# High Information Projections and Independent Components

David Hofmeyr

Dept. Statistics and Actuarial Science,
Stellenbosch University, South Africa

28 November 2022

**Multivariate Data Analysis Group**

**SASA**

# Outline

1. Entropy and Information
2. Independent Components Analysis
3. Direct Entropy Minimisation
4. Playing around in `R` (go to https://github.com/DavidHofmeyr/SASA2022)
5. Online Collision Entropy Minimisation

# Entropy and Information

- Entropy $\equiv$ "randomness" $\Rightarrow$ negative Entropy $\equiv$ Information
- How can we intuit the "information" in a random variable?
  - An observation gives "information" about where other independent copies are likely to occur
  - "Observations tend to occur in high density regions"
  - This is almost just the definition of the density function, BUT some satisfy this property to greater degrees than others, in that

$$E[p(f_X(X))]$$

is relatively large, where $p(\cdot)$ is an increasing function

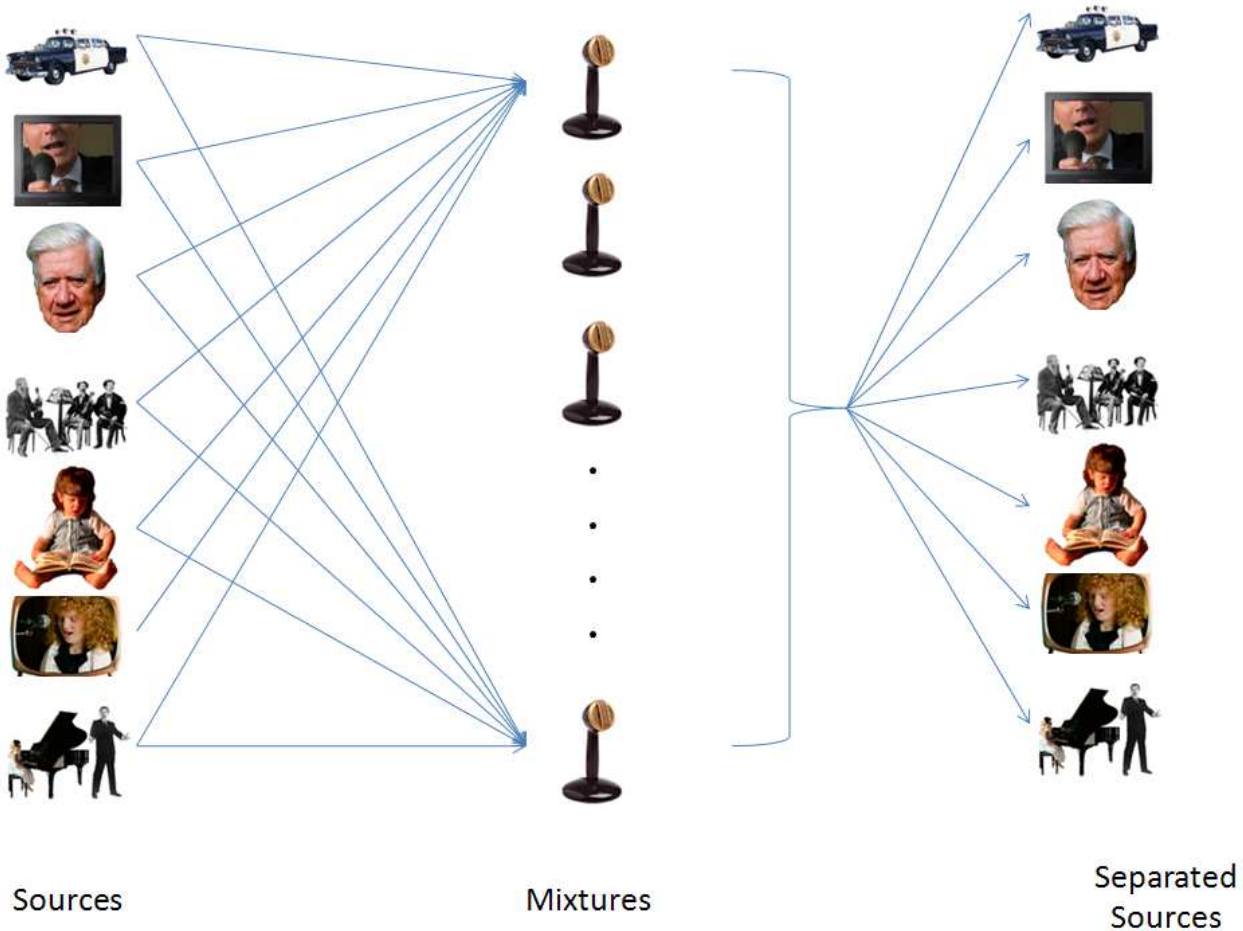
# Rényi Entropy

- Parameterised by $\alpha > 1$,

$$\begin{aligned} H_\alpha(X) &= \frac{1}{1-\alpha} \log \left( \int f_X(x)^\alpha dx \right) \\ &= \frac{1}{1-\alpha} \log \left( E[f_X(X)^{\alpha-1}] \right) \\ H_1(X) := \lim_{\alpha \to 1^+} H_\alpha(X) &= -E[\log(f_X(X))] \end{aligned}$$

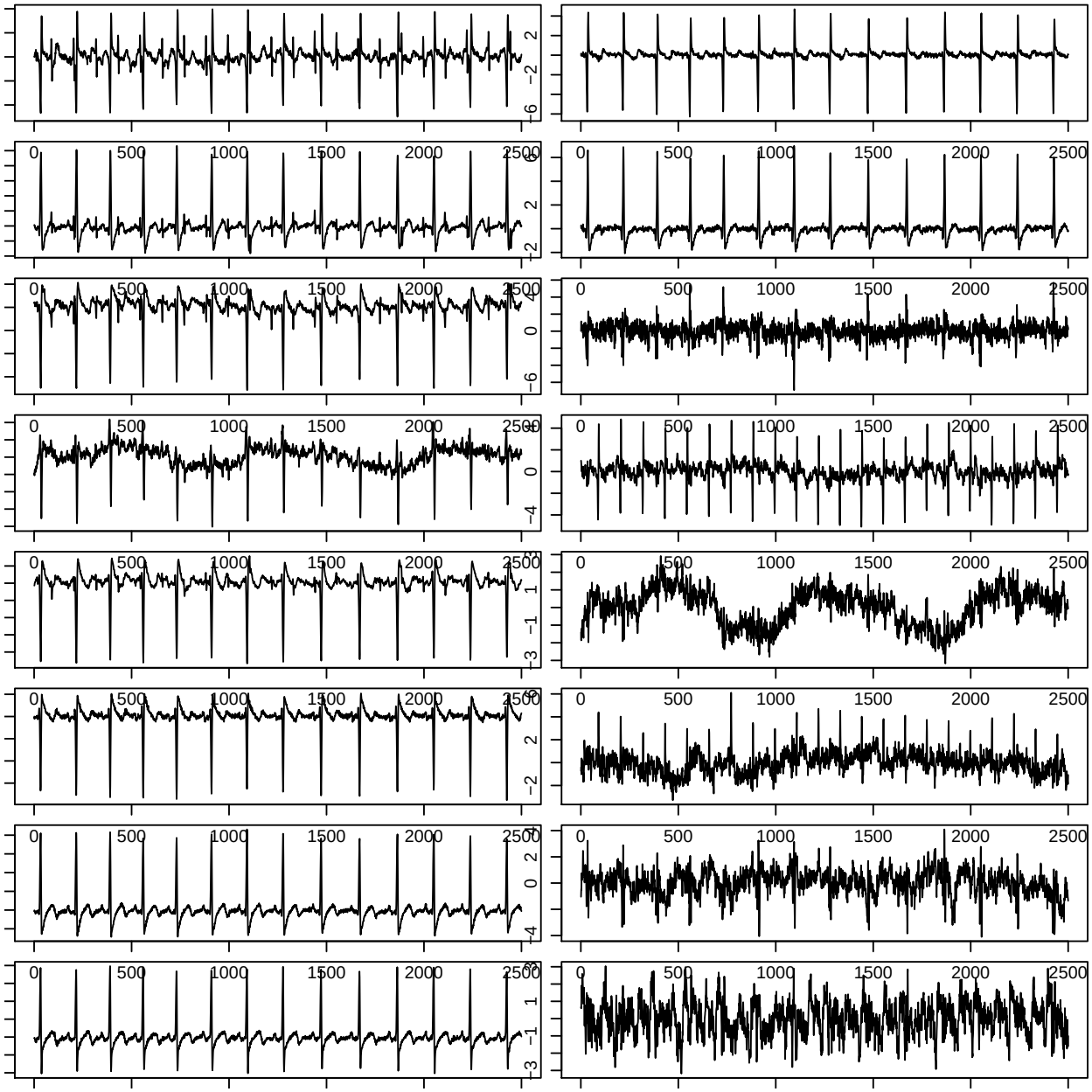
- For $\alpha > 1$ (and in the limit) this satisfies the intuitive definition of -entropy $\equiv$ information

