02471 Machine Learning for Signal Processing

Week 8: Dictionary learning and source separation

This exercise is based on S. Theodoridis: Machine Learning, A Bayesian and Optimization Perspective 2nd edition, section(s) 2,2, 2.5, 19.5.

The objective of this exercise is to understand the derivation behind ICA, and apply ICA as a source separation technique.

NOTE: in this exercise we are going to derive and implement ICA based on mutual information. The resulting algorithm is primarily for education purposes. If you use ICA for "production", use a library e.g the FastICA library, available for both Matlab and Python.

Overview

The exercise have the following structure:

8.1 will investigate why ICA cannot carry out separation for Gaussian signals, and additionally, provide some mathematical tools we need to derive the ICA update formula.

8.2 will derive the ICA update formula eq (19.59) in the book.

8.3 will implement the ICA update and run ICA on simulated data in order to get better acquainted with the performance of ICA.

8.4 are generating a simplified artificial signal resembling the recordings obtained in EEG and apply ICA to separate the sources.

8.5 is using ICA to perform source separation on audio.

You can in principle start with exercise 3, and just use the update formula eq. (19.60) from the book to implement ICA. Exercise 2 assumes that exercise 1 has been completed, and exercise 4 and 5 requires a working ICA algorithm (implemented in exercise 3).

Notation

- $J(\boldsymbol{\theta})$ is a cost function that we are seeking to minimize with respect to $\boldsymbol{\theta}$.
- n indicates the time step as we collect data $(y_n, \boldsymbol{x}_n)$, where y_n is our target, or desired signal, and $\boldsymbol{x}_n$ is our filter input.
- $\mathcal{N}(\mu, \sigma^2)$ denotes a normal (Gaussian) distribution with mean value μ and variance σ^2.
- $U(a, b)$ denotes a uniform distribution with a as lower limit and b as upper limit.
- $\mathbf{x} \sim$ distribution$(\cdot)$ means that $\mathbf{x}$ is a random variable that follows the associated distribution.

Code

The code can be found in the .m and .py files named in the same way as exercises, ie. the code for exercise 8.x.y is in the file 8_x_y.m (or .py).

For coding exercises that requires implementation we will usually write `complete this line` where the implementing should be done.

Solutions

The solution is provided for all derivation exercises, and often hints are provided at the end of the document. If you get stuck, take a look at the hints, and if you are still stuck, take a look in the solution to see the approach being taken. Then try to do it on your own.

Solutions are also provided for some coding exercises. If you get stuck, take a look at the solution, and then try to implement it on your own.

8.1 ICA and Gaussian signals

In this exercise we will provide proof of the claim that ICA cannot identify Gaussian distributed sources. Before completing this exercise, be sure to read section 2.2.5 and 19.5.1.

The ICA model is written as

$$\mathbf{x} = A\mathbf{s}$$

where we have observed a realization of $\mathbf{x}$ ($\mathbf{x}$ is a random vector), and we want to provide an estimate for the sources $\mathbf{s}$, denoted as $\mathbf{z}$, where $\mathbf{z} := \hat{\mathbf{s}}$.

Assuming that A is a square invertible matrix, we can write

$$\mathbf{z} = W\mathbf{x} = WA\mathbf{s}$$

where, if $W = A^{-1}$, we get perfect recovery of $\mathbf{s}$. Hence, ICA consists of estimating W. To do that, we assume that the underlying sources are statistically independent (that is, $p(\mathbf{s}) = \prod_{i=1}^{l} p(\mathrm{s}_i)$) and use that as our optimization criteria.

Now, consider the case where we have l Gaussian distributed sources with a mean of zero and diagonal covariance matrix $\Sigma_s = I$:

$$p_{\mathbf{s}}(\mathbf{s}) = \frac{1}{(2\pi)^{l/2}} \exp\left(-\frac{\|\mathbf{s}\|^2}{2}\right)$$

We will now investigate how that affects the observed signals $\mathbf{x}$.

