The Annals of Applied Statistics
2012, Vol. 6, No. 1, 125–160
DOI: 10.1214/11-AOAS499

A STOCHASTIC ALGORITHM FOR PROBABILISTIC INDEPENDENT COMPONENT ANALYSIS

BY STÉPHANIE ALLASSONNIÈRE AND LAURENT YOUNES[1]

CMAP, Ecole Polytechnique and CIS, and Johns Hopkins University

The decomposition of a sample of images on a relevant subspace is a recurrent problem in many different fields from Computer Vision to medical image analysis. We propose in this paper a new learning principle and implementation of the generative decomposition model generally known as noisy ICA (for independent component analysis) based on the SAEM algorithm, which is a versatile stochastic approximation of the standard EM algorithm. We demonstrate the applicability of the method on a large range of decomposition models and illustrate the developments with experimental results on various data sets.

1. Introduction. Independent Component Analysis (ICA) is a statistical technique that aims at representing a data set of random vectors as linear combinations of a fixed family of vectors with statistically independent coefficients. It was initially designed to solve source separation problems in acoustic signals [Bremond, Moulines and Cardoso (1997)] and rapidly found a large range of applications, in particular, in medical image analysis [Calhoun et al. (2001), Calhoun, Adali and McGinty (2001)], where ICA has become one of the standard approaches. And because it is often valuable to decompose a large set of variables into simple components, ICA applies more generally as well [in computer vision Bartlett, Movellan and Sejnowski (2002), Bell and Sejnowski (1995a), Farid and Adelson (1999), Liu and Wechsler (2003); and in computational biology Liebermeister (2002), Makeig and Jung (1997), Scholz et al. (2004), etc.].

Often in such problems, the data are high dimensional but have small to moderate sample size, which complicates statistical analysis. For example, one challenge in medical imaging is to extract significant information from spatially varying anatomical or functional signals drawn from a relatively small number of individuals. A common way to address this issue is to apply dimension-reduction techniques to reduce the information to a smaller number of highly informative statistics. ICA can be used for this purpose, and, in many cases, the representations it provides are qualitatively very different from those obtained using decorrelation methods such as principal components analysis (PCA) [Üzümcü et al. (2003)].

Received June 2010; revised July 2011.

[1] Supported in part by NSF ITR:0427223.

Key words and phrases. Independent component analysis, independent factor analysis, stochastic approximation, EM algorithm, statistical modeling, image analysis, $\LaTeX\ 2_{\varepsilon}$.

ICA can be formulated in terms of a generative model that approximates the distribution of the data, allowing well-understood statistical methods to be used for training and validation. ICA represents an observed d-dimensional random variable $\mathbf{X}$ as

$$\mathbf{X} = \sum_{j=1}^{d} \beta^j \mathbf{a}_j, \tag{1.1}$$

where $(\mathbf{a}_1, \dots, \mathbf{a}_d) \in \mathbb{R}^{d\times d}$ are parameters (called decomposition vectors) and $\beta^1, \dots, \beta^d$ are *independent* scalar random variables drawn from a specified distribution (or family of distributions). One product of ICA is an estimate of the decomposition matrix $\mathbf{A} = (\mathbf{a}_1, \dots, \mathbf{a}_d)$ based on i.i.d. observations $(\mathbf{X}_1, \dots, \mathbf{X}_n)$. With model (1.1), the independent components $\beta^1, \dots, \beta^d$ can be computed from $\mathbf{X}$ by inverting $\mathbf{A}$. A variety of methods and criteria have been proposed to estimate either $\mathbf{A}$ or $\mathbf{W} = \mathbf{A}^{-1}$ (see http://www.tsi.enst.fr/icacentral/index.html, from which some algorithms may be accessed). For example, in Arie (2002), $\mathbf{A}$ is seen as a joint diagonalizer of a set of estimated correlation matrices. In Bell and Sejnowski (1995b) and Eriksson, Karvanen and Koivunen (2000), standard estimation procedures, like maximum entropy or minimum Kullback–Leibler divergence, are used with specified distributions for the independent components.

We will also use a model-based formulation in this paper, but it is important to mention that a large class of algorithms have also been defined for distribution-free representations (based on the so-called negentropy—non-Gaussian entropy—and cumulant expansion), including the widely used FastICA method [Hyvärinen and Oja (1997)], as well as algorithms proposed in Learned-Miller et al. (2003) or Bach and Jordan (2003), which maximize the independence (with respect to some criteria) of the components using the semi-parametric model of ICA. A comparison study has been made in Cardoso (1999), using high-order measures to assess component independence.

One of the drawbacks of ICA is that it does not come (like PCA does) with a well-defined method to select the most important components. In the original formulation, the number of independent components is equal to the dimension of the variables, so that the decomposition is achieved without dimensional reduction. This leads to computational and overfitting issues when dealing with high-dimensional data and small sample sizes, and a lack of interpretability of the obtained results.

Probabilistic ICA (alternatively called noisy ICA, or independent factor analysis, although we will reserve the latter term to a more specific method in which factors are Gaussian mixtures) assumes a small number of independent components, with a residual term which is modeled as Gaussian noise. The explicit model is therefore given by

$$\mathbf{X} = \sum_{j=1}^{p} \beta^j \mathbf{a}_j + \sigma\boldsymbol{\varepsilon}, \tag{1.2}$$

where $(\mathbf{a}_1, \ldots, \mathbf{a}_p) \in \mathbb{R}^{d \times p}$ now represent $d \times p$ parameters (to be compared to the $d \times d$ matrix, $\mathbf{A}$, estimated in the standard ICA model), $\beta^1, \ldots, \beta^p$ are independent scalar random variables and $\boldsymbol{\varepsilon}$, the noise, follows a standard normal distribution (we will take the standard deviation, σ, to be a fixed scalar, also a parameter). Such models therefore represent the d-dimensional input vector, $\mathbf{X}$, by p scalar components, achieving the required dimensional reduction.

The ICA training algorithms (e.g., estimating $\mathbf{W}$) do not generalize to probabilistic ICA. In particular, the d-dimensional vector $\mathbf{X}$ is modeled as a function of the $(p + d)$-dimensional variable $(\boldsymbol{\beta}, \boldsymbol{\varepsilon})$ and we have partial observations. A possible approach is to first implement some dimension reduction to the data, typically projecting $\mathbf{X}$ on the p first principal components to eliminate the residual, before applying standard ICA to the projection [Côme et al. (2008), Varoquaux et al. (2010)]. But this procedure does not necessarily retrieve the model described in (1.2) (especially when the noise has a large variance), and training probabilistic ICA in a way which is consistent with this statistical model certainly is a more satisfactory approach.

The numerical method described in this paper estimates the maximum likelihood estimator associated to (1.2), where the likelihood is for the observations, $\mathbf{X}$, therefore averaging over the unobserved components $\boldsymbol{\beta}$. This differs from the solution which is often adopted in the literature, which consists in maximizing the joint likelihood of $\mathbf{X}$ and $\boldsymbol{\beta}$, simultaneously in the parameters and in the unobserved variables [Hyvarinen (1999)]. This latter method attempts to solve the parametric estimation and hidden variable reconstruction problems at the same time. However, the estimation of both $\mathbf{X}$ and $\boldsymbol{\beta}$ is not always a good choice, because it can lead to biased estimators: as we will show in our experiments, these approaches have good results when the noise level is small [as already noticed in Valpola Lappalainen and Pajunen (2000)], but these results can significantly degrade otherwise [see Section 5, or Allassonnière, Amit and Trouvé (2007) for a similar observation made in a different context]. In contrast, averaging over the unobserved variables takes the whole distribution into account, which becomes important as soon as the posterior distribution is not unimodal, with its mean equal to its mode. The reconstruction problem (estimating $\boldsymbol{\beta}$ from $\mathbf{X}$), which is also important, for example, to define efficient lossy compression methods, can be solved *afterward* using the estimated parameters. Estimation and reconstruction are, in this regard, two separate problems.

When independent components are modeled as mixtures of Gaussians, as done in Moulines, cois Cardoso and Gassiat (1997) for blind source separation and blind deconvolution, or with independent factor analysis (IFA), as introduced in Attias (1999), maximizing the likelihood of the observations (averaging over the nonobserved independent components) can be done using the expectation–maximization (EM) algorithm. In this particular case, this algorithm can be derived with closed form formulae and explicit computations. But, even in this special case (mixture of

Gaussians), the EM algorithm can become computationally prohibitive, especially when the number of components is large. For general component distributions, the explicit evaluation of conditional expectations given observations constitutes an infeasible task, and only Markov chain Monte Carlo (MCMC) approximations remain available. Replacing explicit formulae by Monte Carlo approximations in the E-step of the EM algorithm leads to the MCEM algorithm, introduced in Moulines, cois Cardoso and Gassiat (1997). MCEM, however, still is a highly computational procedure, with many Monte Carlo samples required at each update of the parameters.

We suggest using an alternative approach for maximum likelihood estimation, relying on a stochastic approximation to the EM algorithm [called SAEM, Delyon, Lavielle and Moulines (1999)] which only requires being able to sample from this conditional distribution. Instead of running a long Monte Carlo simulation at each E-step, as MCEM does, this algorithm interlaces sampling with the M-step, requiring only a single new sample between two parameter updates. This algorithm compensates the larger convergence time (in number of steps) generally associated to stochastic approximations by much simpler iteration steps. This algorithm has been proposed and proved convergent under some weak conditions in Allassonnière, Kuhn and Trouvé (2010). A comparison between the MCEM and SAEM is proposed as part of the experiments provided here.

Another advantage of our learning algorithm is that it applies to many different probabilistic distributions. There are almost no restrictions to the range of statistical models that can be used for the unobserved independent variables. As examples, we will present in this paper different models that all fit into this same framework, but which correspond to different statistical contexts. They will be introduced in Section 2. The parametric estimation method, including the SAEM algorithm, is described in Section 3 and the reconstruction of hidden variables is discussed in Section 4. Experimental results with both synthetic and real data are presented in Section 5 where we also provide some comparison with the EM (when feasible), MCEM and FastICA, three of the most used algorithms.

2. Models.

We start with some general assumptions on the data, that will be made specific in the experiments. We assume that the observation is a set of vectors which take values in $\mathbb{R}^d$. Let $\mathbf{X}_1, \dots, \mathbf{X}_n$ be the training observations, which are assumed to be independent and identically distributed. We will denote by $\mathbf{X}$ a generic variable having the same distribution as the $\mathbf{X}_k$'s. The jth coordinate of $\mathbf{X}$ (resp., $\mathbf{X}_k$) will be denoted X^j (resp., X_k^j).

We assume that $\mathbf{X}$ can be generated in the form

$$\mathbf{X} = \boldsymbol{\mu}_0 + \sum_{j=1}^{p} \beta^j \mathbf{a}_j + \sigma \boldsymbol{\varepsilon}, \tag{2.1}$$

where $\boldsymbol{\mu}_0 \in \mathbb{R}^d$, $\mathbf{a}_j \in \mathbb{R}^d$ for all $j \in \{1, \dots, p\}$, $\boldsymbol{\varepsilon}$ is a standard d-dimensional Gaussian variable and $\beta^1, \dots, \beta^p$ are p independent scalar variables, the distribution of which being specified later. Let $\boldsymbol{\beta}$ denote the p-dimensional variable $\boldsymbol{\beta} = (\beta^1, \dots, \beta^p)$. To each observation $\mathbf{X}_k$ is therefore associated hidden realizations of $\boldsymbol{\beta}$ and $\boldsymbol{\varepsilon}$, which will be denoted $\boldsymbol{\beta}_k$ and $\boldsymbol{\varepsilon}_k$.

Denote $\mathbf{A} = (\mathbf{a}_1, \dots, \mathbf{a}_p)$. It is a d by p matrix and one of the parameters of the model. Another parameter is σ, which will be a scalar in our case (a diagonal matrix being also possible). Additional parameters will appear in specific models of $\boldsymbol{\beta}$ which are described in the following subsections. In some of these models, it will be convenient to build $\boldsymbol{\beta}$ as a function of new hidden variables, which will be denoted $\mathbf{Z}$.

The models that we describe are all identifiable, with the obvious restriction that $\mathbf{A}$ is identifiable up to a permutation and a sign change of its columns (the latter restriction being needed only when the distribution of $\boldsymbol{\beta}$ is symmetrical). This fact derives from identifiability theorems for factor analysis, like Theorem 10.3.1 in Kagan, Linnik and Rao (1973).

2.1. *Logistic distribution* (*Log-ICA*). We start with one of the most popular models, in which each β^j follows a logistic distribution with fixed parameter 1/2. The associated cumulative distribution function is $P(\beta^j \leq t) = 1/(1 + \exp(-2t))$.

For this model, the parameters to estimate are $\theta = (\mathbf{A}, \sigma^2, \boldsymbol{\mu}_0)$. Hidden variables are $\mathbf{Z} = \boldsymbol{\beta}$ and $\boldsymbol{\varepsilon}$. This is the model introduced in the original paper of Bell and Sejnowsky [Bell and Sejnowski (1995a)], and probably one of the most commonly used parametric models for ICA. One reason for this is that the logistic probability density function (p.d.f.) is easy to describe, smooth, with a shape similar to the Gaussian, but with heavier, exponential, tails. Note that, for identifiability reasons, one cannot use Gaussian distributions for the components.

2.2. *Laplacian distribution* (*Lap-ICA*). A simple variant is to take β^j to be Laplacian with density $e^{-|t|}/2$. The parameter still is $\theta = (\mathbf{A}, \sigma^2, \boldsymbol{\mu}_0)$. Hidden variables are $\mathbf{Z} = \boldsymbol{\beta}$ and $\boldsymbol{\varepsilon}$.

The resulting model is very similar to the previous one with similar exponential tails, with the noticeable difference that the Laplacian p.d.f. it is not differentiable in 0. One consequence of this is that it leads to sparse maximum a posteriori reconstruction of the hidden variables (cf. Section 4).

