Online Component Analysis, Architectures and Applications

Other titles in Foundations and Trends® in Signal Processing

An Introduction to Quantum Machine Learning for Engineers
Osvaldo Simeone
ISBN: 978-1-63828-058-3

Bilevel Methods for Image Reconstruction
Caroline Crockett and Jeffrey A. Fessler
ISBN: 978-1-63828-002-6

Operating Characteristics for Classical and Quantum Binary Hypothesis Testing
Catherine A. Medlock and Alan V. Oppenheim
ISBN: 978-1-68083-882-4

Foundations of User-Centric Cell-Free Massive MIMO
Özlem Tugfe Demir, Emil Björnson and Luca Sanguinetti
ISBN: 978-1-68083-790-2

Data-Driven Multi-Microphone Speaker Localization on Manifolds
Bracha Laufer-Goldshtein, Ronen Talmon and Sharon Gannot
ISBN: 978-1-68083-736-0

Online Component Analysis, Architectures and Applications

João B. O. Souza Filho
COPPE/Poli Federal University of Rio de Janeiro
jbfilho@poli.ufrj.br

Lan-Da Van
National Yang Ming Chiao Tung University
ldvan@cs.nycu.edu.tw

Tzyy-Ping Jung
University of California, San Diego
jung@sccn.ucsd.edu

Paulo S. R. Diniz
COPPE/Poli Federal University of Rio de Janeiro
diniz@smt.ufrj.br

now
the essence of knowledge
Boston — Delft

Foundations and Trends® in Signal Processing

Published, sold and distributed by:
now Publishers Inc.
PO Box 1024
Hanover, MA 02339
United States
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

Outside North America:
now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

The preferred citation for this publication is

J. B. O. Souza Filho *et al.*. *Online Component Analysis, Architectures and Applications*. Foundations and Trends® in Signal Processing, vol. 16, no. 3-4, pp. 224–429, 2022.

ISBN: 978-1-63828-117-7

Foundations and Trends® in Signal Processing

Volume 16, Issue 3-4, 2022

Editorial Scope

Topics

Foundations and Trends® in Signal Processing publishes survey and tutorial articles in the following topics:

- Adaptive signal processing
- Audio signal processing
- Biological and biomedical signal processing
- Complexity in signal processing
- Digital signal processing
- Distributed and network signal processing
- Image and video processing
- Linear and nonlinear filtering
- Multidimensional signal processing
- Multimodal signal processing
- Multirate signal processing
- Multiresolution signal processing
- Nonlinear signal processing
- Randomized algorithms in signal processing
- Sensor and multiple source signal processing, source separation
- Signal decompositions, subband and transform methods, sparse representations
- Signal processing for communications
- Signal processing for security and forensic analysis, biometric signal processing
- Signal quantization, sampling, analog-to-digital conversion, coding and compression
- Signal reconstruction, digital-to-analog conversion, enhancement, decoding and inverse problems
- Speech/audio/image/video compression
- Speech and spoken language processing
- Statistical/machine learning
- Statistical signal processing
  - Classification and detection
  - Estimation and regression
  - Tree-structured methods

Information for Librarians

Foundations and Trends® in Signal Processing, 2022, Volume 16, 4 issues. ISSN paper version 1932-8346. ISSN online version 1932-8354. Also available as a combined paper and online subscription.

Contents

Online Component Analysis, Architectures and Applications

João B. O. Souza Filho[1], Lan-Da Van[2], Tzyy-Ping Jung[3] and Paulo S. R. Diniz[4]

[1] *COPPE/Poli Federal University of Rio de Janeiro, Brazil; jbfilho@poli.ufrj.br*
[2] *National Yang Ming Chiao Tung University, Taiwan; ldvan@cs.nycu.edu.tw*
[3] *University of California, San Diego, USA; jung@sccn.ucsd.edu*
[4] *COPPE/Poli Federal University of Rio de Janeiro, Brazil; diniz@smt.ufrj.br*

ABSTRACT

This monograph deals with principal component analysis (PCA), kernel component analysis (KPCA), and independent component analysis (ICA), highlighting their applications to streaming-data implementations. The basic concepts related to PCA, KPCA, and ICA are widely available in the literature; however, very few texts deal with their practical implementation in computationally limited resources. The presentation tries to emphasize the current solutions considering possible constraints in power consumption and desirable computational complexity. For instance, there are good examples in biomedical engineering applications where tools like PCA and ICA can sort out the human body's activities. For example, it is possible to remove noise and undesirable artifacts from a target signal such as EEG and ECG, among others. In turn, KPCA may be a valuable

João B. O. Souza Filho, Lan-Da Van, Tzyy-Ping Jung and Paulo S. R. Diniz (2022), "Online Component Analysis, Architectures and Applications", Foundations and Trends® in Signal Processing: Vol. 16, No. 3-4, pp 224–429. DOI: 10.1561/2000000112.

resource for non-linear image denoising. Nonetheless, many current solutions rely on batch processing implemented in general-purpose computing resources.

In general terms, PCA consists of a sequence of uncorrelated data projections ordered according to their variances and employing mutually orthogonal directions. PCA is mighty in extracting hidden linear structures in high-dimension datasets. The standard PCA implementation computes the eigenvectors of the data-covariance matrix, retaining those directions to which the data exhibit the highest projection variances. This concept can be extended to the so-called Kernel PCA, wherein the data instances are implicitly mapped into a high-dimensional feature space via some non-linear transform, typically unknown. Conversely, ICA strengthens the PCA maximization variance approach by imposing the strict premise of mutual independence on the resulting projections. In fact, ICA comes to rescue the traditional tools when one aims at assessing non-Gaussian sources from data, often not available for direct measurement. Frequently, ICA and KPCA are more powerful tools for solving challenging tasks than PCA since they exploit high-order statistics from data.

All these methods require some simplifications to allow a simple online implementation when coping with streaming data. This monograph describes some state-of-the-art solutions for PCA, KPCA, and ICA, emphasizing their online deployments. Many online PCA and, more recently, KPCA techniques were proposed based on Hebbian learning rules and fixed-point iterative equations. Notably, online KPCA solutions also include data selection strategies to define a compact dictionary over which the kernel components are expanded. The complexity of these dictionaries is controlled by simply setting a single hyperparameter. In both cases, the online extensions proposed rely on simple equations, can track nonstationary environments, and require reduced

storage, enabling its use in real-time applications operating in low-cost embedded hardware.

