

PMOG: The projected mixture of Gaussians model with application to blind source separation

Gautam V. Pendse^{*1}

¹ P.A.I.N Group, Imaging and Analysis Group (IMAG), McLean Hospital, Harvard Medical School

July 22, 2010

*To whom correspondence should be addressed. e-mail: gpendse@mclean.harvard.edu

1 Abstract

We extend the mixtures of Gaussians (MOG) model to the projected mixture of Gaussians (PMOG) model. In the PMOG model, we assume that q dimensional input data points \mathbf{z}_i are projected by a q dimensional vector \mathbf{w} into 1-D variables u_i . The projected variables u_i are assumed to follow a 1-D MOG model. In the PMOG model, we maximize the likelihood of observing u_i to find both the model parameters for the 1-D MOG as well as the projection vector \mathbf{w} . First, we derive an EM algorithm for estimating the PMOG model. Next, we show how the PMOG model can be applied to the problem of blind source separation (BSS). In contrast to conventional BSS where an objective function based on an approximation to differential entropy is minimized, PMOG based BSS simply minimizes the differential entropy of projected sources by fitting a flexible MOG model in the projected 1-D space while simultaneously optimizing the projection vector \mathbf{w} . The advantage of PMOG over conventional BSS algorithms is the more flexible fitting of non-Gaussian source densities without assuming near-Gaussianity (as in conventional BSS) and still retaining computational feasibility.

2 Introduction

The mixture of Gaussians (MOG) is a flexible model with application to many real world problems. The adjustable parameters in a 1-D MOG model include the number of component Gaussian distributions, the mean and variance of each component distribution and their mixing fractions. Given 1-D data points u_i , these distributional parameters can be efficiently estimated by maximum likelihood (ML) using the expectation maximization (EM) algorithm [7]. Now consider the following situation:

- Data points u_i are not given directly but suppose that we are given vectors \mathbf{z}_i . Next, 1-D scalar variables u_i are generated by $u_i = \mathbf{w}^T \mathbf{z}_i$ where \mathbf{w} is an unknown projection vector.
- Suppose that projected variables u_i follow a 1-D MOG model which we will refer to as a projected mixture of Gaussians (PMOG) model in this work.

Can we estimate the PMOG distributional parameters as well as the projection vector \mathbf{w} using ML? In particular, can we derive an EM algorithm for this PMOG model similar to the standard EM algorithm for the conventional MOG model?

While estimating the PMOG model is an interesting problem in its own right, we will show that it is also closely related to the problem of estimating the differential entropy of a random variable. Given this fact, we will show that the PMOG model can be applied to the problem of linear blind source separation (BSS) and linear independent component analysis (ICA). We use the term ICA to refer to "square mixing" where there are equal number of latent sources and mixed signals and BSS to refer to "non-square and noisy" mixing where there are more mixtures corrupted with additive

Gaussian noise than latent sources. From this point of view, ICA is a special "noise free" case of BSS.

BSS or ICA is a well studied problem and we refer the reader to [5, 15, 4] for a detailed overview of relevant work. A central point in BSS is choosing the "measuring function" for differential entropy of a random variable. One of the most widely used algorithm for BSS is the FastICA (FICA) algorithm [13]. The FICA algorithm works by optimizing the differential entropy based contrast functions developed in the seminal work by Hyvarinen et al. [12]. These contrast functions are approximations to the differential entropy of a random variable x under the following conditions:

- Expected values of certain functions $G_i(x)$ are given. The density of x is estimated to be the maximum entropy distribution (MED) $f^{med}(x)$ subject to the these constraints.
- Most importantly, the assumed MED $f^{med}(x)$ is, in the words of Hyvarinen et al. [12] "not very far from a Gaussian distribution". Let us denote this simplified form of $f^{med}(x)$ by $f_{Gaussian}^{med}(x)$.

Hyvarinen et al. calculated expressions for the differential entropy using the distribution $f_{Gaussian}^{med}(x)$ given flexible user defined functions $G_i(x)$. A key question that arises is: are these approximations to differential entropy and BSS solutions based on $f_{Gaussian}^{med}(x)$ adequate when the true density and hence $f^{med}(x)$ is not "near Gaussian"?

In another seminal paper, Attias et al. [1] developed a general solution to the BSS problem where the latent source density was modeled as a "factorial MOG" density. This significant advance removed the "near Gaussianity" assumption on the latent source densities. In addition, [1] developed an EM algorithm for the ML solution of the mixing parameters and MOG source parameters in BSS followed by a posterior mean or maximum a posteriori (MAP) estimate of the sources. However, this algorithm becomes computationally intractable for > 13 sources. Moreover, the solution by Attias et al. assumes "exact independence" between the sources i.e., it does not allow any partial dependence between sources. Can we develop a solution to the BSS problem that retains the flexible latent density modeling of [1], retains computational tractability and can be applied under partial dependence between sources?

In this work, we develop the PMOG model and then apply it to the problem of BSS to address these questions:

1. We describe the PMOG model and derive an EM algorithm for estimating its parameters in **section 5**.
2. We show how PMOG can be applied to solving a BSS problem including cases where partial dependence between sources is allowed in **sections 7, 8**.

3 Code distribution for PMOG

In the spirit of reproducible research, Matlab code (www.mathworks.com), data and step-by-step demos for estimating a PMOG model and performing PMOG based BSS can be downloaded from the following URL:

<http://gautampendse.com/reviewers/pmog/webpage/pmog.html>

This code distribution is currently made available only to reviewers of this manuscript but it will be freely available for all non-commercial uses in the near future.

4 Notation

- Scalars will be denoted in a non-bold font (e.g. γ, θ, L) possibly with subscripts (e.g. π_k, μ_k). We will use bold face lower case letters possibly with subscripts to denote vectors (e.g. $\boldsymbol{\mu}, \mathbf{x}, \mathbf{z}_1$) and bold face upper case letters possibly with subscripts to denote matrices (e.g. $\mathbf{A}, \boldsymbol{\Sigma}, \mathbf{B}_1$). The transpose of a matrix \mathbf{A} will be denoted by \mathbf{A}^T and its inverse will be denoted by \mathbf{A}^{-1} . We will denote the $p \times p$ identity matrix by \mathbf{I}_p . A vector or matrix of all zeros will be denoted by a bold face zero $\mathbf{0}$ whose size should be clear from context.
- The j th component of vector \mathbf{x}_i will be denoted by x_{ij} whereas the j th component of vector \mathbf{x} will be denoted by x_j . The element (i, j) of matrix \mathbf{A} will be denoted by $A(i, j)$. Estimates of variables will be denoted by putting a hat on top of the variable symbol. For example, an estimate of σ^2 will be denoted by $\hat{\sigma}^2$. The 2-norm of a $p \times 1$ vector \mathbf{x} will be denoted by $\|\mathbf{x}\|_2 = +\sqrt{\sum_{i=1}^p x_i^2}$.
- If \mathbf{x} is a random vector with a multivariate Normal distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ then we will denote this distribution by $\mathcal{N}(\mathbf{x} | \boldsymbol{\mu}, \boldsymbol{\Sigma})$. Similarly, if u is a scalar random variable with a Normal distribution with mean μ_k and variance σ_k^2 then we will denote this distribution by $\mathcal{N}(u | \mu_k, \sigma_k^2)$. We will use $U(a, b)$ to denote the uniform random distribution on (a, b) . Unless otherwise stated all logarithms are natural logarithms (i.e., log means \log_e). Probability density functions will be denoted by capital letters with the vector of arguments in parenthesis such as $P(\mathbf{x})$ or $Q(\mathbf{x})$. When necessary we will also indicate the dependence of a probability density on a vector of parameters $\boldsymbol{\theta}$ by using the notation $P(\mathbf{x} | \boldsymbol{\theta})$. The expected value of a function $f(\mathbf{x})$ with respect to the density $P(\mathbf{x})$ will be denoted by $E_x[f(\mathbf{x})]$ or simply by $E[f(\mathbf{x})]$.

5 Projected mixture of Gaussians (PMOG) model

In this section, we try to answer the following questions:

- What is the PMOG model? What is given and what needs to be estimated?
- Can we derive an EM algorithm for estimating the PMOG model?
- How are the MOG parameters and projection vector estimated in the M-step?
- What other precautions need to be taken to make the monotonic likelihood improvement property of EM hold true?

5.1 What is the PMOG model?

Suppose we are given n vectors $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n$ where each \mathbf{z}_i is a $q \times 1$ vector. Suppose \mathbf{Z} is a $q \times n$ matrix formed by assembling these vectors into a matrix i.e.,

$$\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n] \quad (5.1)$$

We can think of these vectors as n realizations of a random vector \mathbf{z} . Suppose \mathbf{w} is an unknown $q \times 1$ vector that defines new "projected" scalars u_1, u_2, \dots, u_n such that

$$u_i = \mathbf{w}^T \mathbf{z}_i = \mathbf{z}_i^T \mathbf{w} \quad (5.2)$$

Again, these n scalars can be thought of as n realizations of the random variable $u = \mathbf{w}^T \mathbf{z}$. Suppose we are also given a $q \times L$ matrix \mathbf{G} of full column rank L with $q > L$. It is given that the unknown vector \mathbf{w} satisfies the constraints

$$\mathbf{w}^T \mathbf{w} = 1 \quad (5.3)$$

$$\mathbf{G}^T \mathbf{w} = \mathbf{0} \quad (5.4)$$

It is assumed that the density of random variable u is a mixture of R Gaussians i.e.,

$$P(u \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2) = \sum_{k=1}^R \pi_k \mathcal{N}(u \mid \mu_k, \sigma_k^2) \quad (5.5)$$

In the above equation, $\boldsymbol{\pi}$ is a $R \times 1$ vector $[\pi_1, \pi_2, \dots, \pi_R]^T$. Similarly $\boldsymbol{\mu} = [\mu_1, \mu_2, \dots, \mu_R]^T$ and $\boldsymbol{\sigma}^2 = [\sigma_1^2, \sigma_2^2, \dots, \sigma_R^2]^T$ are also $R \times 1$ vectors. The class fractions π_k satisfy:

$$\sum_{k=1}^R \pi_k = 1 \quad (5.6)$$

$$0 \leq \pi_k \leq 1 \quad (5.7)$$

Equation 5.5 can be equivalently written in terms of \mathbf{z} as follows:

$$P(\mathbf{w}^T \mathbf{z} | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2) = \sum_{k=1}^R \pi_k \mathcal{N}(\mathbf{w}^T \mathbf{z} | \mu_k, \sigma_k^2) \quad (5.8)$$

We will call the model 5.8 a projected mixture of Gaussians or PMOG model. A pictorial description of the PMOG model is given in Fig. 1.

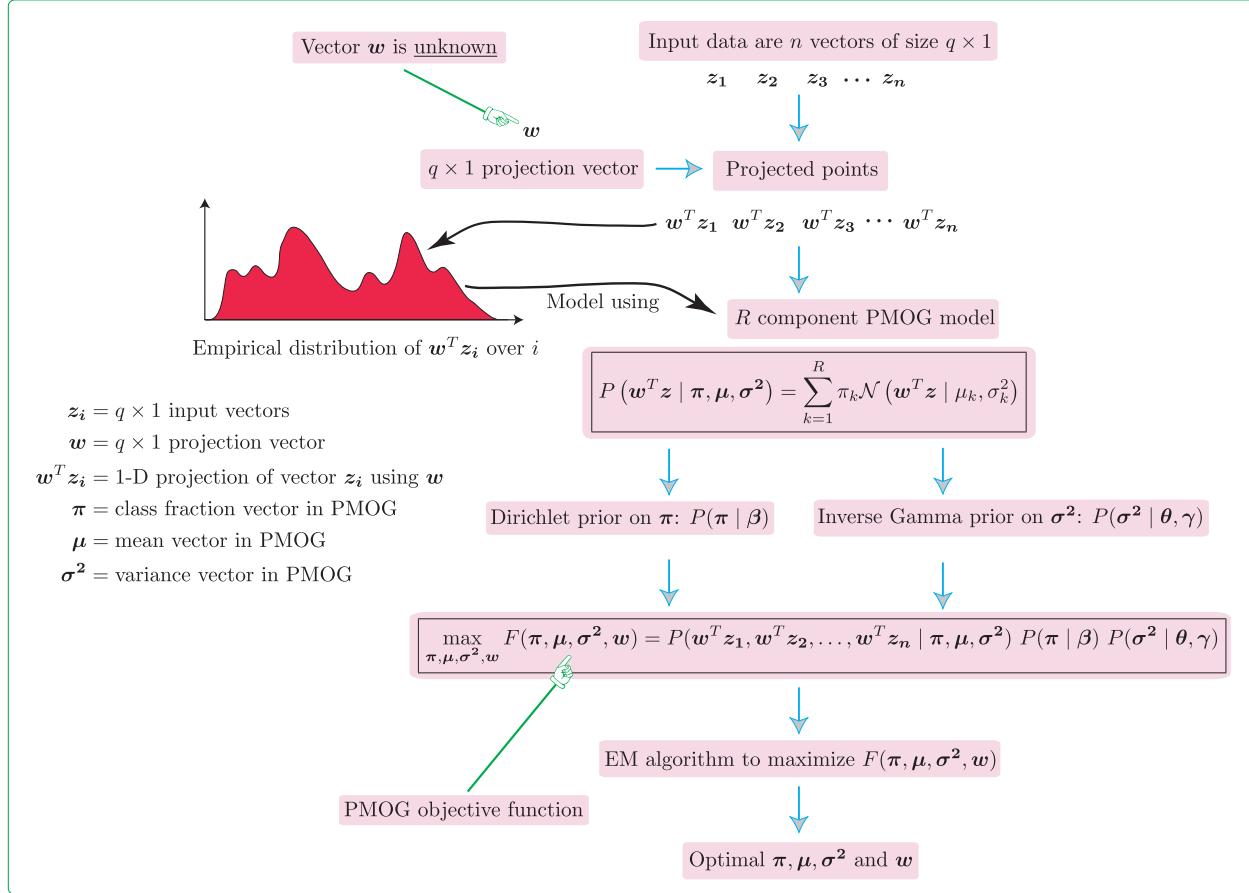


Figure 1: Pictorial depiction of the PMOG model. Input vectors \mathbf{z}_i are projected into 1-D variables using projection vector \mathbf{w} . The 1-D empirical density of projected points $\mathbf{w}^T \mathbf{z}_i$ is modeled using a MOG density. In contrast to conventional MOG, the PMOG model requires estimation of both the MOG parameters $\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2$ as well as the projection vector \mathbf{w} . The overall objective function F also includes the influence of priors for $\boldsymbol{\pi}$ and $\boldsymbol{\sigma}^2$. This is done to prevent the collapse of a component density of PMOG onto a single point.

5.2 Estimating the PMOG model

Assuming that $\mathbf{w}^T \mathbf{z}_i$ for $i = 1, 2, \dots, n$ are n independent realizations of $\mathbf{w}^T \mathbf{z}$, we can write their joint density as

$$P(\mathbf{w}^T \mathbf{z}_1, \mathbf{w}^T \mathbf{z}_2, \dots, \mathbf{w}^T \mathbf{z}_n | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2) = \prod_{i=1}^n P(\mathbf{w}^T \mathbf{z}_i | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2) \quad (5.9)$$

To simplify notation, we will denote the left hand side of 5.9 by $P(\mathbf{Z}^T \mathbf{w} | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2)$. With this notation, we can write:

$$P(\mathbf{Z}^T \mathbf{w} | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2) = \prod_{i=1}^n P(\mathbf{w}^T \mathbf{z}_i | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2) \quad (5.10)$$

The problem is to **maximize** $P(\mathbf{Z}^T \mathbf{w} | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ (equation 5.10) w.r.t $\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2$ and \mathbf{w} . For a given \mathbf{w} this problem is equivalent to the standard mixture of Gaussians (MOG) estimation problem and can be handled efficiently by the expectation maximization (EM) algorithm [7]. In the PMOG model, the novelty is to allow \mathbf{w} to be an unknown that is estimated along with MOG parameters $\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2$. Next we introduce priors on $\boldsymbol{\pi}$ and $\boldsymbol{\sigma}^2$. The purpose of introducing priors on $\boldsymbol{\pi}$ and $\boldsymbol{\sigma}^2$ is simply to prevent the collapse of a Gaussian component from PMOG onto a single data point. Suppose we assume a Dirichlet prior on the vector $\boldsymbol{\pi}$. This prior is described by a $R \times 1$ parameter vector $\boldsymbol{\beta}$ with elements $\beta_1, \beta_2, \dots, \beta_R$.

$$P(\boldsymbol{\pi} | \boldsymbol{\beta}) \propto \prod_{k=1}^R \pi_k^{\beta_k - 1} \quad (5.11)$$

Similarly we assume a product of inverse Gamma prior on $\boldsymbol{\sigma}^2$. Suppose $\boldsymbol{\theta}$ is a vector with elements $\theta_1, \theta_2, \dots, \theta_R$ and similarly $\boldsymbol{\gamma}$ is a vector with elements $\gamma_1, \gamma_2, \dots, \gamma_R$ then

$$P(\boldsymbol{\sigma}^2 | \boldsymbol{\theta}, \boldsymbol{\gamma}) = \prod_{k=1}^R P(\sigma_k^2 | \theta_k, \gamma_k) \propto \prod_{k=1}^R (\sigma_k^2)^{-(\theta_k + 1)} \exp\left(-\frac{1}{\gamma_k \sigma_k^2}\right) \quad (5.12)$$