2.3. *Exponentially scaled Gaussian ICA* (*EG-ICA*). In this model, we let $\beta^j = s^j Y^j$ where $\mathbf{Y}$ is a standard Gaussian vector, $s^1, \dots, s^p$ are independent exponential random variables with parameter 1, also independent from Y and $\boldsymbol{\varepsilon}$. In this case, we can write

$$\mathbf{X} = \boldsymbol{\mu}_0 + \sum_{j=1}^{p} s^j Y^j \mathbf{a}_j + \boldsymbol{\sigma} \varepsilon. \tag{2.2}$$

Hidden variables are $\mathbf{Z} = (\mathbf{s}, \mathbf{Y})$ and $\boldsymbol{\varepsilon}$, and the parameter is $\theta = (\mathbf{A}, \sigma^2, \boldsymbol{\mu}_0)$.

The p.d.f. of $\beta = sY$ is given by $g(\beta) = \int_0^\infty \exp(-\frac{1}{2}y^2 - \frac{\beta}{y})\frac{dy}{y}$. It tends to infinity at $\beta = 0$, and has subexponential tails, because $\log[P(\beta^i > t)]$ is asymptotically proportional to $(-t^{2/3})$ (see the Appendix for details). It therefore allows for higher sparsity and more frequent large values of the component coefficients. This may help to overcome the variability in intensity which appears in medical images for examples. If we think in terms of source separation, the source has its own intensity and observations may require a large range of intensity around this "mean." It is also important to notice that, in spite of its increased complexity, this model can be implemented and learned as simply as the previous two using the algorithm that is proposed here.

2.4. *Independent factor analysis* (*IFA*). The IFA [Attias (1999), Miskin and MacKay (2000), Moulines, cois Cardoso and Gassiat (1997)] model is a special case of probabilistic ICA in which the distribution of each coordinate β^j is assumed to be a mixture of Gaussians. We will here use a restricted definition of the IFA model which will be consistent with the other distributions that we are considering in this paper, ensuring that the β^j's are independent with identical distribution, and that this distribution is symmetrical.

More precisely, we will introduce two new sets of hidden variables, the first one, denoted $(t^1, \dots, t^p)$, represents the class in the mixture model, and the second one, denoted $(b^1, \dots, b^p)$, is a random sign change for each component. Each t^j takes values in the finite set $\{0, 1, \dots, K\}$, with respective probabilities $w_0, \dots, w_K$, and b^j takes values ± 1 with probability $\frac{1}{2}$. We then let

$$\beta^j = b^j \sum_{k=1}^{p} m_k \delta_k(t^j) + Y^j,$$

where Y^j is standard Gaussian. In other terms, β^j is a mixture of $2K + 1$ Gaussians with unit variance, the first one being centered, and the following ones having means $m_1, -m_1, m_2, -m_2, \dots$.

The parameters of this model are therefore $\theta = (A, \sigma^2, (w_k, m_k)_{1 \le k \le K})$. Hidden variables are $\mathbf{Z} = (\boldsymbol{\beta}, \mathbf{b}, \mathbf{t})$. Note that, even if we use a simplified and symmetrized version of the model originally presented in Attias (1999), the stochastic approximation learning algorithm that will be designed in Section 3.2 immediately extends to the general case where the means depend on the index j.

2.5. *Bernoulli-censored Gaussian* (*BG-ICA*). In contrast with the logistic or Laplacian models for which coefficients vanish with probability zero, we now introduce a discrete switch which "turns them off" with positive probability. Here, we model the hidden variables as a Gaussian-distributed scale factor multiplied by a Bernoulli random variable. We therefore define $\beta^j = b^j Y^j$, using the same definition for $\mathbf{Y}$ as in Section 2.3 and letting b^j have a Bernoulli distribution with parameter $\alpha = P(b^j = 1)$. We assume that all variables $b^1, \dots, b^p, Y^1, \dots, Y^p, \varepsilon$ are independent. The complete model for $\mathbf{X}$ has the same structure as before, namely,

$$\mathbf{X} = \boldsymbol{\mu}_0 + \sum_{j=1}^{p} b^j Y^j \mathbf{a}_j + \sigma \boldsymbol{\varepsilon}. \tag{2.3}$$

Parameters in this case are $\theta = (\mathbf{A}, \sigma^2, \alpha, \boldsymbol{\mu}_0)$ and hidden variables are $\mathbf{Z} = (\mathbf{b}, \mathbf{Y})$ and $\boldsymbol{\varepsilon}$.

Using a censoring distribution in the decomposition is a very simple way to enforce sparsity in the resulting model. The population is characterized by a set of p vectors, however, each subject is only described by a subset of these p vectors corresponding to the active ones. The probability of the activation of the vectors is given by α. As α increases, the sparsity in the subject decomposition increases as well, whereas the dimension to explain the whole training set may remain equal to p. Censored models therefore arise naturally in situations where independent components are not expected to always contribute to the observed signals. This often occurs in spatial statistics, in situations for which observations combine basic components in space, not necessarily occurring all together. We will see an example of such a situation with handwritten digits where components can be interpreted as common parts of some of the digits, but not all, and therefore should not be selected every time. Functional magnetic resonance images (fMRIs), for which ICA methods have been extensively used [Calhoun et al. (2001), Calhoun, Adali and McGinty (2001), Makeig and Jung (1997)], are also important examples of similar situations. These three-dimensional images indicate active areas in the brain when a subject executes a specific cognitive task. People generally interpret components as basic processing units that interact in a complex task, but these units are not expected to be involved in every task for every subject. Similarly, genomic data, where a gene can activate a protein or not for particular patients, may fall into this context as well.

We now describe some possible variants within the class of censored models.

2.6. *Exponentially scaled Bernoulli-censored Gaussian* (*EBG-ICA*). Combining EG- and BG-ICA, so that a scale factor and a censoring variable intervene together, we get a new complete model for $\mathbf{X}$ given by

$$\mathbf{X} = \boldsymbol{\mu}_0 + \sum_{j=1}^{p} s^j b^j Y^j \mathbf{a}_j + \sigma \boldsymbol{\varepsilon}. \tag{2.4}$$

Since the exponential law has fixed variance, the parameters of interest are the same as in the BG-ICA model, that is, $\theta = (\mathbf{A}, \sigma^2, \alpha, \boldsymbol{\mu}_0)$. The hidden variables are $\mathbf{Z} = (\mathbf{s}, \mathbf{b}, \mathbf{Y})$ and $\boldsymbol{\varepsilon}$.

2.7. *Exponentially-scaled ternary distribution* (*ET-ICA*). The previous models include a switch which controls whether the component is present in the observation or not. One may want to further qualify this effect as "activating" or "inhibiting," which can be done by introducing a discrete model for $\mathbf{Y}$, each component taking values -1, 0 or 1. We define $\beta^j = s^j Y^j$, where $s^1, \dots, s^p$ are i.i.d. exponential variables with parameter 1. We let $\gamma = P(Y^j = -1) = P(Y^j = 1)$, providing a symmetric distribution for the components of $\mathbf{Y}$. As before, all hidden variables are assumed to be independent. The model is

$$\mathbf{X} = \boldsymbol{\mu}_0 + \sum_{j=1}^{p} s^j Y^j \mathbf{a}_j + \sigma \boldsymbol{\varepsilon}. \tag{2.5}$$

Hidden variables here are $\mathbf{Z} = (\mathbf{s}, \mathbf{Y})$ and $\boldsymbol{\varepsilon}$, the parameter being $\theta = (\mathbf{A}, \sigma^2, \gamma, \boldsymbol{\mu}_0)$.

The interpretation of the decomposition is that each component has a fixed effect, up to scale, which can be positive, negative or null. The model can therefore be seen as a variation of the Bernoulli–Gaussian where the effect can be a weighted inhibitor as well as a weighted activator. This allows selective appearance of decomposition vectors and therefore refines the characterization of the population.

This particular model makes all its sense when trying to model the generation of data with nonzero mean. Going back to our fMRI example, the mean image is more likely to be an active brain since all the patients are subject to the same cognitive task and the activation is always positive or zero. This will create some active areas in the mean brain ($\boldsymbol{\mu}_0$). However, as we already noticed, these areas can be active or not depending on the subject participating to the experiment. This can be modeled by a weighted activation or inhibition of its areas around the mean through the corresponding decomposition vectors. The decomposition vectors are still expected to correspond to the different active zones. This is what this model tries to capture. We will see in the experiments that it also applies to the handwritten digits.

2.8. *Single-scale ternary distribution* (*TE-ICA*). The previous model can be simplified by assuming that the exponential scale factor is shared by all the components, that is, we let $\beta^j = sY^j$, where s is exponential with parameter 1, and Y^j has the same ternary distribution as in the ET-ICA model. The decomposition now is

$$\mathbf{X} = \boldsymbol{\mu}_0 + s \sum_{j=1}^{p} Y^j \mathbf{a}_j + \sigma \varepsilon. \tag{2.6}$$

Hidden variables here are $(s, \mathbf{Y})$, the parameter being $\theta = (\mathbf{A}, \sigma^2, \gamma, \boldsymbol{\mu}_0)$. Notice that this model is not explicitly an ICA decomposition, since the components are only independent given the scale factor. Notice also that we assume that the scaling effect acts on the components, not on the observation noise which remains unchanged.

Probabilistic-ICA in general is obviously a very efficient representation for lossy compression of random variables, since, if the noise is neglected, and as soon as the parameters $\boldsymbol{\mu}_0$ and A are known, one only needs to know the realization of $\boldsymbol{\beta}$ (hopefully with $p \ll d$) to reconstruct an approximation to the signal. In the present model, the transmission of $\boldsymbol{\beta}$ only requires sending the scalar scale factor, s, and p ternary variables. If many components vanish (i.e., if γ is significantly smaller than $1/2$), compression is even more efficient.

In this model (and for the previous two also), the sparsity of the representation will obviously depend on the number of selected components, p, that we suppose given here. When p is too small, it is likely that the model will find that censoring does not help and take $\gamma = 1/2$ (or $\alpha = 1$ in the Bernoulli–Gaussian model). Adding more components in the model generally results in α and γ decreasing, enabling some components to be switched off. This effect is illustrated in Section 5.

Finally, let's remark that, although the results in Kagan, Linnik and Rao (1973) do not directly apply to this model (the components are not independent, since they share the same scale factor), they can be applied to the conditional distribution given the scale to prove identifiability (since the scale factor distribution is fixed).

2.9. *Playing with the average.* Clearly, all the previous models admit a centered submodel in which $\boldsymbol{\mu}_0 = 0$, which might be preferred in some cases. In this case ($\boldsymbol{\mu}_0 = 0$), it may be interesting to allow for some shift in the distribution of the components, replacing β^j by $\mu + \beta^j$ where μ is a one-dimensional parameter. This is therefore equivalent to modeling $\boldsymbol{\mu}_0 = \mathbf{A}\boldsymbol{\mu}$ where μ is a p-dimensional vector with all coordinates equal to μ. When dealing with scaled, or censored models, one can decide to apply the shift before or after censoring or scaling. For example, one can define a shifted Bernoulli–Gaussian model by replacing Y^i by $\mu + Y^j$ in Section 2.5, which results in shifting β^j only when it is not censored.

Another choice that can also be interesting is to model the signal with a random, scalar, offset (or AC component). One way to achieve this is to impose that one of the columns of the matrix $\mathbf{A}$ is the d-dimensional vector $(1, \dots, 1)^T$. In this case, it is natural to separate the distribution of the offset coefficient from the ones of other components, as customary in compression (the offset coefficient should not be censored, e.g.). A simple choice is to provide it with a logistic or Laplacian distribution. This is illustrated in the next model.

2.10. *Single-scale ternary distribution with offset* (*TEoff-ICA*). In this model the mean $\boldsymbol{\mu}_0$ is not a parameter and is not the same for all the observed vectors, so that this random effect (in opposition to the fixed effect it had in the previous models) now is a hidden variable. We furthermore assume that this random variable, denoted $\boldsymbol{\mu}$, takes the form $\boldsymbol{\mu} = (\mu, \dots, \mu) \in \mathbb{R}^d$ where μ is Laplacian. So μ

can be interpreted as an offset acting simultaneously on all coordinates of $\mathbf{X}$. This yields the following model:

$$\mathbf{X} = \boldsymbol{\mu} + s \sum_{j=1}^{p} Y^j \mathbf{a}_j + \sigma \varepsilon, \tag{2.7}$$

where s follows an exponential distribution with parameter 1 and Y^j are ternary variables with $\gamma = P(Y^j = -1) = P(Y^j = 1)$. The hidden variables are $(s, \mathbf{Y}, \mu)$ and the parameters (A, σ^2, γ).

Introducing observation-dependent offset and scale effects is useful when dealing with uncalibrated observations. This is typical, for example, with micro-array data, for which strong variations in calibration can occur among different patients. This is also common for signal and image processing, for which interpretation often needs to be performed in a way which is invariant, or robust, to offset or scale effects.

3. Maximum likelihood estimation.

3.1. *Notation.* The previous models are all built using simple generative relations $\mathbf{Z} \to \boldsymbol{\beta}$ and $(\boldsymbol{\beta}, \boldsymbol{\varepsilon}) \to \mathbf{X}$. Our goal here is to estimate the parameters that maximize the likelihood of the observation of n independent samples of $\mathbf{X}$ that we will denote $\mathbf{x}^{*n} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$.

