Notes on Statistical and Machine Learning

Independent Component Analysis

Chapter: *25* **Prepared by:** *Chenxi Zhou*

This note is prepared based on

- *Chapter 15, Latent Variable Models for Blind Source Separation* in Izenman (2009), and
- *Chapter 14, Unsupervised Learning* in Hastie, Tibshirani, and Friedman (2009).

I. Introduction

1. **Blind Source Separation Problem:** The *blind source separation* (BSS) problem involves decomposing an unknown mixture of non-Gaussian signals into its independent component signals.
2. **Overview:** *Independent component analysis* (ICA) is a multivariate statistical technique that seeks to uncover hidden variables in high-dimensional data and solve the blind source separation problem.
3. **Assumption:**
   (a) The ICA model is a linear mixture of an *unknown* number of *unknown* hidden source variables, where the mixing coefficients are also *unknown*;
   (b) The hidden variables are mutually independent;
   (c) The hidden variables are (with at most one exception) non-Gaussian.
4. **Setup:** Let $X = (X_1, \cdots, X_p)^\top$ be a p-dimensional random vector with mean $\mathbb{E}[X] = \boldsymbol{\mu} \in \mathbb{R}^p$ and covariance matrix $\boldsymbol{\Sigma}_{XX} \in \mathbb{R}^{p\times p}$.
5. **Preprocessing:** We
   (a) center X so that its components have zero mean, and
   (b) whiten the result so that its components are uncorrelated and have unit variances.

   Let $\boldsymbol{\Sigma}_{XX} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top$ be the spectral decomposition of $\boldsymbol{\Sigma}_{XX}$, where $\boldsymbol{\Lambda} \in \mathbb{R}^{p\times p}$ is a diagonal matrix with the eigenvalues of $\boldsymbol{\Sigma}_{XX}$ on the diagonal. Since $\boldsymbol{\Sigma}_{XX} \succeq \mathbf{0}_{p\times p}$, all diagonal elements of $\boldsymbol{\Lambda}$ are nonnegative.
   (a) Assume both $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}_{XX}$ are known. Then, we can achieve the goal of preprocessing by performing

$$X \longleftarrow \boldsymbol{\Lambda}^{-\frac{1}{2}}\mathbf{U}^\top(X - \boldsymbol{\mu}). \tag{1}$$

(b) Typically, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}_{XX}$ are unknown. Let $\mathbf{x}_1, \cdots, \mathbf{x}_n$ be n i.i.d observations from X. We estimate $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}_{XX}$ by

$$\bar{\mathbf{x}} := \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i, \qquad \text{and} \qquad \widehat{\boldsymbol{\Sigma}}_{XX} := \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top,$$

respectively. Let $\widehat{\boldsymbol{\Sigma}}_{XX} = \widehat{\mathbf{U}}\widehat{\boldsymbol{\Lambda}}\widehat{\mathbf{U}}^\top$ be the spectral decomposition of $\widehat{\boldsymbol{\Sigma}}_{XX}$. We can preprocess each $\mathbf{x}_i$ as

$$\mathbf{x}_i \longleftarrow \widehat{\boldsymbol{\Lambda}}^{-\frac{1}{2}}\widehat{\mathbf{U}}^\top(\mathbf{x}_i - \bar{\mathbf{x}}), \qquad \text{for all } i = 1, \cdots, n.$$

II. The General ICA Problem

1. **ICA Model:** The *ICA model* assumes that the p-dimensional random vector X is generated by

$$X = f(S) + \varepsilon, \tag{2}$$

where $S = (S_1, S_2, \cdots, S_m)^\top \in \mathbb{R}^m$ is an unobserved random vector of sources. We assume

   (a) components $S_1, S_2, \cdots, S_m$ are independent latent variables,
   (b) $\mathbb{E}[S_j] = 0$ for all $j = 1, \cdots, m$, and $\mathrm{Var}[S] = \mathbf{I}_m$,
   (c) $f : \mathbb{R}^m \to \mathbb{R}^p$ is an unknown *mixing function*, and
   (d) $\varepsilon \in \mathbb{R}^p$ is an additive component with $\mathbb{E}[\varepsilon] = \mathbf{0}_p$ that represents measurement noise and any other type of variability that *cannot* be directly attributed to the sources.

2. **Goal:** The goal is to invert f and estimate S.

3. **Ill-posedness of the Problem:** With the setup so far, the problem is ill-posed, and needs some additional constraints or regularization to achieve the desired goal. Examples include the following:

   (a) If we let $f(S) = \mathbf{A}S$, where $\mathbf{A}$ is a "mixing" matrix, then (2) is a *linear ICA model*; if f is a nonlinear function, (2) is a *nonlinear ICA model*.
   (b) If we require $\varepsilon = \mathbf{0}_p$, i.e., there is no random noise so that all noise in the model is associated with the components of S, we obtain the *noiseless ICA model*.