We use an improper prior for $\boldsymbol{\mu}$, $P(\boldsymbol{\mu}) = 1$. With this choice of priors the posterior distribution can be written as:

$$P(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2 | \mathbf{Z}^T \mathbf{w}) = \frac{P(\mathbf{Z}^T \mathbf{w} | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2) P(\boldsymbol{\pi} | \boldsymbol{\beta}) P(\boldsymbol{\sigma}^2 | \boldsymbol{\theta}, \boldsymbol{\gamma})}{P(\mathbf{Z}^T \mathbf{w})} \quad (5.13)$$

Maximization of this posterior density w.r.t both $\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2$ and \mathbf{w} is difficult because of the presence of \mathbf{w} in the denominator. We could however, maximize the posterior density w.r.t $\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2$ only and maximize the likelihood term $P(\mathbf{Z}^T \mathbf{w} | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ w.r.t \mathbf{w} . This is equivalent to solving the problem:

$$\max_{\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \mathbf{w}} F(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \mathbf{w}) = P(\mathbf{Z}^T \mathbf{w} | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2) P(\boldsymbol{\pi} | \boldsymbol{\beta}) P(\boldsymbol{\sigma}^2 | \boldsymbol{\theta}, \boldsymbol{\gamma}) \quad (5.14)$$

5.3 EM algorithm for maximizing F

The main development in this subsection is an EM algorithm for maximizing F . It is much easier to deal with the logarithm of F rather than F itself. Now

$$\log F(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \mathbf{w}) = \log P(\mathbf{Z}^T \mathbf{w} | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2) + \log P(\boldsymbol{\pi} | \boldsymbol{\beta}) + \log P(\boldsymbol{\sigma}^2 | \boldsymbol{\theta}, \boldsymbol{\gamma}) \quad (5.15)$$

Substituting 5.10, 5.11 and 5.12 into 5.15 we get:

$$\begin{aligned} & \log F(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \mathbf{w}) \\ & \propto \log \left(\prod_{i=1}^n P(\mathbf{w}^T \mathbf{z}_i | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2) \right) + \log \left(\prod_{k=1}^R \pi_k^{\beta_k - 1} \right) + \log \left(\prod_{k=1}^R (\sigma_k^2)^{-(\theta_k + 1)} \exp \left(-\frac{1}{\gamma_k \sigma_k^2} \right) \right) \end{aligned} \quad (5.16)$$

which can be simplified to:

$$\begin{aligned} & \log F(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \mathbf{w}) \\ & \propto \sum_{i=1}^n \log P(\mathbf{w}^T \mathbf{z}_i | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2) + \sum_{k=1}^R (\beta_k - 1) \log \pi_k + \sum_{k=1}^R \left(-(\theta_k + 1) \log \sigma_k^2 - \frac{1}{\gamma_k \sigma_k^2} \right) \end{aligned} \quad (5.17)$$

In the above equation, we have ignored the constants of the prior densities on $\boldsymbol{\pi}$ and $\boldsymbol{\sigma}^2$ since they do not depend on the unknown parameters. Thus the objective function to be maximized can be written as:

$$H(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \mathbf{w}) = H_1(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \mathbf{w}) + H_2(\boldsymbol{\pi}, \boldsymbol{\sigma}^2) \quad (5.18)$$

where

$$H_1(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \mathbf{w}) = \sum_{i=1}^n \log P(\mathbf{w}^T \mathbf{z}_i | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2) \quad (5.19)$$

and

$$H_2(\boldsymbol{\pi}, \boldsymbol{\sigma}^2) = \sum_{k=1}^R (\beta_k - 1) \log \pi_k + \sum_{k=1}^R \left(-(\theta_k + 1) \log \sigma_k^2 - \frac{1}{\gamma_k \sigma_k^2} \right) \quad (5.20)$$

Substituting 5.8 into 5.19 we get:

$$H_1(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \mathbf{w}) = \sum_{i=1}^n \log \sum_{k=1}^R \pi_k \mathcal{N}(\mathbf{w}^T \mathbf{z}_i | \mu_k, \sigma_k^2) \quad (5.21)$$

Introducing auxillary variables α_{ki} such that:

$$\sum_{k=1}^R \alpha_{ki} = 1 \quad (5.22)$$

$$0 \leq \alpha_{ki} \leq 1 \quad (5.23)$$

we can write

$$H_1(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \mathbf{w}) = \sum_{i=1}^n \log \sum_{k=1}^R \left(\frac{\pi_k \mathcal{N}(\mathbf{w}^T \mathbf{z}_i | \mu_k, \sigma_k^2)}{\alpha_{ki}} \right) \alpha_{ki} \quad (5.24)$$

By using the concavity of the log function, we can write:

$$H_1(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \mathbf{w}) \geq Q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \mathbf{w}, \boldsymbol{\alpha}) = \sum_{i=1}^n \sum_{k=1}^R \alpha_{ki} \log \left(\frac{\pi_k \mathcal{N}(\mathbf{w}^T \mathbf{z}_i | \mu_k, \sigma_k^2)}{\alpha_{ki}} \right) \quad (5.25)$$

From 5.18 we see that

$$H(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \mathbf{w}) = H_1(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \mathbf{w}) + H_2(\boldsymbol{\pi}, \boldsymbol{\sigma}^2) \geq Q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \mathbf{w}, \boldsymbol{\alpha}) + H_2(\boldsymbol{\pi}, \boldsymbol{\sigma}^2) \quad (5.26)$$

The rightmost inequality becomes an equality when we choose:

$$\alpha_{ki} = \frac{\pi_k \mathcal{N}(\mathbf{w}^T \mathbf{z}_i | \mu_k, \sigma_k^2)}{\sum_{k=1}^R \pi_k \mathcal{N}(\mathbf{w}^T \mathbf{z}_i | \mu_k, \sigma_k^2)} \quad (5.27)$$

In the context of EM, the α_{ki} are also known as responsibilities. Suppose $\boldsymbol{\pi}^{(t)}, \boldsymbol{\mu}^{(t)}, \boldsymbol{\sigma}^{2(t)}, \mathbf{w}^{(t)}$ are the parameter values at iteration t and $\boldsymbol{\alpha}^{(t)}$ be the corresponding responsibilities computed from 5.27. Then we have

$$\begin{aligned} & H\left(\boldsymbol{\pi}^{(t+1)}, \boldsymbol{\mu}^{(t+1)}, \boldsymbol{\sigma}^{2(t+1)}, \mathbf{w}^{(t+1)}\right) \\ &= Q\left(\boldsymbol{\pi}^{(t+1)}, \boldsymbol{\mu}^{(t+1)}, \boldsymbol{\sigma}^{2(t+1)}, \mathbf{w}^{(t+1)}, \boldsymbol{\alpha}^{(t+1)}\right) + H_2\left(\boldsymbol{\pi}^{(t+1)}, \boldsymbol{\sigma}^{2(t+1)}\right) \quad (\text{by 5.26 and 5.27}) \\ &\geq Q\left(\boldsymbol{\pi}^{(t+1)}, \boldsymbol{\mu}^{(t+1)}, \boldsymbol{\sigma}^{2(t+1)}, \mathbf{w}^{(t+1)}, \boldsymbol{\alpha}^{(t)}\right) + H_2\left(\boldsymbol{\pi}^{(t+1)}, \boldsymbol{\sigma}^{2(t+1)}\right) \quad (\text{by 5.26 and 5.27}) \end{aligned} \quad (5.28)$$

Now suppose, given $\boldsymbol{\pi}^{(t)}, \boldsymbol{\mu}^{(t)}, \boldsymbol{\sigma}^{2(t)}, \mathbf{w}^{(t)}$ and $\boldsymbol{\alpha}^{(t)}$, we calculate $\boldsymbol{\pi}^{(t+1)}, \boldsymbol{\mu}^{(t+1)}, \boldsymbol{\sigma}^{2(t+1)}, \mathbf{w}^{(t+1)}$ such that

$$\begin{aligned} & Q\left(\boldsymbol{\pi}^{(t+1)}, \boldsymbol{\mu}^{(t+1)}, \boldsymbol{\sigma}^{2(t+1)}, \mathbf{w}^{(t+1)}, \boldsymbol{\alpha}^{(t)}\right) + H_2\left(\boldsymbol{\pi}^{(t+1)}, \boldsymbol{\sigma}^{2(t+1)}\right) \\ &\geq Q\left(\boldsymbol{\pi}^{(t)}, \boldsymbol{\mu}^{(t)}, \boldsymbol{\sigma}^{2(t)}, \mathbf{w}^{(t)}, \boldsymbol{\alpha}^{(t)}\right) + H_2\left(\boldsymbol{\pi}^{(t)}, \boldsymbol{\sigma}^{2(t)}\right) \end{aligned} \quad (5.29)$$

From 5.29 and 5.28 we get:

$$\begin{aligned} & H\left(\boldsymbol{\pi}^{(t+1)}, \boldsymbol{\mu}^{(t+1)}, \boldsymbol{\sigma}^{2(t+1)}, \mathbf{w}^{(t+1)}\right) \\ &\geq Q\left(\boldsymbol{\pi}^{(t)}, \boldsymbol{\mu}^{(t)}, \boldsymbol{\sigma}^{2(t)}, \mathbf{w}^{(t)}, \boldsymbol{\alpha}^{(t)}\right) + H_2\left(\boldsymbol{\pi}^{(t)}, \boldsymbol{\sigma}^{2(t)}\right) \\ &= H\left(\boldsymbol{\pi}^{(t)}, \boldsymbol{\mu}^{(t)}, \boldsymbol{\sigma}^{2(t)}, \mathbf{w}^{(t)}\right) \quad (\text{by 5.26 and 5.27}) \end{aligned} \quad (5.30)$$

This shows that the objective function H increases monotonically from iteration t to iteration $(t+1)$ and hence converges to a local maximum. This gives us the following EM algorithm for the PMOG model:

Initialization Choose initial values of parameters either randomly or by other techniques such as clustering using k -means [17]. The output of this stage is the initial values $\boldsymbol{\pi}^{(0)}, \boldsymbol{\mu}^{(0)}, \boldsymbol{\sigma^2}^{(0)}, \mathbf{w}^{(0)}$. Following initialization, the E and M steps outlined below are performed in an alternating fashion until convergence.

E-step In the E-step, we calculate the responsibilities $\boldsymbol{\alpha}^{(t)}$ using the current estimate of parameters $\boldsymbol{\pi}^{(t)}, \boldsymbol{\mu}^{(t)}, \boldsymbol{\sigma^2}^{(t)}, \mathbf{w}^{(t)}$ and equation 5.27.

M-step In the M-step, we solve the problem:

$$\boldsymbol{\pi}^{(t+1)}, \boldsymbol{\mu}^{(t+1)}, \boldsymbol{\sigma^2}^{(t+1)}, \mathbf{w}^{(t+1)} = \arg \max_{\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma^2}, \mathbf{w}} Q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma^2}, \mathbf{w}, \boldsymbol{\alpha}^{(t)}) + H_2(\boldsymbol{\pi}, \boldsymbol{\sigma^2}) \quad (5.31)$$

In ordinary MOG, the objective function in 5.31 is a convex function of $\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma^2}$ and a closed form solution exists for the M-step. If optimizing also w.r.t \mathbf{w} then the objective function becomes non-convex and hence we need to explicitly impose the post optimization condition 5.29 for EM convergence. We use a relative error criterion to detect EM convergence. Our convergence criterion is:

$$\text{abs} \left[H(\boldsymbol{\pi}^{(t+1)}, \boldsymbol{\mu}^{(t+1)}, \boldsymbol{\sigma^2}^{(t+1)}, \mathbf{w}^{(t+1)}) - H(\boldsymbol{\pi}^{(t)}, \boldsymbol{\mu}^{(t)}, \boldsymbol{\sigma^2}^{(t)}, \mathbf{w}^{(t)}) \right] \leq \varepsilon^* \quad (5.32)$$

$$\varepsilon^* = \varepsilon_{rel} \left(\text{mean} \text{abs} \left[H(\boldsymbol{\pi}^{(t)}, \boldsymbol{\mu}^{(t)}, \boldsymbol{\sigma^2}^{(t)}, \mathbf{w}^{(t)}) \right] \right) \quad (5.33)$$

Here ε_{rel} is a user specified relative error. We used $\varepsilon_{rel} = 10^{-5}$ in our experiments.

5.4 Solving the M-step problem

In this subsection, we discuss in detail the solution of problem 5.31. During the estimation of each variable $\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma^2}$ and \mathbf{w} we account for the relevant constraints using Lagrange multipliers and Karush-Kuhn-Tucker optimality conditions (see [19] for details).

Estimating $\boldsymbol{\pi}$ Differentiating the M-step objective function w.r.t π_k and noting the constraint $\sum_{k=1}^R \pi_k - 1 = 0$ the optimality condition is given by:

$$\frac{\partial}{\partial \pi_k} Q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma^2}, \mathbf{w}, \boldsymbol{\alpha}^{(t)}) + \frac{\partial}{\partial \pi_k} H_2(\boldsymbol{\pi}, \boldsymbol{\sigma^2}) - \lambda \frac{\partial}{\partial \pi_k} \left(\sum_{k=1}^R \pi_k - 1 \right) = 0 \quad (5.34)$$

Now,

$$\frac{\partial}{\partial \pi_k} Q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma^2}, \mathbf{w}, \boldsymbol{\alpha}^{(t)}) = \frac{1}{\pi_k} \sum_{i=1}^n \alpha_{ki}^{(t)} \quad (5.35)$$

$$\frac{\partial}{\partial \pi_k} H_2(\boldsymbol{\pi}, \boldsymbol{\sigma^2}) = \frac{1}{\pi_k} (\beta_k - 1) \quad (5.36)$$

From 5.34 and 5.35 we get:

$$\frac{1}{\pi_k} \sum_{i=1}^n \alpha_{ki}^{(t)} + \frac{1}{\pi_k} (\beta_k - 1) = \lambda \implies \frac{1}{\lambda} \left(\sum_{i=1}^n \alpha_{ki}^{(t)} + (\beta_k - 1) \right) = \pi_k \quad (5.37)$$

Imposing the constraint $\sum_{k=1}^R \pi_k = 1$ and noting that $\sum_{k=1}^R \alpha_{ki}^{(t)} = 1$ we get:

$$\lambda = \sum_{k=1}^R \left(\sum_{i=1}^n \alpha_{ki}^{(t)} + (\beta_k - 1) \right) \implies \lambda = n + \sum_{k=1}^R (\beta_k - 1) \quad (5.38)$$

Thus π_k is given by:

$$\pi_k = \frac{\sum_{i=1}^n \alpha_{ki}^{(t)} + (\beta_k - 1)}{n + \sum_{k=1}^R (\beta_k - 1)} \quad (5.39)$$

We have ignored the inequality constraints $0 \leq \pi_k \leq 1$ on π_k during optimization. By choosing $\beta_k > 1$ we can ensure that these constraints are always satisfied and inactive and hence can be disregarded during optimization.

Estimating $\boldsymbol{\mu}$ Differentiating the M-step objective w.r.t μ_k and noting that $H_2(\boldsymbol{\pi}, \boldsymbol{\sigma^2})$ is not dependent on μ_k , the optimality condition is given by:

$$\frac{\partial}{\partial \mu_k} Q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma^2}, \mathbf{w}, \boldsymbol{\alpha}^{(t)}) = 0 \quad (5.40)$$

Upon substituting the derivative, we get

$$\sum_{i=1}^n \alpha_{ki}^{(t)} \left(\frac{(\mathbf{w}^T \mathbf{z}_i - \mu_k)}{\sigma_k^2} \right) = 0 \quad (5.41)$$

Upon simplifying, μ_k is given by:

$$\mu_k = \frac{\sum_{i=1}^n \alpha_{ki}^{(t)} (\mathbf{w}^T \mathbf{z}_i)}{\sum_{i=1}^n \alpha_{ki}^{(t)}} \quad (5.42)$$

Estimating σ^2 Differentiating the M-step objective w.r.t σ_k^2 the optimality condition is:

$$\frac{\partial}{\partial \sigma_k^2} Q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma^2}, \mathbf{w}, \boldsymbol{\alpha}^{(t)}) + \frac{\partial}{\partial \sigma_k^2} H_2(\boldsymbol{\pi}, \boldsymbol{\sigma^2}) = 0 \quad (5.43)$$

Now,

$$\frac{\partial}{\partial \sigma_k^2} Q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma^2}, \mathbf{w}, \boldsymbol{\alpha}^{(t)}) = \sum_{i=1}^n \alpha_{ki}^{(t)} \left\{ \frac{(\mathbf{w}^T \mathbf{z}_i - \mu_k)^2}{2\sigma_k^4} - \frac{1}{2\sigma_k^2} \right\} \quad (5.44)$$

and

$$\frac{\partial}{\partial \sigma_k^2} H_2(\boldsymbol{\pi}, \boldsymbol{\sigma^2}) = -\frac{(\theta_k + 1)}{\sigma_k^2} + \frac{1}{\gamma_k} \frac{1}{\sigma_k^4} \quad (5.45)$$

Substituting 5.44 and 5.45 in 5.43 we get:

$$\sum_{i=1}^n \alpha_{ki}^{(t)} \left\{ \frac{(\mathbf{w}^T \mathbf{z}_i - \mu_k)^2}{2\sigma_k^4} - \frac{1}{2\sigma_k^2} \right\} - \frac{(\theta_k + 1)}{\sigma_k^2} + \frac{1}{\gamma_k} \frac{1}{\sigma_k^4} = 0 \quad (5.46)$$

Upon simplification, we get:

$$\sigma_k^2 = \frac{2\gamma_k^{-1} + \sum_{i=1}^n \alpha_{ki}^{(t)} (\mathbf{w}^T \mathbf{z}_i - \mu_k)^2}{2(\theta_k + 1) + \sum_{i=1}^n \alpha_{ki}^{(t)}} \quad (5.47)$$