In essence, the ICA model assumes that $\mathbf{x}$ is a linear transformation of $\mathbf{s}$. That can be considered as a change random variables in order to obtain $p_x(\mathbf{x})$ from $p_s(\mathbf{x})$. We need the theorem of change of random variables (eq. 2.45), defined as (where $\mathbf{x}$, and $\mathbf{y}$ is the random variables):

$$\begin{aligned} p_{\mathbf{y}}(\mathbf{y}) &= \frac{p_{\mathbf{x}}(\mathbf{x})}{|\det(J(\mathbf{y}, \mathbf{x}))|} \\ \mathbf{y} &= f(\mathbf{x}) \\ \mathbf{x} &= f^{-1}(\mathbf{y}) \end{aligned}$$

$|\det(\cdot)|$ is the determinant of a matrix, and $J(\mathbf{y}, \mathbf{x})$ is the Jacobian matrix defined as

$$J(\mathbf{y}, \mathbf{x}) = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \cdots & \frac{\partial y_1}{\partial x_l} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \cdots & \frac{\partial y_2}{\partial x_l} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_l}{\partial x_1} & \frac{\partial y_l}{\partial x_2} & \cdots & \frac{\partial y_l}{\partial x_l} \end{bmatrix}.$$

Exercise 8.1.1

Assume for simplicity that A is orthogonal (that implies $A^T A = I$, and $\det(A) = \pm 1$). Apply the change of variable theorem, we need to identify $f(\cdot)$, $f^{-1}(\cdot)$ and derive $J(\mathbf{x}, \mathbf{s})$. Identify these functions given the ICA model $\mathbf{x} = A\mathbf{s}$, and show that $J(\mathbf{x}, \mathbf{s}) = A$.

Exercise 8.1.2

Use the obtained results and plug into the change of random variables theorem to show that the distribution for $p_{\mathbf{x}}(\mathbf{x})$ is

$$p_{\mathbf{x}}(\mathbf{x}) = \frac{1}{(2\pi)^{l/2}} \exp\left(-\frac{\|\mathbf{x}\|^2}{2}\right)$$

Now we can se that $p_x(\mathbf{x})$ has the exact same distribution as $p_s(\mathbf{s})$.

The implication is that the observation variables and the source variables have the same distribution under the ICA model, hence, the ICA model is not able to separate Gaussian sources.

8.2 Derivation of ICA based on mutual information

In this exercise, we will derive the ICA update formula:

$$W^{(i)} = W^{(i-1)} + \mu_i \big(I - \mathbb{E}\big[\boldsymbol{\phi}(\mathbf{z})\mathbf{z}^T\big]\big)(W^{(i-1)})^{-T}$$

Before completing this exercise, be sure to read section 2.5 and 19.5.4. We are going through the exact steps from eq. (19.51)–(19.59) in the book. I feel the book is too brief here, so this exercise is a step-by-step guide on how we go from mutual information to an update formula.

This can be a rather difficult and time-consuming exercise depending on your mathematical background, so consider using this as a tutorial on the ICA derivation (especially if hand-deriving is meaningless for your learning).

The starting point of ICA based on mutual information is the minimization of the expression

$$\hat{W} = \arg\min_{W} I(\mathbf{z})$$

Where $\mathbf{z}$ is the estimate of the sources $\mathbf{s}$ we want to discover.

The function $I(\mathbf{z})$ is called the mutual information (since it measure how much random variables have in common), and $H(\mathbf{z})$ is called the differential entropy. The entropy is inspired from physics, and measures the amount of randomness of the distribution $p(\mathbf{z})$.

Our goal is to minimize $I(\mathbf{z})$ w.r.t. W, since if e.g. $I(\mathrm{z}_1, \mathrm{z}_2) = 0$ the variables z_1 and z_2 have no mutual information and are statistically independent (observing z_1 reveals no information about z_2 and vice versa).