III. Linear Noiseless ICA

1. **Setup:** We consider the ICA model that has the linear mixing and has no additive noise. In this scenario, X is modeled deterministically as

$$X = \mathbf{A}S,$$

where $S = (S_1, S_2, \cdots, S_m)^\top$ is a latent m-dimensional random vector of independent source components, and $\mathbf{A} \in \mathbb{R}^{p \times m}$ is a full-rank mixing matrix of unknown parameters. We require $m \leq p$.

2. **No Solution Exists When S Has Independent Gaussian Component:** Suppose $m = p$. The resulting linear noiseless ICA model, $X = \mathbf{A}S$, can *only* be solved if independent components of S are *not* Gaussian.

   Suppose the contrary that $S_1, S_2, \cdots, S_m$ are independent and Gaussian with mean 0 and variance 1. The joint density function is

$$q_S(\mathbf{s}) := \frac{1}{(2\pi)^{\frac{p}{2}}} \exp\left(-\frac{1}{2}\sum_{j=1}^{p} s_j^2\right).$$

   Recall that X, after the preprocessing, has $\mathrm{Var}[X] = \mathbf{I}_p = \mathrm{Var}[S]$. Then, the mixing matrix $\mathbf{A}$ must satisfy

$$\mathbf{I}_p = \mathbf{\Sigma}_{XX} = \mathbf{A}\mathbf{A}^\top;$$

   in other words, we have $\mathbf{A} = \mathbf{A}^{-1}$, i.e., $\mathbf{A}$ is orthogonal. It follows that the density function of $X = \mathbf{A}S$ is

$$\begin{aligned} q_X(\mathbf{x}) &:= \frac{1}{(2\pi)^{\frac{p}{2}}} \exp\left(-\frac{1}{2}\|\mathbf{A}^\top\mathbf{x}\|_2^2\right)|\det(\mathbf{A}^\top)| \\ &= \frac{1}{(2\pi)^{\frac{p}{2}}} \exp\left(-\frac{1}{2}\|\mathbf{x}\|_2^2\right). \end{aligned}$$

   Thus, the density of X reduces to that of S, and the orthogonal mixing matrix $\mathbf{A}$ *cannot* be identified for independent Gaussian sources.

   Remedy: We need to require that, with the exception of at most one component, the remaining independent source components cannot be Gaussian distributed.

3. **Goal:** Given n i.i.d observations on X, the ICA problem attempts to estimate $\mathbf{A}$ and, hence, recover S.

4. **Solution with a Given A:** For a given $\mathbf{A}$ with full rank, there exists a un-mixing matrix $\mathbf{W}$ such that the sources can be recovered exactly from the observed X by

$$S = \mathbf{W}X, \tag{3}$$

   where $\mathbf{W} = (\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top$.

   Special Cases: If the number of independent sources is equal to the number of measurements, i.e., $m = p$, we have $\mathbf{W} = \mathbf{A}^{-1}$. If X has been centered and sphered, then the square mixing matrix $\mathbf{A}$ is orthogonal, and so $\mathbf{W} = \mathbf{A}^\top$.

5. **Solution When A is Unknown:** In practice, $\mathbf{A}$ is *unknown* and the goal is to estimate $\mathbf{W}$ and the source components based solely upon the observations of X.

Given an estimate $\widehat{\mathbf{W}} = (\hat{\mathbf{w}}_1, \cdots, \hat{\mathbf{w}}_m)^\top \in \mathbb{R}^{m\times p}$ of the un-mixing matrix $\mathbf{W}$, the source component vector S is approximated by

$$Y = \widehat{\mathbf{W}} X,$$

where the elements $Y_1 := \hat{\mathbf{w}}_1^\top X$, $Y_2 := \hat{\mathbf{w}}_2^\top X$, $\cdots$, $Y_m := \hat{\mathbf{w}}_m^\top X$ are taken to be statistically independent and as non-Gaussian as possible.

IV. Non-polynomial Based Approximation

1. **Goal:** In this section, we aim to achieve the following two goals:
   (a) we approximate the density function of a random variable Y under certain moment constraints and the assumption that $\log p_Y$ is a sum of non-polynomial functions,
   (b) under this approximation, we derive the formula for the *entropy*

$$H(Y) = -\int p_Y(y) \log p_Y(y) \mathrm{d}y, \tag{4}$$

   where p_Y is the density function of the random variable Y, and the *negentropy*

$$J(Y) = H(Z) - H(Y), \tag{5}$$

   where Z is a Gaussian random variable with the same variance as Y.

   Achieving these two goals helps us to derive the FastICA algorithm in the next section.