Estimating \mathbf{w} The Lagrangian function for optimizing \mathbf{w} is given by:

$$\mathcal{L}(\mathbf{w}, \lambda_1, \boldsymbol{\lambda_2}) = Q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma^2}, \mathbf{w}, \boldsymbol{\alpha}^{(t)}) - \{\lambda_1(\mathbf{w}^T \mathbf{w} - 1)\} - \{\boldsymbol{\lambda_2}^T \mathbf{G}^T \mathbf{w}\} \quad (5.48)$$

Since \mathbf{w} must satisfy the constraints 5.3, and since $H_2(\boldsymbol{\pi}, \boldsymbol{\sigma^2})$ does not depend on \mathbf{w} the optimality condition is given by:

$$\frac{\partial}{\partial \mathbf{w}} Q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma^2}, \mathbf{w}, \boldsymbol{\alpha}^{(t)}) - \frac{\partial}{\partial \mathbf{w}} \{\lambda_1(\mathbf{w}^T \mathbf{w} - 1)\} - \frac{\partial}{\partial \mathbf{w}} \{\boldsymbol{\lambda_2}^T \mathbf{G}^T \mathbf{w}\} = \mathbf{0} \quad (5.49)$$

In the above equation, λ_1 is the Lagrange multiplier for the constraint $\mathbf{w}^T \mathbf{w} = 1$ and $\boldsymbol{\lambda_2}$ is a $q \times 1$ Lagrange multiplier vector for the constraint $\mathbf{G}^T \mathbf{w} = \mathbf{0}$. Substituting the derivatives, we can write the optimality condition as:

$$\sum_{i=1}^n \sum_{k=1}^R \alpha_{ki}^{(t)} \left\{ -\frac{(\mathbf{w}^T \mathbf{z}_i - \mu_k) \mathbf{z}_i}{\sigma_k^2} \right\} - 2\lambda_1 \mathbf{w} - \mathbf{G} \boldsymbol{\lambda_2} = \mathbf{0} \quad (5.50)$$

Upon simplification, we can write the optimality condition as:

$$\mathbf{b} - \mathbf{A} \mathbf{w} - 2\lambda_1 \mathbf{w} - \mathbf{G} \boldsymbol{\lambda_2} = \mathbf{0} \quad (5.51)$$

where the vector \mathbf{b} and matrix \mathbf{A} are independent of \mathbf{w} and are given by:

$$\mathbf{b} = \sum_{i=1}^n \sum_{k=1}^R \frac{\alpha_{ki}^{(t)} \mu_k}{\sigma_k^2} \mathbf{z}_i \quad (5.52)$$

and

$$\mathbf{A} = \sum_{i=1}^n \sum_{k=1}^R \frac{\alpha_{ki}^{(t)}}{\sigma_k^2} \mathbf{z}_i \mathbf{z}_i^T \quad (5.53)$$

Premultiplying both sides of 5.51 by \mathbf{w}^T and noting the constraints on \mathbf{w} given by 5.3 we get:

$$\mathbf{w}^T \mathbf{b} - \mathbf{w}^T \mathbf{A} \mathbf{w} - 2\lambda_1(\mathbf{1}) - 0 = 0 \quad (5.54)$$

In other words, the λ_1 is given by:

$$\lambda_1 = \frac{1}{2}(\mathbf{w}^T \mathbf{b} - \mathbf{w}^T \mathbf{A} \mathbf{w}) \quad (5.55)$$

Similarly, premultiplying both sides of 5.51 by \mathbf{G}^T and noting the constraint $\mathbf{G}^T \mathbf{w} = \mathbf{0}$ we get:

$$\mathbf{G}^T (\mathbf{b} - \mathbf{A} \mathbf{w}) - 2\lambda_1(\mathbf{0}) - (\mathbf{G}^T \mathbf{G}) \boldsymbol{\lambda}_2 = \mathbf{0} \quad (5.56)$$

Note that \mathbf{G} is of size $q \times L$ with $q > L$ and of full column rank L . This means that $\mathbf{G}^T \mathbf{G}$ is non-singular and invertible. Hence we can solve for $\boldsymbol{\lambda}_2$ to get:

$$\boldsymbol{\lambda}_2 = (\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T (\mathbf{b} - \mathbf{A} \mathbf{w}) \quad (5.57)$$

Substituting λ_1 from 5.55 and $\boldsymbol{\lambda}_2$ from 5.57 into 5.51 we can re-write the optimality condition as:

$$\mathbf{b} - \mathbf{A} \mathbf{w} - (\mathbf{w}^T \mathbf{b} - \mathbf{w}^T \mathbf{A} \mathbf{w}) \mathbf{w} - \mathbf{G} (\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T (\mathbf{b} - \mathbf{A} \mathbf{w}) = \mathbf{0} \quad (5.58)$$

Let

$$\mathbf{P}_G = \mathbf{I}_q - \mathbf{G} (\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T \quad (5.59)$$

Essentially \mathbf{P}_G is an orthogonal projector to the columns of \mathbf{G} . Thus the optimality condition for \mathbf{w} can be simplified to:

$$\mathbf{P}_G (\mathbf{b} - \mathbf{A} \mathbf{w}) - (\mathbf{w}^T \mathbf{b} - \mathbf{w}^T \mathbf{A} \mathbf{w}) \mathbf{w} = \mathbf{0} \quad (5.60)$$

Since \mathbf{w} is a $q \times 1$ vector, for fixed \mathbf{b} and \mathbf{A} this is a system of q **cubic** equations in q variables.

5.5 Breaking up the M-step into two parts

Suppose we have already estimated the current responsibilities $\alpha^{(t)}$ in the E-step. Our goal is to simultaneously solve the M-step equations for π, μ, σ^2 and w . Our strategy for solving the M-step equations is to note that with w fixed, π, μ and σ^2 can be solved explicitly using 5.39, 5.42 and 5.47 respectively. While for fixed π, μ and σ^2 (i.e., fixed b and A), we can calculate w by solving a system of cubic equations 5.60. Hence we can break up the M-step into 2 parts as follows:

1. M-step, Part 1:

In the first part, we keep w fixed at its current value w^* and solve the following maximization problem:

$$\pi^*, \mu^*, \sigma^{2*} = \arg \max_{\pi, \mu, \sigma^2} Q(\pi, \mu, \sigma^2, w^*, \alpha^{(t)}) + H_2(\pi, \sigma^2) \quad (5.61)$$

This is equivalent to solving for π, μ and σ^2 using 5.39, 5.42 and 5.47.

2. M-step, Part 2:

In the second part, we keep π, μ and σ^2 fixed at their current values $\pi^*, \mu^*, \sigma^{2*}$ and solve the maximization problem:

$$w^* = \arg \max_w Q(\pi^*, \mu^*, \sigma^{2*}, w, \alpha^{(t)}) + H_2(\pi^*, \sigma^{2*}) \quad (5.62)$$

This is equivalent to solving the system of cubic equations 5.60 for w .

We repeatedly perform the alternating maximizations in part 1 and part 2 until the absolute value of the change in $H(\pi, \mu, \sigma^2, w)$ from one alternating maximization to the next is below a user specified tolerance ε_M . We used $\varepsilon_M = 10^{-3}$ in our experiments.

There is however a complication that needs to be taken care of when implementing the overall EM step. Note that when excluding w , the M-step objective has closed form solutions for π, μ and σ^2 . In addition, we can show that these closed form solutions result in a maximization of the partial M-step function with fixed w . Can we say the same thing about maximizing w for fixed π, μ and σ^2 ? Does the 2nd part of M-step for optimizing w converge to a maximum? Suppose w^* is a solution to 5.60. The Hessian of the Lagrangian 5.48 is given by:

$$\nabla_{ww}^2 \mathcal{L}(w^*, \lambda_1^*, \lambda_2^*) = -A - 2\lambda_1^* I_q \quad (5.63)$$

Here, λ_1^* is simply the Lagrange multiplier from 5.55 evaluated at w^* . Note that A is a symmetric and positive definite matrix. If $\lambda_1^* > 0$ then we can guarantee that $\nabla_{ww}^2 \mathcal{L}(w^*, \lambda_1^*, \lambda_2^*)$ will be negative definite and the 2nd part of M-step will have found a local maximum. How can we be sure that $\lambda_1^* > 0$? How can we ensure that the M-step solution for π, μ, σ^2 and w is admissible?

One way to solve this problem is to check 5.29 explicitly after convergence of each overall M-step. If 5.29 is not satisfied then we re-initialize the M-step with a random vector w that satisfies $G^T w = 0$ and $\|w\|_2 = 1$ and re-solve the 2 part M-step. This implies we are simply checking for an increase in the objective function 5.31 after each M-step.

5.6 Solving for the projection vector

Solving for \mathbf{w} is equivalent to finding zeros of the set of equations:

$$\mathbf{f}(\mathbf{w}) = \mathbf{P}_G(\mathbf{b} - \mathbf{A}\mathbf{w}) - (\mathbf{w}^T\mathbf{b} - \mathbf{w}^T\mathbf{A}\mathbf{w}) \mathbf{w} \quad (5.64)$$

that satisfy the constraints in 5.3. We consider below three possible cases:

5.6.1 Case 1

Suppose $(\mathbf{b} - \mathbf{A}\mathbf{w}) = \mathbf{0}$. Note that if \mathbf{w} satisfies $(\mathbf{b} - \mathbf{A}\mathbf{w}) = \mathbf{0}$ then equation $\mathbf{f}(\mathbf{w}) = \mathbf{0}$ is satisfied. Since \mathbf{A} is invertible, define a candidate solution for the second part of the M-step as:

$$\mathbf{w}_1 = \mathbf{A}^{-1} \mathbf{b} \quad (5.65)$$

Is this an acceptable solution? If \mathbf{w}_1 satisfies $\mathbf{w}_1^T \mathbf{w}_1 = 1$ and $\mathbf{G}^T \mathbf{w}_1 = \mathbf{0}$ then it is an acceptable solution. However, there is no guarantee that this will be true for \mathbf{w}_1 satisfying 5.65. Thus the solution 5.65 for \mathbf{w}_1 is not admissible.

5.6.2 Case 2

Suppose $(\mathbf{w}^T\mathbf{b} - \mathbf{w}^T\mathbf{A}\mathbf{w}) = 0$. If $(\mathbf{w}^T\mathbf{b} - \mathbf{w}^T\mathbf{A}\mathbf{w}) = 0$ then solving $\mathbf{f}(\mathbf{w}) = \mathbf{0}$ reduces to solving:

$$\mathbf{P}_G(\mathbf{b} - \mathbf{A}\mathbf{w}) = \mathbf{0} \quad (5.66)$$

Another candidate solution for the second part of the M-step is given by:

$$\mathbf{w}_2 = \mathbf{A}^{-1}(\mathbf{b} - \mathbf{G}\boldsymbol{\eta}) \quad (5.67)$$

where $\boldsymbol{\eta}$ is an arbitrary $L \times 1$ vector. Does this \mathbf{w}_2 satisfy the $(L+1)$ distinct constraints $\mathbf{w}_2^T \mathbf{w}_2 = 1$ and $\mathbf{G}^T \mathbf{w}_2 = \mathbf{0}$? The number of variables on the right hand side of 5.67 is L but the number of constraints that they have to satisfy is $(L+1)$ (constraints on \mathbf{w}_2) and so the solution 5.67 for \mathbf{w}_2 is not admissible.

5.6.3 Case 3

Suppose $(\mathbf{w}^T\mathbf{b} - \mathbf{w}^T\mathbf{A}\mathbf{w}) \neq 0$ and we solve for \mathbf{w} that satisfies $\mathbf{f}(\mathbf{w}) = 0$. This means that \mathbf{w} will satisfy:

$$\mathbf{P}_G(\mathbf{b} - \mathbf{A}\mathbf{w}) = (\mathbf{w}^T\mathbf{b} - \mathbf{w}^T\mathbf{A}\mathbf{w}) \mathbf{w} \quad (5.68)$$

Will such a \mathbf{w} satisfy $\mathbf{w}^T \mathbf{w} = 1$ and $\mathbf{G}^T \mathbf{w} = 0$? Premultiplying both sides of 5.68 by \mathbf{G}^T and noting the definition of \mathbf{P}_G in 5.59 we get:

$$\mathbf{G}^T \mathbf{P}_G (\mathbf{b} - \mathbf{A}\mathbf{w}) = \mathbf{0} = (\mathbf{w}^T \mathbf{b} - \mathbf{w}^T \mathbf{A}\mathbf{w}) \mathbf{G}^T \mathbf{w} \quad (5.69)$$

Since $(\mathbf{w}^T \mathbf{b} - \mathbf{w}^T \mathbf{A}\mathbf{w}) \neq \mathbf{0}$, 5.69 gives us:

$$\mathbf{G}^T \mathbf{w} = \mathbf{0} \quad (5.70)$$

Premultiplying both sides of 5.68 by \mathbf{w}^T we get:

$$\mathbf{w}^T \mathbf{P}_G (\mathbf{b} - \mathbf{A}\mathbf{w}) = (\mathbf{w}^T \mathbf{b} - \mathbf{w}^T \mathbf{A}\mathbf{w}) \mathbf{w}^T \mathbf{w} \quad (5.71)$$

From 5.59 and 5.70 we know that $\mathbf{w}^T \mathbf{P}_G = \mathbf{w}^T$ and so 5.71 can be written as:

$$\mathbf{w}^T (\mathbf{b} - \mathbf{A}\mathbf{w}) = (\mathbf{w}^T \mathbf{b} - \mathbf{w}^T \mathbf{A}\mathbf{w}) = (\mathbf{w}^T \mathbf{b} - \mathbf{w}^T \mathbf{A}\mathbf{w}) \mathbf{w}^T \mathbf{w} \quad (5.72)$$

Since $(\mathbf{w}^T \mathbf{b} - \mathbf{w}^T \mathbf{A}\mathbf{w}) \neq \mathbf{0}$, 5.72 gives us:

$$\mathbf{w}^T \mathbf{w} = 1 \quad (5.73)$$

Therefore if $(\mathbf{w}^T \mathbf{b} - \mathbf{w}^T \mathbf{A}\mathbf{w}) \neq \mathbf{0}$ then any solution to $\mathbf{f}(\mathbf{w}) = \mathbf{0}$ will satisfy both $\mathbf{w}^T \mathbf{w} = 1$ and $\mathbf{G}^T \mathbf{w} = 0$. To solve for \mathbf{w} such that $\mathbf{f}(\mathbf{w}) = \mathbf{0}$ we can use Newton's method. The Jacobian of \mathbf{f} w.r.t \mathbf{w} is given by:

$$\mathbf{J}(\mathbf{w}) = -\mathbf{P}_G \mathbf{A} - \mathbf{w} \mathbf{b}^T - (\mathbf{b}^T \mathbf{w}) \mathbf{I}_q + (\mathbf{w}^T \mathbf{A}\mathbf{w}) \mathbf{I}_q + 2 \mathbf{w} \mathbf{w}^T \mathbf{A} \quad (5.74)$$

Assuming, $\mathbf{J}(\mathbf{w})$ is non-singular the basic Newton update is given by:

$$\mathbf{w}^{(i+1)} = \mathbf{w}^{(i)} - [\mathbf{J}(\mathbf{w}^{(i)})]^{-1} \mathbf{f}(\mathbf{w}^{(i)}) \quad (5.75)$$

To avoid problems caused by intermediate singularity of \mathbf{J} we can replace $[\mathbf{J}(\mathbf{w}^{(i)})]^{-1}$ by a modified version $[\mathbf{J}(\mathbf{w}^{(i)}) + \eta \mathbf{I}_q]^{-1}$ for a sufficiently large η such that $[\mathbf{J}(\mathbf{w}^{(i)}) + \eta \mathbf{I}_q]$ is non-singular (see [19] for more details). For instance, we could use the Levenberg-Marquardt modification of the Newton's method which essentially does this replacement. In this case, the update equation for \mathbf{w} can be written as:

$$\mathbf{w}^{(i+1)} = \mathbf{w}^{(i)} - [\mathbf{J}(\mathbf{w}^{(i)}) + \eta \mathbf{I}_q]^{-1} \mathbf{f}(\mathbf{w}^{(i)}) \quad (5.76)$$

The projection vector \mathbf{w} is initialized for the Newton iteration as follows:

$$\mathbf{w}^{init} = \mathbf{P}_G \mathbf{A}^{-1} \mathbf{b} \quad (5.77)$$

$$\mathbf{w}^{init} = \frac{\mathbf{w}^{init}}{\|\mathbf{w}^{init}\|_2} \quad (5.78)$$

The complete EM algorithm pseudocode is given in Fig. 2.