# Independent Component Analysis: The Cocktail Party Problem




[1]image taken from Shubhendu Trivedi's blog post; https://onionesquereality.wordpress.com/2010/01/30/

# My Favourite ICA Success: Foetal ECG



# Independent Component Analysis: The Model

- Identify *source signals* which are observed only indirectly after "mixing"
- Assume $\mathsf{S} \in \mathbb{R}^{n \times k}$ represents realisations of $k$ INDEPENDENT source signals
- Assume observations are given by $X = \mathsf{SM}$
- Task is to estimate *unmixing* matrix, U, s.t., $X\mathsf{U} \approx \mathsf{S}$
- How could we possibly start to address this?
  - We assume nothing about the sources (except independence)
- We could use projection pursuit to "maximise independence"
  - what do we mean by this?
  - measuring independence is not usually straight-forward (and may be computationally demanding)

# Independent Component Analysis: The Method

- Some useful observations massively simplify the problem formulation
  - Orthogonality (zero covariance) is necessary for independence
  - scale doesn't affect independence:
    $X \perp Y \iff aX \perp Y \ \forall a \neq 0$
- We can measure the independence in the components of XU via their mutual information

$$
\begin{aligned}
MI(\mathrm{XU}) &= KL\left(f_{\mathrm{XU}} \middle|\middle| \prod_i f_{\mathrm{Xu}_i}\right) \\
&= E_{\mathrm{XU}}\left[\log(f_{\mathrm{XU}}(\mathrm{X}))\right] - E_{\mathrm{XU}}\left[\log\left(\prod_i f_{\mathrm{Xu}_i}(X_i)\right)\right] \\
&= E_{\mathrm{X}}\left[\log(f_{\mathrm{X}}(\mathrm{X}))\right] + \log(|\det(\mathrm{U})|) \\
&\quad - \sum_{i=1}^{d} E_{\mathrm{Xu}_i}\left[\log\left(f_{\mathrm{Xu}_i}(X)\right)\right]
\end{aligned}
$$

# Independent Component Analysis

- But $E_{\mathsf{X}}[\log(f_{\mathsf{X}}(X))]$ is constant, and we don't care about scale (we can't discriminate based on scale), so can force det(U) to be constant
- We therefore want to minimise

$$-\sum_{i=1}^{d} E_{\mathsf{X}\mathsf{u}_i}[\log(f_{\mathsf{X}\mathsf{u}_i}(X))]$$

- We don't know $f_{\mathsf{X}\mathsf{u}_i}$:
  - we assume we have a sample from $F_{\mathsf{X}}$ for estimation
- minimise the sample estimate

$$-\sum_{i=1}^{d} \frac{1}{n} \sum_{j=1}^{n} \log(\hat{f}_{\mathsf{X}\mathsf{u}_i}(\mathsf{x}_j^\top \mathsf{u}_i))$$

- Notice that this is also a maximum (pseudo) likelihood solution, under the assumption of independence