2. **Setup:** Suppose that g_j, for $j = 1, 2, \cdots, m$, are distinct non-polynomial functions that
   (a) form an orthonormal system with respect to the standard Gaussian density function φ, i.e.,

$$\int \varphi(y) g_i(y) g_j(y) \mathrm{d}y = \delta_{i,j}, \tag{6}$$

   where $\delta_{i,j} = 1$ if and only if $i = j$ and $\delta_{i,j} = 0$ otherwise, and
   (b) are orthogonal to all polynomials of degrees up to 2, i.e., for all $j = 1, 2, \cdots, m$, the following equations hold

$$\int \varphi(y) g_j(y) \mathrm{d}y = 0, \qquad \int \varphi(y) g_j(y) y \mathrm{d}y = 0, \qquad \int \varphi(y) g_j(y) y^2 \mathrm{d}y = 0. \tag{7}$$

   Remark. We can find functions $g_1, g_2, \cdots, g_m$ that satisfy the orthogonality conditions (6) and (7) using the Gram-Schmidt process.

3. **Assumptions:**

(a) The expectations of $g_j(Y)$, for $j = 1, 2, \cdots, m$, are given by the following equations

$$\mathbb{E}[g_j(Y)] = \int g_j(y) p_Y(y) \mathrm{d}y = c_j, \qquad \text{for all } j = 1, \cdots, m; \tag{8}$$

(b) Y has the zero mean and unit variance, i.e.,

$$\mathbb{E}[Y] = 0, \qquad \text{and} \qquad \mathrm{Var}[Y] = 1. \tag{9}$$

4. **Density Function of Y:** If the probability density p_Y satisfies the constraints (6) - (9) and also has the *largest* entropy among all such densities, then p_Y must be of the form

$$p_Y(y) = A \exp\left(\sum_{j=1}^{m+2} a_j g_j(y) \right), \tag{10}$$

where we let $g_{m+1}(y) = y$ and $g_{m+2}(y) = y^2$, A is the normalizing constant to ensure that p_Y is a valid probability density function, and $a_1, \cdots, a_{m+2}$ are chosen so that (8) and (9) are satisfied.

5. **Approximate Maximum Entropy Density:** We further require p_Y to be close to φ, where φ is the density function of the standard normal distribution, then

$$\begin{aligned} p_Y(y) &= A \exp\left(-\frac{y^2}{2} + a_{m+1} y + \left(a_{m+2} + \frac{1}{2} \right) y^2 + \sum_{j=1}^{m} a_j g_j(y) \right) \\ &\approx \widetilde{A} \varphi(y) \left(1 + a_{m+1} y + \left(a_{m+2} + \frac{1}{2} \right) y^2 + \sum_{j=1}^{m} a_j g_j(y) \right) \\ &=: \tilde{p}_Y(y), \end{aligned}$$

where $\widetilde{A} = \sqrt{2\pi} A$ and we use the approximation $e^{\varepsilon} \approx 1 + \varepsilon$ in the last step.

Under our assumptions earlier, $\tilde{p}_Y$ must satisfy the following constraints

$$\begin{aligned} 1 &= \int \tilde{p}_Y(y) \mathrm{d}y = \widetilde{A} \left(1 + a_{m+2} + \frac{1}{2} \right), \\ 0 &= \mathbb{E}[Y] = \int \tilde{p}_Y(y) y \mathrm{d}y = \widetilde{A} a_{m+1}, \\ 1 &= \mathbb{E}[Y^2] = \int \tilde{p}_Y(y) y^2 \mathrm{d}y = \widetilde{A} \left(1 + 3 \left(a_{m+2} + \frac{1}{2} \right) \right), \\ c_j &= \int \tilde{p}_Y(y) g_j(y) \mathrm{d}y = \widetilde{A} a_j, \qquad \text{for all } j = 1, \cdots, m. \end{aligned}$$

From the equations above, we can solve

$$\begin{aligned} a_j &= c_j, \qquad \text{for all } j = 1, \cdots, m, \\ a_{m+1} &= 0, \qquad a_{m+2} = -\frac{1}{2}, \qquad \widetilde{A} = 1. \end{aligned}$$

It follows that the resulting density function $\tilde{p}_Y$ is

$$\tilde{p}_Y(y) = \varphi(y)\left(1 + \sum_{j=1}^{m} c_j g_j(y)\right), \tag{11}$$

which is referred to as the *approximate maximum entropy density*.