EM algorithm for estimating the PMOG model

Require: $q \times n$ matrix \mathbf{Z} , number of Gaussian components in MOG R , $q \times L$ matrix \mathbf{G} ($L < q$), Dirichlet prior parameter vector $\boldsymbol{\beta}$ for $\boldsymbol{\pi}$, Inverse Gamma prior parameter vectors $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}$ for $\boldsymbol{\sigma}^2$ and convergence tolerances ε_{rel} , ε_M

- 1: Select $\mathbf{w}^{(0)}$ randomly such that $\mathbf{G}^T \mathbf{w}^{(0)} = \mathbf{0}$ and $\|\mathbf{w}^{(0)}\|_2 = 1$. Initialize the $R \times 1$ vectors $\boldsymbol{\pi}^{(0)}, \boldsymbol{\mu}^{(0)}, \boldsymbol{\sigma}^{2(0)}$ using the k -means algorithm on the projected points $\mathbf{Z}^T \mathbf{w}^{(0)}$ and set $found = 0$ and $t = 0$
- 2: **while** $found = 0$ **do**
- 3: **E-step:** Calculate the responsibilities $\boldsymbol{\alpha}^{(t)}$ using the current parameter estimates $\boldsymbol{\pi}^{(t)}, \boldsymbol{\mu}^{(t)}, \boldsymbol{\sigma}^{2(t)}, \mathbf{w}^{(t)}$ and equation 5.27.
- 4: To start the **M-step**, set $\mathbf{w}^* = \mathbf{w}^{(t)}$.
- 5: **M-step, Part 1:** Given $\boldsymbol{\alpha}^{(t)}$ and \mathbf{w}^* optimize the M-step objective function w.r.t $\boldsymbol{\pi}, \boldsymbol{\mu}$ and $\boldsymbol{\sigma}^2$:

$$\boldsymbol{\pi}^*, \boldsymbol{\mu}^*, \boldsymbol{\sigma}^{2*} = \arg \max_{\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2} Q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \mathbf{w}^*, \boldsymbol{\alpha}^{(t)}) + H_2(\boldsymbol{\pi}, \boldsymbol{\sigma}^2) \quad (5.79)$$

This problem has an explicit solution given by equations 5.39, 5.42 and 5.47.

- 6: **M-step, Part 2:** Given $\boldsymbol{\alpha}^{(t)}$ and $\boldsymbol{\pi}^*, \boldsymbol{\mu}^*, \boldsymbol{\sigma}^{2*}$ optimize the M-step objective function w.r.t \mathbf{w} :

$$\mathbf{w}^* = \arg \max_{\mathbf{w}} Q(\boldsymbol{\pi}^*, \boldsymbol{\mu}^*, \boldsymbol{\sigma}^{2*}, \mathbf{w}, \boldsymbol{\alpha}^{(t)}) + H_2(\boldsymbol{\pi}^*, \boldsymbol{\sigma}^{2*}) \quad (5.80)$$

Solving 5.80 is equivalent to finding the roots of the equation 5.64 which can be done with Newton or quasi-Newton techniques as in 5.75 and 5.76. We initialize \mathbf{w} as per 5.77 and solve 5.64 to give \mathbf{w}^* . Steps 1 and 2 are alternated repeatedly in each cycle until the absolute value of the change in $H(\boldsymbol{\pi}^*, \boldsymbol{\mu}^*, \boldsymbol{\sigma}^{2*}, \mathbf{w}^*)$ from one cycle to the next is $< \varepsilon_M$. Since 5.80 is a non-concave maximization problem, the solution to 5.64 might converge to a minimum instead of a maximum. Hence we iterate as follows:

- 7: $validsolution = 0$
- 8: **while** $validsolution = 0$ **do**
- 9: **if** equation 5.29 is satisfied **then**
- 10: $validsolution = 1$, set $\boldsymbol{\pi}^{(t+1)} = \boldsymbol{\pi}^*, \boldsymbol{\mu}^{(t+1)} = \boldsymbol{\mu}^*, \boldsymbol{\sigma}^{2(t+1)} = \boldsymbol{\sigma}^{2*}, \mathbf{w}^{(t+1)} = \mathbf{w}^*$
- 11: **else**
- 12: Initialize \mathbf{w}^* randomly such that $\mathbf{G}^T \mathbf{w}^* = \mathbf{0}$ and $\|\mathbf{w}^*\|_2 = 1$. With this initialization re-solve the 2-part **M-step**
- 13: **end if**
- 14: **end while**
- 15: If $\text{abs} [H(\boldsymbol{\pi}^{(t+1)}, \boldsymbol{\mu}^{(t+1)}, \boldsymbol{\sigma}^{2(t+1)}, \mathbf{w}^{(t+1)}) - H(\boldsymbol{\pi}^{(t)}, \boldsymbol{\mu}^{(t)}, \boldsymbol{\sigma}^{2(t)}, \mathbf{w}^{(t)})] \leq \varepsilon^*$, set $found = 1$. In the above equation $\varepsilon^* = \varepsilon_{rel} \left(\text{mean} \text{abs} \left[H(\boldsymbol{\pi}^{(t)}, \boldsymbol{\mu}^{(t)}, \boldsymbol{\sigma}^{2(t)}, \mathbf{w}^{(t)}) \right] \right)$.
- 16: $t \leftarrow t + 1$
- 17: **end while**

Figure 2: EM algorithm for estimating the PMOG model.

6 The BSS problem

In this section, we will try to answer the following two questions;

- What is the BSS problem and why is its solution difficult?
- What is the connection between BSS, differential entropy and source correlation?

6.1 The linear BSS problem

We consider the general version of a linear BSS problem. Vectors $\mathbf{x}_i \in \mathbf{R}^p$ are generated from latent source vectors $\mathbf{s}_i \in \mathbf{R}^q$ as follows:

$$\mathbf{x}_i = \boldsymbol{\mu} + \mathbf{A} \mathbf{s}_i + \boldsymbol{\varepsilon}_i \quad (6.1)$$

In this linear mixing model, \mathbf{A} is the $p \times q$ mixing matrix with $p > q$, $\boldsymbol{\mu}$ is the $p \times 1$ mean vector and $\boldsymbol{\varepsilon}_i$ is the $p \times 1$ noise vector with distribution $\mathcal{N}(\boldsymbol{\varepsilon}_i | \mathbf{0}, \sigma^2 \mathbf{I}_p)$. We assume without loss of generality that each component of \mathbf{s}_i has zero mean and unit variance i.e., $E(s_{ij}) = 0$ and $E(s_{ij}^2) = 1$. Thus, the second order statistics of \mathbf{s}_i can be summarized as:

$$E(\mathbf{s}_i) = \mathbf{0} \quad (6.2)$$

$$E(\mathbf{s}_i \mathbf{s}_i^T) = \boldsymbol{\Sigma}_s \quad (6.3)$$

where $\boldsymbol{\Sigma}_s$ is the unknown $q \times q$ correlation matrix between the source components. Note that the diagonal elements of $\boldsymbol{\Sigma}_s$ are ones (1's). No distributional assumptions are made on the density of vector \mathbf{s}_i . One simply requires that the components of vector \mathbf{s}_i are minimally dependent (or maximally independent) on each other in an information theoretic sense. Given n independent realizations of the vector \mathbf{x}_i (generated from n independent realizations of sources \mathbf{s}_i), the goal is to estimate the unknown sources \mathbf{s}_i when \mathbf{A} , $\boldsymbol{\mu}$ and the noise variance σ^2 are unknown.

6.2 Why is the BSS problem difficult?

The BSS problem is difficult because the estimation of mixing parameters \mathbf{A} , $\boldsymbol{\mu}$ and σ^2 is coupled with the estimation of latent sources \mathbf{s}_i . Note that the equation 6.1 gives us the conditional density of \mathbf{x}_i given \mathbf{s}_i as:

$$P(\mathbf{x}_i | \mathbf{s}_i) \sim \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu} + \mathbf{A} \mathbf{s}_i, \sigma^2 \mathbf{I}_p) \quad (6.4)$$

The standard approach of estimating mixing parameters by maximum likelihood (ML) would require the computation of marginal density $P(\mathbf{x}_i)$. If the joint source density is $P(\mathbf{s}_i)$, then we can write:

$$P(\mathbf{x}_i) = \int P(\mathbf{x}_i | \mathbf{s}_i) P(\mathbf{s}_i) d\mathbf{s}_i \quad (6.5)$$

If the above integral is tractable then we can compute $P(\mathbf{x}_i)$ and subsequently the ML solution for $\boldsymbol{\mu}$, \mathbf{A} and σ^2 . However, we are not given any parametric form for $P(\mathbf{s}_i)$. We are simply given that the components of \mathbf{s}_i are maximally independent with 0 mean, unit-variance and unknown correlation structure $\boldsymbol{\Sigma}_s$. This loose specification of $P(\mathbf{s}_i)$ is the root cause of difficulty in the BSS problem. Even if the mixing parameters $\boldsymbol{\mu}$, \mathbf{A} and σ^2 are known the computation of posterior mean or maximum a posteriori (MAP) estimate of \mathbf{s}_i would require some specification of the density of \mathbf{s}_i since

$$P(\mathbf{s}_i | \mathbf{x}_i) \propto P(\mathbf{x}_i | \mathbf{s}_i) P(\mathbf{s}_i) \quad (6.6)$$

Thus it is clear that the BSS problem is non-trivial.

6.3 Measuring dependence in BSS

We start this subsection with some fundamental definitions from information theory followed by a detailed study of mutual information as a contrast function for BSS.

Definition 6.1. Kullback-Leibler divergence: Given probability density functions $P(\mathbf{y})$ and $Q(\mathbf{y})$, the Kullback-Leibler divergence (or KL distance) between P and Q is given by:

$$KL(P, Q) = \int P(\mathbf{y}) \log \left[\frac{P(\mathbf{y})}{Q(\mathbf{y})} \right] d\mathbf{y} \quad (6.7)$$

For any P and Q it is true that $KL(P, Q) \geq 0$.

Definition 6.2. Differential entropy: Given a random variable \mathbf{y} with density $P(\mathbf{y})$, the differential entropy of \mathbf{y} is defined to be:

$$H(\mathbf{y}) = - \int P(\mathbf{y}) \log P(\mathbf{y}) d\mathbf{y} \quad (6.8)$$

Definition 6.3. Non-Gaussianity: A concept related to differential entropy is the non-Gaussianity (NG) of a distribution. Given a $q \times 1$ random vector \mathbf{y} with density $P(\mathbf{y})$, mean $\boldsymbol{\mu}_{\mathbf{y}}$ and co-variance $\boldsymbol{\Sigma}_{\mathbf{y}}$, suppose $\mathcal{N}(\mathbf{y} | \boldsymbol{\mu}_{\mathbf{y}}, \boldsymbol{\Sigma}_{\mathbf{y}})$ is a Normal density with the same mean and co-variance as \mathbf{y} . Then the non-Gaussianity of \mathbf{y} is defined to be (see [3]):

$$NG(\mathbf{y}) = KL(P(\mathbf{y}), \mathcal{N}(\mathbf{y} | \boldsymbol{\mu}_{\mathbf{y}}, \boldsymbol{\Sigma}_{\mathbf{y}})) \quad (6.9)$$

An important property of non-Gaussianity is invariance to invertible linear transformations i.e.,

$$NG(\mathbf{a} + \mathbf{B}\mathbf{y}) = NG(\mathbf{y}) \quad (6.10)$$

for any non-singular matrix \mathbf{B} and vector \mathbf{a} .

Remark 6.4. Differential entropy and non-Gaussianity: Invoking the definition of KL divergence in 6.9, we get:

$$NG(\mathbf{y}) = \int P(\mathbf{y}) \log \left[\frac{P(\mathbf{y})}{\mathcal{N}(\mathbf{y} | \boldsymbol{\mu}_{\mathbf{y}}, \boldsymbol{\Sigma}_{\mathbf{y}})} \right] d\mathbf{y} \quad (6.11)$$

After some algebraic manipulations and noting that \mathbf{y} has co-variance $\boldsymbol{\Sigma}_{\mathbf{y}}$ we get:

$$NG(\mathbf{y}) = -H(\mathbf{y}) + \frac{q}{2} \log 2\pi e + \frac{1}{2} \log[\det(\boldsymbol{\Sigma}_{\mathbf{y}})] \quad (6.12)$$

Note that in equation 6.12, q is the length of the vector \mathbf{y} .

Definition 6.5. Mutual information: An information theoretic measure of the dependence between components of a $q \times 1$ random vector \mathbf{s} with density $P(\mathbf{s})$ is the KL distance between $P(\mathbf{s})$ and the density of \mathbf{s} when its components are independent i.e., $P^{id}(\mathbf{s}) = \prod_{j=1}^q P_j(s_j)$ where P_j is the marginal density of s_j . This KL distance is also called the mutual information [3] between the components of \mathbf{s} :

$$I(\mathbf{s}) = KL(P(\mathbf{s}), P^{id}(\mathbf{s})) \quad (6.13)$$

Substituting the definitions of KL distance and P^{id} and simplifying we get the classical expression for mutual information between components of \mathbf{s} :

$$I(\mathbf{s}) = \sum_{j=1}^q H(s_j) - H(\mathbf{s}) \quad (6.14)$$

Suppose the vector \mathbf{s} is generated from another $q \times 1$ vector \mathbf{z} with density $P(\mathbf{z})$ via a non-singular linear transformation:

$$\mathbf{s} = \mathbf{W}^T \mathbf{z} = \begin{pmatrix} \mathbf{w}_1^T \\ \mathbf{w}_2^T \\ \vdots \\ \mathbf{w}_q^T \end{pmatrix} \mathbf{z} \quad (6.15)$$

then we can re-write equation 6.14 as:

$$I(\mathbf{s}) = \sum_{j=1}^q H(s_j) - H(\mathbf{z}) - \log[\det(\mathbf{W}^T)] \quad (6.16)$$

where $H(\mathbf{z})$ is the differential entropy of \mathbf{z} . If we are given that \mathbf{s} satisfies $E(\mathbf{s}) = \mathbf{0}$ and $E(\mathbf{s}\mathbf{s}^T) = \boldsymbol{\Sigma}_{\mathbf{s}}$ with 1's on the diagonal then \mathbf{z} must have $\mathbf{0}$ mean and co-variance $\boldsymbol{\Sigma}_{\mathbf{z}}$ satisfying:

$$\boldsymbol{\Sigma}_{\mathbf{s}} = \mathbf{W}^T \boldsymbol{\Sigma}_{\mathbf{z}} \mathbf{W} \quad (6.17)$$

Taking determinants on both sides, it is easy to see that:

$$\text{abs}(\det(\mathbf{W}^T)) = \frac{\det(\boldsymbol{\Sigma}_s)^{\frac{1}{2}}}{\det(\boldsymbol{\Sigma}_z)^{\frac{1}{2}}} \quad (6.18)$$

Substituting 6.18 into 6.16 we see that:

$$I(\mathbf{s}) = \underbrace{\left\{ \sum_{j=1}^q H(s_j) - \frac{1}{2} \log[\det(\boldsymbol{\Sigma}_s)] \right\}}_{\text{dependent on } \mathbf{W}^T} + \underbrace{\left\{ \frac{1}{2} \log[\det(\boldsymbol{\Sigma}_z)] - H(\mathbf{z}) \right\}}_{\text{dependent only on } P(\mathbf{z})} \quad (6.19)$$

This equation is identical to equation (16) in [3]. This can be seen by replacing differential entropy by non-Gaussianity in 6.19 using 6.12, noting that individual components s_j of \mathbf{s} have unit variance and using the invariance property 6.10 for non-Gaussianity. Note that the diagonal elements of $\boldsymbol{\Sigma}_s$ are ones and so the correlation $C(\mathbf{y})$ as defined in [3] is equivalent to $-\frac{1}{2} \log[\det(\boldsymbol{\Sigma}_s)]$. It is clear that given a density $P(\mathbf{z})$ for \mathbf{z} , the second term is independent of the linear transformation \mathbf{W}^T . Thus the dependence between components of \mathbf{s} is fully captured by the first term alone which depends on \mathbf{W}^T (see 6.17 and 6.15).

Remark 6.6. Note on the $-\frac{1}{2} \log[\det(\boldsymbol{\Sigma}_s)]$ term: First, note that $\boldsymbol{\Sigma}_s$ is a correlation matrix with 1s on the diagonal. If $\phi_1, \phi_2, \dots, \phi_q$ are its eigenvalues then it follows that

$$\text{trace}(\boldsymbol{\Sigma}_s) = q = \sum_{i=1}^q \phi_i \quad (6.20)$$

$$\det(\boldsymbol{\Sigma}_s) = \prod_{i=1}^q \phi_i \quad (6.21)$$

Therefore it follows that:

$$\log[\det(\boldsymbol{\Sigma}_s)] = \sum_{i=1}^q \log \phi_i \quad (6.22)$$

$$= q \left[\frac{1}{q} \sum_{i=1}^q \log \phi_i \right] \quad (6.23)$$

$$\leq q \log \left[\frac{1}{q} \sum_{i=1}^q \phi_i \right] \quad \text{by concavity of log function} \quad (6.24)$$

$$= q \log \left[\frac{1}{q} q \right] \quad \text{using the trace condition in 6.20} \quad (6.25)$$

$$= 0 \quad (6.26)$$

Thus we see that

$$\log[\det(\Sigma_s)] \leq 0 \text{ or} \quad (6.27)$$

$$-\frac{1}{2} \log[\det(\Sigma_s)] \geq 0 \quad (6.28)$$

The minimal value of $-\frac{1}{2} \log[\det(\Sigma_s)]$ is 0 and is attained when $\det(\Sigma_s) = 1$. Since Σ_s is a correlation matrix with 1s on the diagonal the only way $\det(\Sigma_s) = 1$ can hold is if $\Sigma_s = \mathbf{I}_q$.

The term $-\frac{1}{2} \log[\det(\Sigma_s)]$ attains a minimal value of 0 when $\Sigma_s = \mathbf{I}_q$.