This monograph discusses the state-of-the-art online PCA and KPCA techniques in a unified and principled manner, presenting solutions that achieve a higher convergence speed and accuracy in many applications, particularly image processing. Besides, this work also explains how to remove various artifacts from data records based on blind source separation (BSS) by ICA, splitting feature identification from feature separation. Herein, three FastICA online hardware architectures and implementation for biomedical signal processing are addressed. The main features are summarized as follows: 1) energy-efficient FastICA using the early determination scheme; 2) cost-effective variable-channel FastICA using the Gram-Schmidt-based whitening algorithm; and 3) moving-window-based online FastICA algorithm with limited memory. The post-layout simulation results with artificial and EEG data validate the design concepts.

In summary, this monograph presents the leading algorithmic solutions for PCA, KPCA, ICA, Iterative PCA, Online KPCA, and Online ICA, focusing on approaches amenable to process streaming signals. Furthermore, it provides some insights into how to choose the right solution for practical systems. Along the way, some implementation examples are provided in a variety of areas.

Full text available at: http://dx.doi.org/10.1561/2000000112

1

Introduction

This section provides a brief introduction to the concepts of Principal Component Analysis (PCA), Kernel Component Analysis and Independent Component Analysis (ICA), setting the stage for the presentation of their variants targeting streaming-data applications. It is fair to say that many textbooks and papers have covered these algorithms in the last three decades [25], [16], [66], [53], [48], [8], [47], [46], [60], [61], [91], [82]. Nevertheless, only very few works address their potential online deployments under a single cover; this work aims at filling this gap.

Component analysis algorithms fall into the realm of algorithms that learn from data, taking part in many textbooks addressing Statistical Inference [33], [41] and Machine Learning [104], [94], [37], [67], [11], [42], [27]. Since belonging to the category of unsupervised learning, such techniques entail extracting information from non-annotated data generated by some unknown distribution. Thus, the central idea is to extract most data information to allow their representation in a much simpler or compact form. An attentive reader may conclude that unsupervised learning directly relates to sparse coding, dimensionality reduction, and independent representations for datasets, topics broadly applied in several fields, from biomedical signal analysis, image processing, and communications to entertainment devices.

This work focuses on describing PCA, KPCA, and ICA related solutions under storage, computational resources, power, and complexity constraints. The continuing integrated circuit technological developments, especially in the microprocessors and field-programmable-gate-arrays (FPGAs) industry opened up room for integrating these algorithms in a wide range of embedded systems.

1.1 General Formulation

A common assumption behind most component analysis methods is that the observed data instances originate from variables not directly observed, named latent variables in the Statistics area. In many practical problems, latent variables are typically low, meaning that data have a reduced intrinsic dimensionality or lie on some low-dimensional subspace [101]. In such cases, one may assume that some latent vector $\mathbf{z} \in \mathbb{R}^M$ $(M \ll N)$ may represent an arbitrary vector data $\mathbf{x} \in \mathbb{R}^N$ as follows

$$\mathbf{z} = \mathbf{T}_{\mathbf{X}}\mathbf{x}, \tag{1.1}$$

where $\mathbf{T}_{\mathbf{X}} \in \mathbb{R}^{M \times N}$ is an orthonormal $(\mathbf{T}_{\mathbf{X}}^T\mathbf{T}_{\mathbf{X}} = \mathbf{I})$ data-dependent feature extraction transformation when considering PCA and ICA techniques. Similarly, KPCA exploits an extension of (1.1) given by

$$\mathbf{z} = \mathbf{T}_{\phi}\boldsymbol{\phi}(\mathbf{x}), \tag{1.2}$$

where $\boldsymbol{\phi}(\mathbf{x})$ represents a function responsible for intrinsically mapping the vector $\mathbf{x}$ into some high-dimensional feature space, and $\mathbf{T}_{\phi}$ is an orthonormal transform in this space [82], aspects further discussed in Section 6.3.

1.2 A Brief Introduction to PCA

By performing a sequence of data projections in a set of orthonormal directions defined such that their variances are maximized, PCA is a popular tool for accessing the low intrinsic dimensions of high-dimension datasets. The standard PCA implementation is based on the eigendecomposition of a defined data-covariance matrix and retains those directions with the highest data-projection variances. As such, PCA does not require any assumption over the data distribution and relies only on the second-order statistics of data. The related statistics can be easily measured or estimated, see [31]. PCA can also be extended to take benefit of high-order statistics from data through nonlinear data transforms into high dimensional space, giving rise to the so-called KPCA [83].

This means that for a set of data $\mathbf{x}_i \in \mathbb{R}^N$ for $i = 1, 2, \ldots, N$, or observations from a random process vector, PCA aims at extracting the main components representing this set. The idea is simple: project the data into l mutually orthonormal directions that maximize the variance of the resulting projections for $l < N$, *i.e.*, directions that better explain the data. This process results in an optimum projection matrix $\mathbf{T}_\mathbf{X}$ in (1.1), in the sense that the main latent variables are taken into consideration in the vector $\mathbf{z}$.

PCA is also strongly related to the Karhunen-Loève (KLT) transform. In the discrete-time domain, one can think of a linear transform that maximizes energy concentration for a given ensemble of signals generated by some statistical distribution, *i.e.*, KLT is optimum regarding energy compaction. The KLT of a discrete random process is defined by the vectors that diagonalize the autocovariance matrix. Therefore, for each distinct signal statistic, there is a corresponding KLT.

In summary, PCA features are defined by a sequence of mutually uncorrelated data projections ordered in decreasing order of their variance. In this sense, PCA is often a mighty tool for extracting hidden structures in high-dimension datasets. This concept can be extended to the so-called Kernel PCA, where the data samples are implicitly mapped to a higher dimension spaces via nonlinear transformations. The standard PCA implementation computes the eigenvectors of the data-correlation matrix, retaining those associated with the highest eigenvalues since they represent directions to which the data presents the highest projection variances.

To derive PCA, we start by assuming that we have available L random data vectors of dimension N assembled in a matrix as follows [37]:

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1(N-1)} & x_{1N} \\ x_{21} & x_{22} & \cdots & x_{2(N-1)} & x_{2N} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{L1} & x_{L2} & \cdots & x_{L(N-1)} & x_{LN} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_L^T \end{bmatrix}.$$

Here, the data vectors are assumed to have zero mean, *i.e.*, $\mathbb{E}[\mathbf{x}_i] = \mathbf{0}$, for $l = 1, 2, \ldots, L$, and the symbol $\mathbb{E}[\cdot]$ represents the expected value operation.