6. **Entropy of $\tilde{p}_Y$:** Using the definition of the entropy, we have

$$\begin{aligned}
H(Y) &= -\int p_Y(t)\log p_Y(y)\mathrm{d}y \\
&\approx -\int \tilde{p}_Y(t)\log \tilde{p}_Y(y)\mathrm{d}y \\
&= -\int \varphi(y)\left(1+\sum_{j=1}^{m} c_j g_j(y)\right)\log\left(\varphi(y)\left(1+\sum_{j=1}^{m} c_j g_j(y)\right)\right) \\
&\approx -\int \varphi(y)\log\varphi(y)\mathrm{d}y - \sum_{j=1}^{m} c_j \int \varphi(y) g_j(y)\log\varphi(y)\mathrm{d}y \\
&\qquad - \int \varphi(y)\left(1+\sum_{j=1}^{m} c_j g_j(y)\right)\log\left(1+\sum_{j=1}^{m} c_j g_j(y)\right)\mathrm{d}y \\
&= H(Z) - \sum_{j=1}^{m} c_j \int \varphi(y) g_j(y)\log\varphi(y)\mathrm{d}y - \sum_{j=1}^{m} c_j \int \varphi(y) g_j(y)\mathrm{d}y \\
&\qquad - \frac{1}{2}\sum_{j=1}^{m} c_j^2 \int \varphi(y) g_j^2(y)\mathrm{d}y - o\left(\sum_{j=1}^{m} c_j^2 \int \varphi(y) g_j^2(y)\mathrm{d}y\right) \\
&= H(Z) - \frac{1}{2}\sum_{j=1}^{m} c_i^2 + o\left(\sum_{j=1}^{m} c_j^2\right),
\end{aligned}$$

where Z is the standard normal random variable, we use the conditions (6) and (7), and the expansion $(1+\varepsilon)\log(1+\varepsilon) \approx \varepsilon + \frac{1}{2}\varepsilon^2 + o(\varepsilon^2)$ for small ε, and $\log\varphi(y) = -\frac{1}{2}\log(2\pi) - \frac{1}{2}y^2$.

Based on the calculation above, we have

$$H(Z) - H(Y) \approx J_m(Y) := \frac{1}{2}\sum_{j=1}^{m}\big(\mathbb{E}[g_j(Y)]\big)^2.$$

Remark. Up to this point, what remains is to choose an appropriate value of m and appropriate basis functions $g_1, g_2, \cdots, g_m$.

7. **Choices of m:**

   (a) If $m = 2$, we can make

      i. g_1 an odd function, reflecting symmetry vs. asymmetry, and

ii. g_2 an even function, reflecting sub-Gaussian (negative kurtosis) vs. super-Gaussian (positive kurtosis) distributions.

In this case, we have

$$J_2(Y) = \beta_1\big(\mathbb{E}[g_1(Y)]\big)^2 + \beta_2\big(\mathbb{E}[g_2(Y)] - \mathbb{E}[g_2(Z)]\big)^2, \tag{12}$$

where $\beta_1 > 0$ and $\beta_2 > 0$.

(b) If $m = 1$, we have

$$J_1(Y) = \beta\big(\mathbb{E}[g_1(Y)] - \mathbb{E}[g_1(Z)]\big)^2, \tag{13}$$

where $\beta > 0$.

8. **Choices of $\{g_j\}_{j=1}^m$:**

(a) logcosh function: $g(y) = \frac{1}{\alpha}\log\cosh(\alpha y)$, where $\alpha \in [1, 2]$;

(b) exp function: $g(y) = -\exp(-\frac{1}{2}y^2)$.

V. FastICA Algorithm for a Single Source Component

1. **Goal:** Consider a single ($m = 1$) source component $Y = \mathbf{w}^\top X$, where the p-vector $\mathbf{w}$ represents a direction for a one-dimensional projection. We wish to find $\mathbf{w}$ that maximizes the approximation (13) subject to the constraint $\mathbb{E}[(\mathbf{w}^\top X)^2] = \|\mathbf{w}\|_2^2 = 1$ on the projection.

Remark. In the criterion above, $\mathbf{w}$ is to be the direction that makes the density of the one-dimensional projection $Y = \mathbf{w}^\top X$ as *far away* from the Gaussian density as possible.

2. **Problem Formulation:** We solve the following optimization problem

$$\begin{aligned} &\text{maximize } J_1(Y) = \beta\big(\mathbb{E}[g_1(Y)] - \mathbb{E}[g_1(Z)]\big)^2 \\ &\text{subject to } \|\mathbf{w}\|_2^2 = 1. \end{aligned}$$

Because the maxima of $J_1(\mathbf{w}^\top X)$ are typically obtained at certain maxima of $\mathbb{E}[g_1(\mathbf{w}^\top X)]$, we work with

$$F(\mathbf{w}) := \mathbb{E}[g_1(\mathbf{w}^\top X)] - \frac{\lambda}{2}(\|\mathbf{w}\|_2^2 - 1), \tag{14}$$

where $\lambda > 0$ is the Lagrangian multiplier.