The optimization problem for minimizing dependence between components of s can therefore be written as:

$$\min_{\mathbf{W}^T} f(s) = \underbrace{\left\{ \sum_{j=1}^q H(s_j) \right\}}_{\text{encourages non-Gaussianity}} + \psi \underbrace{\left\{ -\frac{1}{2} \log[\det(\Sigma_s)] \right\}}_{\geq 0, \text{ encourages } \Sigma_s = \mathbf{I}_q} \quad (6.29)$$

where: (6.30)

$$s_j = \mathbf{w}_j^T \mathbf{z}, \quad \text{abs}(\det(\mathbf{W}^T)) = \frac{\det(\Sigma_s)^{\frac{1}{2}}}{\det(\Sigma_z)^{\frac{1}{2}}} \quad (6.31)$$

subject to: (6.32)

$$\Sigma_s = \mathbf{W}^T \Sigma_z \mathbf{W} \quad (6.33)$$

Since the sources s_j have unit variance, using the property 6.12 we get:

$$NG(s_j) = -H(s_j) + \frac{1}{2} \log 2\pi e \quad (6.34)$$

Minimizing $f(s)$ is therefore a problem of minimizing the sum of 2 different terms. The first term measures the differential entropy of each component s_j . As seen from 6.34, minimizing the differential entropy under the unit variance constraint is equivalent to maximizing the non-Gaussianity $NG(s_j)$. Thus minimizing the first term encourages non-Gaussianity. The second term measures the 2nd order cross-correlation between sources. Since this term is minimized when sources are uncorrelated, it encourages uncorrelatedness of the sources. The weighting constant ψ serves to balance the two terms in the objective function. It is instructive to note the form of the objective function for various values of ψ :

- When $\psi = 0$, the 2nd term drops out. In this case, minimization of differential entropy (or equivalently maximization of non-Gaussianity) of sources is considered much more important than minimizing correlatedness. Without making any additional assumptions on the correlation structure of s , the projection vectors \mathbf{w}_i are constrained by the relation $\mathbf{w}_i^T \Sigma_z \mathbf{w}_i = 1$.

-
- When $\psi = 1$, then minimizing 6.29 is equivalent to minimization of mutual information $I(\mathbf{s})$. In this case, equal importance is given to maximizing non-Gaussianity and uncorrelatedness. As before, the projection vectors \mathbf{w}_i satisfy $\mathbf{w}_i^T \Sigma_{\mathbf{z}} \mathbf{w}_i = 1$.
 - When $\psi = \infty$, the 2nd term is forced to become 0 at $\Sigma_{\mathbf{s}} = \mathbf{I}_q$. In this case, non-Gaussianity is maximized under the uncorrelated source assumption. This means that the projection vectors are constrained by: $\mathbf{w}_i^T \Sigma_{\mathbf{z}} \mathbf{w}_j = \delta_{ij}$ where $\delta_{ij} = 1$ if $i = j$ and 0 otherwise.

The objective function for minimization in 6.29 can also be re-written as:

$$\min_{\mathbf{W}^T} f(\mathbf{s}) = \underbrace{\left\{ \sum_{j=1}^q H(s_j) \right\}}_{\text{encourages non-Gaussianity}} + \psi \underbrace{\left\{ -\frac{1}{2} \log[\det(\Sigma_{\mathbf{z}})] - \log[\text{abs}(\det(\mathbf{W}^T))] \right\}}_{\geq 0, \text{ encourages } \Sigma_{\mathbf{s}} = \mathbf{I}_q} \quad (6.35)$$

where: (6.36)

$$s_j = \mathbf{w}_j^T \mathbf{z}, \quad \text{abs}(\det(\mathbf{W}^T)) = \frac{\det(\Sigma_{\mathbf{s}})^{\frac{1}{2}}}{\det(\Sigma_{\mathbf{z}})^{\frac{1}{2}}} \quad (6.37)$$

subject to: (6.38)

$$\Sigma_{\mathbf{s}} = \mathbf{W}^T \Sigma_{\mathbf{z}} \mathbf{W} \quad (6.39)$$

7 Solving the BSS problem

Solving the BSS problem means estimating both the mixing parameters $\boldsymbol{\mu}, \mathbf{A}$ and σ^2 and the latent sources \mathbf{s}_i given n mixed vectors \mathbf{x}_i . We will try to answer the following questions in this section:

- What are the main solution approaches when the sources are assumed to be uncorrelated and how does the work of Hyvärinen et al. [12] relate to the problem of BSS?
- How should the solution approach change when non-zero second order source correlation is allowed?

7.1 Case I: Uncorrelated sources

In this case, we assume that $\Sigma_{\mathbf{s}} = \mathbf{I}_q$ i.e., the sources are uncorrelated. There are two main solution techniques in this case.

7.1.1 (1) The solution by Attias et al. under exact independence

Attias et al. [1] in a seminal paper described a general solution to the linear BSS problem under exact independence. He assumed that each component of the vector \mathbf{s}_i is described by a mixture of Gaussians (MOG) density. The logic for this was that since the MOG is a very flexible density (given sufficient components in the mixture), it should be able to describe more complicated and non-Gaussian source densities. Given the fact that components of \mathbf{s}_i are exactly independent, we have

$$P(\mathbf{s}_i) = \prod_{j=1}^q P_j(s_{ij}) \quad (7.1)$$

where s_{ij} is the j th component of vector \mathbf{s}_i and P_j is the marginal density of the j th component of \mathbf{s}_i . In other words, the source density $P(\mathbf{s}_i)$ has a very flexible parametric form of a product of q MOG densities. This type of a density for \mathbf{s}_i is also called a "factorial" MOG which is a special case of a mixture of co-adaptive Gaussians densities. Attias et al. also showed that under this parametric form the integral 6.5 is analytically tractable and the density $P(\mathbf{x}_i)$ is also a mixture of co-adaptive Gaussians (although not "factorial" MOG as in the case of \mathbf{s}_i). Attias et al. also derive an exact EM algorithm for obtaining the ML solution for $\mathbf{A}, \boldsymbol{\mu}, \sigma^2$ as well as the MOG parameters for the source density $P(\mathbf{s}_i)$. The only limitation of this algorithm is the computational intractability for large problems. Attias et al. note that a 13 source mixture with each source described by a 3 component MOG would require 3^{13} sums in each E-step, making it computationally intractable. To overcome this problem Attias et al. propose a variational approximation to the ML solution instead of the EM algorithm. In summary, an exact solution for the BSS problem under exact independence exists, but is computationally intractable for large problems (greater than 13 sources) and therefore one has to resort to approximate solutions.

7.1.2 (2) PPCA and least squares based solution

An alternative and much simpler approach is to use an approximation to the density of \mathbf{s}_i only for the purposes of computing $\mathbf{A}, \boldsymbol{\mu}$ and σ^2 . This approximation should be such that it simplifies computation of the integral 6.5 making the ML solution possible under the approximate density of \mathbf{s}_i . A density that satisfies this requirement is $P(\mathbf{s}_i) = \mathcal{N}(\mathbf{s}_i | \mathbf{0}, \mathbf{I}_q)$. Under this assumption, the model 6.1 simply reduces to the probabilistic PCA (PPCA) model of Tipping et al. [20]. The ML solution for $\mathbf{A}, \boldsymbol{\mu}$ and noise variance σ^2 under the PPCA model (when \mathbf{s}_i is Gaussian) is known. Let the $p \times 1$ vector $\bar{\mathbf{x}}$ be the mean of observations \mathbf{x}_i and \mathbf{S}_x be the $p \times p$ sample co-variance i.e.,

$$\bar{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \quad (7.2)$$

$$\mathbf{S}_{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}}) (\mathbf{x}_i - \bar{\mathbf{x}})^T \quad (7.3)$$

Suppose

$$\mathbf{S}_{\mathbf{x}} = \mathbf{U} \Lambda \mathbf{U}^T \quad (7.4)$$

is the eigen decomposition of the symmetric matrix $\mathbf{S}_{\mathbf{x}}$. Here \mathbf{U} is a $p \times p$ matrix of eigenvectors of $\mathbf{S}_{\mathbf{x}}$ and Λ is a $p \times p$ diagonal matrix containing the corresponding eigenvalues of $\mathbf{S}_{\mathbf{x}}$.

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \dots & \ddots & 0 \\ 0 & \dots & \dots & \lambda_p \end{pmatrix} \quad (7.5)$$

Suppose we order the eigenvalues of $\mathbf{S}_{\mathbf{x}}$ such that $\lambda_1 \geq \lambda_2 \dots \geq \lambda_p \geq 0$. Let \mathbf{U}_q be a $p \times q$ submatrix of \mathbf{U} containing the eigenvectors corresponding to the q largest eigenvalues $\lambda_1, \dots, \lambda_q$ and Λ_q be the $q \times q$ submatrix of Λ with $\lambda_1, \dots, \lambda_q$ on the diagonal. Then as shown in Tipping et al. the ML solution for mixing parameters under the PPCA model is given by:

$$\hat{\mu} = \bar{\mathbf{x}} \quad (7.6)$$

$$\hat{\sigma}^2 = \left(\frac{1}{p-q} \right) \sum_{i=q+1}^p \lambda_i \quad (7.7)$$

$$\hat{\mathbf{A}} = \mathbf{U}_q (\Lambda_q - \hat{\sigma}^2 \mathbf{I}_q)^{1/2} \mathbf{Q}^T \quad (7.8)$$

$$\mathbf{Q}^T \mathbf{Q} = \mathbf{I}_q \quad (7.9)$$

The ML solution is unique upto an arbitrary $q \times q$ orthogonal matrix \mathbf{Q} . Following this step, we can estimate the sources \hat{s}_i using least squares to get:

$$\hat{s}_i = \mathbf{Q} (\Lambda_q - \hat{\sigma}^2 \mathbf{I}_q)^{-\frac{1}{2}} \mathbf{U}_q^T (\mathbf{x}_i - \hat{\mu}) \quad (7.10)$$

The advantage of this approach is that it decouples the estimation of mixing parameters $\mathbf{A}, \mu, \sigma^2$ from the sources \mathbf{s}_i . This approach is used for example in Hyvarinen et al. ([14]) with $\sigma^2 = 0$ and Beckmann et al. ([2]) with $\sigma^2 \neq 0$.

Now suppose

$$\mathbf{z}_i = (\boldsymbol{\Lambda}_q - \hat{\sigma}^2 \mathbf{I}_q)^{-\frac{1}{2}} \mathbf{U}_q^T (\mathbf{x}_i - \hat{\mu}) \text{ and} \quad (7.11)$$

$$\mathbf{Q} = \mathbf{W}^T = \begin{pmatrix} \mathbf{w}_1^T \\ \mathbf{w}_2^T \\ \vdots \\ \mathbf{w}_q^T \end{pmatrix} \quad (7.12)$$

then

$$\hat{\mathbf{s}}_i = \mathbf{W}^T \mathbf{z}_i = \begin{pmatrix} \mathbf{w}_1^T \mathbf{z}_i \\ \mathbf{w}_2^T \mathbf{z}_i \\ \vdots \\ \mathbf{w}_q^T \mathbf{z}_i \end{pmatrix} \quad (7.13)$$

Since \mathbf{Q} is orthogonal the vectors \mathbf{w}_i satisfy the constraints

$$\mathbf{w}_i^T \mathbf{w}_j = \delta_{ij} \quad (7.14)$$

where $\delta_{ij} = 1$ if $i = j$ and 0 otherwise. Since \mathbf{W}^T is orthogonal, the second term in 6.35 becomes independent of \mathbf{W}^T and so the objective function $f(\hat{\mathbf{s}}_i)$ reduces to a sum of the differential entropies of the components of $\hat{\mathbf{s}}_i$:

$$\min_{\mathbf{W}^T} f(\hat{\mathbf{s}}_i) \propto \underbrace{\left\{ \sum_{j=1}^q H(\hat{s}_{ij}) \right\}}_{\text{encourages non-Gaussianity}} \quad (7.15)$$

where: (7.16)

$$\hat{s}_{ij} = \mathbf{w}_j^T \mathbf{z}_i \quad (7.17)$$

It is worth noting that since \mathbf{W}^T is constrained to be orthogonal, no apriori assumptions can be made on the correlation structure $\Sigma_{\hat{\mathbf{s}}_i}$ of $\hat{\mathbf{s}}_i$. Thus the components of $\hat{\mathbf{s}}_i$ will have co-variance $\Sigma_{\hat{\mathbf{s}}_i} = \mathbf{W}^T \Sigma_{\mathbf{z}_i} \mathbf{W}$ where $\Sigma_{\mathbf{z}_i}$ is the co-variance of \mathbf{z}_i . Now $\Sigma_{\mathbf{z}_i} = \mathbf{I}_q$ only if $\hat{\sigma}^2 = 0$. This means that unless $\hat{\sigma}^2 = 0$, the components of $\hat{\mathbf{s}}_i$ will **not** have unit variance and 0 cross-correlation.

7.1.3 Differential entropy approximations of Hyvarinen et al.

In a seminal paper [12], Hyvarinen et al. proposed approximations to the differential entropy function $H(x)$ of a random variable with density $P(x)$. The key idea in this work is to approximate

$P(x)$ by a maximum entropy distribution (MED) given estimates of the expectations of m functions G_i of x i.e., given $E_x[G_i(x)] = c_i$. The solution to this problem is well known [6]:

$$p^{med}(x) = a_0 \exp \left[\sum_{i=1}^m a_i G_i(x) \right] \quad (7.18)$$

where a_0, a_1, \dots, a_m are constants. These $(m+1)$ constants can be solved for by simultaneously solving the m expectation equations under the MED density $E_x[G_i(x)] = c_i$ together with the normalizing equation $\int p^{med}(x) dx = 1$. Simultaneous solution of this $(m+1)$ system of non-linear equations is difficult. Hence, Hyvarinen et al. proposed a solution in which it is assumed that the density $P(x)$ is "not very far from a Gaussian distribution". Under these conditions, using the simplified form of $p^{med}(x)$, Hyvarinen et al. derived approximations to the differential entropy $H(x)$ (see [11] for details). In summary, suppose x has mean 0 and variance σ^2 . Let ν be a Gaussian variable with the same mean and variance as x . The "near Gaussian" MED density approximation of Hyvarinen et al. for a single expectation constraint using a function $\bar{G}(x)$ is given by:

$$p_{Gaussian}^{med}(x) = \mathcal{N}(x | 0, \sigma^2) \{1 + c G(x)\} \quad (7.19)$$

The function $\bar{G}(x)$ is related to its normalized version $G(x)$ by the equation:

$$G(x) = \frac{1}{\delta}(\bar{G}(x) + \alpha_1 x + \alpha_2 x^2 + \gamma) \quad (7.20)$$

Here $\bar{G}(x)$ is any function of x not necessarily even or odd. The 4 constants $\alpha_1, \alpha_2, \gamma$ and δ are determined from the relations:

$$\int \mathcal{N}(x | 0, \sigma^2) x^k G(x) dx = 0, \text{ where } k = 0, 1, 2 \quad (7.21)$$

$$\int \mathcal{N}(x | 0, \sigma^2) G(x) G(x) dx = 1 \quad (7.22)$$

Then the differential entropy approximation developed in Hyvarinen et al. using $p_{Gaussian}^{med}(x)$ from 7.19 is given by:

$$H(x) \approx H(\nu) - \frac{1}{2\delta^2} \{E_x[\bar{G}(x)] - E_\nu[\bar{G}(\nu)]\}^2 \quad (7.23)$$

Note that the term $E_\nu[\bar{G}(\nu)]$ is 0 in case $\bar{G}(x)$ is an odd function. Substituting the differential entropy of a Gaussian random variable with mean 0 and variance σ^2 in 7.23 we get:

$$H(x) \approx \frac{1}{2} \log 2\pi e + \frac{1}{2} \log \sigma^2 - \frac{1}{2\delta^2} \{E_x[\bar{G}(x)] - E_\nu[\bar{G}(\nu)]\}^2 \quad (7.24)$$

$$\nu = \text{Random variable with density } \mathcal{N}(\nu | 0, \sigma^2) \quad (7.25)$$

$$\sigma^2 = \text{Var}(x) \quad (7.26)$$

Given a set of observed samples of x : x_1, x_2, \dots, x_n , the objective function in 7.24 can be estimated by replacing the expected values by sample averages.

Hyvärinen et al. proposed a solution in which the first component of \hat{s}_i is estimated by minimizing the differential entropy of the empirical density of $\hat{s}_{i1} = \mathbf{w}_1^T \mathbf{z}_i$ over n realizations using the approximation $H(\hat{s}_{i1})$ from 7.24. This minimization is carried out under the constraint $\mathbf{w}_1^T \mathbf{w}_1 = 1$. The second component is estimated similarly by minimizing the differential entropy of the empirical density of $\hat{s}_{i2} = \mathbf{w}_2^T \mathbf{z}_i$ over n realizations while imposing the constraint $\mathbf{w}_2^T \mathbf{w}_1 = 0$ and $\mathbf{w}_2^T \mathbf{w}_2 = 1$. In general, the m th component is estimated by minimizing the differential entropy of the empirical density of $\mathbf{w}_m^T \mathbf{z}_i$ over n realizations subject to the constraint:

$$\mathbf{w}_m^T \mathbf{w}_m = 1 \quad (7.27)$$

$$\mathbf{G}^T \mathbf{w}_m = 0 \quad (7.28)$$

$$(7.29)$$

where $\mathbf{G} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_{m-1}]$ is a $q \times (m-1)$ matrix with $q > (m-1)$. This is essentially the FastICA (FICA) algorithm proposed by Hyvärinen et al. [14, 13] and remains the most popular ICA algorithm to date.