Singular-Value Decomposition (SVD)[1] can be applied in this case. Let us start by proceeding the SVD of $\mathbf{X}$ as follows

$$\mathbf{X} = \boldsymbol{\mathcal{U}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{V}}^T, \tag{1.3}$$

where, since $\mathbf{X} \in \mathbb{R}^{L \times N}$, assuming $L \geq N$, the rank of $\mathbf{X}$ will be lower or equal to N, and the dimensions of the matrices $\boldsymbol{\mathcal{U}}$ and $\boldsymbol{\mathcal{V}}$ will be $L \times L$ and $N \times N$, respectively. These matrices are orthogonal, that is

$$\boldsymbol{\mathcal{U}}^T\boldsymbol{\mathcal{U}} = \mathbf{I}_L, \tag{1.4}$$

$$\boldsymbol{\mathcal{V}}^T\boldsymbol{\mathcal{V}} = \mathbf{I}_N, \tag{1.5}$$

and the rectangular diagonal matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{L \times N}$ with non-negative singular values σ_i $(1 \leq i \leq L)$ is given by

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1 & 0 & 0 & & \cdots & 0 & \cdots & 0 \\ 0 & \sigma_2 & 0 & & \cdots & 0 & \cdots & 0 \\ 0 & 0 & \sigma_3 & & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & & \ddots & & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & \sigma_L & 0 & \cdots & 0 \end{bmatrix}, \tag{1.6}$$

under the premise of data being real and and $\sigma_1 \geq \sigma_1 \geq \cdots \sigma_L \geq 0$.

The deterministic and unbiased estimate of the covariance matrix related to the sample matrix $\mathbf{X}$ is defined as

$$\mathrm{cov}[\mathbf{X}] = \frac{1}{L-1}\mathbf{X}^T\mathbf{X}. \tag{1.7}$$

[1] It is also possible to consider the ordinary eigenvalue decomposition of $\mathbf{X}^T\mathbf{X}$ to solve PCA, simply considering the dominant eigenvectors of this matrix for defining the columns of the matrix $\mathbf{T}_{\mathbf{X}}$ in Equation (1.1).

Expressing this covariance with the aid of the SVD it follows that

$$\begin{aligned}\operatorname{cov}[\mathbf{X}] &= \frac{1}{L-1}\mathbf{X}^T\mathbf{X}\\ &= \frac{1}{L-1}\left[\boldsymbol{\mathcal{U}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{V}}^T\right]^T\boldsymbol{\mathcal{U}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{V}}^T\\ &= \frac{1}{L-1}\boldsymbol{\mathcal{V}}\boldsymbol{\Sigma}^T\boldsymbol{\mathcal{U}}^T\boldsymbol{\mathcal{U}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{V}}^T\\ &= \frac{1}{L-1}\boldsymbol{\mathcal{V}}\boldsymbol{\Sigma}^2\boldsymbol{\mathcal{V}}^T, \end{aligned} \tag{1.8}$$

given that (1.4) is true. If the observation matrix $\mathbf{X}$ is transformed through the orthogonal matrix $\boldsymbol{\mathcal{V}}$, representing a rotation applied to the signal space, as

$$\mathbf{Z} = \boldsymbol{\mathcal{V}}\mathbf{X}, \tag{1.9}$$

the covariance of the transformed matrix becomes

$$\begin{aligned}\operatorname{cov}[\mathbf{Z}] &= \frac{1}{L-1}\mathbf{Z}^T\mathbf{Z}\\ &= \frac{1}{L-1}\left[\boldsymbol{\mathcal{V}}\mathbf{X}\right]^T\left[\boldsymbol{\mathcal{V}}\mathbf{X}\right]\\ &= \frac{1}{L-1}\mathbf{X}^T\boldsymbol{\mathcal{V}}^T\boldsymbol{\mathcal{V}}\mathbf{X}\\ &= \frac{1}{L-1}\mathbf{X}^T\mathbf{X} = \operatorname{cov}[\mathbf{X}]. \end{aligned} \tag{1.10}$$

As observed, data expansion using all PCA components maintains the same covariance structure as the original data. However, it may constitute a powerful tool for reducing the data dimensionality if one selects only the principal components, *i.e.*, those related with the highest eigenvalues of the covariance matrix. From another point of view, PCA provides a compact representation of the data in $\mathbf{z}$, constituted of component representations that are *linearly uncorrelated.*

A simple illustration of how principal component analysis finds applications in numerous fields, let us consider a communication system where the received block $\mathbf{y} \in \mathbb{R}^{N\times 1}$ will suffer from entanglement of the transmitted data vector $\mathbf{s}$ according to the channel model described as

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{n}, \tag{1.11}$$

where $\mathbf{n} \in \mathbb{R}^{N\times 1}$ represents the environment noise. Here, we are assuming that all remaining vectors and matrices have proper dimensions.

If the vector to be transmitted is multiplied by $\boldsymbol{\mathcal{V}}$ before transmission (precoder) and by $\boldsymbol{\mathcal{U}}^T$ at the receiver, we obtain as the received signal $\mathbf{r} \in \mathbb{R}^{N\times 1}$:

$$\begin{aligned} \mathbf{r} &= \boldsymbol{\mathcal{U}}^T\mathbf{X} \\ &= \boldsymbol{\mathcal{U}}^T\mathbf{H}\boldsymbol{\mathcal{V}}\,\mathbf{s} + \boldsymbol{\mathcal{U}}^T\mathbf{n} \\ &= \boldsymbol{\Sigma}\,\mathbf{s} + \boldsymbol{\mathcal{U}}^T\mathbf{n}. \end{aligned} \tag{1.12}$$

In practice, we need some estimation of the channel matrix to design the transceiver. The second equality of (1.12) shows the decoupling effect of the SVD, allowing an independent detection of the transmitted data. After this decomposition, the overall system consists of independent subsystems so that the detection of each entry of the transmitted data vector can be performed individually. Depending on the actual communication system deployment, the signal-to-noise ratio perceived in each subchannel might allow maximizing the channel capacity through information-theoretic tools.

PCA solely relies on the covariance function to represent the dependency among the data entries. However, there are more complex relations that the learning algorithms might exploit. Moreover, the family of solutions for dealing with complex problems requires more powerful transformations than simple linear ones.

1.3 A Brief Introduction to KPCA

KPCA [83] represents an elegant extension of PCA. The central idea resides in assuming the existence of a typically unknown function $\boldsymbol{\phi}(\cdot)$ responsible for mapping any input data vector $\mathbf{x}$ into some high dimensional feature space $\mathcal{H}$. If $\mathcal{H}$ represents a reproducing Kernel Hilbert space (RKHS) [82], there is a *kernel function* κ such that $\kappa(\mathbf{x}_k, \mathbf{x}_l) =< \boldsymbol{\phi}(\mathbf{x}_k), \boldsymbol{\phi}(\mathbf{x}_l) >_{\mathcal{H}}= \boldsymbol{\phi}^T(\mathbf{x}_k)\boldsymbol{\phi}(\boldsymbol{x}_l)$. This means that using such a function, the dot data products in $\mathbf{H}$ can be computed by simple kernel function evaluations in the data space, a procedure popularly known as the *kernel trick* [82].