3. **Newtow-Raphson Algorithm:** We apply the Newton-Raphson algorithm to maximize (14). The iterations are

$$\mathbf{w} \longleftarrow \mathbf{w} - \left(\frac{\partial^2 F(\mathbf{w})}{\partial \mathbf{w}\partial \mathbf{w}^\top}\right)^{-1}\left(\frac{\partial F(\mathbf{w})}{\partial \mathbf{w}}\right). \tag{15}$$

Note that

$$\frac{\partial F(\mathbf{w})}{\partial \mathbf{w}} = \mathbb{E}[X g'(\mathbf{w}^\top X)] - \lambda \mathbf{w}.$$

Any stationary point must satisfy $\frac{\partial F(\mathbf{w})}{\partial \mathbf{w}} = \mathbf{0}_p$. Premultiplying both sides of the preceding equation by $\mathbf{w}$ yields

$$\lambda = \mathbb{E}[\mathbf{w}^\top X g'(\mathbf{w}^\top X)].$$

In addition, we have

$$\begin{aligned}\frac{\partial^2 F(\mathbf{w})}{\partial \mathbf{w} \partial \mathbf{w}^\top} &= \mathbb{E}[XX^\top g''(\mathbf{w}^\top X)] - \lambda \mathbf{I}_p \\ &\approx \mathbb{E}[XX^\top]\,\mathbb{E}[g''(\mathbf{w}^\top X)] - \lambda \mathbf{I}_p \\ &= \big(\mathbb{E}[g''(\mathbf{w}^\top X)] - \lambda\big)\mathbf{I}_p,\end{aligned}$$

where we use the fact that X has been sphered.

It follows that the iterations of $\mathbf{w}$ are

$$\mathbf{w} \longleftarrow \mathbf{w} - \frac{\mathbb{E}[X g'(\mathbf{w}^\top X)] - \lambda \mathbf{w}}{\mathbb{E}[g''(\mathbf{w}^\top X)] - \lambda}. \tag{16}$$

In practice, the expectation can be approximated using the sample average.

4. **Alternative Expression of (16):** The k-th iterate of (16) is

$$\mathbf{w}_k = \mathbf{w}_{k-1} - \frac{\mathbb{E}[X g'(\mathbf{w}_{k-1}^\top X)] - \lambda \mathbf{w}_{k-1}}{\mathbb{E}[g''(\mathbf{w}_{k-1}^\top X)] - \lambda}.$$

Multiplying both sides by $\mathbb{E}[g''(\mathbf{w}_{k-1}^\top X)] - \lambda$ and rearranging terms yields

$$\mathbf{w}_k(\lambda - \mathbb{E}[g''(\mathbf{w}_{k-1}^\top X)]) = \mathbb{E}[X g'(\mathbf{w}_{k-1}^\top X)] - \mathbf{w}_{k-1}\,\mathbb{E}[g''(\mathbf{w}_{k-1}^\top X)].$$

Since we divide $\mathbf{w}_k$ by its norm $\|\mathbf{w}_k\|$ at each step, the factor $(\lambda - \mathbb{E}[g''(\mathbf{w}_{k-1}^\top X)])$ on the left-hand side is *not* necessary, and the update equation at the k-th iterate becomes

$$\mathbf{w}_k = \mathbb{E}[X g'(\mathbf{w}_{k-1}^\top X)] - \mathbf{w}_{k-1}\,\mathbb{E}[g''(\mathbf{w}_{k-1}^\top X)].$$

5. **Convergence Criterion:** The values of $\mathbf{w}$ can change *substantially* from iteration to iteration; this is because the ICA model *cannot* determine the sign of $\mathbf{w}$, so that $-\mathbf{w}$ and $\mathbf{w}$ become equivalent and define the same direction.

   Hence, "convergence" of the FastICA algorithm is taken to mean that successive iterative values of $\mathbf{w}$ are oriented in the same direction, i.e., the inner product between two iterations of $\mathbf{w}$ is very close to 1.

6. **Complete FastICA Algorithm:**

Algorithm 1 FactICA Algorithm for a Single Source Component

1: Center and whiten the data to give X;
2: Choose an initial version of the p-vector $\mathbf{w}$ with unit norm;
3: Choose g to be any non-quadratic density with the first and second partial derivatives g' and g'', respectively.
4: Let

$$\mathbf{w} \longleftarrow \mathbb{E}[Xg'(\mathbf{w}^\top X)] - \mathbf{w}\,\mathbb{E}[g''(\mathbf{w}^\top X)].$$

In practice, the expectations are estimated using sample averages.
5: Let $\mathbf{w} \leftarrow \mathbf{w}/\|\mathbf{w}\|_2$;
6: Iterate between steps 4 and 5. Stop when convergence is attained.