7.2 Case II: Correlated sources

Sometimes the assumption of zero second order correlation between the components of s_i might be unrealistic. The modified assumption is that of maximal independence between components of s_i under unknown and potentially non-zero correlation i.e., $\Sigma_s \neq \mathbf{I}_q$. If we introduce a change of variables:

$$s_i^* = \Sigma_s^{-\frac{1}{2}} s_i \quad (7.30)$$

$$\mathbf{A}^* = \Sigma_s^{\frac{1}{2}} \mathbf{A} \quad (7.31)$$

then this case can be reduced to the standard PPCA model:

$$\mathbf{x}_i = \boldsymbol{\mu} + \mathbf{A} s_i + \boldsymbol{\varepsilon}_i \quad (7.32)$$

$$= \boldsymbol{\mu} + \mathbf{A}^* s_i^* + \boldsymbol{\varepsilon}_i \quad (7.33)$$

The transformed sources s_i^* satisfy $E(s_i^*) = 0$ and $E(s_i^* s_i^{*T}) = \mathbf{I}_q$, however maximal independence requirement will be imposed on \hat{s}_i and not on s_i^* . Therefore, following the development in the

previous section, the PPCA solution is given by:

$$\hat{\boldsymbol{\mu}} = \bar{\boldsymbol{x}} \quad (7.34)$$

$$\hat{\sigma}^2 = \left(\frac{1}{p-q} \right) \sum_{i=q+1}^p \lambda_i \quad (7.35)$$

$$\hat{\mathbf{A}}^* = \mathbf{U}_q (\Lambda_q - \hat{\sigma}^2 \mathbf{I}_q)^{1/2} \mathbf{Q}^T \quad (7.36)$$

$$\mathbf{Q}^T \mathbf{Q} = \mathbf{I}_q \quad (7.37)$$

The least squares source estimates are given by:

$$\hat{s}_i^* = \Sigma_s^{-\frac{1}{2}} \hat{s}_i = \mathbf{Q} (\Lambda_q - \hat{\sigma}^2 \mathbf{I}_q)^{-\frac{1}{2}} \mathbf{U}_q^T (\mathbf{x}_i - \hat{\boldsymbol{\mu}}) \quad (7.38)$$

From 7.38 and 7.11

$$\hat{s}_i = \Sigma_s^{\frac{1}{2}} \mathbf{Q} (\Lambda_q - \hat{\sigma}^2 \mathbf{I}_q)^{-\frac{1}{2}} \mathbf{U}_q^T (\mathbf{x}_i - \hat{\boldsymbol{\mu}}) \quad (7.39)$$

$$= \Sigma_s^{\frac{1}{2}} \mathbf{Q} \mathbf{z}_i \quad (7.40)$$

$$= \mathbf{W}^{*T} \mathbf{z}_i \quad (7.41)$$

where

$$\mathbf{W}^{*T} = \begin{pmatrix} \mathbf{w}_1^{*T} \\ \mathbf{w}_2^{*T} \\ \vdots \\ \mathbf{w}_q^{*T} \end{pmatrix} = \Sigma_s^{\frac{1}{2}} \mathbf{Q} \quad (7.42)$$

is the unmixing matrix for recovering the sources $\hat{\mathbf{s}}_i$. Note that when the sources are not exactly uncorrelated, the unmixing matrix satisfies $\mathbf{W}^{*T} \mathbf{W}^* = \Sigma_s$ instead of $\mathbf{W}^T \mathbf{W} = \mathbf{I}_q$ as in the exactly uncorrelated case. In other words, sources $\hat{\mathbf{s}}_i$ are extracted by non-orthogonal projections of vectors \mathbf{z}_i . In this case, the optimization problem can be written as (ignoring terms independent of \mathbf{W}^{*T}):

$$\min_{\mathbf{W}^{*T}} f(\hat{\mathbf{s}}_i) \propto \underbrace{\left\{ \sum_{j=1}^q H(\hat{s}_{ij}) \right\}}_{\text{encourages non-Gaussianity}} + \psi \underbrace{\left\{ -\log[\text{abs}(\det(\mathbf{W}^{*T}))] \right\}}_{\geq 0, \text{ encourages } \Sigma_s = \mathbf{I}_q} \quad (7.43)$$

where: (7.44)

$$\hat{s}_{ij} = \mathbf{w}_j^{*T} \mathbf{z}_i, \quad \text{abs}(\det(\mathbf{W}^{*T})) = \frac{\det(\Sigma_{\hat{\mathbf{s}}_i})^{\frac{1}{2}}}{\det(\Sigma_z)^{\frac{1}{2}}} \quad (7.45)$$

subject to: (7.46)

$$\Sigma_{\hat{\mathbf{s}}_i} = \mathbf{W}^{*T} \Sigma_z \mathbf{W}^* \quad (7.47)$$

Note that $\Sigma_{\hat{s}_i} = \Sigma_s$ only if $\Sigma_z = I_q$. Thus the individual components of \hat{s}_i will **not** have unit variance unless $\hat{\sigma}^2 = 0$. In the case of potentially correlated sources, the 2nd term in the objective function 7.43 does not drop out. Hence sequential extraction of sources is only possible in the case when $\psi = 0$. As discussed before equation 6.35, this corresponds to the case when minimization of differential entropy is considered to be much more important than minimizing correlatedness. We consider the solution of 7.43 with $\psi = 0$ in the correlated case.

Since $\mathbf{W}^{*T}\mathbf{W}^* = \Sigma_s$ and the diagonal elements of Σ_s are ones, the m th component is estimated by minimizing the differential entropy of the empirical density of $\hat{s}_{im} = \mathbf{w}_m^{*T}\mathbf{z}_i$ over n realizations subject to the only constraint:

$$\mathbf{w}_m^{*T}\mathbf{w}_m^* = 1 \quad (7.48)$$

Since the off diagonal elements of Σ_s are allowed to have non-zero values, no additional mutual orthogonality constraints are imposed on the projection vectors. An important question is "How do you impose non-orthogonality?". It is possible that 2 vectors \mathbf{w}_1^* and \mathbf{w}_2^* are identical, since the algorithm does not explicitly enforce non-orthogonality. A simple workaround is to randomly initialize each vector \mathbf{w}_m^* by making \mathbf{w}_m^* orthogonal to the previously estimated vectors. In our experiments, this normally prevents convergence to a previously found solution, but again this is not guaranteed. In general, one must continue to randomly initialize solution for the m th component until it is found to be different from the previous $(m - 1)$ solution vectors.

8 PMOG based BSS

In this section, we try to answer the following questions:

- What is the relationship between the PMOG objective function and differential entropy?
- How can PMOG be applied to the problem of BSS?

8.1 PMOG objective and differential entropy

Suppose u is a random variable with density $P(u)$. The differential entropy of u is given by:

$$H(u) = - \int P(u) \log P(u) du = -E_u[\log P(u)] \quad (8.1)$$

The exact density $P(u)$ of u is not known but suppose we approximate it by a parameterized highly flexible density. It is well known that a mixture of Gaussians with a sufficient number of components R can approximate any probability density with any desired accuracy. Suppose $P_R^{mog}(u | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ is an R -component MOG density as described in 5.5 and 5.6. Then we can write:

$$P(u) \approx P_R^{mog}(u | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2) \quad (8.2)$$

Substituting 8.2 in 8.1 we see that:

$$H(u) = -E_u[\log P(u)] \approx -E_u[\log P_R^{mog}(u | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2)] \quad (8.3)$$

Suppose the exact density of u is not known but n independent samples of u : u_1, u_2, \dots, u_n drawn as per $P(u)$ are given. Then we can approximate the expectation in 8.3 by a sample average using the law of large numbers. With this approximation we get:

$$H(u) \approx -E_u[\log P_R^{mog}(u | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2)] \approx -\frac{1}{n} \sum_{i=1}^n [\log P_R^{mog}(u_i | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2)] \quad (8.4)$$

Suppose random variable u is related to another random variable \mathbf{z} via a linear transform $u = \mathbf{w}^T \mathbf{z}$ and so the samples u_i are generated by a linear transformation $u_i = \mathbf{w}^T \mathbf{z}_i$ then we can write:

$$H(u) = H(\mathbf{w}^T \mathbf{z}) \approx -\frac{1}{n} \sum_{i=1}^n [\log P_R^{mog}(\mathbf{w}^T \mathbf{z}_i | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2)] \quad (8.5)$$

Suppose our goal is to find a projection vector \mathbf{w} such that the differential entropy of the projection $\mathbf{w}^T \mathbf{z}$ is minimized given realizations of \mathbf{z} : $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n$. Then as shown in equation 8.5, **minimizing** $H(\mathbf{w}^T \mathbf{z})$ is equivalent to **maximizing** $\sum_{i=1}^n [\log P_R^{mog}(\mathbf{w}^T \mathbf{z}_i | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2)]$ which is essentially the log likelihood of the PMOG model. In addition to the log likelihood term, the PMOG objective in 5.17 also includes prior terms for $\boldsymbol{\pi}$ and $\boldsymbol{\sigma}^2$. These prior terms simply prevent the collapse of a PMOG component density onto a single point and thus make the PMOG estimation well conditioned.

8.2 Sequential source extraction in PMOG based BSS

As discussed in the previous section, the assumption of uncorrelated sources $\boldsymbol{\Sigma}_s = \mathbf{I}_q$ results in the dropping out of the second term in 7.43. If the sources are correlated, $\boldsymbol{\Sigma}_s \neq \mathbf{I}_q$ then the second term drops out only if $\psi = 0$ i.e., when minimizing differential entropy is considered to be much more important than minimizing correlatedness. The important point to note is that in the uncorrelated source case, a sequential estimation algorithm for BSS using the PMOG model always exists. In the potentially correlated source case, such a sequential algorithm will exist only when $\psi = 0$.

In PMOG based BSS, we simply model the empirical density of $\mathbf{w}_m^T \mathbf{z}_i$ by a flexible R -component MOG density. As shown in 8.5, this is equivalent to minimizing the differential entropy of the projected points. We can use the PMOG EM algorithm 2 to estimate \mathbf{w}_m along with the MOG parameters under constraints 7.27 or 7.48 depending on whether we want to enforce $\boldsymbol{\Sigma}_s = \mathbf{I}_q$ or $\boldsymbol{\Sigma}_s \neq \mathbf{I}_q$ respectively. Since a MOG density with sufficiently large R can accurately model any non-Gaussian density, we should be able to extract complex non-Gaussian source densities after solving the PMOG problem. Thus in PMOG based BSS:

-
- We replace the variational approximation step in the work of Attias et al. [1] by an ML step for the PPCA model of Tipping et al. [20] to compute the mixing parameters. This step is similar to that of Beckmann et al. [2].
 - For latent source estimation, we use a least squares approach after mixing parameters have been estimated. However, we replace the objective functions based on approximation to the differential entropy of Hyvarinen et al. [12] with the PMOG model in which the latent sources are assumed to be described by an R component MOG density.
 - PMOG could have a potential advantage in cases where the latent source densities are very complicated. By choosing a sufficiently large R in 8.5, this complicated density can be approximated accurately by the PMOG model.

As discussed above, sequential source estimation is possible under the PMOG model even under partial source dependence (for $\psi = 0$). Thus, the PMOG based BSS retains flexible source density modeling of Attias et al. [1], is computationally tractable and can be applied under partial second order dependence between sources.

9 Experiments and Results

To illustrate the performance of PMOG based BSS, we performed 2 experiments.

- In Experiment 1, we generate several artificial "sources" using a MOG model. Next we create mixed data multiple times using the same "sources" but with different random mixing matrices. Note this is a case of "non-square" and "noise free" mixing. For each mixture, we run both FICA and PMOG followed by a statistical comparison of the performance of PMOG with FICA across multiple runs.
- In Experiment 2, we use real 2-D pictures from a standard image repository as "sources" and mix them using random mixing matrices. Note that this is a case of "square" and "noise free" mixing. We then run FICA and PMOG on the mixed picture data and compare the resulting recovered sources using each method to the "true sources" by visual inspection.

In each experiment, we used the implementation of FICA from the FastICA package of Hyvarinen et al. ([13],[16]) available from <http://www.cis.hut.fi/projects/ica/fastica/index.shtml>.

9.1 Experiment1

We generated synthetic data with $q = 7$ MOG sources using a mixture of $R = 5$ Gaussians. Each MOG source had $n = 1000$ sample points. For each MOG source, elements of the class fraction vector π were chosen from $U(0, 1)$ and then normalized to have a sum of 1. Elements of the mean

vector μ were chosen from $U(-10, 10)$ and the elements of the variance vector σ^2 were chosen from $U(1, 5)$.

Suppose \mathbf{S} is the $q \times n$ matrix made up of the n realizations of the source vectors \mathbf{s}_i , $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_n]$. Let \mathbf{e}_n be a $n \times 1$ vector of all 1's and suppose the eigen-decomposition of the sample co-variance of \mathbf{s}_i is:

$$\boldsymbol{\mu}_s = \frac{1}{n} \sum_{i=1}^n \mathbf{s}_i = \frac{1}{n} \mathbf{S} \mathbf{e}_n \quad (9.1)$$

$$\mathbf{U}_s \boldsymbol{\Lambda}_s \mathbf{U}_s^T = \frac{1}{n} (\mathbf{S} - \boldsymbol{\mu}_s \mathbf{e}_n^T) (\mathbf{S} - \boldsymbol{\mu}_s \mathbf{e}_n^T)^T \quad (9.2)$$

Here \mathbf{U}_s is a $q \times q$ orthogonal matrix and $\boldsymbol{\Lambda}_s$ is a diagonal matrix. We construct whitened sources that have $\mathbf{0}$ sample mean and identity sample co-variance as follows:

$$\tilde{\mathbf{S}} = \boldsymbol{\Lambda}_s^{-\frac{1}{2}} \mathbf{U}_s^T (\mathbf{S} - \boldsymbol{\mu}_s \mathbf{e}_n^T) \quad (9.3)$$

The $q \times n$ matrix $\tilde{\mathbf{S}} = [\tilde{\mathbf{s}}_1, \tilde{\mathbf{s}}_2, \dots, \tilde{\mathbf{s}}_n]$ holds n samples of the whitened source vectors $\tilde{\mathbf{s}}_i$. These whitened sources were mixed by random $p \times q$ mixing matrices and embedded into a $p = 20$ dimensional space. We generated a mixture of blind sources $m = 50$ times, each time using a different mixing matrix as follows:

$$\mathbf{X}^{(m)} = \mathbf{A}^{(m)} \tilde{\mathbf{S}} \quad (9.4)$$

Here $\mathbf{A}^{(m)}$ is a random $p \times q$ mixing matrix with elements drawn from $U(0, 1)$. The $p \times n$ matrix $\mathbf{X}^{(m)} = [\mathbf{x}_1^{(m)}, \mathbf{x}_2^{(m)}, \dots, \mathbf{x}_n^{(m)}]$ contains the mixed signal vectors in p dimensional space for the m th run. We analyzed each $\mathbf{X}^{(m)}$ as follows:

- Run FICA on $\mathbf{X}^{(m)}$ to get q sources as rows of the $q \times n$ matrix $\tilde{\mathbf{S}}_{FICA}^{(m)}$. We used the default settings for the FICA algorithm from the FastICA package with the odd function $G(x) = \tanh(x)$.
- Run orthogonal PMOG on $\mathbf{X}^{(m)}$ to get q sources as rows of the $q \times n$ matrix $\tilde{\mathbf{S}}_{PMOG}^{(m)}$.

Definition 9.1. Match between two matrices

Given two $q \times n$ matrices \mathbf{A} and \mathbf{B} , let $\rho_{ij}(\mathbf{A}, \mathbf{B})$ denote the correlation coefficient between the i th row of \mathbf{A} and the j th row of \mathbf{B} . We define the "match" between \mathbf{A} and \mathbf{B} as a quantity that measures the average value of the best absolute correlation coefficient of the rows of \mathbf{A} with \mathbf{B} as follows:

$$\text{Match}(\mathbf{A}, \mathbf{B}) = \frac{1}{q} \sum_{i=1}^q \max_j [\text{abs } \rho_{ij}(\mathbf{A}, \mathbf{B})] \quad (9.5)$$

Our goal is to compare $\text{Match}(\tilde{\mathbf{S}}, \tilde{\mathbf{S}}_{FICA}^{(m)})$ with $\text{Match}(\tilde{\mathbf{S}}, \tilde{\mathbf{S}}_{PMOG}^{(m)})$ across the $m = 50$ runs of FICA and PMOG. Since the distribution of the Match values is potentially non-Gaussian, we apply

a transformation to Normality to both Match $(\tilde{S}, \tilde{S}_{FICA}^{(m)})$ and Match $(\tilde{S}, \tilde{S}_{PMOG}^{(m)})$ as described in [21] followed by a statistical comparison of the transformed values:

- In brief, we simply pass the empirical distribution of Match $(\tilde{S}, \tilde{S}_{FICA}^{(m)})$ across m through the inverse CDF of a Normal distribution with the same sample mean and variance. A similar transformation to Normality is applied to Match $(\tilde{S}, \tilde{S}_{PMOG}^{(m)})$ and this transforms the non-Gaussian Match values to Gaussianity.
- Let us denote the Normally distributed Match values for FICA and PMOG using the notation Match_{Normal} $(\tilde{S}, \tilde{S}_{FICA}^{(m)})$ and Match_{Normal} $(\tilde{S}, \tilde{S}_{PMOG}^{(m)})$ respectively. We performed a 2-sample t -test with unequal variance to compare the mean of Match_{Normal} $(\tilde{S}, \tilde{S}_{FICA}^{(m)})$ and Match_{Normal} $(\tilde{S}, \tilde{S}_{PMOG}^{(m)})$.

Results are shown in Fig. 3. PMOG produces results that are statistically significantly better than FICA with a p -value of $\sim 1.13 \times 10^{-5}$ across the $m = 50$ runs.

9.2 Experiment 2

To illustrate the performance of PMOG based BSS on real data, we selected 2 separate sets of 3 images from the Berkeley Segmentation Dataset and Benchmark [18]. First, each image was demeaned and standardized to have unit variance. In each case, the 3 identical size images were mixed by a random 3×3 mixing matrix \mathbf{A} and mean vector $\boldsymbol{\mu}$ whose elements were drawn from a $\mathcal{N}(0, 1)$ Normal distribution. Note that this is a case of "square" and "noise free" mixing. In both cases, the 3 mixed images were subjected to 3 different analyses:

- FICA analysis with the standard orthogonality constraints on the projection vectors. We used the default settings for the FICA algorithm from the FastICA package with the odd function $G(x) = \tanh(x)$.
- PMOG analysis with orthogonality constraints on the projection vectors.
- PMOG analysis without orthogonality constraints on the projection vectors.

