Independent Components Analysis[1]

David Barber

University College London

[1] These slides accompany the book *Bayesian Reasoning and Machine Learning*. The book and demos can be downloaded from `www.cs.ucl.ac.uk/staff/D.Barber/brml`. Feedback and corrections are also available on the site. Feel free to adapt these slides for your own purposes, but please include a link the above website.

Independent Components Analysis

We seek a linear coordinate system in which the coordinates are independent. Such independent coordinate systems arguably form a natural representation of the data and can give rise to very different representations than PCA (which assumes the directions are orthogonal).

$$p(\mathbf{v},\mathbf{h}|\mathbf{A}) = p(\mathbf{v}|\mathbf{h},\mathbf{A})\prod_i p(h_i)$$

For technical reasons, the most convenient practical choice is to use

$$\mathbf{v} = \mathbf{A}\mathbf{h}$$

where $\mathbf{A}$ is a square mixing matrix so that the likelihood of an observation $\mathbf{v}$ is

$$p(\mathbf{v}) = \int p(\mathbf{v}|\mathbf{h},\mathbf{A})\prod_i p(h_i)d\mathbf{h} = \frac{1}{|\det\left(\mathbf{A}\right)|}\prod_i p(\left[\mathbf{A}^{-1}\mathbf{v}\right]_i)$$

The underlying independence assumptions are then the same as for PPCA (in the limit of zero output noise). Below, however, we will choose a non-Gaussian prior $p(h_i)$.

ICA versus PCA

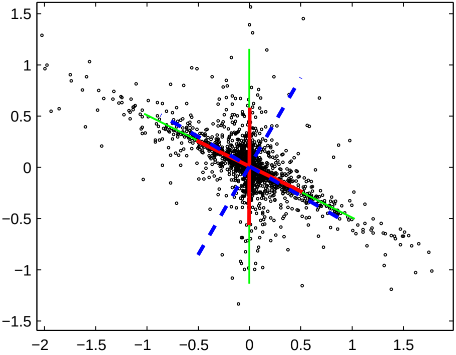

Figure: Latent data is sampled from the prior $p(x_i) \propto \exp(-5\sqrt{|x_i|})$ with the mixing matrix $\mathbf{A}$ shown in green to create the observed two dimensional vectors $\mathbf{y} = \mathbf{A}\mathbf{x}$. The red lines are the mixing matrix estimated by `ica.m` based on the observations. For comparison, PCA produces the blue (dashed) components. Note that the components have been scaled to improve visualisation. As expected, PCA finds the orthogonal directions of maximal variation. ICA however, correctly estimates the directions in which the components were independently generated.

Maximum Likelihood

For a given set of data $\mathcal{V} = \left(\mathbf{v}^1, \ldots, \mathbf{v}^N\right)$ and prior $p(h)$, our aim is to find $\mathbf{A}$. For i.i.d. data, the log likelihood is conveniently written in terms of $\mathbf{B} = \mathbf{A}^{-1}$,

$$L(\mathbf{B}) = N \log \det(\mathbf{B}) + \sum_n \sum_i \log p([\mathbf{B}\mathbf{v}^n]_i)$$

Rotational invariance for a Gaussian prior

Note that for a Gaussian prior

$$p(h) \propto e^{-h^2}$$

the log likelihood becomes

$$L(\mathbf{B}) = N \log \det(\mathbf{B}) - \sum_n (\mathbf{v}^n)^{\mathsf{T}} \mathbf{B}^{\mathsf{T}} \mathbf{B} \mathbf{v}^n + \text{const.}$$

which is invariant with respect to an orthogonal rotation $\mathbf{B} \to \mathbf{R}\mathbf{B}$, with $\mathbf{R}^{\mathsf{T}}\mathbf{R} = \mathbf{I}$. This means that for a Gaussian prior $p(h)$, we cannot estimate uniquely the mixing matrix. To break this rotational invariance we therefore need to use a non-Gaussian prior.

Maximum Likelihood

Assuming we have a non-Gaussian prior $p(h)$, taking the derivative *w.r.t.* B_{ab} we obtain

$$\frac{\partial}{\partial B_{ab}} L(\mathbf{B}) = N A_{ba} + \sum_n \phi([\mathbf{B}\mathbf{v}]_a) v_b^n$$

where

$$\phi(x) \equiv \frac{d}{dx} \log p(x) = \frac{1}{p(x)} \frac{d}{dx} p(x)$$

A simple gradient ascent learning rule for $\mathbf{B}$ is then

$$\mathbf{B}^{new} = \mathbf{B} + \eta \left(\mathbf{B}^{-\mathsf{T}} + \frac{1}{N} \sum_n \boldsymbol{\phi}(\mathbf{B}\mathbf{v}^n) \left(\mathbf{v}^n\right)^{\mathsf{T}} \right)$$

Natural Gradient

An alternative ‘natural gradient’ algorithm is given by multiplying the gradient by $\mathbf{B}^{\mathsf{T}}\mathbf{B}$ on the right to give the update

$$\mathbf{B}^{new} = \mathbf{B} + \eta \left(\mathbf{I} + \frac{1}{N} \sum_{n} \boldsymbol{\phi}(\mathbf{B}\mathbf{v}^{n}) \left(\mathbf{B}\mathbf{v}^{n}\right)^{\mathsf{T}} \right) \mathbf{B}$$

Here η is an empirically set learning rate.
The standard derivation of Natural Gradient for ICA is somewhat opaque.

Natural Gradient Derivation

Consider a modified objective

$$\tilde{L}(\mathbf{B}) = N \log \det (\mathbf{B})$$

which neglects the second term of the standard objective. Then applying Newton's method to this objective, one can show that the ICA Natural Gradient update is equivalent to

$$\eta \tilde{H}^{-1} \mathbf{g}$$

where $\tilde{H}$ is the Hessian of the above modified objective and $\mathbf{g}$ is the gradient of the standard objective.

ICA in practice

Choosing the non-linearity

In practice it is common to use the non-linearity

$$\phi(x) = x + c \tanh(x)$$

where $c = 1$ for a super-Gaussian (fatter tails than a Gaussian) component and $c = -1$ for a sub-Gaussian (thinner tails than a Gaussian) component. There are automatic ways to determine this.

---

fast ICA

A popular alternative estimation method is FastICA and can be related to an iterative Maximum Likelihood optimisation procedure, based on an approximate Newton procedure.