Since a range of machine learning algorithms can be expressed as a linear combination of dot products between input data and dataset elements, the kernel trick is useful for producing powerful nonlinear generalizations of such techniques. This can be achieved by replacing products as $\mathbf{x}^T\mathbf{x}_l$ by kernel function evaluations like $\kappa(\mathbf{x}, \mathbf{x}_l)$. This is the case of KPCA, allowing it to inherit many interesting geometrical properties of PCA.

KPCA extracts nonlinear features from data in $\mathbf{z}$ by using (please refer to Section 6.3 for a detailed coverage)

$$\mathbf{z} = \bar{\mathbf{A}}^T \begin{bmatrix} \boldsymbol{\kappa}(\mathbf{x}, \mathbf{x}_1); & \boldsymbol{\kappa}(\mathbf{x}, \mathbf{x}_2); & \cdots & \boldsymbol{\kappa}(\mathbf{x}, \mathbf{x}_L) \end{bmatrix}, \tag{1.13}$$

with

$$\bar{\mathbf{A}} = \begin{bmatrix} \bar{\boldsymbol{\alpha}}_1 & \bar{\boldsymbol{\alpha}}_2 & \cdots & \bar{\boldsymbol{\alpha}}_N \end{bmatrix}, \tag{1.14}$$

where the vectors $\bar{\boldsymbol{\alpha}}_i$, $1 \leq i \leq N$, are computed by

$$\mathbf{K}_N \boldsymbol{\alpha}_i = \lambda_i \boldsymbol{\alpha}_i, \tag{1.15}$$

$$\bar{\boldsymbol{\alpha}}_i = \frac{\boldsymbol{\alpha}_i}{\sqrt{\lambda_i}}. \tag{1.16}$$

Naturally, the solution of (1.15) is given by $\boldsymbol{\alpha}_i = \mathbf{e}_i$, where $\mathbf{e}_i$ is the eigenvector associated with the ith eigenvalue of $\mathbf{K}_N$, considering that such eigenvectors sorted according to $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N \geq 0$, while $\mathbf{K}_N$ corresponds to the Gram matrix [82] defined as

$$\mathbf{K}_L = \begin{bmatrix} \kappa(\mathbf{x}_1, \mathbf{x}_1) & \cdots & \kappa(\mathbf{x}_1, \mathbf{x}_L) \\ & \ddots & \\ \kappa(\mathbf{x}_L, \mathbf{x}_1) & \cdots & \kappa(\mathbf{x}_L, \mathbf{x}_L) \end{bmatrix}. \tag{1.17}$$

1.4 A Brief Introduction to ICA

ICA adopts a much stronger premise behind the data generation process than PCA. Both methods assume that the observed data consists of a linear combination of a set of basis vectors weighted by some latent variables. However, ICA assumes these weighting factors as independent, not simply as mutually decorrelated variables. The independence between two random variables is a more stringent assumption than zero correlation. Any two independent random variables are uncorrelated, *i.e.*, have zero correlation. Besides, for random variables having zero mean, zero correlation corresponds to zero covariance. Otherwise, such variables can be centered. However, decorrelation does not imply independence, except for Gaussian distributed variables.

Therefore, according to ICA, any dataset instance results from statistically independent sources or data representations. Thus, this technique aims to disentangle such representations to provide more explicit information about the data structures, which may eventually be even more compact. In synthesis, two basic ICA assumptions refer to the independence of the sources and that the mixture must include non-Gaussian distributed sources.

Let us assume that the observed data originate from the following linear and noiseless mixing model as

$$\mathbf{X} = \mathbf{A}\mathbf{S}, \tag{1.18}$$

where the observed information is assembled in $\mathbf{X} \in \mathbb{R}^{N \times L}$, the unknown mixing matrix is $\mathbf{A} \in \mathbb{R}^{N \times N}$, and the source matrix is $\mathbf{S} \in \mathbb{R}^{N \times L}$. The primary objective of the ICA algorithm is finding $\mathbf{S}$ with no knowledge about $\mathbf{A}$. Another point is that no more than one Gaussian-distributed source must integrate $\mathbf{S}$.

The most straightforward solution to unmix $\mathbf{X}$ would be estimating the mixing matrix $\mathbf{A}$ by some $\hat{\mathbf{A}}$ and then untangle the independent sources by premultiplying $\mathbf{X}$ by the inverse of $\hat{\mathbf{A}}$. Another way is to indirectly estimate the inverse of $\mathbf{A}$ using some matrix $\mathbf{W}$ such as

$$\begin{aligned}\hat{\mathbf{S}} &= \mathbf{W}^T\mathbf{X} \\ &= \mathbf{W}^T\mathbf{A}\mathbf{S}, \\ &\approx \mathbf{S},\end{aligned} \tag{1.19}$$

utilizing some preprocessing and some proper ICA algorithm to ensure that $\mathbf{W}^T\mathbf{A} \approx \mathbf{I}$. The matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$ is the weight matrix that provides the source separation over $\mathbf{X}$, also known as *unmixing matrix.*

A general model encompassing many ICA problems is depicted in Figure 1.1. In this figure, an estimate of $\mathbf{S}$, denoted as $\hat{\mathbf{S}}$, is produced over the noisy observed mixture $\mathbf{Y}$, modeled as $\mathbf{Y} = \mathbf{X}+\mathbf{N} = \mathbf{A}\mathbf{S}+\mathbf{N}$, where

N represents the additive noise. The weight matrix **W** is computed in a blind form, taking into consideration a nonlinearly modified version of $\hat{\mathbf{S}}$, named as $\tilde{\mathbf{S}}$, obtained by applying it to an entry-wise nonlinear transformation provided by **G**.

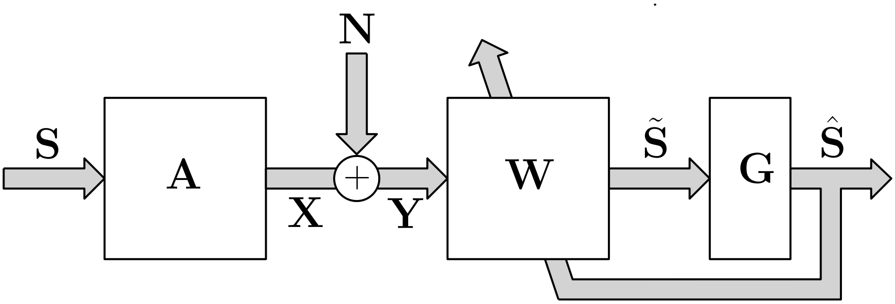

Figure 1.1: General model for ICA data processing.

Figure 1.2 illustrates the detailed structural description of the building blocks of the description in Figure 1.1 in the vector case.