# Independent Component Analysis: The Computation

- Computational problem: Estimating $f_{\mathsf{X}\mathsf{u}}$, and evaluating it at all $\mathsf{x}_j^\top \mathsf{u}$ is expensive if approached naïvely
  - Approximations from "negentropy" (measure of departure from Gaussianity)
  - ... or... or
  - or suck it up and get better at computation
- Non-Gaussianity $\approx\Rightarrow$ independence (in this context)
  - For fixed scale, Gaussian random variables have maximum (Shannon) entropy
  - If observations are linear combinations of independent random variables, these "tend to look more Gaussian" (think CLT, but without the L)
  - The actual sources won't have this convolution effect
  - (ICA can't distinguish Gaussian sources)

# FastICA

- (Seemingly) the most popular approach is to maximise

$$L(E[c(Xu)] - E[c(Z)]),\ Z \sim N(0,1)$$

  for symmetric "loss" function $L$ and "contrast" function $c$
  - Usually $L(x) = x^2$
  - Popular optons for $c$ are
    - $c(x) = x^4$
    - $c(x) = \log(\cosh(x))$
    - $c(x) = exp(-x^2/2)$
- Very fast and theoretically consistent, but inaccurate in many practical examples

# Non-parametric Entropy Minimisation

- I like kernels, they're more intuitive (to me) and amenable to differentiation
- The kernel estimate of a density, $f_X$, using a sample from its distribution, $\{x_1, ..., x_n\}$, is given by

$$\hat{f}_X(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \Rightarrow$$

$$\nabla_{\mathrm{u}_k} \left( -\sum_{i=1}^{d} \frac{1}{n} \sum_{j=1}^{n} \log(\hat{f}_{X\mathrm{u}_i}(\mathrm{x}_j^\top \mathrm{u}_i)) \right) =$$

$$\frac{1}{n^2 h^2} \sum_{i=1}^{n} x_i \sum_{j=1}^{n} K'\left(\frac{x_j^\top \mathrm{u}_k - x_i^\top \mathrm{u}_k}{h}\right) \left( \frac{1}{\hat{f}_{X\mathrm{u}_k}(\mathrm{x}_j^\top \mathrm{u}_k)} + \frac{1}{\hat{f}_{X\mathrm{u}_k}(\mathrm{x}_i^\top \mathrm{u}_k)} \right)$$

# Fast Kernel Smoothing

- Evaluating this gradient "should" cost $\mathcal{O}(n^2)$
- Can be made $\mathcal{O}(n \log(n))$ if the kernel looks like this:

$$K(x) = \sum_{i=0}^{m} \beta_i |x|^i \exp(-|x|)$$

- Fast computation boils down to
  1. trivial factorisation of $\exp(a + b)$
  2. binomial expansion of $(a + b)^i$

  and gets us ...

# Fast Kernel Smoothing

$$\sum_{i=1}^{n} |x - x_i|^p \exp\left(-|x - x_i|\right) \omega_i$$

$$= \sum_{k=0}^{p} \binom{p}{k} \left( x^{p-k} \ell(k, n(x)) \exp\left( \frac{x_{(n(x))} - x}{h} \right) \right.$$

