Chapter 5
Independent Component Analysis

Part II: Algorithms
ICA definition

• Given $n$ observations of $m$ random variables in matrix $X$, find $n$ observations of $m$ independent components in $S$ and $m$-by-$m$ invertible mixing matrix $A$ s.t. $X = SA$

• Components are statistically independent

• At most one is Gaussian

• We can assume $A$ is orthogonal (by whitening $X$)
Maximal non-Gaussian
Central limit theorem

- Average of i.i.d. variables converges to normal distribution
- \[ \sqrt{n} \left( \left( \frac{1}{n} \sum_{i=1}^{n} X_i \right) - \mu \right) \xrightarrow{d} N(0, \sigma^2) \] as \( n \to \infty \)
- Hence \( (X_1 + X_2)/2 \) is “more Gaussian” than \( X_1 \) or \( X_2 \) alone
  - For i.i.d. zero-centered non-Gaussian \( X_1 \) and \( X_2 \)
  - Hence, we can try to find components \( s \) that are “maximally non-Gaussian”
Re-writing ICA

• Recall, in ICA $\mathbf{x} = \mathbf{sA} \iff \mathbf{s} = \mathbf{xA}^{-1}$

  • Hence, $s_j$ is a linear combination of $x_i$

• Approximate $s_j \approx y = \mathbf{xb}^T$ ($\mathbf{b}$ to be determined)

  • Now $y = \mathbf{sAb}^T$ so $y$ is a lin. comb. of $\mathbf{s}$

• Let $\mathbf{q}^T = \mathbf{Ab}^T$ and write $y = \mathbf{xb}^T = \mathbf{sq}^T$
More re-writings

• Now \( s_j \approx y = xb^T = sq^T \)

• If \( b^T \) is a column of \( A^{-1} \), \( s_j = y \) and \( q_j = 1 \) and \( q \) is 0 elsewhere

• CLT: \( sq^T \) is least Gaussian when \( q \) looks correct
  • We don’t know \( s \), so we can’t vary \( q \)
  • But we can vary \( b \) and study \( xb^T \)

• **Approach:** find \( b \) s.t. \( xb^T \) is least Gaussian
Kurtosis

• One way to measure how Gaussian a random variable is is its **kurtosis**
  
  • \( \text{kurt}(y) = E[(y - \mu)^4] - 3(E[(y - \mu)^2])^2 \)
  
  • \( E[y] = \mu \)
  
  • Normalized version of the fourth central moment \( E[(y - \mu)^4] \)
  
  • If \( y \sim N(\mu, \sigma^2) \), \( \text{kurt}(y) = 0 \), most other distributions have non-zero kurtosis (positive or negative)
Computing with kurtosis

• If $x$ and $y$ are independent random variables:
  • $\text{kurt}(x + y) = \text{kurt}(x) + \text{kurt}(y)$
  • Homework

• If $\alpha$ is a constant:
  • $\text{kurt}(\alpha x) = \alpha^4 \text{kurt}(x)$
  • $E[(\alpha x)^4] - 3(E[(\alpha x)^2])^2 = \alpha^4 E[x^4] - \alpha^4 3(E[x^2])^2$
Sub- and super-Gaussian distributions

- Distributions with negative kurtosis are **sub-Gaussian** (or **platykurtic**)
  - Flatter than Gaussian
- Distributions with positive kurtosis are **super-Gaussian** (or **leptokurtic**)
  - Spikier than Gaussian
Examples

![Standard symmetric PDFs](https://en.wikipedia.org/wiki/Kurtosis#/media/File:Standard_symmetric_pdf.png)
Negentropy