VI. FastICA Algorithm for Multiple Source Components

1. **Goal:** Extract multiple independent projections of X.
2. **Method 1 — Deflation Method:** A single component that is orthogonal to *all* previously found components (using the Gram-Schmidt process), and then the resulting new component is normalized.

Algorithm 2 FactICA Algorithm for Multiple Source Components (Deflation Method)

1: Center and whiten the data to give X;
2: Decide on the number, m, of independent components to be extracted;
3: For $j = 1, 2, \cdots, m$,

(a) Initialize (e.g., randomly) the p-vector $\mathbf{w}_j$ to have unit norm;

(b) Let

$$\mathbf{w}_j \longleftarrow \mathbb{E}[Xg'(\mathbf{w}_j^\top X)] - \mathbf{w}_j\,\mathbb{E}[g''(\mathbf{w}_j^\top X)].$$

be the FastICA single component update for $\mathbf{w}_j$. In practice, the expectations are estimated using sample averages;

(c) Use the Gram-Schmidt process to orthogonalize $\mathbf{w}_j$ with respect to the previously chosen $\mathbf{w}_1, \cdots, \mathbf{w}_{j-1}$ as

$$\mathbf{w}_j \longleftarrow \mathbf{w}_j - \sum_{k=1}^{j-1} (\mathbf{w}_j^\top \mathbf{w}_k)\mathbf{w}_k;$$

(d) Let $\mathbf{w}_j \leftarrow \mathbf{w}_j / \|\mathbf{w}_j\|_2$;

(e) Iterate $\mathbf{w}_j$ until convergence.

4: Set $j \leftarrow j + 1$. If $j \leq m$, return to Step 3.

3. Method 2 — Parallel Method: The single component routine is carried out *in parallel* for each independent component to be extracted, and then a symmetric orthogonalization is carried out on all components simultaneously.

Algorithm 3 FactICA Algorithm for Multiple Source Components (Parallel Method)

1: Center and whiten the data to give X;
2: Decide on the number, M, of independent components to be extracted;
3: Initialize (e.g., randomly) the p-vectors $\mathbf{w}_1, \cdots, \mathbf{w}_m$, each to have unit norm. Let $\mathbf{W} = (\mathbf{w}_1, \cdots, \mathbf{w}_m)^\top$;
4: Carry out a symmetric orthogonalization of $\mathbf{W}$ by

$$\mathbf{W} \longleftarrow (\mathbf{W}\mathbf{W}^\top)^{-\frac{1}{2}}\mathbf{W};$$

5: For each $j = 1, 2, \cdots, m$, let

$$\mathbf{w}_j \longleftarrow \mathbb{E}[Xg'(\mathbf{w}_j^\top X)] - \mathbf{w}_j\,\mathbb{E}[g''(\mathbf{w}_j^\top X)].$$

be the FastICA single-component update for $\mathbf{w}_j$. In practice, the expectations are estimated using sample averages;
6: Carry out another symmetric orthogonalization of $\mathbf{W}$;
7: If convergence has not occurred, return to Step 5.

4. **Comparison of the Deflation and Parallel Methods:**
   (a) The deflation method extracts independent components sequentially one at a time, whereas
   (b) the parallel method extracts all the independent components at the same time.

VII. Maximum Likelihood ICA

1. **Main Idea:** Specify a parametric distribution, p_S, for the latent source variables S and then apply the maximum-likelihood (ML) method to estimate the parameters of that distribution.

2. **Setup:** We only consider the square mixing case (i.e., $m = p$) and the linear mixing case.

3. **Density Functions of X:** Let p_S be the density function of S. Since $X = \mathbf{A}S$, where $\mathbf{A} \in \mathbb{R}^{p\times p}$ is nonsingular, we let $\mathbf{W} = \mathbf{A}^{-1}$ and the density function of X is

$$p_X(\mathbf{x}) = |\det(\mathbf{W})| p_S(\mathbf{s}).$$

Since the sources are assumed to be independent, we have

$$p_X(\mathbf{x}) = |\det(\mathbf{W})| \prod_{j=1}^{m} p_{S_j}(\mathbf{w}_j^\top \mathbf{x}), \tag{17}$$

where p_{S_j} is the density of S_j and $\mathbf{w}_j^\top$ is the j-th row of $\mathbf{W}$.