We want to apply the ICA model to $\mathbf{z}$, compute the gradients of $I(\mathbf{z})$ w.r.t. W (the matrix we want to estimate). Then we can apply gradient descent estimate W. In order do that, we need to rewrite $I(\mathbf{z})$ as we cannot optimize it in the current form.

Exercise 8.2.1

First we rewrite $I(\mathbf{z})$ to a more convenient form. The mutual information in the two-dimensional case is (eq. 2.158):

$$I(\mathrm{x};\mathrm{y}) := \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x,y)\ln\frac{p(x,y)}{p(x)p(y)}dxdy$$

Show that this expression can be rewritten to $I(\mathrm{x};\mathrm{y}) = -H(\mathrm{x};\mathrm{y}) + H(\mathrm{x}) + H(\mathrm{y})$ which, for the l–dimensional case leads to

$$\begin{aligned} I(\mathbf{z}) =& - H(\mathbf{z}) + \sum_{i=1}^{l} H(\mathrm{z}_i) \\ H(\mathbf{z}) =& - \int_{-\infty}^{\infty} p(\mathbf{z})\ln p(\mathbf{z})d\mathbf{z} \end{aligned}$$

Exercise 8.2.2 (This is a difficult exercise)

We need to have W in our expression as well in order to optimize w.r.t that, so we need to perform additional rewrites. Using the result from the change of variables in the previous exercise (8.1), we have $p_{\mathbf{z}}(\mathbf{z}) = p_{\mathbf{x}}(\mathbf{x})/|\det(W)|$. Show that, by applying these rewrites, the mutual information can be written as

$$I(\mathbf{z}) = -H(\mathbf{x}) - \ln|\det(W)| + \sum_{i=1}^{l} H(\mathrm{z}_i)$$

Exercise 8.2.3

Using the previously found results, argue or show that our original minimization problem is equivalent to the following maximization problem

$$\begin{aligned} \hat{W} &= \arg\min_{W} I(\mathbf{z}) \qquad \Rightarrow \\ \hat{W} &= \arg\max_{W}\ J(W) \\ J(W) &:= \ln|\det(W)| + \mathbb{E}\left[\sum_{i=1}^{l}\ln p_i(\mathrm{z}_i)\right] \end{aligned}$$

We now have the cost function in a form we need, so now the focus can turn to deriving the gradient.

Exercise 8.2.4 (Optional)

We will take the derivative w.r.t $J(W)$ term by term. Use the following formula:

$$\frac{d}{dW}\det(W) = W^{-T}\det(W), \quad W^{-T} := (W^{-1})^T$$

and the chain rule to show that

$$\frac{d}{dW}\ln|\det(W)| = W^{-T}$$

Exercise 8.2.5 (Optional)

Now focusing on the second term of the cost function $J(W)$, we can, since $\ln x$ is integrable for $x > 0$, interchange the expectations with the derivative[1], that is

$$\frac{d}{dx}\mathbb{E}[\ln x] = \mathbb{E}\left[\frac{d}{dx}\ln x\right]$$

Argue that, in our case, we have $x > 0$. Then, compute $\frac{d}{dW}\ln p_i(\mathrm{z}_i)$ and show (by using the chain rule) that we obtain

$$\frac{d}{dW}\ln p_i(\mathrm{z}_i) = \frac{1}{p(\mathrm{z}_i)}\frac{\partial p_i(\mathrm{z}_i)}{\partial \mathrm{z}_i}\frac{d\mathrm{z}_i}{dW}$$

Exercise 8.2.6 (Optional - this is a difficult exercise)