Let $q_m(\mathbf{z}; \theta)$ denote the prior likelihood of the hidden (or missing) variable $\mathbf{Z}$ that generates $\boldsymbol{\beta}$. Denote by $q_c(\mathbf{x}|\mathbf{z}; \theta)$ the conditional distribution of $\mathbf{X}$ given $\mathbf{Z} = \mathbf{z}$ which is, in all our models, a Gaussian distribution centered at the ICA decomposition. The joint density is

$$q(\mathbf{x}, \mathbf{z}; \theta) = q_c(\mathbf{x}|\mathbf{z}; \theta) q_m(\mathbf{z}; \theta)$$

and the marginal distribution of $\mathbf{X}$ has density

$$q_{\text{obs}}(\mathbf{x}; \theta) = \int q_c(\mathbf{x}|\mathbf{z}; \theta) q_m(\mathbf{z}; \theta)\, d\mathbf{z}.$$

Our goal is to maximize the likelihood of the observations, namely, to find

$$\hat{\theta}_n = \arg\max_{\theta} q_{\text{obs}}^{*n}(\mathbf{x}^{*n}; \theta) \qquad \text{with } q_{\text{obs}}^{*n}(\mathbf{x}^{*n}; \theta) = \prod_{k=1}^{n} q_{\text{obs}}(\mathbf{x}_k; \theta). \tag{3.1}$$

3.2. *SAEM algorithm.* This problem can, in principle, be solved using the expectation–maximization (EM) algorithm. With the EM, a local maximum of the likelihood is computed recursively while replacing the missing variables with a conditional expectation. For each observation $\mathbf{x}_k$ and parameter θ, we define the conditional density of $\mathbf{Z}$ by

$$\nu_{k,\theta}(\mathbf{z}) = q(\mathbf{z}|\mathbf{X} = \mathbf{x}_k; \theta). \tag{3.2}$$

The EM algorithm iterates the following two steps, where t indexes the current iteration:

E: expectation. Compute $\ell_{t+1} : \theta \mapsto \ell_{t+1}(\theta) = \sum_{k=1}^{n} \mathbb{E}_{\nu_{k,\theta_t}}[\log q(\mathbf{x}_k, \mathbf{Z}; \theta)]$.
M: maximization. Set $\theta_{t+1} = \arg\max_{\theta \in \Theta} \ell_{t+1}(\theta)$.

The models we have discussed for ICA belong to the curved exponential family, in the sense that the joint distribution of hidden and observed variables for a given parameter can be expressed as

$$\log q(\mathbf{x}, \mathbf{z}; \theta) = \phi(\theta) \cdot \mathbf{S}(\mathbf{x}, \mathbf{z}) - \log C(\theta),$$

where $\mathbf{S}$ is a multidimensional sufficient statistic, ϕ is a fixed, vector-valued function of the parameters, C is a normalizing constant and the dot refers to the usual Euclidean dot product. This implies that

$$\ell_{t+1}(\theta) = \phi(\theta) \cdot \left(\sum_{k=1}^{n} \mathbb{E}_{\nu_{k,\theta_t}} \mathbf{S} \right) - n \log C(\theta).$$

Thus, the E-step only requires computing the conditional expectations of the sufficient statistic, and the M-step is equivalent to maximum likelihood for a fully observed model, with the empirical expectation of the sufficient statistic equal to

$$\bar{\mathbf{S}}_{t+1} = \frac{1}{n} \sum_{k=1}^{n} \mathbb{E}_{\nu_{k,\theta_t}} \mathbf{S}.$$

This is an important property (satisfied by our models) for the numerical feasibility of the EM algorithm.

However, this is not enough, since one must also be able to explicitly compute the conditional expectations. For several of our models, there is no closed form expression for the densities $\nu_{k,\theta}$. For others, like IFA, for which such an expression can be derived, its computational complexity is exponential in the number of components and rapidly becomes intractable (details are given in the Appendix).

A common way to overcome this difficulty is to approximate these conditional distributions by Dirac measures at their mode. The resulting algorithm is sometimes called EM-MAP or FAM-EM (for "Fast Approximation with Mode") [Allassonnière, Amit and Trouvé (2007), Allassonnière, Kuhn and Trouvé (2008)]. At each iteration of the algorithm, one computes the most likely hidden variables $\hat{\mathbf{z}}_k$, $1 \le k \le n$, with respect to the current parameters:

$$\hat{\mathbf{z}}_{t,k} = \arg\max_{\mathbf{z}} \big[\log\big(q(\mathbf{z} | \mathbf{X} = \mathbf{x}_k, \theta_t)\big)\big]. \tag{3.3}$$

The M-step then maximizes the likelihood for the "completed observations" $\mathbf{x}^{*n}$ and $\hat{\mathbf{z}}_{t,1}, \ldots, \hat{\mathbf{z}}_{t,n}$.

The statistical accuracy of this approximation is unclear, since it estimates a number of parameters that scales like the number of observations. Consistency of the obtained estimator when n goes to infinity cannot be proved in general. Some experimental evidence of asymptotic bias is demonstrated in Section 5 below.

In spite of these remarks, this approach (or approaches similar to it) is the most common choice for training probabilistic ICA models [Grimes and Rao (2005), Hyvarinen (1999), Olshausen and Field (1996a, 1996b)]. In the under-determined problem ($p \gg d$), this algorithm has also been implemented in Bremond, Moulines and Cardoso (1997).

Although the conditional distribution is not explicit, it is still possible (as we shall see later) to sample from it. The conditional expectation of the sufficient statistics ($\bar{\mathbf{S}}_{t+1}$) can therefore be approximated by Monte Carlo simulation, as proposed in Tanner (1996) and Wei and Tanner (1990) with the MCEM (Monte Carlo EM) algorithm. The resulting method, however, is heavily computational. Also, there is no guarantee that the errors resulting from the approximation to the E-step will cancel out to provide an estimator converging to a local maximum of the likelihood.

In this regard, a more interesting procedure, which has been proposed in Delyon, Lavielle and Moulines (1999), is a stochastic approximation of the EM algorithm, called SAEM. It replaces the E-step by a stochastic approximation step for the conditional likelihood (or, in practice, for the conditional expectation of the sufficient statistics), on which the M-step is based. More precisely, based on a sequence Δ_t of positive numbers decreasing to 0, the algorithm iterates the following two steps (assuming the tth iteration):

SAE step. For $k = 1, \dots, n$, sample a new hidden variable $\mathbf{z}_{t+1,k}$ according to the conditional distribution ν_{k,θ_t} and define

$$\ell_{t+1}(\theta) = \ell_t(\theta) + \Delta_t \left(\sum_{k=1}^{n} \log q(\mathbf{x}_k, \mathbf{z}_{t+1,k}; \theta) - \ell_t(\theta) \right).$$

M step. Set

$$\theta_{t+1} = \arg\max_{\theta \in \Theta} \ell_{t+1}(\theta).$$

For exponential families, the SAE step is more conveniently (and equivalently) replaced by an update of the estimation of the conditional expectation of the sufficient statistics, namely,

$$\bar{\mathbf{S}}_{t+1} = \bar{\mathbf{S}}_t + \Delta_t \left(\frac{1}{n} \sum_{k=1}^{n} \mathbf{S}(\mathbf{x}_k, \mathbf{z}_{t+1,k}) - \bar{\mathbf{S}}_t \right)$$

with

$$\ell_{t+1}(\theta) = \phi(\theta) \cdot \bar{\mathbf{S}}_{t+1} - \log C(\theta)$$

being maximized in the M-step. Note that this algorithm is fundamentally distinct from the SEM method [Celeux and Diebolt (1985)] in which the E-step directly defines $\ell_{t+1}(\theta) = \sum_{k=1}^{n} \log q(\mathbf{x}_k, \mathbf{z}_{t+1,k}; \theta)$.

A final refinement may be needed in the SAEM algorithm, when directly sampling from the posterior distribution is infeasible, or inefficient, but can be done using Markov Chain Monte Carlo (MCMC) methods. In this situation, there exists, for each θ and $\mathbf{x}$, a transition probability $z \mapsto \Pi_{\mathbf{x},\theta}(z, \cdot)$ such that the associated Markov chain is ergodic and has the posterior probability $q(\cdot|\mathbf{X} = \mathbf{x}; \theta)$ as stationary distribution. The corresponding variant of the SAEM (which we shall still call SAEM) replaces the direct sampling operation

$$\mathbf{z}_{t+1,k} \sim \nu_{k,\theta_t} = q(\cdot|\mathbf{X} = \mathbf{x}_k, \theta_t)$$

by a single Markov chain step

$$\mathbf{z}_{t+1,k} \sim \Pi_{\mathbf{x}_k,\theta_t}(\mathbf{z}_{t,k}, \cdot).$$

This procedure has been introduced and proved convergent for bounded missing data in Kuhn and Lavielle (2004). This result has been generalized to unbounded hidden random variables in Allassonnière, Kuhn and Trouvé (2010).

To ensure the convergence of this algorithm in the noncompact case (which is our case in the models above), one needs, in principle, to introduce a truncation on random boundaries as in Allassonnière, Kuhn and Trouvé (2010). This would add a new operation between the stochastic approximation and the maximization steps, with the following *truncation step*. Let $\mathcal{S}$ be the range of the sufficient statistic, S. Let $(\mathcal{K}_q)_{q\geq 0}$ be an increasing sequence of compact subsets of $\mathcal{S}$ such as $\bigcup_{q\geq 0} \mathcal{K}_q = \mathcal{S}$ and $\mathcal{K}_q \subset \operatorname{int}(\mathcal{K}_{q+1}), \forall q \geq 0$. Let $(\delta_t)_t$ be a decreasing sequence of positive numbers. If $\bar{S}_{t+1}$ wanders out of $\mathcal{K}_{t+1}$ or if $|\bar{S}_{t+1} - \bar{S}_t| \geq \delta_t$, then the algorithm is reinitialized in a fixed compact set.

More details can be found in Andrieu, Moulines and Priouret (2005) and Allassonnière, Kuhn and Trouvé (2010). In practice, however, our algorithms work properly without this technical hedge.

3.3. *Application to our models.* To complete the description of the SAEM algorithm for a given model, it remains to make explicit (i) the specific form of the sufficient statistic $\mathbf{S}$; (ii) the corresponding maximum likelihood estimate for complete observations; and (iii) the transition kernel for the MCMC simulation. Formulae for (i) and (ii) are provided in the Appendix for the ICA models we have described here. For (iii), we have used a Metropolis–Hastings procedure, looping over the components (sometimes called "Metropolis–Hastings within Gibbs Sampling") that we now describe. This is for a fixed observation $\mathbf{x}_k$ and parameter θ, although we do not let them appear in the notation. So we let $\nu = \nu_{k,\theta}$ be the probability that needs to be sampled from.

In the Metropolis–Hastings procedure, one must first specify a candidate transition probability $\rho(\mathbf{z}, \tilde{\mathbf{z}})$. A Markov chain $(\mathbf{Z}_t, t = 0, 1, \dots)$ can then be defined by the two iteration steps, given $\mathbf{Z}_t$:

(1) Sample $\mathbf{z}$ from $\rho(\mathbf{Z}_t, \cdot)$.

(2) Compute the ratio

$$r(\mathbf{Z}_t, \mathbf{z}) = \frac{\nu(\mathbf{z})\rho(\mathbf{z}, \mathbf{Z}_t)}{\nu(\mathbf{Z}_t)\rho(\mathbf{Z}_t, \mathbf{z})}$$

and set $\mathbf{Z}_{t+1} = \mathbf{z}$ with probability $\min(1, r)$ and $\mathbf{Z}_{t+1} = \mathbf{Z}_t$ otherwise.

An interesting special case is when ρ corresponds to a Gibbs sampling procedure for the prior distribution, $q_m(\mathbf{z}; \theta)$. Given the current simulation $\mathbf{z}$, one randomly selects one component z^j and generates $\tilde{\mathbf{z}}$ by only changing z^j, replacing it by $\tilde{z}^j$ sampled from the conditional distribution $q_m(\tilde{z}^j | z^i, i \neq j; \theta)$. In this case, it is easy to see that the ratio r is then given by

$$r(\tilde{\mathbf{z}}, \mathbf{z}) = \frac{q(\mathbf{x}_k | \tilde{\mathbf{z}})}{q(\mathbf{x}_k | \mathbf{z})}.$$

The Markov kernel is then built by successively applying the previous kernel to each component.

Our implementation follows this procedure whenever the current set of parameters leads to an irreducible transition probability ρ. This is always true, except for the censored models, in which parameters $\alpha \in \{0, 1\}$ or $\gamma \in \{0, \frac{1}{2}\}$ are degenerate and must be replaced by some fixed values α_0 and γ_0 in the definition of ρ.

4. Reconstruction.

Assuming that the parameters in the model are known or have been estimated, the reconstruction problem consists in estimating the hidden coefficients of the independent components, $\hat{\beta} \in \mathbb{R}^p$, based on a new observation of $\mathbf{x} \in \mathbb{R}^d$. As noticed in the Introduction, this is a separate problem. Estimating model parameters is based on the likelihood of the observation, which integrates out hidden variables. In contrast, reconstructing hidden decomposition vectors from data is typically done by minimizing a chosen loss function for a fixed choice of model parameters, and is based on the posterior likelihood (proportional to the complete likelihood). Even if this does not constitute our main focus here, we briefly describe in this section how the MAP estimator, based on maximizing the complete likelihood, can be achieved using the models presented in this paper.

Reconstruction with probabilistic ICA models is not as straightforward as with complete ICA, for which the operation reduces to solving a linear system. A natural approach is maximum likelihood, that is, (with our notation) find $\hat{\mathbf{z}} = \arg\max_{\mathbf{z}} \phi(\theta) \cdot S(\mathbf{x}, \mathbf{z})$ and deduce $\hat{\boldsymbol{\beta}}$ from it.

This maximization is not explicit, although simpler for our first two models. Indeed, for Log-ICA, this requires minimizing

$$\frac{1}{2\sigma^2} |\mathbf{x} - \mathbf{A}\boldsymbol{\beta}|^2 + 2 \sum_{j=1}^{p} \log(e^{\beta^j} + e^{-\beta^j}).$$

(We take $\boldsymbol{\mu}_0 = 0$ in this section, replacing, if needed, $\mathbf{x}$ by $\mathbf{x} - \boldsymbol{\mu}_0$.)