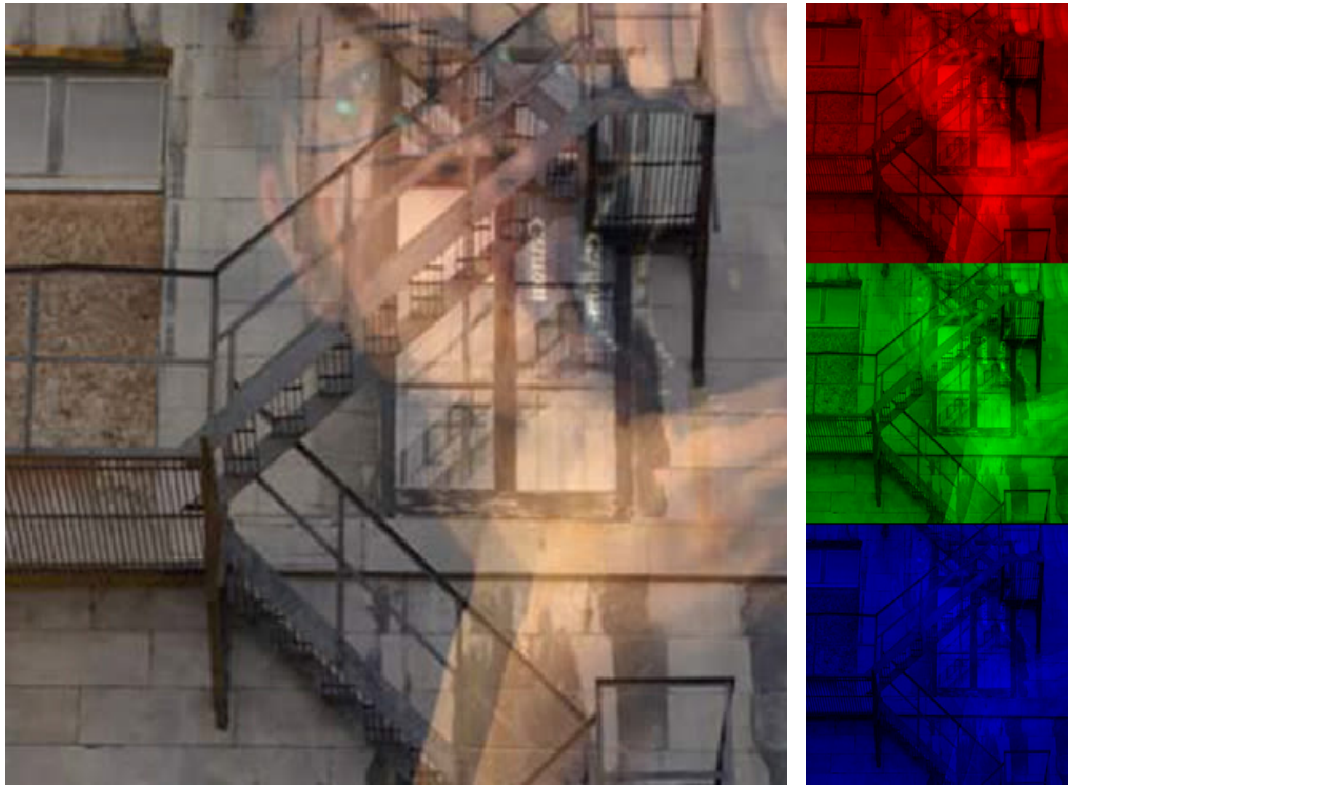
$$\left. + (-x)^{p-k} r(k, n(x)) \exp\left( \frac{x - x_{(n(x))}}{h} \right) \right),$$

$$\ell(k,j) := \sum_{i=1}^{j} (-x_{(i)})^k \exp\left( \frac{x_{(i)} - x_{(j)}}{h} \right) \omega_{(i)}$$

$$r(k,j) := \sum_{i=j+1}^{n} x_{(i)}^k \exp\left( \frac{x_{(j)} - x_{(i)}}{h} \right) \omega_{(i)}$$

where $\ell(\cdot,\cdot)$'s and $r(\cdot,\cdot)$'s can be computed recursively

# A (relative) ICA success



# Some Simulation Results

- Some popular benchmark marginals taken from Bach and Jordan (2002), $n = 2000, d = 4$

| amari_fk | amari_fast | amari_ProDen | amari_Pearson |
|---|---|---|---|
| 0.084677 | 0.151334 | 0.049196 | 0.187573 |
| **t_fk** | **t_fast** | **t_ProDen** | **t_Pearson** |
| 0.143280 | 0.012280 | 2.178800 | 0.030220 |

- A (pseudo) cocktail-party problem, $n = 50000, d = 4$

| amari_fk | amari_fkbin | amari_fast | amari_ProDen | amari_Pearson |
|---|---|---|---|---|
| 0.141679 | 0.139295 | 0.160589 | 0.185482 | 0.186624 |
| **t_fk** | **t_fkbin** | **t_fast** | **t_ProDen** | **t_Pearson** |
| 2.433450 | 0.38030 | 0.07330 | 4.98875 | 0.21880 |