Results are shown in Fig. 4 - 7. For the example shown in Fig. 4, FICA failed to converge using the default settings. Upon experimenting with various settings, we found that the FICA algorithm was able to converge for the data in Fig. 4 when we change the "decorrelation" approach to "symm" (see the FICA package for more details). The default settings of FICA resulted in successful convergence for the example in Fig. 6.

It can be seen visually from Fig. 5 and Fig. 7 that the results of PMOG are better than that of FICA. Since there are dependencies between the intensities of corresponding pixels in the 3 images,

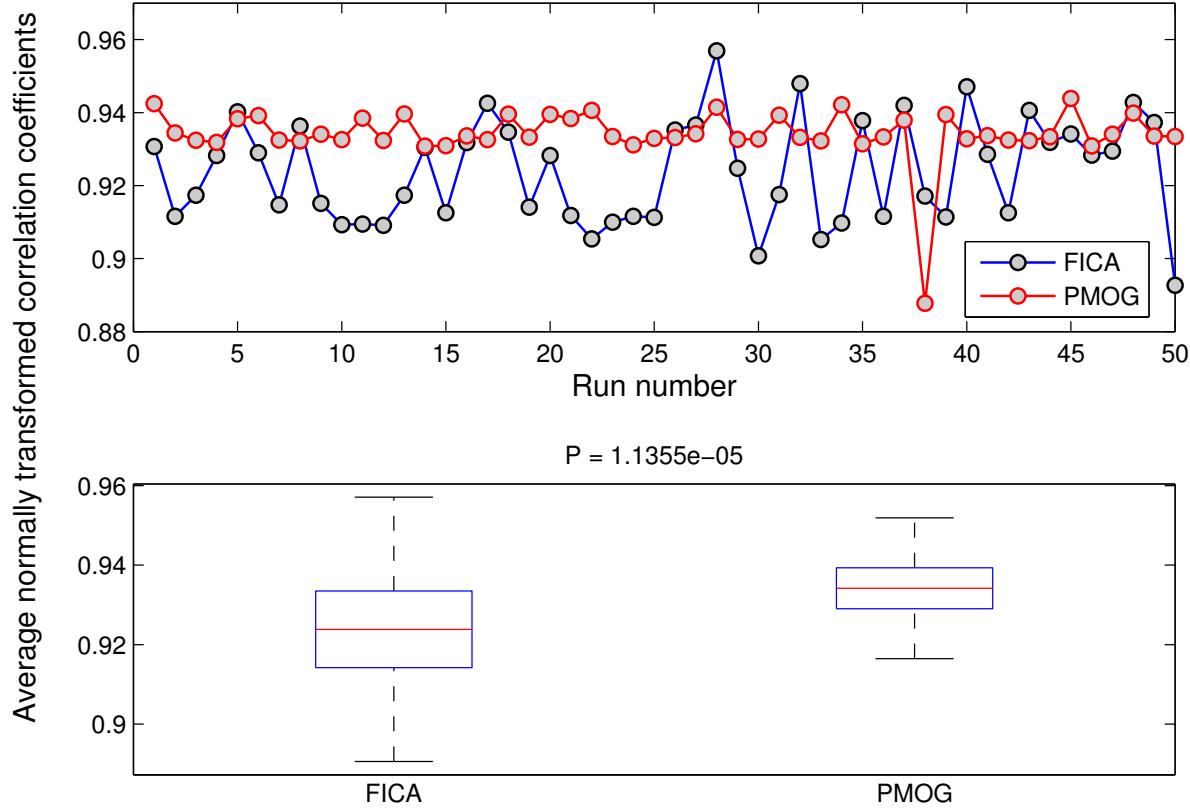


Figure 3: The top figure shows the quantities $\text{Match}_{\text{Normal}} \left(\tilde{S}, \tilde{S}_{\text{FICA}}^{(m)} \right)$ (blue) and $\text{Match}_{\text{Normal}} \left(\tilde{S}, \tilde{S}_{\text{PMOG}}^{(m)} \right)$ (red) across the m runs. The bottom figure shows the results of a 2-sample t -test with unequal variance on the data from the top figure. The results from PMOG are statistically significantly better than those of FICA with a p -value of $\sim 1.13 \times 10^{-5}$.

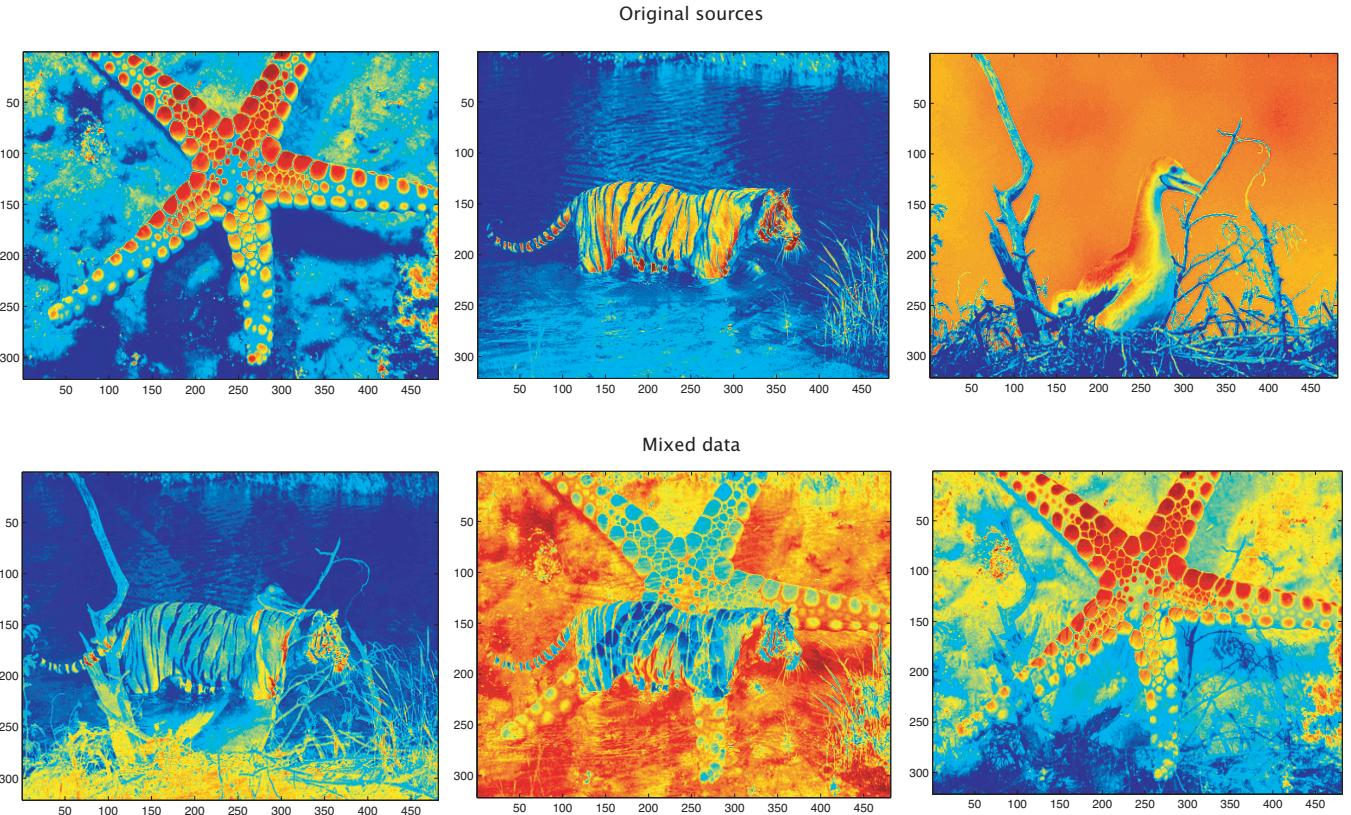


Figure 4: The top row shows 3 pictures of natural images i.e., the original sources. The bottom row shows mixed pictures obtained after mixing image intensities from each pixel of the 3 images using a 3×3 random mixing matrix and adding a random mean offset. This mixed data was analyzed using FICA and PMOG. Results are shown in Fig. 5.

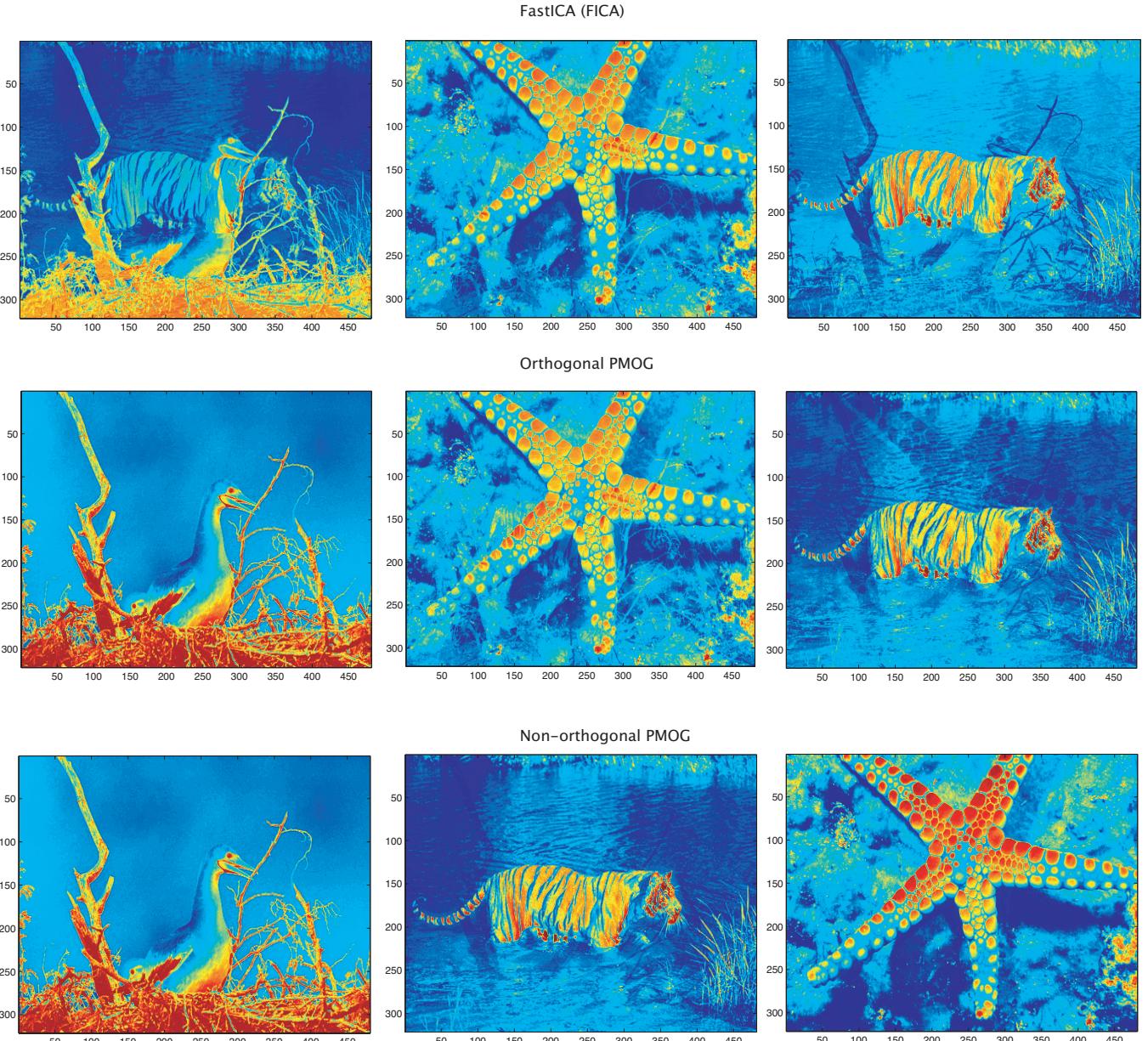


Figure 5: This figure shows the results of running FICA and PMOG on the data from Fig. 4. The top row shows 3 estimated sources using the FICA algorithm. For this dataset the 'defl' approach in FICA fails and hence we used the 'symm' approach. The middle row shows 3 estimated sources using PMOG algorithm with orthogonality constraint on the 3 projections. The bottom row shows estimated sources using the PMOG algorithm without imposing the orthogonality constraint on the 3 projections.

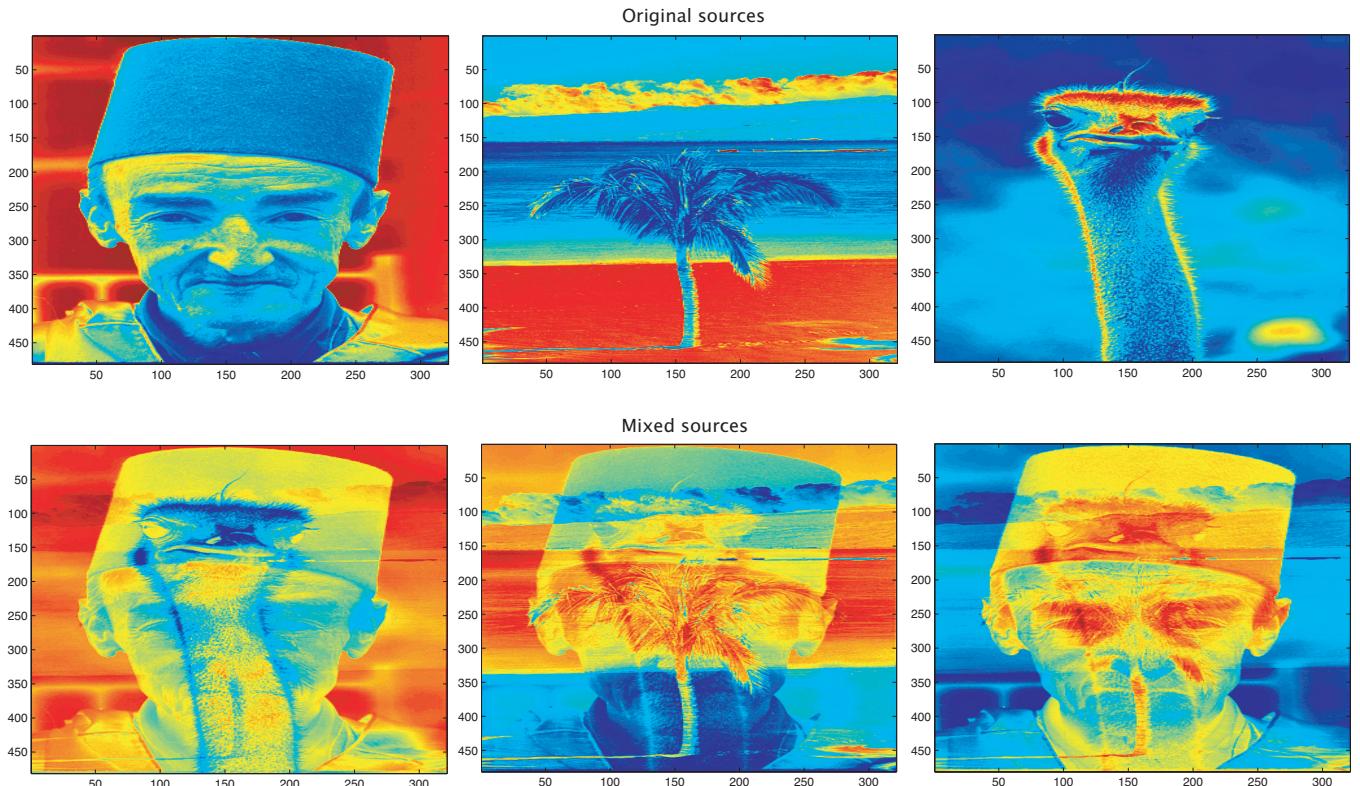


Figure 6: The top row shows 3 pictures of natural images i.e., the original sources. The bottom row shows mixed pictures obtained after mixing image intensities from each pixel of the 3 images using a 3×3 random mixing matrix and adding a random mean offset. This mixed data was analyzed using FICA and PMOG. Results are shown in Fig. 7.