The Laplacian case, Lap-ICA, gives

$$\frac{1}{2\sigma^2}|\mathbf{x} - \mathbf{A}\boldsymbol{\beta}|^2 + \sum_{j=1}^{p} |\beta^j|.$$

Both cases can be solved efficiently by convex programming. The Laplacian case is similar (up to the absence of normalization of the columns of **A**) to the Lasso regression algorithm [Tibshirani (1996)], and can be minimized using an incremental procedure on the set of vanishing β^j's [Efron et al. (2004)].

The other models also involve some form of quadratic integer programming, the general solution of which being NP-complete. When dealing with large numbers of components, one must use generally suboptimal optimization strategies (including local searches) that have been developed for this context [see Li and Sun (2006), e.g.].

The EG-ICA problem requires minimizing

$$\frac{1}{2\sigma^2}\left|\mathbf{x} - \sum_{j=1}^{p} s^j y^j \mathbf{a}_j\right|^2 + \sum_{j=1}^{p} s^j + \frac{1}{2}\sum_{j=1}^{p}(y^j - \mu)^2$$

with $s^1, \dots, s^p \geq 0$. This is not convex, but one can use in this context an alternate minimization procedure, minimizing in **y** with fixed **s** and in **s** with fixed **y**. The first problem is a straightforward least squares and the second requires quadratic programming.

The symmetrized IFA model leads to minimize

$$\frac{1}{2\sigma^2}|\mathbf{x} - \mathbf{A}\boldsymbol{\beta}|^2 + \frac{1}{2}\sum_{j=1}^{p}(\beta^j - b^j m_{t_j})^2 + \sum_{j=1}^{p} \log w_{t_j}$$

with respect to $\boldsymbol{\beta}$, the unobserved configuration of labels **t**, and the sign change **b**. When labels and signs are given, the problem is quadratic in $\boldsymbol{\beta}$. Given $\boldsymbol{\beta}$ and **t**, the optimal **b** is explicit, and for fixed $\boldsymbol{\beta}$ and **b**, the search for an optimal **t** reduces to a quadratic integer programming problem. For small dimensions, it is possible to make an exhaustive search of all $(2K + 1)^p$ possible configurations of labels and signs.

For the BG-ICA, we must minimize

$$\frac{1}{2\sigma^2}\left|\mathbf{x} - \sum_{j=1}^{p} b^j y^j \mathbf{a}_j\right|^2 + \rho\sum_{j=1}^{p} b^j + \frac{1}{2}\sum_{j=1}^{p}(y^j - \mu)^2$$

with $\rho = \log((1 - \alpha)/\alpha)$ and $b^j \in \{0, 1\}$. The minimization in **b** is a $(0, 1)$-quadratic programming problem, an exhaustive search being feasible for small p. Given **b**, the optimal **y** is provided by least squares.

Concerning the EBG-ICA, we must minimize

$$\frac{1}{2\sigma^2}\left|\mathbf{x}-\sum_{j=1}^{p}s^j b^j y^j \mathbf{a}_j\right|^2+\sum_{j=1}^{p}s^j+\rho\sum_{j=1}^{p}b^j+\frac{1}{2}\sum_{j=1}^{p}(y^j-\mu)^2$$

with $\rho=\log((1-\alpha)/\alpha)$, $s^1,\dots,s^p>0$ and $b^j\in\{0,1\}$. This is again a (0, 1)-quadratic programming problem in $\mathbf{b}$ and, given $\mathbf{b}$, the optimal $\mathbf{y}$ and $\mathbf{s}$ are computed similarly to the EG-ICA model.

With ET-ICA, the objective function is

$$\frac{1}{2\sigma^2}\left|\mathbf{x}-\sum_{j=1}^{p}s^j y^j \mathbf{a}_j\right|^2+\sum_{j=1}^{p}s^j+\rho\sum_{j=1}^{p}|y^j|$$

with $\rho=\log((1-2\gamma)/2\gamma)$, $y^1,\dots,y^p\in\{-1,0,1\}$ and $s^1,\dots,s^p>0$. This is a quadratic integer programming in $\mathbf{y}$, with a complexity of 3^p for an exhaustive search. Given $\mathbf{y}$, computing $\mathbf{s}$ is a standard quadratic programming problem.

The TE-ICA problem, requiring to minimize

$$\frac{1}{2\sigma^2}\left|\mathbf{x}-s\sum_{j=1}^{p}y^j \mathbf{a}_j\right|^2+s+\rho\sum_{j=1}^{p}|y^j|$$

is slightly simpler since, in this case, the computation of $s\geq 0$ given y is straightforward.

The TEoff-ICA model involves a third hidden variable μ. This leads to the following objective function to minimize both in s, $\mathbf{y}$ and μ:

$$\frac{1}{2\sigma^2}\left|\mathbf{x}-\boldsymbol{\mu}-s\sum_{j=1}^{p}y^j \mathbf{a}_j\right|^2+s+\rho\sum_{j=1}^{p}|y^j|$$

with $\boldsymbol{\mu}=(\mu,\dots,\mu)\in\mathbb{R}^d$, and $s>0$. Given $\boldsymbol{\mu}$, the minimization with respect to s and $\mathbf{y}$ is done as in the previous TE-ICA model. The minimization over μ has a closed form:

$$\mu=\frac{1}{d}\sum_{i=1}^{d}\left(\mathbf{x}^j-s\sum_{j=1}^{p}y^j\mathbf{a}_{i,j}\right).$$

5. Experiments.

5.1. *Synthetic image data.*

5.1.1. *Data set.* We first provide an experimental analysis using synthetic data, which allows us to work in a controlled environment with a known ground

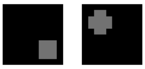


FIG. 1. *Two decomposition images used for sampling synthetic data.*

truth. In this setting, we assume that the true distribution is the Bernoulli–Gaussian (BG) model, with two components ($p = 2$). The probability α of each component to be "on" is set to 0.8. We run experiments based on 30, 50 or 100 observations, and vary the standard deviation of the noise using $\sigma = 0.1, 0.5, 0.8, 1.5$.

The components are represented as two-dimensional binary images (grey levels being either 0 or 1). The first one is a black image (grey level equals 0) with a white cross (grey level 1) in the top left corner. The second one has a white square (same grey level) in the bottom right corner. These two images are shown in Figure 1. Figure 2 presents 30 images sampled from this model with the different noise levels. The training sets were sampled once and used in all the comparative experiments below. We used a fixed color map for all figures to allow for comparisons across experiments (this explains why the patterns in Figure 1 appear as grey instead of white).

5.1.2. *Interpretation of the results.* We have compared the following estimation strategies: (1) FAM-EM algorithm [Grimes and Rao (2005), Olshausen and Field (1996a), Tenenbaum and Freeman (2002)] (which maximizes the likelihood with respect to parameters and hidden variables together) with the Log-ICA model (Logistic distribution); (2) SAEM with the same Log-ICA model; (3) SAEM for the IFA model, and (4) EM with the IFA model [Attias (1999), Welling and Weber (2001)]; (5) SAEM for the true BG-ICA model; (6) finally, we also ran a standard ICA decomposition using fast-ICA [Hyvärinen and Oja (1997)] with a requirement of computing only two components (with a preliminary dimension reduction based on PCA). Models (3) and (4) are theoretically equivalent, and our experiments evaluate how they differ numerically. We reemphasize that the EM algo-

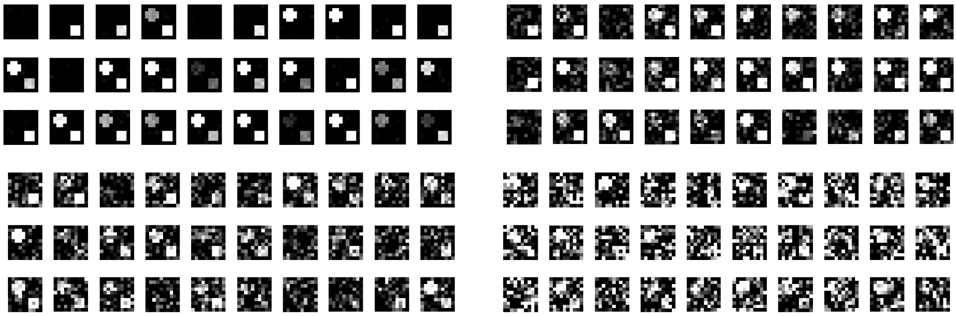


FIG. 2. *Samples of the training sets used for synthetic data with different level of noise.* $\sigma = 0.1$, 0.5, 0.8, 1.5 *are upper left, upper right, lower left and lower right, respectively.*

rithm for the IFA model is only feasible for a reasonably small number of components, p, and number of mixtures, K (with a complexity in K^p), whereas this limitation does not apply to the SAEM algorithm (see the Appendix for more details). For other alternative approaches to the EM for the IFA model (including the use of the FAM-EM strategy), see Brandt Petersen and Winther (2005), Côme et al. (2008), Grimes, Shon and Rao (2003), Valpola Lappalainen and Pajunen (2000), Varoquaux et al. (2010). The fast-ICA algorithm used in (6) is nonparametric (and maximizes an approximation of the negentropy of the model).

We also notice that (1), which requires minimizing in A, σ^2 and $\mathbf{b}$, is ill-posed because a transformation $(A, \mathbf{b}) \to (\lambda A, \mathbf{b}/\lambda)$ always decreases the likelihood when $\lambda > 1$, which implies that the optimal A is unbounded. To address this, one solution is to use a prior distribution for A, or enforce some normalization. We chose the latter option, enforcing the empirical mean square of all $\mathbf{b}$'s to be equal to $\log 2$ as implied by the logistic distribution.

Table 1 provides mean-square errors for the estimation of A based on different models and algorithms, and for different noise levels and sample size. Each error is computed from 50 repeats of the full experiment (sampling from the true model followed by estimation). The mean square error (MSE) is defined by

$$\mathrm{MSE} = \frac{1}{|\Lambda|} \sum_{x \in \Lambda} \big[\big(A_{\mathrm{est}}(x,1) - A_{\mathrm{true}}(x,1)\big)^2 + \big(A_{\mathrm{est}}(x,2) - A_{\mathrm{true}}(x,2)\big)^2\big],$$

where Λ is the grid of pixels, $|\Lambda|$ its cardinality, A_{est} is the estimated decomposition matrix and A_{true} the true one (up to a permutation and a change of sign). The lack of monotonicity in mean squared errors with respect to σ may come from the small number of simulations which are averaged here. The estimation is to proceed 50 times but a larger number of simulations would solve the problem.

We also evaluated the accuracy of the estimation of σ^2. The results are presented in Table 2. A surprising result is that σ^2 is always well estimated even when the decomposition vectors are not. This is an important observation which indicates

TABLE 1
Mean-square estimation error for various combinations of algorithms and models based on estimations based on 30/100 *samples and several noise levels. Each mean-square error is an average over* 50 *independent repeats*

	30 images per training set				**100 images per training set**			
Algo/model	$\sigma = 0.1$	$\sigma = 0.5$	$\sigma = 0.8$	$\sigma = 1.5$	$\sigma = 0.1$	$\sigma = 0.5$	$\sigma = 0.8$	$\sigma = 1.5$
FAM-EM/Log	0.55	0.49	0.51	0.82	0.52	0.47	0.46	0.62
SAEM/Log	0.05	0.06	0.10	0.26	0.03	0.06	0.06	0.11
SAEM/IFA	0.19	0.18	0.16	0.20	0.16	0.16	0.09	0.10
EM/IFA	0.05	0.04	0.06	0.15	0.05	0.03	0.03	0.06
SAEM/BG	0.09	0.13	0.16	0.6	0.07	0.07	0.05	0.25

TABLE 2

Estimated noise variance with the different models and the two different algorithms for 30, 50 *and* 100 *images in the training set. These variances correspond to the estimated decomposition vectors presented in Figure* 3

	Algo/model					
	True σ^2	**FAM-EM/Log**	**SAEM/Log**	**EM/IFA**	**SAEM/IFA**	**SAEM/BG**
30 images in the training set	0.001	0.0088	0.0086	0.0097	0.0089	0.0087
	0.2500	0.2253	0.2224	0.2240	0.2410	0.2226
	0.6400	0.5685	0.5577	0.5534	0.6092	0.5569
	2.2500	2.0375	1.9978	2.1199	2.0735	2.0009
50 images in the training set	0.001	0.0095	0.0092	0.0095	0.0094	0.0092
	0.2500	0.2400	0.2399	0.2363	0.2524	0.2399
	0.6400	0.5831	0.5798	0.6381	0.6429	0.5795
	2.2500	2.1544	2.1377	2.2061	2.2112	2.1366
100 images in the training set	0.001	0.0176	0.0097	0.0095	0.0098	0.0097
	0.2500	0.2432	0.2459	0.2455	0.2564	0.2456
	0.6400	0.6225	0.6282	0.6336	0.6388	0.6280
	2.2500	2.1268	2.1479	2.1767	2.1970	2.1490

that one should not evaluate the final convergence of any of these algorithms based on the convergence of σ^2 only.

A visual illustration of these results is provided in Figure 3, in which a single (typical) experiment is displayed for each noise level and sample size. The algorithms that maximize the likelihood of the observed data (SAEM, MCEM and EM for the IFA) all provide results that are consistent with the ground truth, even when the model used for the estimation differs from the true one. This statement does not apply to the FAM-EM algorithm (which maximizes the likelihood with respect to parameters and hidden variables together), or to FastICA, which degrade significantly when the noise is high. We also experienced numerical failures when running the publicly available software with high noise (we had, in fact, to resample a new 100-image training set to be able to present results from this method).

