Independent Component Analysis Tutorial

September 17, 2025

1 Introduction

Independent Component Analysis (ICA) separates mixed signals into statistically independent sources using higher-order statistics. In contrast to PCA, which decorrelates by second-order moments, ICA seeks directions where components exhibit maximal non-Gaussianity, enabling blind source separation in audio, imaging, and biomedical data.

2 Theory and Formulas

2.1 Linear Mixing Model

Assume observed data $\mathbf{x} = \mathbf{A}\mathbf{s}$, where $\mathbf{s}$ contains independent sources with unit variance and $\mathbf{A}$ is an invertible mixing matrix. ICA estimates an unmixing matrix $\mathbf{W}$ so that $\hat{\mathbf{s}} = \mathbf{W}\mathbf{x}$ approximates the original sources.

2.2 Non-Gaussianity Maximization

Many ICA algorithms maximize a contrast function $J(\mathbf{w})$ measuring non-Gaussianity of projections. FastICA employs negentropy approximations with non-linearities such as $g(u) = \tanh(u)$ or $g(u) = u^3$:

$$\mathbf{w}_{\text{new}} = \mathbb{E}[\mathbf{x} g(\mathbf{w}^\top \mathbf{x})] - \mathbb{E}[g'(\mathbf{w}^\top \mathbf{x})]\mathbf{w}. \tag{1}$$

Vectors are decorrelated after each update to enforce orthogonality across components.

2.3 Whitening and Convergence

Whitening the data ($\mathbf{z} = \mathbf{V}^{-1/2}\mathbf{x}$) simplifies estimation by ensuring uncorrelated unit-variance mixtures. Convergence is tracked via change in $\mathbf{w}$ or the likelihood. Scale and permutation indeterminacies remain: components may be recovered up to sign and ordering.

3 Applications and Tips

- **Blind source separation**: unmix audio recordings, EEG signals, or hyperspectral pixels.

- **Feature extraction**: provide statistically independent factors for downstream models.
- **Anomaly detection**: analyze independent components for sparse activations indicative of events.
- **Best practices**: center and whiten data, choose component counts carefully, and validate with reconstruction or domain knowledge.

4 Python Practice

The script `gen_ica_figures.py` creates synthetic source signals, mixes them, applies FastICA, and visualizes both the separated signals and the mixing/unmixing matrices.

Listing 1: Excerpt from gen$_{i}ca_{f}igures.py$

```python
from sklearn.decomposition import FastICA

ica = FastICA(n_components=3, whiten='unit-variance', random_state=0)
sources_hat = ica.fit_transform(mixed_signals)

mixing = ica.mixing_
unmixing = ica.components_
```

5 Result

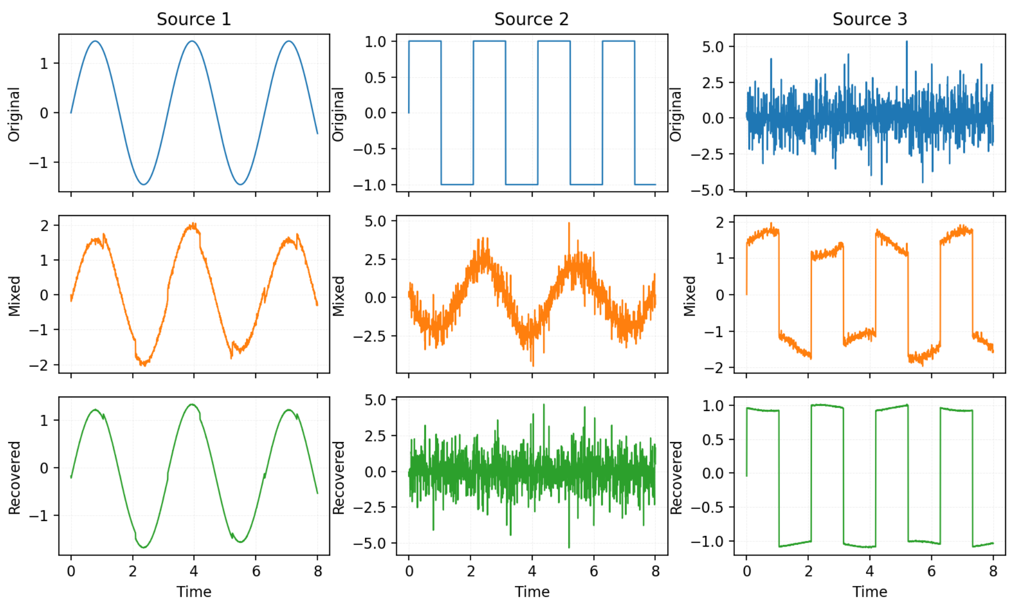

Figure 1: Comparison between original sources, mixed signals, and ICA-recovered components

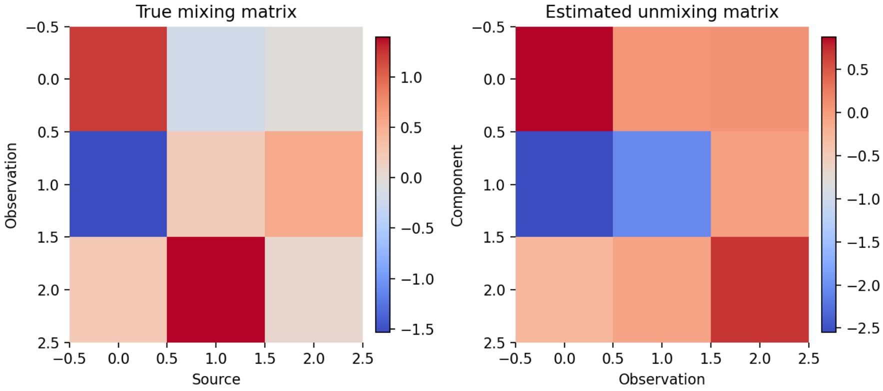

Figure 2: Heatmaps of true mixing matrix and estimated unmixing matrix

6 Summary

ICA goes beyond decorrelation by leveraging non-Gaussianity to recover latent sources. Whitening, iterative contrast optimization, and post-hoc validation are key to reliable separation. The example demonstrates how time-series visualizations and matrix comparisons help verify ICA performance.