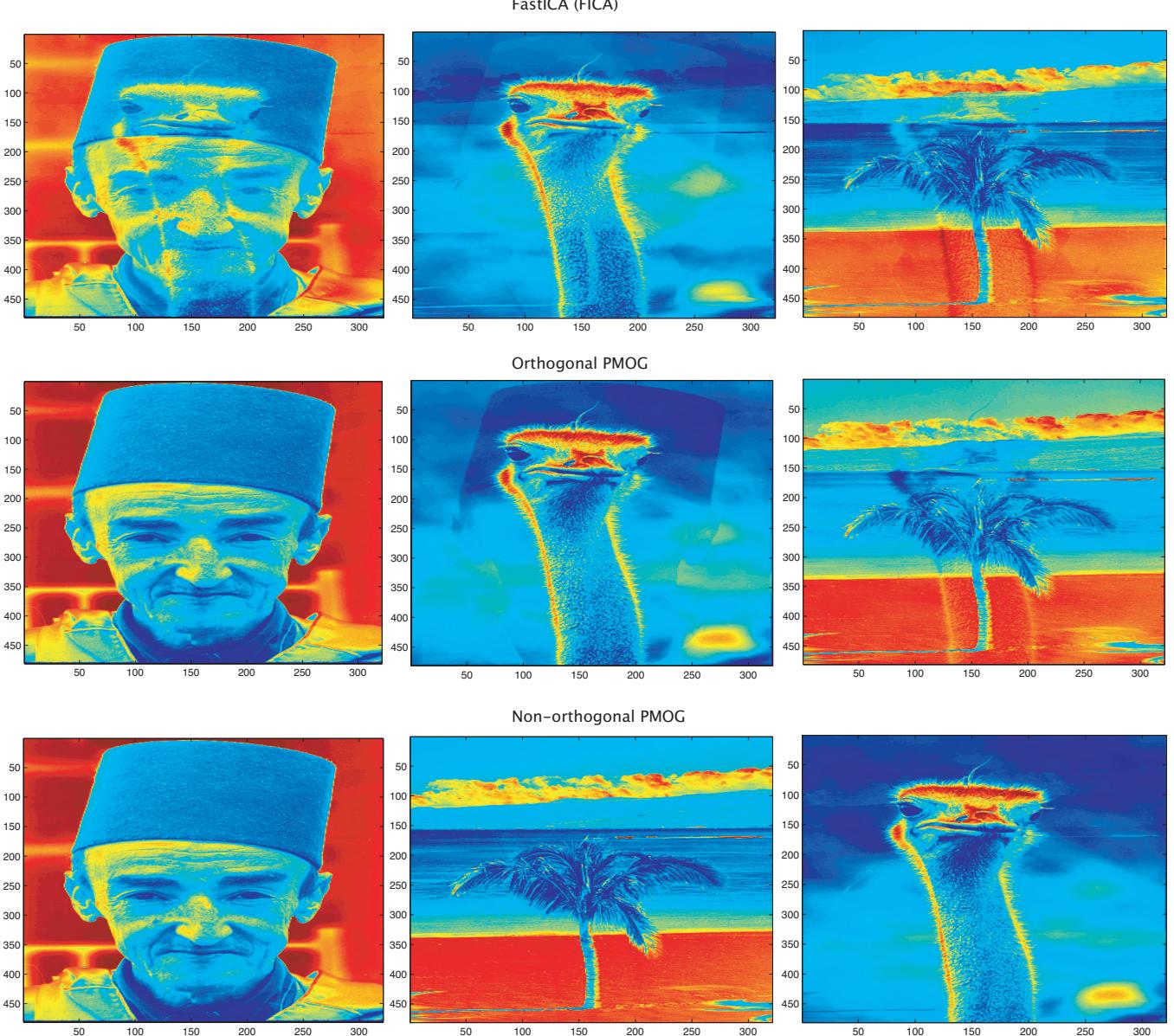


Figure 7: This figure shows the results of running FICA and PMOG on the data from Fig. 6. The top row shows 3 estimated sources using the FICA algorithm. The middle row shows 3 estimated sources using PMOG algorithm with orthogonality constraint on the 3 projections. The bottom row shows estimated sources using the PMOG algorithm without imposing the orthogonality constraint on the 3 projections.

they are not exactly independent. Thus, we see that PMOG without orthogonality constraint is able to achieve better source separation compared to PMOG with orthogonality constraint.

As an example, we also show the fitted PMOG model for one of the extracted sources (the bottom left image in Fig. 7) in Fig. 8. This figure shows the monotonic increase in PMOG objective function $H(\pi^{(t)}, \mu^{(t)}, \sigma^2(t), w^{(t)})$ from 5.30 over EM iterations t and the final fitted PMOG model using $R = 5$ Gaussian distributions. It is interesting to note that this distribution is highly non-Gaussian and hence approximations to negentropy used in FICA based on the "near Gaussianity" assumption in [11] might not be adequate in such cases.

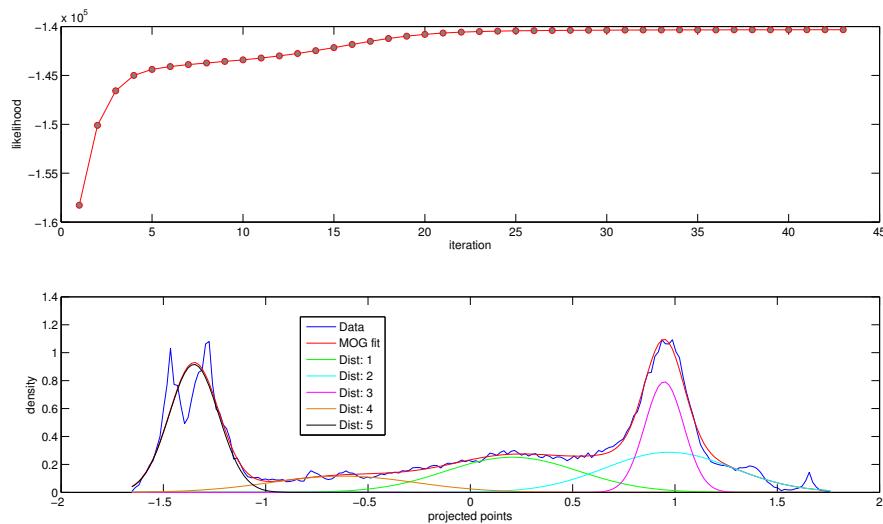


Figure 8: This figure shows the fitted PMOG model for one of the extracted sources (the bottom left image in Fig. 7). Top figure shows the evolution of PMOG log likelihood over iterations illustrating the monotonic increase property of the EM algorithm for estimating the PMOG model. Bottom figure shows the 1-D projected density (corresponding to the estimated projection vector) fitted by a 5 component PMOG model. In PMOG, both the projection vector and the MOG parameters are jointly optimized.

10 Discussion

In this work, we posed the problem of estimating a projection of input data such that the projection is well described by an R component MOG density. We showed that it is possible to derive an EM algorithm for solving this problem. Since the estimation of projection vector is coupled with the estimation of distributional parameters, we break up the M-step into two parts:

-
- In part 1 of the M-step, we estimate the distributional parameters for a fixed projection vector.
 - In part 2 of the M-step, we optimize for the projection vector while fixing the distributional parameters.

We show that solving the M-step problem in part 2 is equivalent to finding the roots of a particular cubic equation for the projection vector. Since the objective in part 2 of the M-step is non-concave, the solution to the set of cubic equation in part 2 might converge to a minimum instead of a maximum. To enforce the monotonic likelihood increase property of the EM algorithm, we explicitly check equation 5.29 after each M-step. If this condition is not satisfied, we randomly re-initialize \mathbf{w} and repeat the M-step.

Article	BSS with Gaussian noise?	Mixing parameter estimation	Source estimation	Second order source correlation allowed?
Hyvärinen et al. (1999, IEEE Trans. on Neural Networks)	No	PCA upto orthogonal rotation	Least squares + Optimization of negentropy based contrast functions in Hyvärinen et al. (1998)	No
Attias et al. (1999, Neural Computation)	Yes	EM based ML using factorial MOG as source density	Posterior source density mean or MAP estimators	No
Beckmann et al. (2004, IEEE Trans. on Medical Imaging)	Yes	PPCA of Tipping et al. (1999) upto orthogonal rotation	Least squares + Optimization of negentropy based contrast functions in Hyvärinen et al. (1998)	No
This work	Yes	PPCA of Tipping et al. (1999) upto linear transformation	Least squares + Optimization of PMOG objective function	Yes

Figure 9: A comparison of the features and estimation techniques of four different BSS algorithms.

Next we considered the application of the PMOG model to the solution of the BSS problem. If the mixing process is without Gaussian noise then the estimation of mixing parameters and latent sources is uncoupled [14]. In this case, the BSS problem can be reduced to one of estimating an orthogonal matrix by using principal component analysis (PCA) as a preprocessing step. However, in the presence of additive Gaussian noise, the estimation of mixing parameters and latent sources is coupled and this makes the BSS problem a difficult one.

Attias et al. [1] developed a very general solution for the BSS problem under the assumption of exact independence between latent sources. In particular, [1] assumed an MOG density for each latent source. This implies a "factorial MOG" joint density for the vector of latent sources under independence. Further, [1] derived an EM algorithm for joint estimation of both the mixing parameters and "factorial MOG" parameters. While this work provides an exact solution to the BSS problem, it assumes exact independence between sources and becomes computationally intractable for > 13 latent sources. As a computationally tractable alternative, [1] propose a variational approximation to compute the BSS parameters followed by using a MAP or posterior mean estimate for the sources.

Beckmann et al. [2] proposed a BSS solution in which the estimation of mixing parameters and latent sources is de-coupled by assuming that the sources have joint distribution $\mathcal{N}(\mathbf{0}, \mathbf{I}_q)$. This effectively reduces the BSS model to the PPCA model of Tipping et al. [20]. In [2], the mixing parameters are chosen to be the ML solution for this PPCA model after which the BSS problem reduces to one of estimating an orthogonal matrix. In both [14] and [2], the sources are estimated as orthogonal projections that minimize the sum of differential entropies of the projections as in equation 7.15. Both techniques [14] and [2] use approximations to the differential entropy developed in [12] based on the assumption of "near Gaussian" source densities.

While the solution by Attias et al. [1] is very attractive (at least under exact independence) since it allows flexible source density modeling via a "factorial MOG", it suffers from computational intractability for > 13 sources. Our work bridges the gap between the approach of Attias et al. [1] and that of Hyvärinen et al. [14] and Beckmann et al. [2]. On the one hand, it retains computational tractability by using the PPCA approach of Beckmann et al. and on the other hand it allows for flexible source density modeling of Attias et al. via the PMOG model. As we show in 8.5, minimizing the differential entropy is equivalent to maximizing the PMOG likelihood function. In the PMOG model, we jointly estimate both the projection vector as well as MOG distributional parameters to maximize the likelihood of observing an MOG in the projected space. The overall algorithm retains computational feasibility for sources > 13 while retaining the flexibility of modeling arbitrarily complex source densities (given enough Gaussians in PMOG).

In the work of Attias et al. [1] an approximation is made to true ML objective (based on "factorial MOG" sources) using variational arguments to retain computational tractability for > 13 sources. In the present work and in Beckmann et al. [2], exact ML solution is used for the analytically tractable PPCA model (based on $\mathcal{N}(\mathbf{0}, \mathbf{I}_q)$ sources). Both methods use approximations. Is the variational approximation more accurate than the PPCA approximation for BSS parameters? This question is open for discussion and beyond the scope of this work. However, based on our experiments with real and synthetic data the PPCA approximation followed by application of the PMOG results in good performance. We also show that when the latent sources are only approximately independent, then we should simply run the PMOG algorithm with only the unit norm constraints on the projection vectors (i.e. without the orthogonality constraints). Thus PMOG based BSS generalizes elegantly to cases where non-zero second order correlation exists between sources. Fig. 9 shows a comparison of PMOG based BSS with alternative approaches.

It is worth noting the difference between the BSS algorithms in Fig. 9 and a related technique from statistics - projection pursuit (PP) [9, 10, 8]. In BSS, the problem is to estimate both the mixing parameters and latent sources given the linear mixing model 6.1. As shown in [1], this is a coupled estimation problem. However, if we use a PPCA step to estimate mixing parameters then this problem becomes decoupled and the latent sources can be estimated using least squares upto a linear transformation. If sources are assumed to be uncorrelated then this linear transformation is orthogonal. At this point, the problem becomes similar to a PP problem, where linear transformations (not necessarily orthogonal) are sought that optimize general contrast functions. Whereas in

PP any contrast function can be used, the focus in BSS is to maximize the independence between sources and so a linear transformation that minimizes the mutual information between the sources is sought. Thus PMOG based BSS can also be thought of as a combination of PPCA preprocessing followed by PP using the special PMOG objective function which is optimized using an EM algorithm with or without orthogonality conditions on the projections.

Both FICA and PMOG solve an optimization problem where different objective functions are maximized. In the case of FICA, the objective function is an approximation to differential entropy developed in [12] whereas in the case of PMOG the objective function is the PMOG log likelihood 8.5. In both FICA and PMOG, the objective functions are not concave and so the solutions to the maximization problems are not unique. An ideal solution would be to find the global solution to the maximization problems in FICA and PMOG and compare the results. Since finding the global solution itself is a difficult problem an alternative suboptimal method of comparing FICA and PMOG is to run each algorithm multiple times on mixtures generated using the same "sources" and then compare the quality of results across runs. This is the approach we used in Experiment 1 9.1. It was found that on average PMOG produced significantly better results compared to FICA across runs. We would like to mention that since the true "sources" in Experiment 1 are in fact MOG sources, the results could be slightly biased towards PMOG. Nevertheless, Experiment 1 does illustrate that when the true "sources" have a complicated density (such as a MOG) then PMOG might show improved performance.

We also ran 2 illustrative experiments on publicly available natural image datasets [18] in Experiment 2 9.2 and compared the results of FICA and PMOG by visual inspection. FICA showed relatively poor performance compared to orthogonal PMOG. We postulate that this is because the source densities are multimodal (see Fig. 8) and so the PMOG model captures them more faithfully compared to the approximate cost function [12] used in FICA. We also noticed that assumption of orthogonality of projection vectors also hurts the separation performance because the source images have non-zero second order correlation and hence are not exactly independent. The best performing algorithm was non-orthogonal PMOG in which maximal source independence is enforced with potentially non-zero second order correlation (i.e., a minimal level of dependence is allowed).

A limitation of PMOG based BSS is the running time of the PMOG algorithm. It is clear from numerical experiments that PMOG is much slower than FICA. For the experiments presented in this work, PMOG takes upto 7 minutes per source estimation versus only a fraction of a second for FICA using a computer with 2×2.93 GHz Quad-Core Intel Xeon processor with 16 GB RAM. Another drawback is the potential re-initialization required to enforce "non-orthogonality" in the form of PMOG where partial source dependence is allowed. A natural solution to this problem is to perform a joint PMOG estimation with $\psi \neq 0$ in 7.43 which would automatically encourage orthogonality while not explicitly enforcing it.

Finally, note that the approximation in 8.5 becomes increasingly accurate as n increases and thus ideally we would like the number of available data samples n to be large. However, this dependence

of the quality of approximation on n is not unique to PMOG based BSS. For instance, even in the differential entropy approximations of Hyvarinen et al. [12] such as 7.24, the expectations on the right hand side are approximated by sample averages which also become increasingly accurate as n increases.

11 Conclusions

We propose that the non-square linear BSS model with Gaussian noise can be estimated by first using the PPCA model of Tipping et al. [20] followed by application of the PMOG algorithm. The MOG density provides a flexible model for the unknown source density and simulations on illustrative data sets indicate that this approach might be a useful alternative to well established approaches such FICA. Furthermore, it is possible to allow for non-zero second order correlation between latent sources simply by relaxation of the orthogonality requirement in PMOG. This could result in better performance since the assumption of exact statistical independence is unlikely to be true for real world data-sets.

The current version of PMOG based BSS solves the optimization problem in 7.43 for $\psi = 0$ since this decouples the estimation of individual projection vectors. Future work would involve the joint solution of all projection vectors using a PMOG approach for $\psi = 1$.

Acknowledgements

We gratefully acknowledge financial support from the Pain and Analgesia Imaging and Neuroscience (P.A.I.N) group, McLean Hospital, Harvard Medical School, Belmont MA, USA under a grant from the Louis Herlands Fund for Pain Research (DB and LB).

References

- [1] H. Attias. Independent factor analysis. *Neural Computation*, 11:803–851, 1999.
- [2] C.F. Beckmann and S.M. Smith. Probabilistic independent component analysis for functional magnetic resonance imaging. *IEEE Trans. on Medical Imaging*, 23(2):137–152, 2004.
- [3] Jean-Francois Cardoso. Dependence, Correlation and Gaussianity in Independent Component Analysis. *Journal of Machine Learning Research*, 4:1177–1203, 2003.
- [4] A. Cichocki and S. Amari. *Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications*. John Wiley, Chichester, UK, 2003.
- [5] P. Comon. ‘Independent Component Analysis, a new concept? *Signal Processing*, 36(3):287–314, 1994.
- [6] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley and Sons, 1991.
- [7] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum Likelihood From Incomplete Data via EM Algorithm. *Journal of Royal Statistical Society., Series B*, 39:1–38, 1977.
- [8] J. H. Friedman. Exploratory projection pursuit. *Journal of the American Statistical Association*, 82(397):249–266, 1987.
- [9] J. H. Friedman and J. W. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Trans. of Computers*, 23(9):881–890, 1974.
- [10] P. J. Huber. Projection pursuit. *The Annals of Statistics*, 13(2):435–475, 1985.
- [11] A. Hyvärinen. New Approximations of Differential Entropy for Independent Component Analysis and Projection Pursuit. Technical Report Report A47, Department of Computer Science and Engineering, Helsinki University of Technology, 1997.
- [12] A. Hyvärinen. New approximations of differential entropy for independent component analysis and projection pursuit. *Advances in Neural Information Processing Systems*, 10:273–279, 1998.
- [13] A. Hyvärinen. Fast and Robust Fixed-Point Algorithms for Independent Component Analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999.
- [14] A. Hyvärinen. Survey on Independent Component Analysis. *Neural Computing Surveys*, 2:94–128, 1999.
- [15] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John Wiley and Sons, 2001.
- [16] A. Hyvärinen and E. Oja. Independent Component Analysis: Algorithms and Applications. *Neural Networks*, 13(4-5):411–430, 2000.

-
- [17] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. volume 1, pages 281–297, Berkeley, University of California Press, 1967. Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability.
 - [18] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.
 - [19] J. Nocedal and S.J Wright. *Numerical Optimization, 2nd Edition*. New York:Springer, 2006.
 - [20] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61(3):611–622, 1999.
 - [21] S. J. van Albada and P. A. Robinson. Transformation of arbitrary distributions to the normal distribution with application to EEG test-retest reliability. *Journal of Neuroscience Methods*, 161:205–211, 2007.