COMS30035, Machine learning: Independent component analysis (ICA)

James Cussens
james.cussens@bristol.ac.uk

School of Computer Science
University of Bristol

October 23, 2023

The cocktail party problem

- Imagine you are at a cocktail party where many people are talking.
- What you hear is a mixture, with the mixing coefficients depending on how close you are to the speaker (and how loud they are talking).
- The *cocktail party problem* is to separate out the signal you are hearing into its constituent parts (i.e. the different speakers).
- This is an example of a *blind source separation* problem.
- Independent component analysis (ICA) aims to solve this problem.

The ICA model

- I will be using the notation used by Murphy [Mur23, §28.6] since Bishop is very brief on ICA.
- Let $\mathbf{x}_n \in \mathbb{R}^D$ be the signal received at "time" n.
- Let $\mathbf{z}_n \in \mathbb{R}^D$ be the vector of source signals at "time" n.
- We simply assume there is an (unknown) $D \times D$ *mixing matrix* $\mathbf{A}$:

$$\mathbf{x}_n = \mathbf{A}\mathbf{z}_n \tag{1}$$

- So the signal received is a linear mixture of signals.
- We ignore the temporal dependence between signals at different timepoints and treat the data as iid.
- (NB. Equation (28.143) in [Mur23, §28.6] is wrong, use (28.142) to understand the ICA model.)

ICA and PPCA

$$\mathbf{x}_n = \mathbf{A}\mathbf{z}_n \tag{2}$$

- This looks like probabilistic PCA.
- But in PPCA we assume that the 'sources' are independent Gaussian distributions.
- In ICA the sources z_{tj} are again assumed independent, but are required to be *non-Gaussian*.

$$p(\mathbf{z}_t) = \prod_{j=1}^{L} p_j(z_{tj}) \tag{3}$$

$p_j(z_{tj})$ not Gaussian.

What's wrong with Gaussian?

Recall from PPCA:

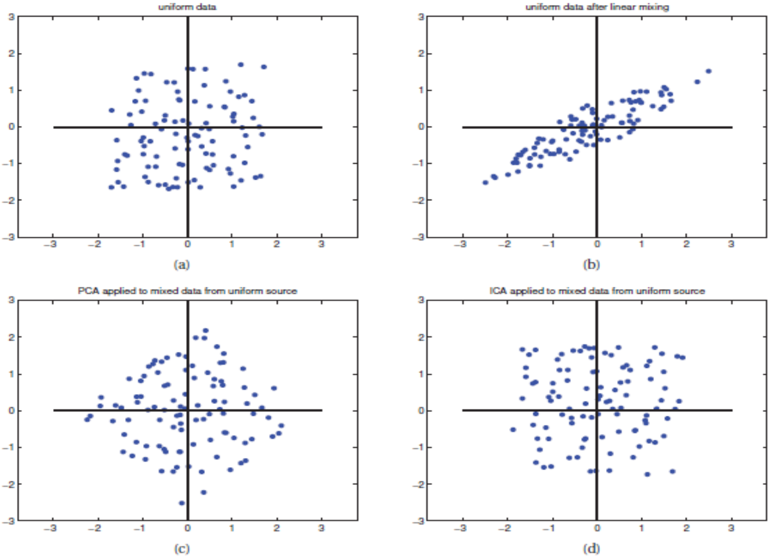
$$p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}|\mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \sigma^2\mathbf{I}) \tag{4}$$

The marginal distribution over **x** is [Bis06, pp.572–573] is:

$$\begin{aligned} p(\mathbf{x}) &= \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \mathbf{C}) && (5) \\ \mathbf{C} &= \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I} && (6) \end{aligned}$$

- Let **R** be an orthogonal matrix where $\mathbf{R}\mathbf{R}^T = \mathbf{I}$ so that $\mathbf{W} \to \mathbf{W}\mathbf{R}$ is a rotation of the latent variables.
- The likelihood for PPCA is unchanged if we replace **W** by **WR**, so PPCA can't distinguish between different rotations of the coordinate system in the latent space, and so can't get the 'right' one.

PCA vs ICA (Murphy Fig 28.33



(a) (b) (c) (d)

Solving ICA

Suppose we have no noise and the same number of signals and sensors so we have:

$$\mathbf{x}_t = \mathbf{W}\mathbf{z}_t$$

$$p(\mathbf{z}_t) = \prod_{j=1}^{L} p_j(z_{tj})$$

where W is a square matrix.

- Given observed values for $\mathbf{x}_t$ and a particular choice for the p_j we can use MLE to get a value for W.
- scikit-learn uses FastICA, an approximate Newton method.

Which non-Gaussian source distribution?

- There is some 'true' collection of source distributions p_j (e.g. for cocktail parties).
- But it is not critical for ICA to get the correct source distributions.
- The default for scikit-learn's FastICA is to use the same source distribution $p(z)$ for all j and set $G(z) = \log\cosh(z)$ where $G(z) = \log p(z)$.
- But you have other options or can supply your own.

- Christopher M. Bishop.
  Pattern Recognition and Machine Learning.
  Springer, 2006.
- Kevin P. Murphy.
  Probabilistic Machine Learning: Advanced Topics.
  MIT Press, 2023.