• Another measure of non-Gaussianity
• Entropy of discrete r.v. $X$ is $H(X) = -\sum_i \Pr[X=i] \log \Pr[X=i]$
• The differential entropy of continuous random vector $\mathbf{x}$ with density $f(\mathbf{x})$ is $H(\mathbf{x}) = -\int f(\mathbf{x}) \log f(\mathbf{x}) \, d\mathbf{x}$
  • Gaussian $\mathbf{x}$ has the largest entropy over all random variables of equal variance
• Negentropy is $J(\mathbf{x}) = H(\mathbf{x}_{\text{Gauss}}) - H(\mathbf{x})$
  • $\mathbf{x}_{\text{Gauss}}$ is a Gaussian r.v. of the same covariance matrix as $\mathbf{x}$
Approximating negentropy

• Computing the negentropy requires estimating the (unknown) pdfs

• It can be approximated as

\[ J(y) \approx \sum_i k_i (E[G_i(y)] - E[G_i(v)])^2 \]

• \( v \sim N(0, 1) \), \( k_i \) are positive constants and \( G_i \) are some non-quadratic functions

• With only one function \( G(y) = y^4 \), this is kurtosis

• One choice: \( G_1(y) = \log(\cosh(ay))/a \), \( G_2(y) = -\exp(-y^2/2) \)
Back to optimization (using kurtosis)

• Recall: with two components
\[ y = xb^T = sq^T = q_1s_1 + q_2s_2 \]

• \( s_i \) have unit variance

• We want to find \( \pm b = \text{argmax } |\text{kurt}(xb^T)| \)

• We can’t determine the sign

• We want \( y \) to be either \( s_1 \) or \( s_2 \), hence
\[ E[y^2] = q_1^2 + q_2^2 = 1 \]
Whitening, again

• Generally, $||q||^2 = 1$

• Recall: $Z = U = XV\Sigma^{-1}$ is the whitened version of $X$

• Target becomes $\pm w = \arg\max |\text{kurt}(zw^T)|$

• Now $||q||^2_2 = (wU^T)(Uw^T) = ||w||^2_2$

• Hence we have constraint $||w||^2 = 1$
Gradient-based algorithm

• Gradient with kurtosis is

\[ \frac{\partial \text{kurt}(zw^T)}{\partial w} = 4 \text{sign}(\text{kurt}(zw^T))(E[(zw^T)^3 z] - 3w \|w\|_2^2) \]

• \( E[(zw^T)^2] = \|w\|^2 \) for whitened data

• We can optimize this using standard gradient methods

• To satisfy the constraint \( \|w\|^2 = 1 \), we divide \( w \) with its norm after every update
FastICA for one IC and kurtosis

- Noticing that $||\mathbf{w}||^2 = 1$ by constraint and taking infinite step update, we get
  
  $\mathbf{w} \leftarrow E[(\mathbf{z}\mathbf{w}^T)^3 \mathbf{z}] - 3\mathbf{w}$

- Again set $\mathbf{w} \leftarrow \mathbf{w}/||\mathbf{w}||$ after every update

- Expectation has naturally to be estimated

- No theoretical guarantees but works in practice
FastICA with approximations of negentropy

• Let $g$ be the derivative of a function used to approximate the negentropy

  • $g_1(x) = G_1'(x) = \tanh(ax)$

• The general fixed-point update rule is

  $$w \leftarrow E[g(zw^T)z] - E[g'(zw^T)]w$$
Multiple components

• So far we have found only one component

  • To find more, remember that vectors $w_i$ are orthogonal (columns of invertible $A$)

• General idea:

  • Find one vector $w$

  • Find second that is orthogonal to the first one

  • Find third that is orthogonal to the two previous ones, etc.
Symmetric orthogonalization

• We can compute $w_i$'s in parallel
  • Update $w_i$'s independently
  • Run orthogonalization after every update step
    • $W \leftarrow (WW^T)^{-1/2}W$
  • Iterate until convergence
Maximum Likelihood
Maximum-likelihood algorithms

• **Idea:** We are given observations $\mathbf{X}$ that are drawn from some parameterized family of distributions $D(\Theta)$

• The **likelihood** of $\mathbf{X}$ given $\Theta$, $L(\Theta; \mathbf{X}) = p_D(\mathbf{X}; \Theta)$, where $p_D(\cdot; \Theta)$ is the probability density function of $D$ with parameters $\Theta$