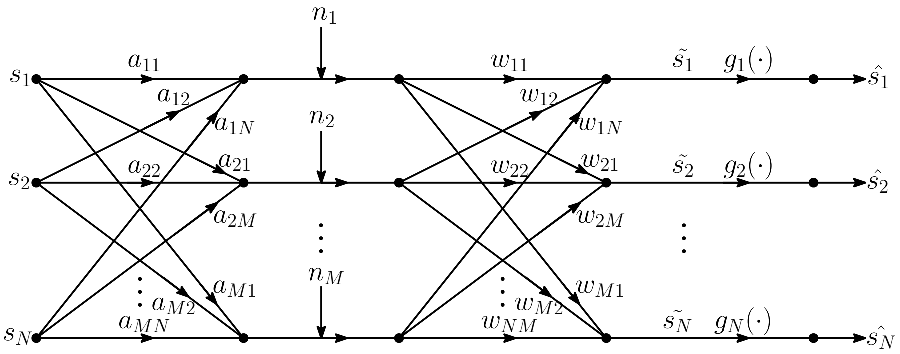

Figure 1.2: Inner structure of ICA building blocks.

ICA enhances the variance-maximization cost function, unlike PCA, by imposing a much stronger mutual independence assumption among the representations. In many practical cases, the data representations contributing to the observed data are not available for direct measurement, turning the strategy exploited in Equation (1.19) the only feasible. If the source representations are non-Gaussian, the ICA becomes a viable solution in such cases. Several cost functions enable independent component analysis, such as high-order statistics, particularly the kurtosis, defined as

$$\text{kurt}[z_i] = \mathbb{E}\left[z_i^4\right] - 3\left(\mathbb{E}\left[z_i^2\right]\right)^2, \tag{1.20}$$

for $1 \leq i \leq N$, wherein one may verify the use of fourth-order statistics. The motivation to employ the kurtosis is that its value is zero for variables with Gaussian distributions and non-zero for most non-Gaussian cases.

1.5 PCA × ICA

ICA does not distinguish independent Gaussian sources since, for any mixing matrix, the resulting mixture will be Gaussian distributed. Thus, using PCA, a simple decorrelation over such mixture signals will lead to independent variables. In fact, ICA relies on the assumption of independence among the latent representations, while the reduction in the number of representations is less critical, unlike PCA. Any nonlinear correlations linking variables must be removed to achieve independent representation, leading to inherently specialized learning algorithms. However, PCA is an essential preprocessing tool for ICA.

1.6 General Picture

As expected, PCA, KPCA, and ICA methods must undergo adaptations to process streaming data inherent to real-time implementations. This monograph introduces some state-of-the-art solutions for PCA, KPCA, and ICA, enabling their implementation on low-complex embedded hardware platforms. Some application examples are included to illustrate the exposed tools.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, S. I., K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, O. Viégas F. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: Large-scale machine learning on heterogeneous systems," in 2015, URL: https://www.tensorflow.org/.

[2] A. Acharyya, K. Maharatna, B. M. Al-Hashimi, and J. Reeve, "Coordinate rotation based low complexity n-d fastica algorithm and architecture," *IEEE Transactions on Signal Processing*, vol. 59, 2011, pp. 3997–4011. DOI: 0.1109/TSP.2011.2150219.

[3] S. Amari, "Neural theory of association and concept formation," *Biological Cybernetics*, 1977, pp. 175–185. DOI: 10.1007/BF00365229.

[4] R. Andraka, "A survey of cordic algorithms for fpga based computers," 1998, pp. 192–200. DOI: 10.1145/275107.275139.

[5] S. Bannour and M. R. Azimi-Sadjadi, "Principal component extraction using recursive least squares learning," *IEEE Transactions on Neural Networks*, vol. 6, 1995, pp. 457–469. DOI: 10.1109/72.363480.

[6] X. Bao, J. Lucas, S. Sachdeva, and R. Grosse, "Regularized linear autoencoders recover the principal components, eventually," in *Proceedings of the 34th Conference on Neural Information Processing Systems*, vol. 97, pp. 3560–3569, 2019.

[7] X. Bao, J. Lucas, S. Sachdeva, and R. B. Grosse, "Regularized linear autoencoders recover the principal components, eventually," in *Advances in Neural Information Processing Systems*, vol. 33, pp. 6971–6981, 2020.

[8] A. J. Bell and T. J. Sejnowski, "An information maximization approach to blind separation and blind deconvolution," *Neural Computation*, vol. 7, 1995, pp. 1129–1159. DOI: 10.1162/neco.1995.7.6.1129.

[9] D. P. Bertsekas, "Incremental least squares methods and the extended kalman filter," *SIAM J. Optim*, vol. 6, 1996, pp. 807–822. DOI: 10.1137/S1052623494268522.

[10] S. Bhardwaj, S. Raghuraman, and A. Acharyya, "Simplex fastica: An accelerated and low complex architecture design methodology for nd fastica," *IEEE Trans. Very Large Scale Integration (VLSI) Systems*, vol. 27, 2019, pp. 1124–1137. DOI: 10.1109/TVLSI.2018.2886357.

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

[12] H. Bourland and Y. Kamp, "Auto-association by multilayer perceptrons and singular value decomposition," *Biological Cybernetics*, vol. 59, 1988, pp. 291–294. DOI: 10.1007/BF00332918.

[13] T. Bouwmans, S. Javed, H. Zhang, Z. Lin, and R. Otazo, "On the applications of robust pca in image and video processing," *Proceedings of the IEEE*, vol. 106, no. 8, 2018, pp. 1427–1457. DOI: 10.1109/JPROC.2018.2853589.

[14] R. W. Brockett, “Dynamical systems that sorts lists, diagonalize matrices and solve linear programming problems,” *Linear algebra and its applications*, 1991, pp. 79–91. DOI: 10.1016/0024-3795(91)90021-N.

[15] E. J. Candés, X. Li, Y. Ma, and J. Wright, “Robust principal component analysis,” *Journal of the ACM*, vol. 581, no. 11, 2011. DOI: 10.1145/1970392.1970395.

[16] J.-F. Cardoso, “Blind signal separation: Statistical principles,” *Proceedings of the IEEE*, vol. 86, 1998, pp. 2009–2025. DOI: 10.1109/5.720250.

[17] F. Carrizosa-Corral, A. Vazquez-Cervantes, J. R. Montes, T. Hernandez-Diaz, J. C. Solano-Vargas, L. Barriga-Rodriguez, J. A. Soto-Cajiga, and H. Jimenez-Hernandez, “Fpga-soc implementation of an ica-based background subtraction method,” *International Journal of Circuit Theory and Applications*, vol. 46, 2018, pp. 1703–1722. DOI: 10.1002/cta.2544.