# Online Estimation with SGD

- Kernel methods, and non-parametrics in general, don't lend themselves well to online estimation, unless fairly simple examples
- To perform SGD we need (approximately) unbiased estimates for the gradient of the objective
- The Shannon entropy doesn't give us such an option (that I can find), since it seems invariably an estimate for

$$\nabla_{\mathbf{u}} E_{X_{\mathbf{u}}}[\log(f_{X_{\mathbf{u}}}(X))]$$

  will require knowledge of $f_{X_{\mathbf{u}}}$
  - If we look back at the offline gradient, we'll see this explicitly

# Implementing Your Own Projection Pursuit

- Choices to make:
  - Joint estimation of components vs. Sequential
  - Deflation vs orthogonalisation vs information removal
  - normalisation or projected gradient descent
- I like sequential + deflation + normalisation
- Sequential estimation is straightforward
- Deflation:
  - once you obtain $u_k$; the $k$-th projection vector, determine $u^*$ by optimising over the modified data set $XU^0_{1:k}$
    - $U^0_{1:k} \in \mathbb{R}^{p\times(p-k)}$ is a "basis" for the null-space of the of $u_1, ..., u_k$
  - $u_{k+1} = U^0_{1:k} u^*$

# Implementing Your Own Projection Pursuit

- Normalisation: Let's assume $PI(u)$ is our projection index
  - We can write $PI(u) = I(p(\vec{u}))$, where $I(p)$ is the interestingness of the univariate $p$ and $p(u) = Xu$.
  - This gives

$$\nabla_u PI(u) = D_u p' \nabla_p I(p)\big|_{p=X\vec{u}}$$

$$(D_u p)_{ij} = \frac{\partial p_i}{\partial u_j} = \frac{\partial}{\partial u_j} X_{i:}\vec{u} = \frac{1}{||u||}\left(X_{ij} - \frac{1}{||u||} p_i u_j\right)$$

$$\Rightarrow D_u p = \frac{1}{||u||}\left(X - \frac{1}{||u||} p u'\right)$$

# Online Estimation with SGD

- All online ICA methods (of which I am aware) use the fastICA framework
  - Assume knowledge of sub/super-Gaussianity of sources
  - or even stronger assumptions, like equal first, second and fourth moments of all sources
- If we allow deviation from Shannon entropy, then progress can be made
- Although other Rényi's entropies are not consistent for ICA, they can have very strong performance
- The Rényi-2 entropy, also called collision entropy since in the discrete case is $-P(X = Y)$ for $X, Y$ i.i.d., is such an example

# Online Minimum Collision Entropy

- Since

$$\operatorname{argmin}_{\mathrm{u}} \left(-\log\left(E_{X\mathrm{u}}[f_{X\mathrm{u}}(X)]\right)\right) = \operatorname{argmax}_{\mathrm{u}} E_{X\mathrm{u}}[f_{X\mathrm{u}}(X)],$$

and we can show that

$$\begin{aligned} E_{X\perp Y\sim F_X}\left[\frac{\mathrm{u}^\top(X-Y)}{h^3}K\left(\frac{\mathrm{u}^\top(X-Y)}{h}\right)(Y-X)\right] \\ = \nabla_{\mathrm{u}} E_{X\sim X\mathrm{u}}[f_{X\mathrm{u}}(X)] + \mathcal{O}(h^2), \end{aligned}$$

we can achieve convergence to a stationary point of the collision entropy if we can take i.i.d. pairs from the data stream, and choose $h \to 0^+$ approriately

# Online Minimum Collision Entropy

- Some intuition for those who aren't familiar such bias derivations in kernel smoothing:
  - The online updates for PCA are of the form

$$\mathsf{u}^\top \leftarrow (1-\eta)\mathsf{u}^\top + \eta \mathsf{u}^\top (X-\mu)(X-\mu)^\top,$$

    ... (loosely speaking) update u to push the observations away from $\mu$
  - For collision entropy we have