Since these two algorithms both rely on Monte Carlo sampling, we have compared the performances of our SAEM with a Log-ICA model and of the Monte Carlo (MC) EM algorithm. The expectation step of the EM is replaced by an approximation of the expected value of the sufficient statistics using a Monte Carlo sum. Therefore, at each iteration of the algorithm, MCEM requires repeated samples from the posterior distribution of the hidden variables given the observations. Larger samples yield a better approximation and generally result in fewer EM iterations to achieve convergence. Of course, this also implies a computational cost per iteration which grows linearly in the sample size. Notice also that we cannot generate independent samples from the posterior distribution, but only Markov

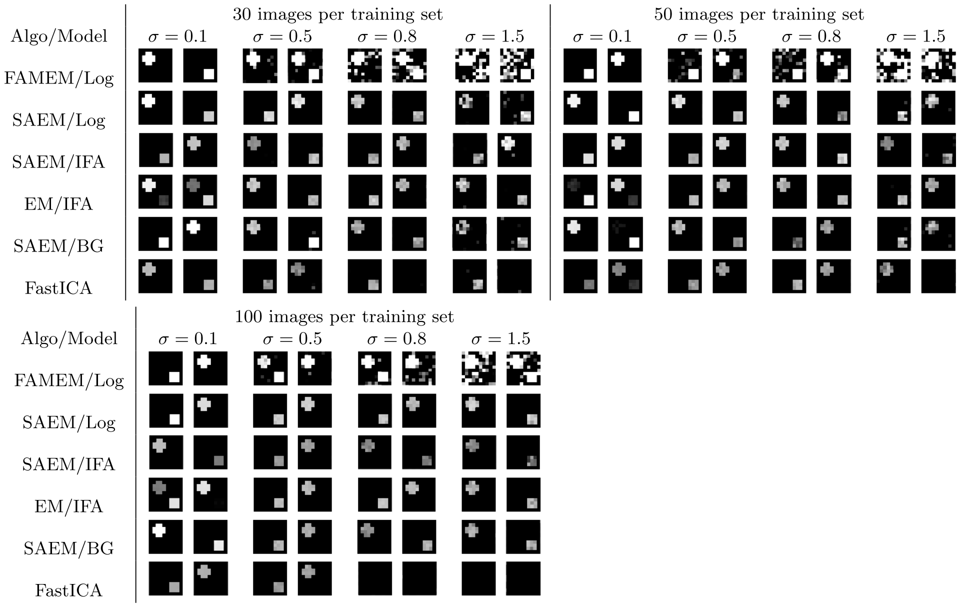


FIG. 3. *Estimated decomposition images with different models and algorithm. The estimation becomes less and less satisfactory as the noise variance increases. It is even more pregnant for the FAMEM and FastICA algorithms for which increasing the number of observation does not address this problem. The other models estimated using our algorithm provide similar results and the noise does not drastically affect the estimation.*

chain samples resulting from the MCMC sampler described in Section 3.3. These samples are therefore correlated and only asymptotically sample from the posterior distribution. A comparison of the output of this algorithm and of the proposed (MCMC-)SAEM is displayed in Figure 4. We ran 1,000 iterations, using 10 and 30 samples in each Monte Carlo approximation, and the estimation is based on 100 observations. The results are similar, whereas the time cost is about the number of samples (10 or 30) times longer for the MCEM than for SAEM. Decreasing the number of samples accelerates the estimation but degrades the estimations, in particular, when the noise level is high.

We have also made a broad comparison of the required computation time associated to each algorithm. One must remember, when interpreting these results, that each algorithm optimizes its own objective function, and only the ones of the EM/IFA and SAEM/IFA coincide. While the objective functions of the SAEM models can be considered as similar, the one associated with the FAM-EM is quite different, and the comparison must be done with this in mind.

Another difficulty in computing these numbers is that the true solution (maximum likelihood, or mode) is unknown, and even if it were known, all methods are prone to converge to a local maximum and never get close to it. Because of this, we have used an empirical definition of the convergence time as the first time at which

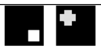
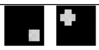
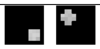
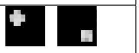
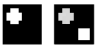
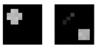
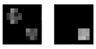
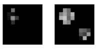
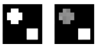
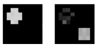
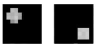
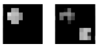
	100 training images			
Algo/Model	$\sigma = 0.1$	$\sigma = 0.5$	$\sigma = 0.8$	$\sigma = 1.5$
SAEM/Log				
MCEM w/ 10 samp./Log				
MCEM w/ 30 samp./Log				
	MSE / Time cost in seconds			
SAEM/Log	0.0926 / 13.51	0.0515 / 13.43	0.0646 / 13.29	0.0459 / 14.17
MCEM w/ 10 samp./Log	0.1558 / 124.01	0.0343 / 125.82	0.0511 / 125.13	0.0847 / 125.18
MCEM w/ 30 samp./Log	0.2189 / 369.65	0.0643 / 370.65	0.0339 / 370.15	0.0614 / 372.65

FIG. 4. *Comparison between MCEM and SAEM with the Log-ICA model. The top images show the decomposition vectors estimated with either model and for different numbers of Monte Carlo samples used to approximate the expectation in the MCEM. The table presents the mean square error* (*MSE*) *and the time cost of each estimation. The results look very similar* (*except for low noise variance where the MCEM seems to behave like the FAM-EM*), *while the time cost of the MCEM increases linearly in the sample size.*

the maximal subsequent variation of the current solution is less than 1/1,000 of what it was initially. More precisely, if $A(t)$ is the estimated component matrix at step t, and $d(t) = \max_{t' \geq t} |A(t') - A(t_{\max})|$, the convergence time defined as

$$t_{\text{conv}} = \min\{t : d(t) \leq d(1)/1{,}000\} \tag{5.1}$$

($t_{\max}$ being the maximal number of iterations, equal to 5,000 in our experiments).

These results are summarized in Table 3. As expected, the times per iteration of the SAEM-based methods are much smaller than with other approaches. This is only partially compensated by an increased number of iterations in order to achieve convergence. Note that, in this table, the number of steps to convergence for the MCEM is close to $t_{\max} = 5{,}000$, which indicates that (5.1) has not been satisfied before the maximal number of iterations. Note also that the studied model, with

TABLE 3
Comparison of computation costs. Second column: *average time*, *in seconds*, *for* 1,000 *iterations of each algorithm. Columns* 3 *to* 6: *average number of iterations to achieve convergence*

		Number of iterations to convergence			
Algo/model	**Time for 1,000 iterations**	$\boldsymbol{\sigma = 0.1}$	$\boldsymbol{\sigma = 0.5}$	$\boldsymbol{\sigma = 0.8}$	$\boldsymbol{\sigma = 1.5}$
FAM-EM/Log	206 s	3,700	3,000	1,300	600
MCEM/Log	140 s	(4,800)	(4,900)	(5,000)	(5,000)
SAEM/Log	14 s	230	980	1,720	2,390
EM/IFA	1,600 s	4,400	350	140	50
SAEM/IFA	33 s	1,510	2,240	2,920	3,670
SAEM/BG	26 s	530	1,210	1,900	3,660

two independent components, is the most favorable for the EM/IFA algorithm, which would become intractable with a higher number of components.

Another interesting (and difficult to explain) observation from this table is that the deterministic algorithms (FAM-EM and EM for the IFA) seem to require fewer iterations at high noise level, while the trend is opposite for the stochastic methods. A final remark is that the fastICA algorithm is much faster than any of these methods when run after reducing the model dimension using PCA.

5.2. *Effect of the number of estimated components.* We now illustrate, with a different model, how, for censored models, the estimation of the censoring coefficient evolves with the number of components. In this experiment we have generated 1,000 samples of a shifted Bernoulli–Gaussian model (see Section 2.9) with 8 components (the components being represented as indicators of 8 nonoverlapping intervals). The true value of α is 0.5, and we took $\mu = 2$. In Figure 7 we plot the value of the estimated α as a function of the number of components in the model, p. We can see that this value seems to decrease to zero, at a rate which is, however, not linear in $1/p$. The expected number of nonzero components grows from 2 for $p = 2$, to 8 when $p = 8$ (correct value—pointed in red in the plot), to about 10 when $p = 50$. The estimated components for $p = 6$, 8 and 15 are plotted in Figures 5 and 6. This illustrates the effect of under-dimensioning the model, in which some of the estimated components must share some of the features of several true components (pointed by arrows), and of over-dimensioning, in which some of the estimates components are essentially noise (clearly indicating overfitting of the data—in red rectangles), while some other estimated components,

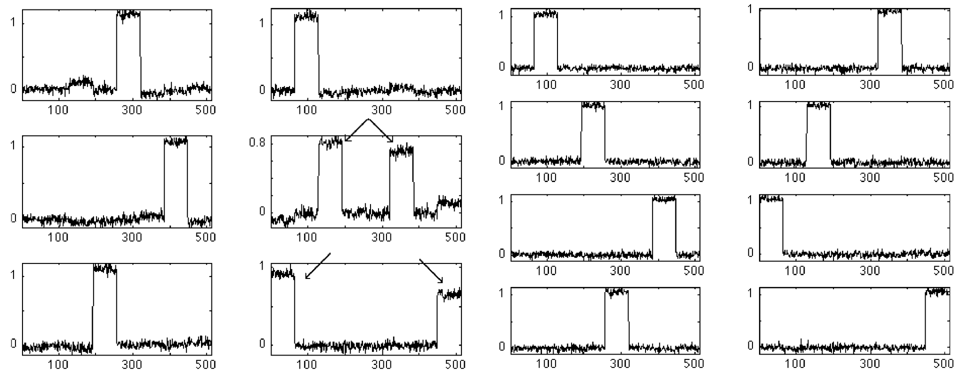


FIG. 5. *Estimated components with probabilistic ICA. The sample contains* 1,000 *signals generated by a shifted Bernoulli–Gaussian model* (*see Section* 2.9) *with* 8 *components* (*the components being represented as indicators of* 8 *nonoverlapping intervals*). *The true value of* α *is* 0.5, *and we took* $\mu = 2$. *Left*: *components estimated with* $p = 6$. *Right*: *components estimated with* $p = 8$. *When estimating only* 6 *components*, *two sources appear in the same component which will make them always appear together with the same weight. This is what can be seen pointed by the arrows. However, when estimating* 8 *components*, *the* 8 *sources are recovered.*

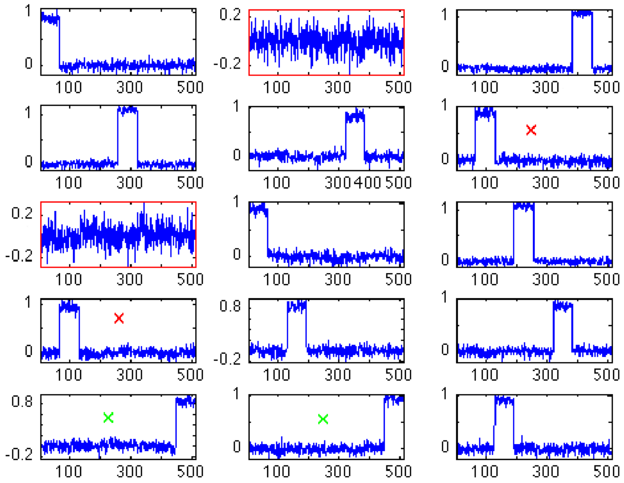


FIG. 6. *Estimated components with probabilistic ICA. The sample contains* 1,000 *signals generated by a shifted Bernoulli–Gaussian model* (*see Section* 2.9) *with* 8 *components* (*the components being represented as indicators of* 8 *nonoverlapping intervals*). *The true value of α is* 0.5, *and we took $\mu = 2$. Components estimated with $p = 15$. We can see that* 15 *is too many since among the* 15 *sources found*, *we can recognize some noise* (*squared by red rectangles*) *and repeated components* (*red crosses and green crosses are similar with each other*).

which correspond to true ones, are essentially repeated (twice for the components marked with red and green crosses). Components are correctly estimated when the estimated model coincides with the true model ($p = 8$).

Although we are not addressing the estimation of the number of components in this paper, these results clearly indicate that this issue is important. We refer the reader to standard approaches in this context, based on penalizing model complexity, using, for example, the Akaike Information Criterion (AIC) [Akaike (2003)] or the Bayesian Information Criterion (BIC) [Maugis, Celeux and Martin-Magniette

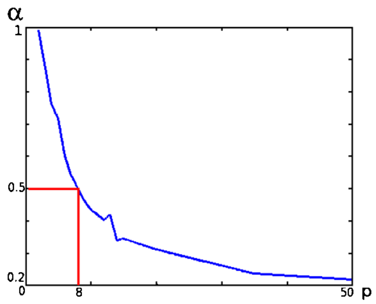


FIG. 7. *Estimated component activation probability* (α) *as a function of the model size for a Bernoulli Gaussian model estimated on the* 1,000 *signals of a shifted Bernoulli–Gaussian model. Ground truth is $p = 8$ and $\alpha = 0.5$* (*red point*).

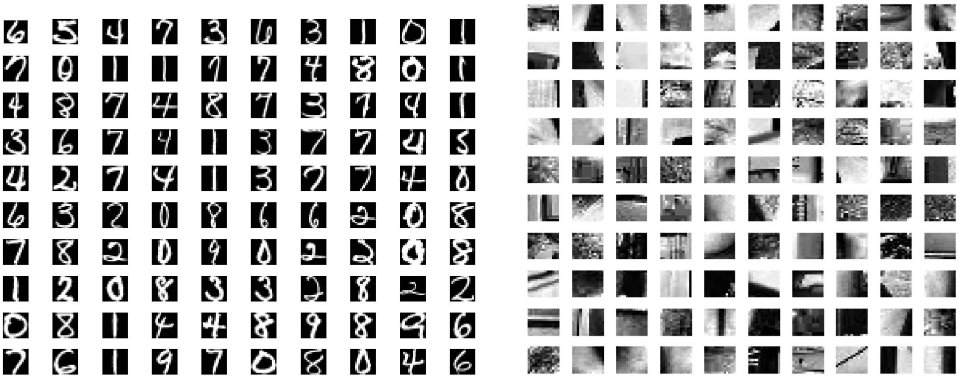


FIG. 8. 100 *images randomly extracted from the USPS database* (*left*) *and from the face category in the Caltech101 data set* (*right*).