[18] S. Chakraverty, D. M. Sahoo, and N. R. Mahato, “Hebbian learning rule,” in *Concepts of Soft Computing: Fuzzy and ANN with Programming*, pp. 175–182, Singapore, 2019. DOI: 10.1007/978-981-13-7430-2_12.

[19] C. Chatterjee, Z. Kang, and V. P. Roychowdhury, “Algorithms for accelerated convergence of adaptive pca,” *IEEE Transactions on Neural Networks*, vol. 11, 2000, pp. 338–355. DOI: 10.1109/72.839005.

[20] T.-J. Chin and D. Suter, “Incremental kernel principal component analysis,” *IEEE Transactions on Image Processing*, vol. 16, 2007, pp. 1662–1674. DOI: 10.1109/TIP.2007.896668.

[21] R. R. Chowdhury, M. A. Adnan, and R. K. Gupta, “Real-time principal component analysis,” *ACM/IMS Trans. Data Sci.*, vol. 1, no. 2, 2020. DOI: 10.1145/3374750.

[22] A. Cichocki and S. Amari, *Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications*. New York, 2002. DOI: 10.1002/0470845899.

[23] A. Cichocki, W. Kasprzak, and W. Skarbek, “Adaptive learning algorithm for principal component analysis with partial data,” *Austrian Society for Cybernetic Studies*, vol. 2, 1996, pp. 1014–1019.

[24] A. Cichocki, S. Osowski, and K. Siwek, “Prewhitening algorithms of signals in the presence of white noise,” *Proceedings of VI International Workshop “Computational Problems of Electrical Engineering”*, 2004, pp. 205–208.

[25] P. Comon, “Independent component analysis: A new concept?” *Signal Processing*, vol. 36, 1994, pp. 11–20. DOI: 10.1016/0165-1684(94)90029-9.

[26] B. Dai, Y. Wang, J. Aston, G. Hua, and D. Wipf, “Connections with robust pca and the role of emergent sparsity in variational autoencoder models,” *Journal of Machine Learning Research*, vol. 19, no. 41, 2018, pp. 1–42.

[27] M. P. Deisenroth, A. A. Faisal, and C. S. Ong, *Mathematics for Machine Learning*. Cambridge, UK, 2020. DOI: 10.1017/9781108679930.

[28] A. Delorme and S. Makeig, “Eeglab: An open source toolbox for analysis of single-trial eeg dynamics including independent component analysis,” *Journal of Neuroscience Methods*, 2004, pp. 9–21. DOI: 10.1016/j.jneumeth.2003.10.009.

[29] P. S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*, 5th ed. Cham, 2020.

[30] P. S. R. Diniz, M. L. R. d. Campos, W. A. Martins, M. V. S. Lima, and J. A. Apolinário Jr., *Online Learning and Adaptive Filters*, 1st ed. Cambridge, 2022.

[31] P. S. R. Diniz, E. A. B. Silva, and S. L. Netto, *Digital Signal Processing: System Analysis and Design*, 2nd ed. Cambridge, 2010.

[32] “EEGLAB Wiki – SCCN,” Tech. Rep., URL: http://sccn.ucsd.edu/wiki/EEGLAB.

[33] B. Efron and T. Hastie, *Computer Age Statistical Inference*. Cambridge, UK, 2016. DOI: 10.1017/CBO9781316576533.

[34] Y. Engel, S. Mannor, and R. Meir, "The kernel recursive least-squares algorithm," *IEEE Transactions on Signal Processing*, vol. 52, 2004, pp. 2275–2285. DOI: 10.1109/TSP.2004.830985.

[35] R. Esfandiari, *Numerical Methods for Engineers and Scientists Using MATLAB®*, 2st. 2017. DOI: 10.1201/9781315152417.

[36] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 4th ed. Baltimore, 2013.

[37] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, 2016.

[38] S. Günter, N. N. Schraudolph, and S. V. N. Vishwanathan, "Fast iterative kernel principal component analysis," *Journal of Machine Learning Research*, vol. 8, 2007, pp. 1893–1918, URL: https://jmlr.org/papers/volume8/guenter07a/guenter07a.pdf.

[39] G. Hacohen and D. Weinshall, "Principal components bias in over-parameterized linear models, and its manifestation in deep neural networks," *Journal of Machine Learning Research*, vol. 23, no. 155, 2022, pp. 1–46.

[40] M. A. Hasan, "Low-rank approximations with applications to principal singular component learning systems," in *47th IEEE Conference on Decision and Control*, IEEE, pp. 3293–3298, Cancun, Mexico, 2008. DOI: 10.1109/CDC.2008.4739112.

[41] T. Hastie, R. Tibshirani, and J. Friedman, *Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY, 2016.

[42] S. Haykin, *Neural Networks and Learning Machines*. Englewood Cliffs, NJ, 2009.

[43] L. Hoegaerts, L. De Lathauwer, I. Goethal, J. A. K. Suykens, J. Vandewalle, and D. M. B., "Efficiently updating and tracking the dominant kernel principal components," *Neural Networks*, vol. 20, 2007, pp. 220–229. DOI: 10.1016/j.neunet.2006.09.012.

[44] S. Hoffmann, *Independent component analysis of ocular artifacts*. Germany, 2009.

[45] P. Honeine, "Online kernel principal component analysis: A reduced-order model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, 2012, pp. 1814–1826. DOI: 10.1109/TPAMI.2011.270.

[46] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. New York, 2001. DOI: 10.1002/0471221317.

[47] A. Hyvärinen and E. Oja, "A fast fixed-point algorithm for independent component analysis," *Neural Computation*, vol. 9, 1997, pp. 1483–1492. DOI: 10.1162/neco.1997.9.7.1483.

[48] A. Hyvärinen and E. Oja, "Independent component analysis: Algorithms and applications," *Neural Networks*, vol. 13, 2000, pp. 411–430. DOI: 10.1016/S0893-6080(00)00026-5.

[49] "ICALAB for Signal Processing," Tech. Rep., URL: http://www.bsp.brain.riken.jp/ICALAB/ICALABSignalProc/.

[50] "ICALAB for Signal Processing – benchmarks," Tech. Rep., URL: http://www.bsp.brain.riken.jp/ICALAB/ICALABSignalProc/benchmarks/.

[51] M. Jianchang and A. Jain, "Artificial neural networks for feature extraction and multivariate data projection," *IEEE Transactions on Neural Networks*, vol. 6, no. 2, 1995, pp. 296–317. DOI: 10.1109/72.363467.

[52] C. J. Jolliffe IT, "Principal component analysis: A review and recent developments," *Philos Trans A Math Phys Eng Sci*, vol. 374, 2016. DOI: 10.1098/rsta.2015.0202.

