COMS30035, Machine learning: Independent component analysis (ICA)

James Cussens
james.cussens@bristol.ac.uk

Department of Computer Science, SCEEM
University of Bristol

October 29, 2020

Agenda

- ICA

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 [Mur12, §12.6] since Bishop is very brief on ICA.
- Let $\mathbf{x}_t \in \mathbb{R}^D$ be the signal received at time t.
- Let $\mathbf{z}_t \in \mathbb{R}^L$ be the vector of source signals at time t.
- We simply assume that:

$$\mathbf{x}_t = \mathbf{W}\mathbf{z}_t + \epsilon_t \tag{1}$$

- So signal received is a linear mixture of signals sent plus some noise. ϵ_t is a zero-mean Gaussian.
- We ignore the temporal dependence between signals at different timepoints and treat the data as iid.

ICA and PPCA

$$\mathbf{x}_t = \mathbf{W}\mathbf{z}_t + \epsilon_t \tag{2}$$

- If we assumed there was no noise (i.e. delete ϵ_t) then this looks like probabilistic PCA.
- But in PPCA we assume that the L '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.

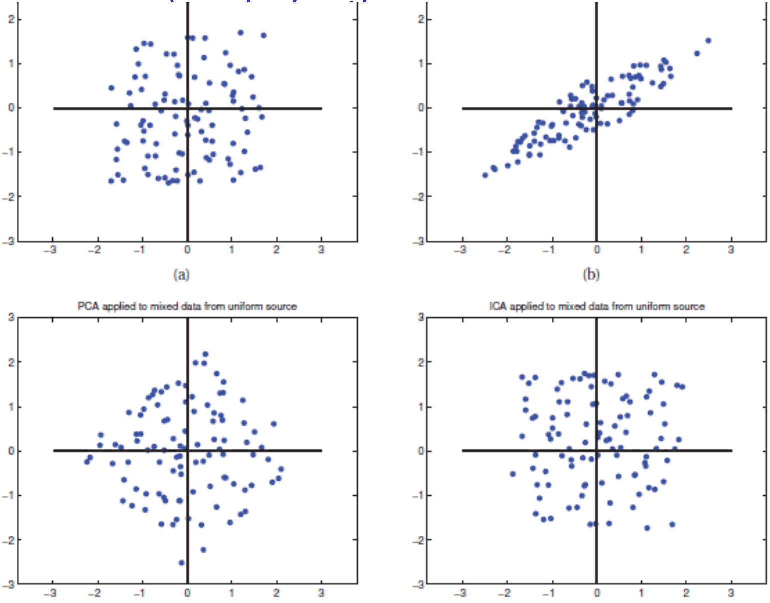
What's wrong with Gaussian?

Recall from PPCA:

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

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

PCA vs ICA (Murphy Fig 12.21



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.

Now do the quiz!

Yes, please do the quiz for this lecture on Blackboard!

Kevin Murphy.
Machine learning: A probabilistic perspective.
MIT Press, 2012.