(2009), Schwarz (1978)]. Using a Bayes prior on parameters would be possible, too, with a straightforward adaptation of the SAEM algorithm.

5.3. *Handwritten digits*. We now test our algorithms on some 2D real images. The first training set we use is the USPS database, which contains 7,291 grey-level images of size 16 × 16. We used the whole database as a training set and computed 20 decomposition vectors. Some images from this data set are presented in Figure 8(left).

The different decomposition vectors and the estimated means (when it is a parameter) are presented in Figure 9. Each 10 by 2 set of 20 images on the right column corresponds to one run of the algorithm for a given model (selecting the most representative results).

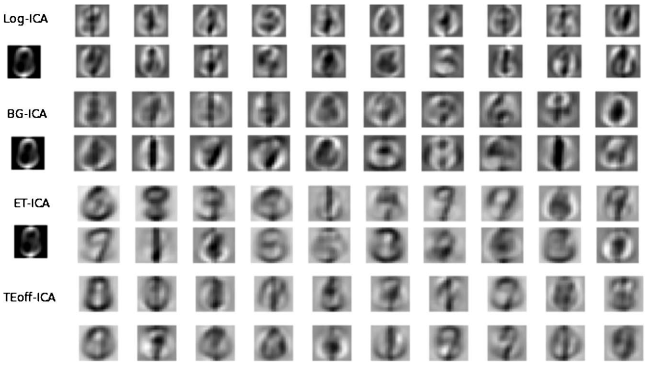


FIG. 9. *Results of the independent component estimation on the USPS database using four selected models. The training set is composed of* 7,291 *images containing the* 10 *digits randomly spread. Left column*: *mean image* μ_0. *Right column*: 20 *estimated decomposition vectors*. (*See Figure* 1 *in the supplementary file* [*Allassonnière and Younes* (2011)] *for a larger image*.)

Interestingly, the results highlight the advantage of the censored models compared to the continuous ones in such situations. Modeling component coefficients that can vanish with positive probability (such in BG and ET-ICA) enables to have decompositions which do not involve vectors shared by all the training sample. Considering a data set such as USPS, one image in one class is not easily expressed as a mixture of images from other classes. Therefore, it is not appropriate to express it as a linear combination of all the decomposition vectors with nonzero coefficients. This means that we expect the decomposition vectors to be separated digits and appearing only for samples belonging to the corresponding class. This is what we see with the censored models (Figure 9, lines 2 to 4), many decomposition vectors represent well-formed digits, whereas decomposition vectors for other models (Figure 9, line 1) mix several digits more often to be able to cancel the nonexpected features. These binary or ternary models seem to be very adequate in such situations.

Note that the USPS data set does not have the same number of images of each digit. There are about twice as many 0's or 1's as other digits. This fact explains the "bias" one can see on the mean, on which the shape of the zero is noticeable. In all experiments, the trace of each digit can be (more or less easily) detected in at least one of the components, at the exception of digit 2. This is probably due to the large geometrical variability of the 2's, which is much higher than other digits (changes of topology-loop or not, changes in global shape) and therefore difficult to capture.

5.4. *Face images.* We have run a similar experiment on a data set of face images (extracted from the Caltech101 data set). Each of these images has been decomposed into patches of size 13 × 13, with some of them presented in Figure 8(right). The resulting database contains 499,697 small images and we estimated 20 decomposition vectors. Results are presented in Figure 10. The patterns which emerge from the estimations are quite similar from one model to another: vertical, horizontal and diagonal separation of the image into black and white, blobs, regular texture like a regular mesh, etc.

We also ran the same estimation with two of the previous models looking for 100 decomposition vectors. The results are presented in Figure 11. We selected the Log and BG-ICA since one has a continuous density and the second has a semi-discrete one. The results are rather different. While the Log-ICA model tends to capture some textures, the BG-ICA captures some shapes. In this example, as well as with the digit case, the sparsity of the decomposition makes sense and plays an important role. This database is composed of discrete features which can hardly be approximated by a linear combination of continuous patterns. The models generating sparse representations again seems to be better adapted to this kind of data.

5.5. *Anatomical surfaces.* We finally consider a data set containing a family of 101 hippocampus surfaces that have been registered to a fixed template using Large

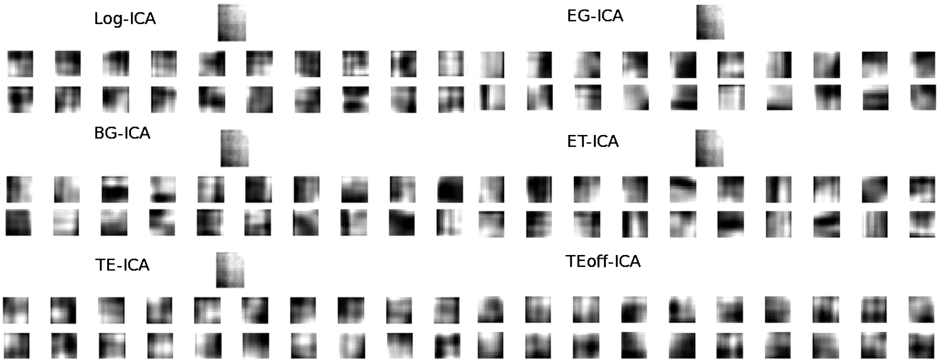


FIG. 10. *Decomposition vectors from six selected models. From left to right and top to bottom: Log-ICA, Lap-ICA, EG-ICA, BG-ICA, EBG-ICA, ET-ICA, TE-ICA, TEoff-ICA. For each model the top row is the mean image and the bottom rows are the* 20 *corresponding decomposition vectors.* (*See Figure* 2 *in the supplementary file* [*Allassonnière and Younes* (2011)] *for a larger image.*)

Deformation Diffeomorphic Metric Mapping [Miller, Trouve and Younes (2002, 2006), Trouvé (1998), Trouvé and Younes (2002)]. We here analyze the logarithm of the Jacobian determinant of the estimated deformations, represented (for each image) as a scalar field over the surface of the template, described by a triangulated mesh. These vectors have fixed length ($d = 3{,}223$) equal to the number of vertices in the triangulation.

The 101 subjects in the data set are separated in 3 groups with 57, 32 and 12 patients, containing healthy patients in the first group and patients with Alzheimer's disease and semantic dementia (denoted the AD group later) at different stages in the last two groups.

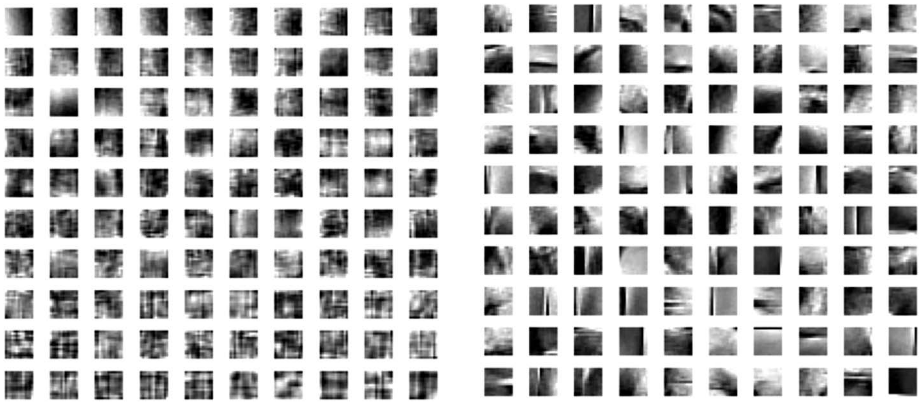


FIG. 11. 100 *decomposition vectors from* 2 *models. Left*: *Log-ICA. Right*: *BG-ICA.* (*See Figure* 3 *in the supplementary file* [*Allassonnière and Younes* (2011)] *for a larger image.*)

Using our algorithm, we have computed $p = 5$ decomposition vectors based on the complete data set. Figures 13 to 15 present these decomposition vectors mapped on the meshed hippocampus for six selected models. The estimated mean is shown on the left side and the five corresponding decomposition vectors are on the right-hand side. Images are presented with different color maps to facilitate the visualization of the patterns. In particular, even if the means seem to contain a lot of information, their intensities vary on a very small scale compared to all the decomposition vectors (they are actually close to 0).

Although results vary with the chosen model, we can see common features emerging. First of all, the means are very similar to each other. The patterns which we can notice on each of them is the same. For example, there is a noticeable contraction on the top part and an extension on the bottom left-hand side of the shape. These deformations, however, have a small amplitude and can be interpreted as the "bias" of the training set with respect to the template. Concerning the decomposition vectors themselves, the pattern of the first vector of the Logistic model is present in all other models [e.g., in position 1 for the Laplacian, EG, TE and TEoff models (not shown here), 4 for the BG model, 5 for the EBG (not shown here) and 2 for the ET model]. Other patterns occur also, like a contraction or a growth of the tail part [in vector 3 of Log, Lap, EG, BG, EBG, TEoff (not shown here) and 5 of TE] or on the bottom of the left part of the image [in vectors 4 and 5 of Log, 5 of Lap, EG, BG and TEoff (not shown here) and in vector 1 otherwise]. These common features seem to be characteristic of this population.

Even if a careful justification of the following statements would require a more thorough study, which would fall out of the scope of the present paper, these ICA patterns seem to correlate with anatomical hippocampus regions, such as those introduced in Miller et al. (2009) and Wang et al. (2006), in the sense that the supports of the decomposition vectors are located within subregions of the anatomical segmentation. For example, the first and third components from the log-ICA decomposition significantly overlap with what authors in Wang et al. (2006) refer to as the hippocampus *lateral zone*, while components 3 and 5 are contained in the *superior zone*, and component 2 in the *interior-medial zone*. Similar conclusions apply with most decomposition vectors obtained with other ICA methods.

In Tables 4 and 5 we provide the p-values obtained from the comparison of the five ICA coefficients (β) among the three subgroups. The test is based on a Hotelling T-statistic evaluated on the coefficients, the p-value being computed using permutation sampling. The test is performed for two different comparisons: first we compare the healthy group with respect to the two pathological groups. This is what is shown in Table 4. The second test compares the healthy group with the group of 32 mild AD patients. The results are presented in Table 5.

Because SAEM is stochastic and only expected to converge to a critical point of the likelihood (which may not be unique), different runs of the algorithm starting from the same initial point can lead to different limits. To evaluate the effect of

TABLE 4
Mean and standard deviation of the p-values for the eight models with the five decomposition vectors shown in Figures 13 to 15. The mean and the standard deviation are computed over 50 *samples of the posterior distributions of the hidden variables to separate the first group* (*Control*) *with respect to the two others* (*AZ*)

Model	**Log-ICA**	**Lap-ICA**	**EG-ICA**	**BG-ICA**	**EBG-ICA**
Mean on log $10^{-3}\times$	0.31	0.29	0.27	0.33	0.9
Std deviation on log $10^{-3}\times$	0.16	0.19	0.12	0.25	1.2

Model	**ET-ICA**	**TE-ICA**	**TEoff-ICA**
Mean on log $10^{-3}\times$	0.27	2.4	75.7
Std deviation on log $10^{-3}\times$	0.14	2.9	126.2

this variability, we ran the algorithm for each model 50 times, with the same initial conditions, and computed an average and a standard deviation of the p-values.

The results are mostly significant. Indeed, almost all methods yield p-values under 1% when we compare the control population to the AD groups and less than 3% for the comparison of the control versus mild AD.

The only model which does not yield significant p-values is the offset case. Both the mean and standard deviation are high (even higher when we focus on the mild AD population). This suggests that this model on this database is unstable. One run can lead to significant decomposition vectors and a second one can lead to very different results. This particular model, which worked well with the USPS database, for example, does not seem to be adapted to this type of data that is considered here. The mean is very close to zero and is therefore not a relevant variable for this application. The additional variability in the model may have an

TABLE 5
Mean and standard deviation of the p-values for the eight models with the five decomposition vectors shown in Figures 13 to 15. The mean and the standard deviation are computed over 50 *samples of the posterior distributions of the hidden variables to separate the first group* (*Control*) *with respect to the second one* (*mild AZ*)

Model	**Log-ICA**	**Lap-ICA**	**EG-ICA**	**BG-ICA**	**EBG-ICA**
Mean on log $10^{-3}\times$	9.0	9.6	8.3	10.9	18.7
Std deviation on log $10^{-3}\times$	3.8	4.8	2.7	7.6	17.7

Model	**ET-ICA**	**TE-ICA**	**TEoff-ICA**
Mean on log $10^{-3}\times$	8.9	30.8	148.7
Std deviation on log$10^{-3}\times$	4.6	28.8	160.4

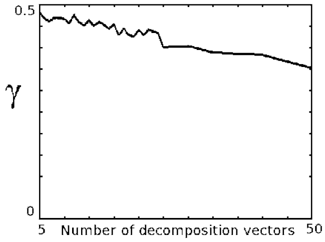


FIG. 12. *Evolution of the probability of one component to activate or inhibit the corresponding decomposition vector in the ET-model with respect to the number of decomposition vectors. The training set is the set of* 101 *hippocampi.*

adverse effect on the estimation. In cases where the dimension of the data is much larger than the number of samples in the training set, it is natural to think that adding more variability in the estimation process may lead to unstable results and therefore large variance of estimated parameters. Depending on this paradigm, the user may prefer to reduce the number of random variables to the decomposition vector weights only.