4. **Log-likelihood Function:** Given n i.i.d. observations, $\mathbf{x}_1, \cdots, \mathbf{x}_n$, the average log-likelihood function for $\mathbf{W}$ is

$$L(\mathbf{W}) := \log|\det(\mathbf{W})| + \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m}\log p_{S_j}(\mathbf{w}_j^\top \mathbf{x}_i). \tag{18}$$

5. **Algorithm:** We derive a fixed-point algorithm that maximizes (18) numerically. Note that

$$\begin{aligned}\frac{\partial L(\mathbf{W})}{\partial \mathbf{W}} &= (\mathbf{W}^\top)^{-1} + \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m}\frac{\partial \log p_{S_j}(\mathbf{w}_j^\top \mathbf{x}_i)}{\partial \mathbf{w}_j} \\ &= (\mathbf{W}^\top)^{-1} + \frac{1}{n}\sum_{i=1}^{n}\mathbf{g}(\mathbf{W}\mathbf{x}_i)\mathbf{x}_i^\top,\end{aligned} \tag{19}$$

where

$$\begin{aligned}\mathbf{g}(\mathbf{W}\mathbf{x}) &= \big(g_1(\mathbf{w}_1^\top \mathbf{x}), g_2(\mathbf{w}_2^\top \mathbf{x}), \cdots, g_m(\mathbf{w}_m^\top \mathbf{x})\big), \\ g_j(\mathbf{w}_j^\top \mathbf{x}) &= \frac{p'_{S_j}(\mathbf{w}_j^\top \mathbf{x})}{p_{S_j}(\mathbf{w}_j^\top \mathbf{x})}.\end{aligned}$$

The update for the k-th iteration of $\mathbf{W}$ is

$$\mathbf{W}_k = \mathbf{W}_{k-1} - \alpha \frac{\partial L(\mathbf{W})}{\partial \mathbf{W}}\bigg|_{\mathbf{W}=\mathbf{W}_{k-1}}, \tag{20}$$

where $\alpha > 0$ is the step size.

Set $\Delta\mathbf{W} = \mathbf{W}_k - \mathbf{W}_{k-1}$. Then, we can rewrite (20) as

$$\Delta\mathbf{W} \propto (\mathbf{W}^\top)^{-1} + \mathbb{E}_{\widehat{F}_n}[\mathbf{g}(\mathbf{W}X)X^\top],$$

where $\mathbb{E}_{\widehat{F}_n}$ denotes the sample average. Post-multiplying the right-hand side of the preceding equation by $\mathbf{W}^\top\mathbf{W}$ gives the *fixed-point algorithm*

$$\mathbf{W} \longleftarrow \mathbf{W} + \alpha_0(\mathbf{I}_m + \mathbb{E}_{\widehat{F}_n}[\mathbf{g}(\mathbf{W}X)X^\top\mathbf{W}^\top])\mathbf{W}, \tag{21}$$

where $\alpha_0 > 0$ is the step size which may be reduced in size until convergence.

Remark. The modification above produces an algorithm that avoids the matrix inversions in (20) and speeds up convergence considerably.

VIII. Product Density ICA

1. **Goal:** This section presents the *product density ICA*, abbreviated as ProDenICA, which has a similar flavor as the maximum likelihood ICA presented in the preceding section.

2. **Setup:** We only consider the square mixing case (i.e., $m = p$) and the linear mixing case.

3. **Tilted Gaussian Density Function:** Since components of $S = (S_1, S_2, \cdots, S_m)$ are independent, we can write the joint density function of S as

$$p_S(\mathbf{s}) = \prod_{j=1}^{m} p_{S_j}(s_j)$$

as before. For each component, in order to represent the departure from the Gaussian distribution as far as possible, we let each component density as

$$p_{S_j}(s_j) = \varphi(s_j)e^{g_j(s_j)}, \qquad \text{for all } j = 1, 2, \cdots, m, \tag{22}$$

where φ is the standard Gaussian density and g_j's satisfy the normalization conditions required by a density function. The density in (22) is known as the *tilted* Gaussian density function.

4. **Problem Formulation:** Let $X = \mathbf{A}S$, where $A \in \mathbb{R}^{m\times m}$ is assumed to be an *orthogonal* matrix so that $\mathbf{A}^\top = \mathbf{A}^{-1}$. Then, with the data $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n \in \mathbb{R}^m$, the log-likelihood function is

$$L(\mathbf{A}) = \sum_{i=1}^{n}\sum_{j=1}^{m}\big(\log\varphi_j(\mathbf{a}_j^\top\mathbf{x}_i) + g_j(\mathbf{a}_j^\top\mathbf{x}_i)\big),$$

where $\mathbf{a}_j$ is the j-th column of $\mathbf{A}$. We maximize L above under the constraints that $\mathbf{A}$ is orthogonal and

$$\int \varphi(s)e^{g_j(s)}\mathrm{d}s = 1, \qquad \text{for all } j = 1, 2, \cdots, m.$$

Combining these constraints, we maximize the following objective function

$$\sum_{j=1}^{m}\left[\frac{1}{n}\sum_{i=1}^{n}\big(\log\varphi_j(\mathbf{a}_j^\top\mathbf{x}_i) + g_j(\mathbf{a}_j^\top\mathbf{x}_i)\big) - \int\varphi(s)e^{g_j(s)}\mathrm{d}s - \lambda_j\int(g_j'''(s))^2\mathrm{d}s\right], \tag{23}$$

where $\lambda_j > 0$ is the penalty parameter and. In (23), for each j, two penalty terms have subtracted:

(a) the first penalty enforces the density constraint $\int\varphi(s)e^{g_j(s)}\mathrm{d}s = 1$, and

(b) the second is a roughness penalty, which guarantees that the maximizer $\hat{g}_j$ is a quartic spline with knots at the observed values of $s_{i,j} = \mathbf{a}_j^\top\mathbf{x}_i$.