• In **maximum-likelihood estimation** (MLE) we try to find $\Theta$ that maximizes the likelihood given $\mathbf{X}$
ICA as MLE

• If $p_x(x)$ is the pdf of $x = sA$, then

$$p_x(x) = p_s(s) |\text{det } B| = |\text{det } B| \prod_i p_i(s_i) = |\text{det } B| \prod_i p_i(xb_i^T)$$

• here $B = A^{-1}$

• In general, if $x$ is r.v. with pdf $p_x(x)$ and $y = Bx$, then $p_y(y) = p_x(Bx)|\text{det } B|$

• For $T$ observations $x_1, x_2, \ldots, x_T$ the log-likelihood of $B$ given $X$ is

$$\log L(B; X) = \sum_{t=1}^T \sum_{i=1}^m \log p_i(x_t b_i^T) + T \log |\text{det } B|$$
Problems with MLE

• The likelihood is expressed as a function of $B$

• But we also need to estimate the pdfs $p_i()$
  
  • Non-parametric problem, infinite number of different pdfs

• Very hard problem…
If we know the pdfs

- Sometimes we know the pdfs of the components
  - We only need to estimate their parameters and $B$
- Sometimes we know only that the pdfs are super-Gaussian (for example)
  - We can use $\log p_i(s_i) = -\log \cosh(s_i)$
  - Requires normalization
\(-\log \cosh(x) \approx -\lvert x \rvert\)
Nothing on the pdfs is known

• We might not know whether the pdfs of the components are sub- or super-Gaussian
  • It is enough to estimate which one they are!
• For super-Gaussian,
  \[ \log p_i^+(s_i) = \alpha_1 - 2\log \cosh(s_i) \]
• For sub-Gaussian,
  \[ \log p_i^-(s_i) = \alpha_2 - (s_i^2/2 - \log \cosh(s_i)) \]

\(\alpha_i\) are only needed to make these logs of pdfs – not in optimization
Log-likelihood gradient

- The gradient is \( \frac{\partial \log L}{\partial B} = (B^T)^{-1} + \sum_{t=1}^{T} g(x_t B^T) x_t \)
  - Here \( g(y) = (g_i(y_i))_{i=1}^{n} \) with \( g_i(y_i) = \log p_i(y_i) \)’ = \( p_i'(y_i)/p_i(y_i) \)
  - This gives us \( B \leftarrow B + \delta((B^T)^{-1} + \sum_t g(x_t B^T) x_t) \)
  - Multiplying from right with \( B^T B \) and defining \( y_t = x_t B^T \) gives \( B \leftarrow B + \delta(I + \sum_t g(y_t) y_t) B \)
  - So-called infomax algorithm

Step size
Setting $g()$

- We compute $E[-\tanh(s_i)s_i + (1 - \tanh(s_i)^2)]$
  - If positive, set $g(y) = -2\tanh(y)$
  - If negative (or zero), set $g(y) = \tanh(y) - y$
- Use current estimates of $s_i$
Putting it all together

• Start with random $B$ and $\gamma$, choose learning rates $\delta$ and $\delta_\gamma$

• Iterate until convergence
  
  • $y \leftarrow Bx$ and normalize $y$ to unit variance
  
  • $\gamma_i \leftarrow (1 - \delta_\gamma)\gamma_{i-1} + \delta_\gamma E[-\tanh(y_i)y_i + (1 - \tanh(y_i)^2)]$
    
    • if $\gamma_i > 0$, use super-Gaussian $g$; o/w sub-Gaussian $g$
  
  • $B \leftarrow B + \delta(I + \sum_t g(y_t)^T y_t)B$
ICA summary

• ICA can recover independent source signals
  • if they are non-Gaussian
• Does not reduce rank
• Many applications, special case of blind source separation
  • Standard algorithmic technique is to maximize non-Gaussianity of the recovered components
ICA literature