Figure 12 provides some insight in the way components are turned on/off by the ET-ICA model, by plotting the estimated probability, $\gamma = P(Y_k^j = -1) = P(Y_k^j = 1)$, against the number of decomposition vectors, p. As already noticed in Section 5.2, for small p, all components are needed, yielding $\gamma \simeq 1/2$. When more components are added, they do not need to appear all the time, yielding a decreasing value of γ.

6. Conclusion and discussion. This paper presents a new solution for probabilistic independent component analysis. Probabilistic ICA enables to estimate a small number of features (compared to the dimension of the data) which characterize a data set. Compared to plain ICA, this avoids the instability of the computation of the decomposition matrix when the number of observations is much smaller than

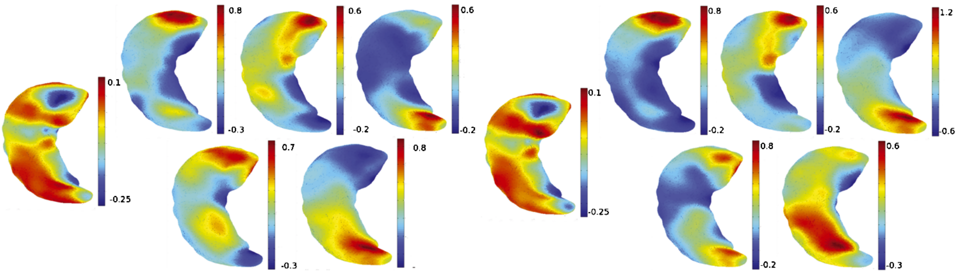


FIG. 13. *Left*: *mean* (*left*) *and* 5 *decomposition vectors estimated with the Log-ICA model. Right*: *mean* (*left*) *and* 5 *decomposition vectors estimated with the Lap-ICA model. Each image has its own color map to highlight the major patterns.*

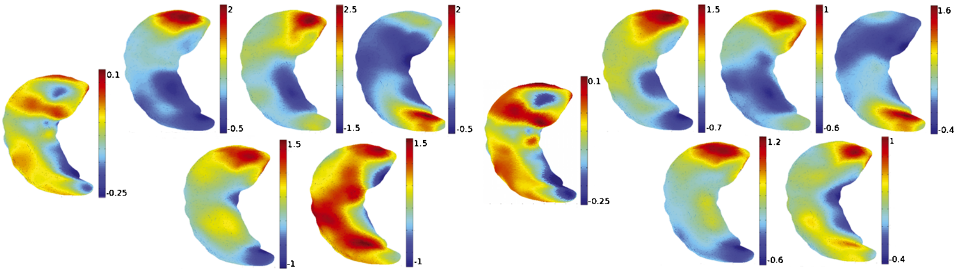


FIG. 14. *Left*: *mean* (*left*) *and* 5 *decomposition vectors estimated with the EG-ICA model. Right*: *mean* (*left*) *and* 5 *decomposition vectors estimated with the BG-ICA model. Each image has its own color map to highlight the major patterns.*

their dimension. We have demonstrated that the stochastic approximation EM algorithm is an efficient and powerful tool which provides a convergent method that estimates the decomposition matrix. We have shown that this procedure does not restrict the large choice of distributions for the independent components, as illustrated by eight models with different properties, mixing continuous and discrete probability measures, that we have introduced and studied.

Future works will be devoted to the analysis of nonlinear generative models that allow for the analysis of data on Riemannian manifolds, including the important case of shape spaces in which the models generate nonlinear deformation of given templates. Generalizations of the methods proposed in Allassonnière, Amit and Trouvé (2007) and Allassonnière and Kuhn (2010) will be developed, in order to estimate both the templates and the generative parameters.

APPENDIX A: PROOF OF THE SUBEXPONENTIAL TAIL OF THE EG-DISTRIBUTION

Let (Y, S) be a pair of independent random variables where Y and S have a standard normal distribution and an exponential distribution, respectively. Let $\beta =$

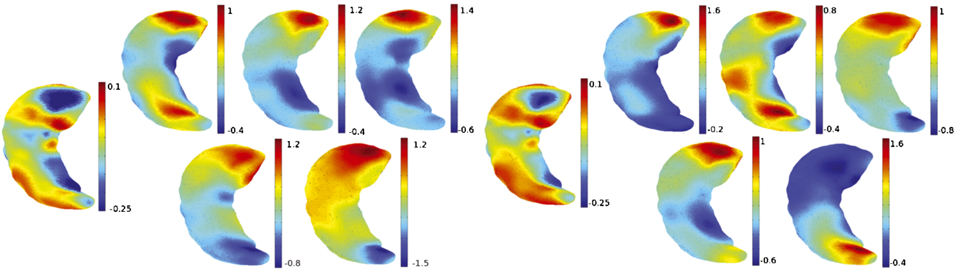


FIG. 15. *Left*: *mean* (*left*) *and* 5 *decomposition vectors estimated with the ET-ICA model. Right*: *mean* (*left*) *and* 5 *decomposition vectors estimated with the TE-ICA model. Each image has its own color map to highlight the major patterns.*

YS and assume $t > 0$ so that $\beta > t$ implies $Y > 0$. We have [letting $C = (2\pi)^{-1/2}$]

$$\mathbb{P}(\beta > t) = \mathbb{P}(s > t/y, y > 0) = C\int_0^\infty \mathbb{P}\left(s > \frac{t}{y}\right)\exp\left(-\frac{1}{2}y^2\right)dy$$
$$= C\int_0^\infty \exp\left(-\frac{1}{2}y^2 - \frac{t}{y}\right)dy.$$

Let $h_t(y) = \frac{1}{2}y^2 + \frac{t}{y}$. We can write, letting $z = y/t^{1/3}$,

$$h_t(y) = \frac{3}{2}t^{2/3} + t^{1/3}\alpha(z)$$

with $\alpha(z) = ((z-1)^2 + z^{-1} + z - 2)/2$. Making the change of variables $y \to z$ in the integral yields

$$\mathbb{P}(\beta > t) = Ct^{1/3}e^{-(3/2)t^{2/3}}\int_0^\infty e^{-t^{1/3}\alpha(z)}\,dz.$$

Using Laplace's method, we find that the second integral is equivalent to $\sqrt{2\pi/(6t^{1/3})}$, proving that $\mathbb{P}(\beta > t$ decays like $t^{1/6}\exp(-3t^{2/3}/2)$ when $t \to +\infty$. Note that the density of β, which is $g(\beta) = \int_0^\infty \exp(-\frac{1}{2}y^2 - \frac{\beta}{y})\,\frac{dy}{y}$, has a singularity at $\beta = 0$.

APPENDIX B: MAXIMUM LIKELIHOOD FOR THE COMPLETE MODELS

The M-step in our models requires solving the equation $E_\theta(\mathbf{S}) = [\mathbf{S}]$ where $[\mathbf{S}]$ is a prescribed value of the sufficient statistic (an empirical average for complete observations, or what we have denoted $\bar{\mathbf{S}}_t$ in the M-step of the learning algorithm). In the next sections we provide the expressions of $\mathbf{S}$ for the family of models we consider and give the corresponding solution of the maximum likelihood equations. Notice that these are closed-form expressions, ensuring the simplicity of each iteration of the SAEM algorithm.

B.1. Log-ICA and Lap-ICA models.

For these models, the log-likelihood is

$$-\sum_{j=1}^{p}\xi(\beta^j) - \frac{1}{2\sigma^2}\left|X - \boldsymbol{\mu}_0 - \sum_{j=1}^{p}\beta^j\mathbf{a}_j\right|^2 - \log C(\sigma^2, \mathbf{A}),$$

where $\xi(\beta) = 2\log(e^\beta + e^{-\beta})$ in the logistic case, and $\xi(\beta) = |\beta|$ in the Laplacian case. As customary, and to lighten the formulae, we let $\beta^0 = 1$ and $\mathbf{a}_0 = \boldsymbol{\mu}_0$, so that $\boldsymbol{\beta}$ and $\mathbf{A}$ have size $d+1$, and remove $\boldsymbol{\mu}_0$ from the expressions for this model and the following ones. We will also leave to the reader the easy modifications of the algorithms in the case of shifted models described in Section 2.9.

The likelihood can be put in exponential form using the sufficient statistic $\mathbf{S} = (\boldsymbol{\beta}\boldsymbol{\beta}^T, \mathbf{X}\boldsymbol{\beta}^T)$, from which the maximum likelihood estimator can be deduced using

$$\begin{cases} \mathbf{A} = [\mathbf{X}\boldsymbol{\beta}^T]([\boldsymbol{\beta}\boldsymbol{\beta}^T])^{-1}, \\ \sigma^2 = \dfrac{1}{d}([|\mathbf{X}|^2] - 2\langle \mathbf{A}, [\mathbf{X}\boldsymbol{\beta}^T]\rangle_F + \langle \mathbf{A}^T\mathbf{A}, [\boldsymbol{\beta}\boldsymbol{\beta}^T]\rangle_F = [|\mathbf{X} - \mathbf{A}\boldsymbol{\beta}|^2]/d), \end{cases}$$

where $\langle \cdot, \cdot \rangle_F$ refers to the Frobenius dot product between matrices (the sum of products of coefficients).

B.2. EG-ICA model. The likelihood is

$$-\frac{1}{2}\sum_{j=1}^{p}(Y^j)^2 - \frac{1}{2}\sum_{j=1}^{p} s^j - \frac{1}{2\sigma^2}\left|X - \sum_{j=1}^{p} s^j Y^j \mathbf{a}_j\right|^2 - \log C(\sigma^2, \mathbf{A})$$

with sufficient statistic $\mathbf{S} = (\boldsymbol{\beta}\boldsymbol{\beta}^T, \mathbf{X}\boldsymbol{\beta}^T)$ with $\beta^j = s^j Y^j$. The maximum likelihood then is

$$\begin{cases} \mathbf{A} = [\mathbf{X}\boldsymbol{\beta}^T]([\boldsymbol{\beta}\boldsymbol{\beta}^T])^{-1}, \\ \sigma^2 = [|\mathbf{X} - \mathbf{A}\boldsymbol{\beta}|^2]/d. \end{cases}$$

B.3. IFA model. The complete log-likelihood of the Independent Factor Analysis model for a single observation X is

$$-\frac{1}{2\sigma^2}\left|X - \sum_{j=1}^{p} \beta^j \mathbf{a}_j\right|^2 - \frac{1}{2}\sum_{j=1}^{p}(\beta^j - b^j m_{t^j})^2 + \sum_{j=1}^{p} \log w_{t^j} - \log C(\mathbf{A}, \sigma, \mathbf{m}, \mathbf{w}).$$

This formulation leads to the following sufficient statistics:

$$S = \left(S_0 = \sum_{j=1}^{p} \mathbb{1}_{t_j = k}, S_1 = \sum_{j=1}^{p} \mathbb{1}_{t_j = k} b^j \beta^j, \boldsymbol{\beta}\boldsymbol{\beta}^T, \mathbf{X}\boldsymbol{\beta}^T \right).$$

The estimator associated to averaged values of these statistics (denoted as above with brackets) is

$$\begin{cases} \mathbf{A} = [\mathbf{X}\boldsymbol{\beta}^T]([\boldsymbol{\beta}\boldsymbol{\beta}^T])^{-1}, \\ \sigma^2 = [|\mathbf{X} - \mathbf{A}\boldsymbol{\beta}|^2]/d, \\ m_k = [S_1]/[S_0], \\ w_k = [S_0]/p. \end{cases}$$

For this model, it is also possible to compute the conditional distribution of the hidden variables, $\boldsymbol{\beta}, \mathbf{t}$ and $\mathbf{b}$ given observed values of X [Attias (1999)]. Indeed, for given $\mathbf{b}$ and $\mathbf{t}$, let $\mu_{\mathbf{b},\mathbf{t}} = (b^1 m_{t^1}, \ldots, b^p m_{t^p})$. Let $\Lambda = (\mathrm{Id}_{\mathbb{R}^p} + \frac{A^T A}{\sigma^2})$ and, for a given $\mathbf{X}$, $\mu_{\mathbf{b},\mathbf{t},\mathbf{X}} = \Lambda(A^T\mathbf{X} + \mu_{\mathbf{b},\mathbf{t}})$. Then, a rewriting of the likelihood above shows that the conditional distribution of $\boldsymbol{\beta}$ given $\mathbf{X}, \mathbf{T}$ and $\mathbf{b}$ is Gaussian with mean $\mu_{\mathbf{b},\mathbf{t},X}$ and covariance Λ, and that the conditional distribution of $(\mathbf{t}, \mathbf{b})$ is

$$\pi(\mathbf{t}, \mathbf{b}|X) \propto \exp\biggl(-\frac{1}{2}(|\mu_{\mathbf{b},\mathbf{t}}|^2 - (A^T X + \mu_{\mathbf{b},\mathbf{t}})^T \Lambda (A^T X + \mu_{\mathbf{b},\mathbf{t}}))\biggr) \prod_{j=1}^{p} w_{tj}.$$

Using these expressions, the E-step of the EM-algorithm can be computed exactly, but it requires computing all $(2K+1)^p$ conditional probabilities $\pi(\mathbf{t}, \mathbf{b}|X)$, which becomes intractable for large dimensions. In contrast, each step of the SAEM algorithm only requires sampling from the conditional distributions, and has complexity of order $p(2K+1)$.

The same remark on the feasibility of the EM algorithm holds for all our models with discrete variables (BG-ICA, ET-ICA, etc.), for which the E-step of the algorithm can be made explicit by conditioning on the discrete variables, with a cost that grows exponentially in the number of components, whereas the sampling part of SAEM only grows linearly.

