likelihood of the means and factors of our input components (up to a correct normalization, and with respective means μ_k and 0).

$$\mathcal{L}_{lk} \equiv \left[\mathcal{N}(\mu_l|\mu_k,C_k)\prod_j\mathcal{N}(\Lambda_l^{\cdot j}|0,C_k)\right]^{N\omega_l} \tag{14}$$

Let us underline that our asymptotical approximation leads to a likelihood term with no dependence on an input data set, solely relying on the parameters of the input MPPCA. Furthermore, the functional form of the Gaussian is preserved, allowing the usual derivation of a lower bound that founds the variational algorithm [1, 2, 4]. Combining result (14) with (8) :

$$p(\mathbf{y}) = \prod_{l}^{L}\sum_{k}^{K}\left(\omega_k\mathcal{N}(\mu_l|\mu_k,C_k)\prod_{j}^{q}\mathcal{N}(\Lambda_l^{\cdot j}|0,C_k)\right)^{N\omega_l} \tag{15}$$

We may introduce the latent variables $\mathbf{x}$, as the covariance matrices in (15) permit it. In the classical scheme [1, 8], there is a single variable x per item. Now, each input component is associated with $1+q$ items, so $\mathbf{x}$ scales accordingly :

$$\begin{aligned}p(\mathbf{y}) = \prod_{l}^{L}\sum_{k}^{K}(\omega_k\int dx_{1l}p(x_{1l})\mathcal{N}(\mu_l|\Lambda_k x_{1l}+\mu_k,\tau_k^{-1}I_d)\\ \prod_{j}^{q}\int dx_{2lj}p(x_{2lj})\mathcal{N}(\Lambda_l^{\cdot j}|\Lambda_k x_{2lj},\tau_k^{-1}I_d))^{N\omega_l}\end{aligned} \tag{16}$$

The lower bound formulation proposed in [1] is employed with (16) as its likelihood term, and this leads to a tractable set of update formulas. Thorough mathematical and implementation details may be found in [5]
In section 2.2, we mentioned the usage of uninformative priors. These are still used here, but we may also jointly exploit some prior knowledge. Indeed we noticed that the standard estimation procedure was able to recover the scaled eigenvectors ordered by decreasing magnitude in the columns of Λ_{ML}. We also remark that the additional latent variables are associated with the columns of the Λ input matrices. Under the assumption of appropriately ordered input Λ, intuitively we would associate the first column of the input Λ to the first column of the output Λ and so on. As x variables denote the combination of columns of Λ, we therefore choose to initialize x_2.. estimates to canonical vectors, so as to reflect this belief. Experimentally this principle was found to improve the results very significantly.

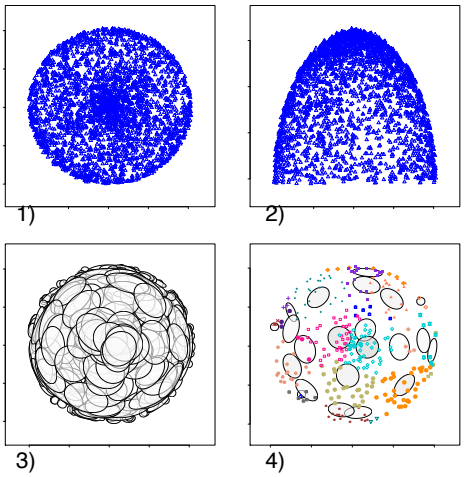

Figure 1. 1) Hemisphere data seen from above and 2) seen from one side. 3) Concatenation of several MPPCAs, forming the input for our algorithm. 4) Result after aggregation, represented along with some data points. Gaussian components are represented by ellipses, with their background color indicating the respective weights.

4 Experimental results

We report experimental results on two synthetic data sets (one clustering-oriented, one with a non-linear manifold):

- a mixture of 3 multivariate Gaussians with random covariance matrices. The sample is produced from 2D Gaussians, and 6 additional dimensions are generated by various linear combinations of the 2D signal with additive noise.
- a hemisphere. For sampling each point, random angles and additive noise are chosen (fig. 1)

For each data set, 3000 points are sampled and :

- subsamples of 300 points are randomly selected. A MPPCA model is fitted to each subsample (fig. 1);
- the first experiment is repeated 100 times. We monitor the number of clusters, and more importantly, the dimensionality of the subspace discovered for each cluster. Indeed, our ground truth tells us we should find 2D manifolds;
- we assess our aggregation technique by choosing randomly 20 models and using them as an input (fig. 1). This experiment is repeated 50 times. The quality of the obtained models will be assessed by monitoring the cluster subspaces dimensionality, and the Jensen-Shannon divergence between the model aggregation and the model obtained using the whole data set.

Correctly, 2D subspaces are always detected for all our clusters when fitting MPPCA to the subsamples. After the aggregations, this property is preserved. 3 clusters are always detected for the 2D Gaussians subsamples, and after aggregations these 3 clusters are found again. 15.2 clusters are found on average when fitting MPPCA to hemisphere subsamples. Aggregations of these models produce 22.1 clusters on average, thus validating the parsimony property.

We use $JS(\text{model1}\|\text{model2})$ to denote the Jensen-Shannon divergence between two models. This divergence measure is a symmetrized and normalized variant of the KL divergence. $JS(\text{model1}\|\text{model2}) \in [0, 1]$, and values below 0.2 generally indicate very similar distributions. For the 2D Gaussians and the hemisphere data sets, we respectively have $JS(\text{aggregation}\|\text{full data}) = 0.15$ and 0.20 on average, indicating strong similarity.

References

[1] M. J. Beal. *Variational Algorithms for approximate inference*. PhD thesis, University of London, 2003.

[2] C. M. Bishop. Variational principal components. *Proceedings of 9th ICANN*, 1:509–514, 1999.

[3] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, 2006.

[4] P. Bruneau, M. Gelgon, and F. Picarougne. Parsimonious reduction of Gaussian mixture models with a variational-Bayes approach. *Pattern Recognition*, 43:850–858, March 2010.

[5] P. Bruneau, M. Gelgon, and F. Picarougne. A variational algorithm for PPCA mixtures aggregation. Technical report, INRIA, available during Feb. 2010.

[6] Z. Ghahramani and M. J. Beal. Variational inference for Bayesian mixtures of factor analysers. *Advances in Neural Information Processing Systems*, 2000.

[7] D. MacKay. Probable networks and plausible predictions — a review of practical Bayesian methods for supervised neural networks. *Network: Computation in Neural Systems*, 6:469–505, 1995.

[8] M. E. Tipping and C. M. Bishop. Mixtures of probabilistic principal component analyzers. *Neural Computation*, 11(2):443–482, 1999.

[9] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society - B Series*, 61(3):611–622, 1999.

[10] N. Vasconcelos. Image indexing with mixture hierarchies. *Proceedings of IEEE Conference in Computer Vision and Pattern Recognition*, 1:3–10, 2001.