[53] T.-P. Jung, S. Makeig, C. Humphries, T. W. Lee, M. J. McKeown, V. Iragui, and T. J. Sejnowski, "Removing electroencephalographic artifacts by blind source separation," *Psychophysiology*, vol. 37, 2000, pp. 163–178. DOI: 10.1111/1469-8986.3720163.

[54] A. Kachenoura, L. Albera, L. Senhadji, and P. Comon, "Ica: A potential tool for bci systems," *IEEE Signal Processing Magazine*, vol. 25, 2008, pp. 57–68. DOI: 10.1109/MSP.2008.4408442.

[55] J. Karhunen and J. Joutsensalo, "Generalizations of principal components analysis, optimizations problems, and neural networks," *Neural Networks*, vol. 8, 1995, pp. 549–562. DOI: 10.1016/0893-6080(94)00098-7.

[56] K. I. Kim, M. O. Franz, and B. Scholkopf, "Iterative kernel principal component analysis for image modeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, 2005, pp. 1351–1366. DOI: 10.1109/TPAMI.2005.181.

[57] M. Kim, K. Ichige, and H. Arai, "Design of jacobi evd processor based on cordic for doa estimation with music algorithm," *PIMRC Conf.*, vol. 1, 2002, pp. 120–124. DOI: 10.1109/PIMRC.2002.1046673.

[58] S. Kimura, S. Ozawa, and S. Abe, "Incremental kernel PCA for online learning of feature space," in *International Conference on Computational Intelligence for Modeling, Control and Automation*, vol. 1, pp. 595–600, 2005. DOI: 10.1109/CIMCA.2005.1631328.

[59] M. K. Kochenderfer and T. L. Wheeler, *Algorithms for Optimization*, 1st ed. Cambrige, MA, 2019.

[60] E. Kofidis, "Blind source separation: Fundamentals and recent advances: A tutorial overview presented at sbrt-2001," Brazilian Society of Telecommunications, Tech. Rep., 2001, arXiv:1603.03089 (accessed in 2016).

[61] S. Y. Kung, *Kernel Methods and Machine Learning*. 2014. DOI: 10.1017/CBO9781139176224.

[62] S. Y. Kung, K. Y. Diamantaras, and J. S. Taur, "Adaptative principal component extraction (apex) and applications," *Signal Processing*, vol. 42, 1994, pp. 1202–1217. DOI: 10.1109/78.295198.

[63] W. Liao, A. Pizurica, W. Philips, and Y. Pi, "A fast iterative kernel PCA feature extraction for hyperspectral images," in *IEEE International Conference on Image Processing (ICIP)*, pp. 1317–1320, 2010. DOI: 10.1109/ICIP.2010.5651670.

[64] J. Lucas, G. Tucker, R. B. Grosse, and M. Norouzi, "Don't blame the elbo! a linear vae perspective on posterior collapse," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[65] S. Makeig, M. Westerfield, J. P. Jung, J. Covington, J. Townsend, T. J. Sejnowski, and E. Courchesne, "Functionally independent components of the late positive event-related potential during visual spatial attention," *Journal of Neuroscience*, 1999, pp. 2665–2680. DOI: 10.1523/JNEUROSCI.19-07-02665.1999.

[66] M. J. McKeown, S. Makeig, G. G. Brown, T.-P. Jung, S. S. Kindermann, A. J. Bell, and T. J. Sejnowski, "Analysis of fmri data by blind separation into independent spatial components," *Human Brain Mapping*, vol. 6, 1998, pp. 160–188. DOI: https://doi.org/10.1002/(SICI)1097-0193(1998)6:3<160::AID-HBM5>3.0.CO;2-1.

[67] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, 2012.

[68] E. Oja, "A simplified neuron model as a principal component analyser," *Journal of Mathematical Biology*, 1982, pp. 267–273. DOI: 10.1007/BF00275687.

[69] E. Oja, "Principal components, minor components and linear neural networks," *Neural Networks*, vol. 5, 1992, pp. 927–935. DOI: 10.1016/S0893-6080(05)80089-9.

[70] E. Oja and J. Karhunen, "On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix," *Journal of Mathematical Analysis and Applications*, 1985, pp. 69–54. DOI: 10.1016/0022-247X(85)90131-3.

[71] E. Oja, H. Ogawa, and J. Wangviwattana, "Principal components analysis by homogeneous neural networks. part i: The weighted subspace criterion," *IEICE Transactions on Information and Systems*, vol. 3, 1992, pp. 366–375.

[72] J. Onton, M. Westerfield, J. Townsend, and S. Makeig, "Imaging human eeg dynamics using independent component analysis," *Neuroscience and Biobehavioral Reviews*, vol. 30, 2006. DOI: 10.1016/j.neubiorev.2006.06.007.

[73] S. Ouyang and Z. Bao, "Fast principal component extraction by a weighted information criterion," *IEEE Transactions on Signal Processing*, vol. 50, 2002, pp. 1994–2002. DOI: 0.1109/TSP.2002.800395.

[74] S. Papaharalabos, P. Sweeney, B. G. Evans, G. Albertazzi, A. Vanelli-Coralli, and G. E. Corazza, "Performance evaluation of a modified sum-product decoding algorithm for ldpc codes," *Proc. Int. Symp. Wireless Commun. Syst. (ISWCS)*, 2005, pp. 800–804. DOI: 10.1109/ISWCS.2005.1547819.

[75] A. Papoulis and U. Pillai, *Probability, Random Variables and Stochastic Processes*, 4th ed. New York, 2004.

[76] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, 2019, pp. 8024–8035, URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[77] Y. N. Rao and J. C. Principe, "A fast on-line generalized eigendecomposition algorithm for time series segmentation," in *IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pp. 266–271, 2000. DOI: 10.1109/ASSPCC.2000.882483.

[78] T. Roh, K. Song, K. H. Cho, D. Shin, and H. J. Yoo, "A wearable neuro-feedback system with eeg-based mental status monitoring and transcranial electrical stimulation," *IEEE Trans. Biomedical Circuits and Systems*, vol. 8, 2014, pp. 755–764. DOI: 10.1109/TBCAS.2014.2384017.

[79] J. Rubner and P. Tavan, "A self-organizing network for principal component analysis," *Europhysics Letters*, vol. 10, 1989, pp. 693–698. DOI: 10.1209/0295-5075/10/7/015.

[80] J. Sanders and E. Kandrot, *CUDA by Example: An Introduction to General-Purpose GPU Programming*. Upper Saddle River, NJ, 2010, URL: https://www.bibsonomy.org/bibtex/27431903b5946bbf782f30227f3107349/ytyoun.

[81] T. D. Sanger, "Optimal unsupervised learning in a single linear feedfoward network," *Neural Networks*, vol. 2, 1989, pp. 459–473. DOI: 10.1016/0893-6080(89)90044-0.

[82] B. Schölkopf and A. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, 2002.