B.4. BG-ICA and EBG-ICA models. These two models have the same parameters and maximize the same function. The likelihood is

$$-\frac{1}{2}\sum_{j=1}^{p}(Y^j)^2 + \log\biggl(\frac{\alpha}{1-\alpha}\biggr)\sum_{j=1}^{p} b^j - \frac{1}{2\sigma^2}\biggl|X - \sum_{j=1}^{p} b^j Y^j \mathbf{a}_j\biggr|^2 - \log C(\sigma^2, \mathbf{A}, \mu, \alpha)$$

with sufficient statistic $\mathbf{S} = (\boldsymbol{\beta}\boldsymbol{\beta}^T, \mathbf{X}\boldsymbol{\beta}^T, \nu)$ with $\beta^j = b^j Y^j$ and $\nu = b^1 + \cdots + b^p$. The optimal parameters are

$$\begin{cases} \mathbf{A} = [\mathbf{X}\boldsymbol{\beta}^T]([\boldsymbol{\beta}\boldsymbol{\beta}^T])^{-1}, \\ \sigma^2 = [|\mathbf{X} - \mathbf{A}\boldsymbol{\beta}|^2]/d, \\ \alpha = [\nu]/p. \end{cases}$$

B.5. ET-ICA, TE-ICA and TEoff-ICA models. We turn to the ternary models which share the same parameters (up to μ_0 for the offset model). The likelihood to maximize is

$$\log\biggl(\frac{\gamma}{1-\gamma}\biggr)\sum_{j=1}^{d}|Y^j| - \frac{1}{2\sigma^2}\biggl|X - \sum_{j=1}^{p} s^j Y^j \mathbf{a}_j\biggr|^2 - \log C(\sigma^2, \mathbf{A}, \gamma)$$

with sufficient statistic $\mathbf{S} = (\boldsymbol{\beta}\boldsymbol{\beta}^T, \mathbf{X}\boldsymbol{\beta}^T, \zeta)$, $\beta^j = s^j Y^j$, $\zeta = |Y^1| + \cdots + |Y^p|$. The optimal parameters are

$$\begin{cases} \mathbf{A} = [\mathbf{X}\boldsymbol{\beta}^T]([\boldsymbol{\beta}\boldsymbol{\beta}^T])^{-1}, \\ \sigma^2 = [|\mathbf{X} - \mathbf{A}\boldsymbol{\beta}|^2]/d, \\ \gamma = [\zeta]/p. \end{cases}$$

The maximum likelihood estimator for the single scale model is given by the same formulae, using $\beta^j = sY^j$.

SUPPLEMENTARY MATERIAL

Supplement to "A stochastic algorithm for probabilistic independent component analysis" (DOI: 10.1214/11-AOAS499SUPP; .pdf). This file presents a larger version of some of the images contained in this paper.

REFERENCES

AKAIKE, H. (2003). A new look at the statistical model identification. *IEEE Trans. Automat. Control* **19** 716–723.

ALLASSONNIÈRE, S., AMIT, Y. and TROUVÉ, A. (2007). Towards a coherent statistical framework for dense deformable template estimation. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **69** 3–29. MR2301497

ALLASSONNIÈRE, S., KUHN, E. and TROUVÉ, A. (2008). MAP estimation of statistical deformable templates via nonlinear mixed effects models: Deterministic and stochastic approaches. In *Proc. of the International Workshop on the Mathematical Foundations of Computational Anatomy* (*MFCA*), *New York* (X. Pennec and S. Joshi, eds.) 80–91. Available at http://www.inria.fr/sophia/asclepios/events/MFCA08/Proceedings/MFCA08_Proceedings.pdf.

ALLASSONNIÈRE, S. and KUHN, E. (2010). Stochastic algorithm for Bayesian mixture effect template estimation. *ESAIM Probab. Stat.* **14** 382–408.

ALLASSONNIÈRE, S., KUHN, E. and TROUVÉ, A. (2010). Construction of Bayesian deformable models via a stochastic approximation algorithm: A convergence study. *Bernoulli* **16** 641–678. MR2730643

ALLASSONNIÈRE, S. and YOUNES, L. (2011). Supplement to "A stochastic algorithm for probabilistic independent component analysis." DOI:10.1214/11-AOAS499SUPP.

ANDRIEU, C., MOULINES, É. and PRIOURET, P. (2005). Stability of stochastic approximation under verifiable conditions. *SIAM J. Control Optim.* **44** 283–312. MR2177157

ARIE, Y. (2002). Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation. *IEEE Trans. Signal Process* **50** 1545–1553.

ATTIAS, H. (1999). Independent factor analysis. *Neural Comput.* **11** 803–851.

BACH, F. and JORDAN, I. M. (2003). Kernel independent component analysis. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing* (*ICASSP*). Hong Kong, China. Available at http://www.di.ens.fr/~fbach/kernelICA-icassp03.pdf.

BARTLETT, M. S., MOVELLAN, J. R. and SEJNOWSKI, T. J. (2002). Face recognition by independent component analysis. *IEEE Trans. Neural Netw.* **13** 1450–1464.

BELL, A. J. and SEJNOWSKI, T. J. (1995a). An information maximisation approach to blind separation and blind deconvolution. *Neural Comput.* **7** 1004–1034.

BELL, A. J. and SEJNOWSKI, T. J. (1995b). An information maximisation approach to blind separation and blind deconvolution. *Neural Comput.* **7, 6** 1129–1159.

BRANDT PETERSEN, K. and WINTHER, O. (2005). The EM algorithm in independent component analysis. In *Proc. of the ICASSP Conference* 169–172. IEEE, Philadelphia, PA.

BREMOND, O., MOULINES, É. and CARDOSO, J.-F. (1997). Séparation et déconvolution aveugle de signaux bruités: Modélisatin par mélange de gaussiennes. *GRETSI*, *Grenoble* 1427–1430.

CALHOUN, V., ADALI, T. and MCGINTY, V. (2001). fMRI activation in a visual-perception task: Network of areas detected using the general linear model and independent components analysis. *NeuroImage* **14** 1080–1088.

CALHOUN, V. D., ADALI, T., PEARLSON, G. D. and PEKAR, J. J. (2001). A method for making group inferences from functional MRI data using independent component analysis. *Hum. Brain Mapp.* **14** 140–151.

CARDOSO, J.-F. (1999). High-order contrasts for independent component analysis. *Neural Comput.* **11** 157–192.

CELEUX, G. and DIEBOLT, J. (1985). The SEM algorithm: A probabilistic teacher algorithm derived from the EM algorithm for the mixture problem. *Comp. Statis. Quaterly* **2** 73–82.

CÔME, E., CHERFI, Z., OUKHELLOU, L. and AKNIN, P. (2008). Semi-supervised IFA with prior knowledge on the mixing process. An application to railway device diagnosis. In *Proc. of the International Conference on Machine Learning and Applications* 415–420. IEEE, Washington, DC.

DELYON, B., LAVIELLE, M. and MOULINES, E. (1999). Convergence of a stochastic approximation version of the EM algorithm. *Ann. Statist.* **27** 94–128. MR1701103

EFRON, B., HASTIE, T., JOHNSTONE, I. and TIBSHIRANI, R. (2004). Least angle regression. *Ann. Statist.* **32** 407–499.

ERIKSSON, J., KARVANEN, J. and KOIVUNEN, V. (2000). Source distribution adaptive maximum likelihood estimation of ICA model. In *Proc. of 2nd International Workshop on Independent Component Analysis and Blind Signal Separation*, *Helsinki* 227–232.

FARID, H. and ADELSON, E. (1999). Separating reflections and lighting using independent components analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, *Fort Collins*, *CO*.

GRIMES, D. B. and RAO, R. P. N. (2005). Bilinear sparse coding for invariant vision. *Neural Comput.* **17** 47–73.

GRIMES, D. B., SHON, A. P. and RAO, R. P. N. (2003). Probabilistic bilinear models for appearance-based vision. In *Proc. of the Ninth IEEE International Conference on Computer Vision* (*ICCV*'03), *Beijing*, *China* **2** 1478–1486.

HYVARINEN, A. (1999). Survey on independent component analysis. *Neural Computing Surveys* **2** 94–128.

HYVÄRINEN, A. and OJA, E. (1997). A fast fixed-point algorithm for independent component analysis. *Neural Comput.* **9** 1483–1492.

KAGAN, A. M., LINNIK, Y. V. and RAO, C. R. (1973). *Characterization Problems in Mathematical Statistics*. Wiley, New York. MR0346969

KUHN, E. and LAVIELLE, M. (2004). Coupling a stochastic approximation version of EM with an MCMC procedure. *ESAIM Probab. Stat.* **8** 115–131 (electronic). MR2085610

LEARNED-MILLER, E. G. and FISHER III, J. W. (2003). ICA using spacings estimates of entropy. *J. Mach. Learn. Res.* **4** 1271–1295.

LI, D. and SUN, X. (2006). *Nonlinear Integer Programming. International Ser. Operations Res. Management Sci.* **84**. Springer, New York.

LIEBERMEISTER, W. (2002). Linear modes of gene expression determined by independent component analysis. *Bioinformatics* **18** 51–60.

LIU, C. and WECHSLER, H. (2003). Independent component analysis of Gabor features for face recognition. *IEEE Trans. Neural Netw.* **4** 919–928.

MAKEIG, S. and JUNG, T. (1997). Blind separation of auditory event-related brain responses into independent components. *Proc. Natl. Acad. Sci. USA* **94** 10979–10984.

MAUGIS, C., CELEUX, G. and MARTIN-MAGNIETTE, M. L. (2009). Variable selection in model-based clustering: A general variable role modeling. *Comput. Statist. Data Anal.* **53** 3872–3882. MR2749931

MILLER, M. I., TROUVE, A. and YOUNES, L. (2002). On the metrics and Euler–Lagrange equations of computational anatomy. *Annu. Rev. Biomed. Eng.* **4** 375–405.

MILLER, M. I., TROUVÉ, A. and YOUNES, L. (2006). Geodesic shooting for computational anatomy. *J. Math. Imaging Vision* **24** 209–228. MR2227097

MILLER, M. I., PRIEBE, C. E., QIU, A., FISCHL, B., KOLASNY, A., BROWN, T., PARK, Y., RATNANATHER, J. T., BUSA, E., JOVICICH, J., YU, P., DICKERSON, B. C. and BUCKNER, R. L. (2009). Morphometry BIRN. Collaborative computational anatomy: An MRI morphometry study of the human brain via diffeomorphic metric mapping. *Hum. Brain Mapp.* **30** 2132–2141.

MISKIN, J. W. and MACKAY, D. J. C. (2000). Ensemble learning for blind source separation and deconvolution. In *Advances in Independent Component Analysis*: *Principle and Practice* (M. Girolami, ed.) 209–233. Springer, Berlin.

MOULINES, E., COIS CARDOSO, J.-F. and GASSIAT, E. (1997). Maximum likelihood for blind separation and deconvolution of noisy signals using mixture models. In *International Conf. Acoustics*, *Speech*, *and Signal Processing ICASSP*-97 *Munich*, *Germany* **5** 3617–3620.

OLSHAUSEN, B. A. and FIELD, D. J. (1996a). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* **381** 607–609.

OLSHAUSEN, B. A. and FIELD, D. J. (1996b). Natural images statistics and efficient coding. *Networks*: *Computation in Neural Systems* **7** 333–339.

SCHOLZ, M., GATZEK, S., STERLING, A., FIEHN, O. and SELBIG, J. (2004). Metabolite fingerprinting: Detecting biological features by independent component analysis. *Bioinformatics* **20** 2447–2454.

SCHWARZ, G. (1978). Estimating the dimension of a model. *Ann*. *Statist*. **6** 461–464. MR0468014

TANNER, M. A. (1996). *Tools for Statistical Inference*. Springer, New York.

TENENBAUM, J. B. and FREEMAN, W. T. (2002). Separating style and content with bilinear models. *Neural Comput*. **12** 1247–1283.

TIBSHIRANI, R. (1996). Regression shrinkage and selection via the lasso. *J*. *Roy*. *Statist*. *Soc*. *Ser*. *B* **58** 267–288. MR1379242

TROUVÉ, A. (1998). Diffeomorphism groups and pattern matching in image analysis. *Int*. *J*. *Comput*. *Vis*. **28** 213–221.

TROUVÉ, A. and YOUNES, L. (2002). Local geometry of deformable templates. Technical report, Univ. Paris 13.

ÜZÜMCÜ, M., FRANGI, A. F., REIBER, J. H. C. and LELIEVELDT, B. P. F. (2003). Independent component analysis in statistical shape models. *SPIE Medical Image Analysis* 375–383.

VALPOLA LAPPALAINEN, H. and PAJUNEN, P. (2000). Fast algorithms for Bayesian independent component analysis. In *Proc. of the Second International Workshop on Independent Component Analysis and Blind Signal Separation*, *ICA 2000*, *Helsinki*, *Finland* 233–237.

VAROQUAUX, G., SADAGHINI, S., POLINE, J. B. and THIRION, B. (2010). A group model for stable multi-subject ICA on fMRI datasets. *NeuroImage* **51** 288—299.

WANG, L., MILLER, J. P., GADO, M. H., MCKEEL, D. W., ROTHERMICH, M., MILLER, M. I., MORRIS, J. C. and CSERNANSKY, J. G. (2006). Abnormalities of hippocampal surface structure in very mild dementia of the Alzheimer type. *Neuroimage* **30** 52–60.

WEI, G. C. and TANNER, M. A. (1990). A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *J*. *Amer*. *Statist*. *Assoc*. **85** 699–704.

WELLING, M. and WEBER, M. (2001). A constrained EM algorithm for independent component analysis. *Neural Comput*. **13** 677–689.

CENTRE DE MATHÉMATIQUES APPLIQUÉES
ECOLE POLYTECHNIQUE
ROUTE DE SACLAY
91128 PALAISEAU
FRANCE
E-MAIL: Stephanie.Allassonniere@polytechnique.edu

CENTER FOR IMAGING SCIENCE
JOHNS HOPKINS UNIVERSITY
3400 N. CHARLES STREET
BALTIMORE, MARYLAND 21218
USA
E-MAIL: Laurent.Younes@jhu.edu