Remark 1. Note that, as $\lambda_j \to \infty$ for all $j = 1, 2, \cdots, m$, the resulting density function is approaching the standard Gaussian density.

Remark 2. It can be shown that each solution densities $\hat{p}_{S_j} = \varphi e^{\hat{g}_j}$ has mean zero and variance one.

5. **Algorithm:** We fit the functions g_j and directions $\mathbf{a}_j$ by optimizing (23) in an alternating fashion, as described in the following algorithm.

Algorithm 4 ProDenICA Algorithm

1: Initialize $\mathbf{A}$ (random Gaussian matrix followed by orthogonalization);
2: Alternate until convergence of $\mathbf{A}$:

(a) Given $\mathbf{A}$, optimize (23) with respect to g_j (separately for each j);

(b) Given g_j, for each $j = 1, 2, \cdots, p$, perform one step of a fixed point algorithm towards finding the optimal $\mathbf{A}$.

6. **Details of Step 2(a) in Algorithm 4:** In Step 2(a), with the matrix $\mathbf{A}$ being fixed, we maximize with respect to g_j's, which corresponds to m semi-parametric density estimation problems.

   Since m components in (23) are separable, we can just consider a single j-th component and maximize

$$\frac{1}{n}\sum_{i=1}^{n}\big(\log\varphi(s_i)+g(s_i)\big)-\int\varphi(s)e^{g(s)}\mathrm{d}s-\lambda\int(g'''(s))^2\mathrm{d}s. \tag{24}$$

   Even though the second integral leads to a smoothing spline, the first integral is problematic and requires an approximation.

   We construct a fine grid of T values s_t^* in increments Δ covering the observed values s_i's, and count the number of s_i in the resulting bins

$$y_t^*=\frac{\big|\{s_i\,|\,s_i\in(s_t^*-\Delta/2,s_t^*+\Delta/2)\}\big|}{n}.$$

   Then, we can approximate (24) as

$$\sum_{t=1}^{T}\Big[y_t^*\big(\log\varphi(s_t^*)+g(s_t^*)\big)-\Delta\varphi(s_t^*)e^{g(s_t^*)}\Big]-\lambda\int(g'''(s))^2\mathrm{d}s.$$

7. **Details of Step 2(b) in Algorithm 4:** In Step 2(b), with g_j's being fixed, we maximize with respect to $\mathbf{A}$. By algebra and the assumption that $\mathbf{A}$ is orthogonal, it is easy to show that the terms involving φ do *not* depend on $\mathbf{A}$. We only need to maximize

$$\frac{1}{n}\sum_{j=1}^{m}\sum_{i=1}^{n}g_j(\mathbf{a}_j^\top\mathbf{x}_i),\qquad\text{with respect to }\mathbf{a}_1,\mathbf{a}_2,\cdots,\mathbf{a}_m.$$

   Then, for each j, we update $\mathbf{a}_j$ using the Newton-Raphson algorithm given by

$$\mathbf{a}_j\longleftarrow\mathbb{E}_{\widehat{F}_n}[Xg_j'(\mathbf{a}_j^\top X)]-\mathbf{a}_j\,\mathbb{E}_{\widehat{F}_n}[g_j''(\mathbf{a}_j^\top X)].$$

Since g_j is a fitted quartic (or cubic) spline, the first and second derivatives are readily available.

In order to make $\mathbf{A}$ satisfy the orthogonality assumption, we orthogonalize $\mathbf{A}$ using the symmetric square-root transformation

$$\mathbf{A} \longleftarrow (\mathbf{A}\mathbf{A}^\top)^{-\frac{1}{2}}\mathbf{A}.$$

If, in particular, $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$ is the SVD of $\mathbf{A}$, we have

$$\mathbf{A} \longleftarrow \mathbf{U}\mathbf{V}^\top.$$

References

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2009). *The Elements of Statistical Learning*. Vol. 1. Springer Series in Statistics. New York, NY, USA: Springer New York Inc.

Izenman, Alan J (Mar. 2009). *Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning*. en. Springer Science & Business Media.