[83] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, 1998, pp. 1299–1319. DOI: 10.1162/089976698300017467.

[84] M. J. Schulte and K. E. Wires, "High-speed inverse square roots," *Proc. 14th Int'l Symp. Computer Arithmetic (ARITH14)*, 1999, pp. 124–131. DOI: 10.1109/ARITH.1999.762837.

[85] W. Shi and W. Zhang, "The accelerated power method for kernel principal component analysis," in *Applied Informatics and Communication*, vol. 225, 2011, pp. 563–570. DOI: 10.1007/978-3-642-23220-6_71.

[86] K. K. Shyu, M. H. Lee, Y. T. Wu, and P. L. Lee, "Implementation of pipelined fastica on fpga for real-time blind source separation," *IEEE Trans. Neural Networks*, vol. 19, 2008, pp. 958–970. DOI: 10.1109/TNN.2007.915115.

[87] J. B. Souza Filho and P. S. R. Diniz, "A fixed-point online kernel principal component extraction algorithm," *IEEE Transactions on Signal Processing*, vol. 65, 2017, pp. 6244–6259. DOI: 10.1109/TSP.2017.2750119.

[88] J. B. Souza Filho and P. S. R. Diniz, "A recursive least square algorithm for online kernel principal component extraction," *Neuralcomputing*, vol. 237, 2017, pp. 255–264. DOI: 10.1016/j.neucom.2016.12.031.

[89] J. B. Souza Filho and P. S. R. Diniz, "Improving kpca online extraction by orthonormalization in the feature space," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, 2018, pp. 2162–2388. DOI: doi.org/10.1109/TNNLS.2017.2660441.

[90] J. Souza Filho, L. Calôba, and J. Seixas, "An accurate and fast neural method for pca extraction," in *Proceedings of the International Joint Conference on Neural Networks, 2003.*, vol. 1, pp. 797–802, 2003. DOI: 10.1109/IJCNN.2003.1223484.

[91] J. A. Suykens, M. Signoretto, and A. Argyriou, *Regularization, Optimization, Kernels, and Support Vector Machines*. 2015.

[92] T. Tanaka, Y. Washizawa, and A. Kuh, "Adaptive kernel principal components tracking," in *IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP 2012*, pp. 1905–1908, 2012. DOI: 10.1109/ICASSP.2008.4517641.

[93] "The FastICA package for MATLAB," Aalto University, Tech. Rep., URL: https://research.ics.aalto.fi/ica/fastica/.

[94] S. Theodoridis, *Machine Learning: A Bayesian Optimization Perspective*. Oxford, UK, 2nd Edition, 2020.

[95] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society, Series B*, vol. 61, no. 3, 1999, pp. 611–622.

[96] *USPS handwritten digits*, URL: http://www.cs.nyu.edu/~roweis/data.html.

[97] L.-D. Van, P.-Y. Huang, and T.-C. Lu, "Cost-effective and variable-channel fastica hardware architecture and implementation for eeg signal processing," *Journal of Signal Processing Systems*, vol. 82, 2016, pp. 91–113. DOI: 10.1007/s11265-015-0988-2.

[98] L.-D. Van, T.-C. Lu, and T.-P. Jung, "Hardware-oriented memory-limited online fastica algorithm and hardware architecture for signal separation," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, pp. 1438–1442, Brighton, 2019. DOI: 10.1109/ICASSP.2019.8682997.

[99] L.-D. Van, D.-Y. Wu, and C.-S. Chen, "Energy-efficient fastica implementation for biomedical signal separation," *IEEE Transactions on Neural Networks*, vol. 22, 2011, pp. 1809–1822. DOI: 10.1109/TNN.2011.2166979.

[100] L.-D. Van, T. J. Wang, S. J. Tzeng, and T. P. Jung, "A computation-aware tpl utilization procedure for parallelizing the fastica algorithm on a multi-core cpu," *Proc. IEEE International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC)*, 2021, pp. 171–177.

[101] R. Vidal, Y. Ma, and S. S. Sastry, *Generalized Principal Component Analysis*, 1st. New York, 2016.

[102] J. E. Volder, "The cordic trigonometric computing technique," *IRE Transactions on Electronic Computers*, vol. 8, 1959, pp. 330–334. DOI: 10.1109/TEC.1959.5222693.

[103] Y. Washizawa, "Adaptive subset kernel principal component analysis for time-varying patterns," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, 2012, pp. 1961–1973. DOI: 10.1109/TNNLS.2012.2214234.

[104] J. Watt, R. Borhani, and A. K. Katsaggelos, *Machine Learning Refined: Foundations, Algorithms, and Applications*. Cambridge, UK, 2nd Edition, 2020.

[105] F. Wen, L. Chu, P. Liu, and R. C. Qiu, "A survey on nonconvex regularization-based sparse and low-rank recovery in signal processing, statistics, and machine learning," *IEEE Access*, vol. 6, 2018, pp. 69 883–69 906. DOI: 10.1109/ACCESS.2018.2880454.

[106] M. S. Woolfson, C. Bigan, J. A. Crowe, and B. R. Hayes-Gill, "Method to separate sparse components from signal mixtures," *Digital Signal Processing*, 2008, pp. 985–1012. DOI: 10.1016/j.dsp.2008.05.005.

[107] L. Xu, "Least mean square error reconstruction principle for self-organizing neural-nets," *Neural Networks*, vol. 6, 1993, pp. 627–648. DOI: 10.1016/S0893-6080(05)80107-8.

[108] B. Yang, "Projection approximation subspace tracking - past," *IEEE Transactions on Signal Processing*, vol. 43, 1995, pp. 95–107. DOI: 10.1109/78.365290.

[109] C. H. Yang, Y. H. Shih, and H. Chiueh, "An 81.6uw fastica processor for epileptic seizure detection," *IEEE Trans. Biomedical Circuits and Systems*, vol. 9, 2015, pp. 60–71. DOI: 10.1109/TBCAS.2014.2318592.

[110] J. Yang, Y. Zhao, and H. Xi, "Weigthed rule based adaptive algorithm for simultaneously extracting generalized eigenvectors," *IEEE Transactions on Neural Networks*, vol. 22, 2011, pp. 800–806. DOI: 10.1109/TNN.2011.2113354.

[111] A. Yazdanbakhsh, K. Seshadri, B. Akin, J. Laudon, and R. Narayanaswami, "An evaluation of edge TPU accelerators for convolutional neural networks," *CoRR*, vol. abs/2102.10423, 2021, URL: https://arxiv.org/abs/2102.10423.

[112] Q. Zhao, D. Meng, Z. Xu, and L. Zhang, “Robust principal component analysis with complex noise,” in *Proceedings of the 31th International Conference on Machine Learning*, vol. 32, pp. 55–63, 2014.