We need to compute two gradients now, $\frac{\partial p_i(\mathrm{z}_i)}{\partial \mathrm{z}_i}$ and $\frac{d\mathrm{z}_i}{dW_{k,m}}$, which we will do seperately. To compute $\frac{d\mathrm{z}_i}{dW_{k,m}}$, we need an expression for z_i that includes $W_{k,m}$. In exercise 8.1, we showed that, under the ICA model, $\mathrm{z}_i = W_{i,:}^T\mathbf{x}$ where $W_{i,:}$ denotes the i'th row in W. Use this result to obtain

$$\frac{d\mathrm{z}_i}{dW_{k,m}} = \mathrm{x}_m, \quad i = k, \text{ otherwise zero}$$

If we define a short-hand notation:

$$\boldsymbol{\phi}(\mathbf{z}) := \begin{bmatrix}\frac{p_1'(\mathrm{z}_1)}{p(\mathrm{z}_1)} & \frac{p_2'(\mathrm{z}_2)}{p(\mathrm{z}_2)} & \cdots & \frac{p_l'(\mathrm{z}_l)}{p(\mathrm{z}_l)}\end{bmatrix}^T, \qquad p_i'(\mathrm{z}_i) := \frac{\partial p_i(\mathrm{z}_i)}{\partial \mathrm{z}_i}$$

show that result can be rewritten in matrix notation as

$$\frac{d}{dW}\sum_{i=1}^{l}\ln p_i(\mathrm{z}_i) = \boldsymbol{\phi}(\mathbf{z})\mathbf{x}^T$$

Exercise 8.2.7

By substiting the derivatives we have found in the previous two exercises, we obtain the final result for the gradient

$$\frac{d}{dW}J(W) = W^{-T} - \mathbb{E}\big[\boldsymbol{\phi}(\mathbf{z})\mathbf{x}^T\big]$$

Use the result and apply gradient descent to obtain the following update rule

$$W^{(i)} = W^{(i-1)} + \mu_i\big(I - \mathbb{E}\big[\boldsymbol{\phi}(\mathbf{z})\mathbf{z}^T\big]\big)(W^{(i-1)})^{-T}$$

As is noted in the book, the use of the natural gradient descent leads to replacing $(W^{(i-1)})^{-T}$ with $W^{(i-1)}$, and this is the update formula we will use in the subsequent exercise, i.e

$$W^{(i)} = W^{(i-1)} + \mu_i\big(I - \mathbb{E}\big[\boldsymbol{\phi}(\mathbf{z})\mathbf{z}^T\big]\big)W^{(i-1)}$$

A missing piece in applying the algorithm is that we need to choose an expression for $\boldsymbol{\phi}(\mathbf{z})$. From the book "Independent component analysis" by Erkki Oja, Juha Karhunen and Aapo Hyvarinen, two choices are suggested; for super-Gaussian distributed $\mathbf{z}$, use $\boldsymbol{\phi}(\mathbf{z}) = 2\tanh(\mathbf{z})$ and for sub-Gaussian distributed $\mathbf{z}$, use $\boldsymbol{\phi}(\mathbf{z}) = \mathbf{z} - \tanh(\mathbf{z})$.

[1] `https://en.wikipedia.org/wiki/Leibniz_integral_rule#Measure_theory_statement`

8.3 ICA for simulated data

In this exercise we are going to test ICA on a simple dataset and compare to PCA.

Consider the following generative model

$$\begin{aligned} \mathrm{s}_1 &\sim U(0,1) \\ \mathrm{s}_2 &\sim U(0,1) \\ \mathbf{x} &= A \cdot [\mathrm{s}_1 \; \mathrm{s}_2]^T \end{aligned}$$

where A is a mixing matrix, $A \in \mathbb{R}^{2\times 2}$.

Exercise 8.3.1

Use the code associated with this exercise. Modify the code to use a mixing matrix as $A = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix}$, generate a X matrix of size $2 \times N$, and visualize the results. Apply PCA using a built-in/library function on X, and create a scatter plot with principal component 1 and 2 as the axes.

Exercise 8.3.2

Use the code associated with this exercise (`ICA.m` for matlab). There is a template implementation for ICA. Relate the code to the optimization algorithm we have previously derived and complete the implementation.

Exercise 8.3.3

Use ICA to unmix X, and recover the original source. Try and use both a sub Gaussian and a super Gaussian activation function.The algorithm can only recover the source signal for one of the activation functions. Was the “good” activation function the one you expected? What happens if you don’t zero-mean the data first?

Exercise 8.3.4

Consider different mixing matrices A and/or different distributions for s. E.g. you found in the first exercise that ICA cannot recover multiple Gaussian sources, try to produce that result. Also check for one Gaussian distributed source.

8.4 ICA as source separation on simulated EEG

Electroencephalography (EEG) is an electrophysiological monitoring method to record electrical activity of the brain. Generally the data collection is a non-invasive procedure that takes over a period of time readings from electrodes placed at specific locations on the scalp. These electrodes measure voltage fluctuations that are caused by brain activity. There are multiple applications based on EEG data such as epilepsia diagnosis or brain diseases or disorders, among others.

In this exercise we are generating a simplified artificial signal resembling the recordings obtained in EEG. In EEG, multiple electrodes are placed at different locations on the scalp and thus capturing a combination of waves generated by brain activity.

Run the code associated with this exercise. A sinusoidal signal is generated alongside with a spike signal similar to the ones present when a subject blinks. The EEG reading is obtained from 3 locations (parietal, central and frontal), modeled as having different amplitudes. Use ICA to remove the blink signal. In order to perform the blink removal, identify and remove the component obtained after applying ICA and use the mixing matrix to project back the signals.

Try to add a third source (beta) that is highest at central location. What is the mixing matrix being used? Play with different signals to get comfortable with ICA and how the time series are unmixed.

8.5 ICA as source separation on audio

In this exercise we will work on a set of mixed audio recordings which have been created using instantaneous mixing. Inspect and run the code associated with this exercise.

The audio file contains two channels. Listen to the two channels individually, mixed and after separation. Consider to create your own mixture of audio to run ICA on. Can you create a mixture where ICA is not able to unmix the signals?

Load the `mixture_convolutive.wav` and listen to it. Run ICA on this sound file? Does it still work well? Why/Why not?

Note: Convolutive blind source seperation has been developed to handle the above case, see e.g. “A survey of convolutive blind source separation methods”, by Michael Syskind Pedersen, Jan Larsen, Ulrik Kjems, and Lucas C. Parra.

HINTS

Exercise 8.1.1

Since the model is $\mathbf{x} = A\mathbf{s}$, we get $\mathbf{s} = A^T\mathbf{x}$ (since we assumed $A^{-1} = A^T$), hence $\mathbf{x} = f(\mathbf{s}) = A\mathbf{s}$, and $\mathbf{s} = f^{-1}(\mathbf{x}) = A^T\mathbf{x}$.

Exercise 8.1.2

By using the change of random variable theorem you should get the intermediate result

$$p_{\mathbf{x}}(\mathbf{x}) = \frac{1}{(2\pi)^{l/2}} \exp\left(-\frac{\|A^T\mathbf{x}\|^2}{2}\right) |\det(A^T)|$$

You will also need the rule (Appendix A.1) $\det(A^{-1}) = \frac{1}{\det(A)}$ to obtain the result.

Exercise 8.2.2

The entropy can be written as $H(\mathbf{z}) = -\mathbb{E}[\ln p(\mathbf{z})]$ (show this by the definition of the expectation).

By applying the logarithm to both side and then the expectation we get

$$\mathbb{E}[\ln p_{\mathbf{z}}(\mathbf{z})] = \mathbb{E}[\ln p_{\mathbf{x}}(\mathbf{x})] - \ln|\det(W)|$$

and then this leads to

$$-H(\mathbf{z}) = -H(\mathbf{x}) - \ln|\det(W)|$$