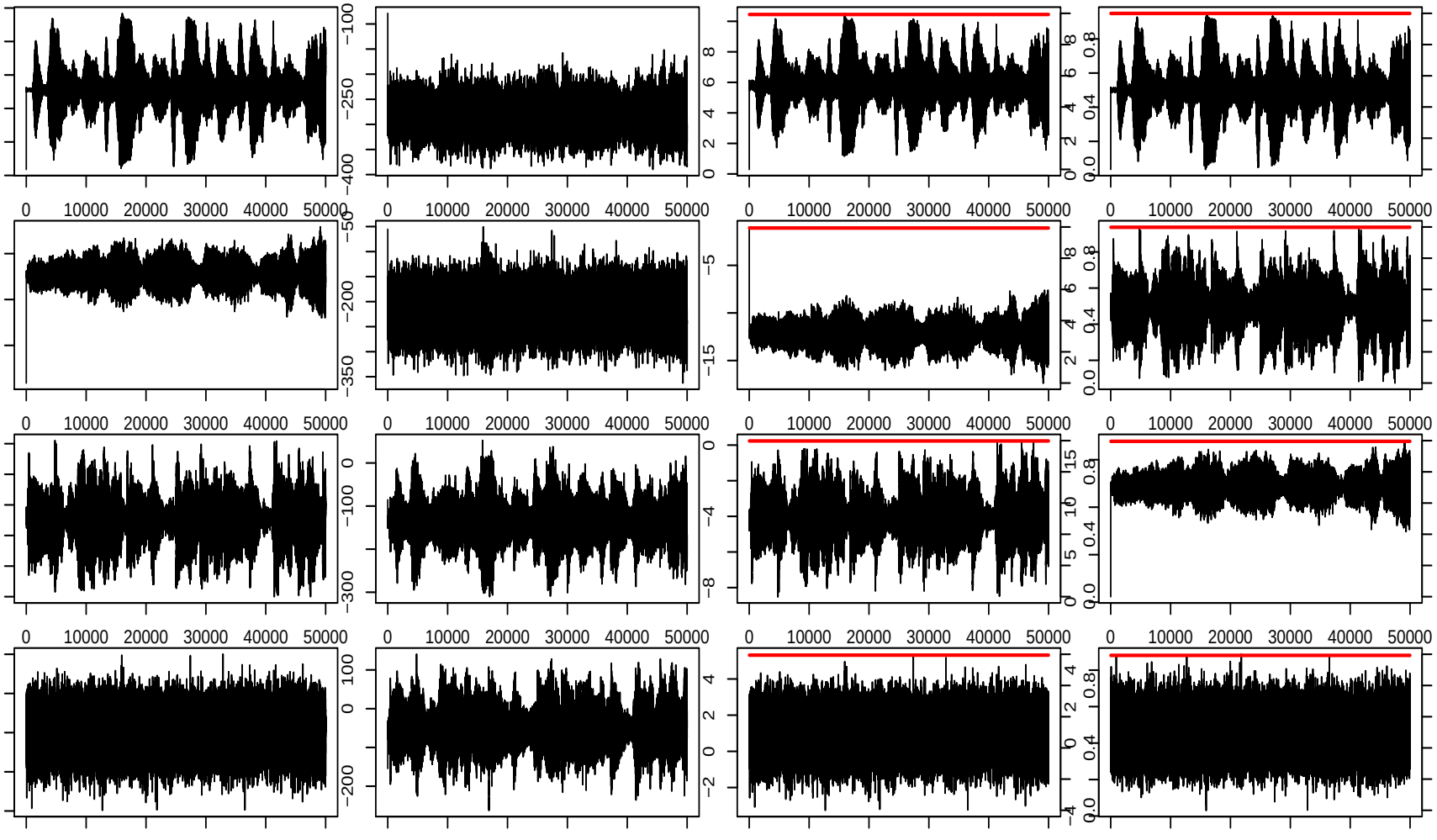
$$\mathsf{u}^\top \leftarrow (1-\eta)\mathsf{u}^\top - \eta K\left(\frac{\mathsf{u}^\top (X-Y)}{h}\right)\mathsf{u}^\top (X-Y)(X-Y)^\top,$$

    ... push the observations *towards one another*, but place emphasis on those which are already nearer along u

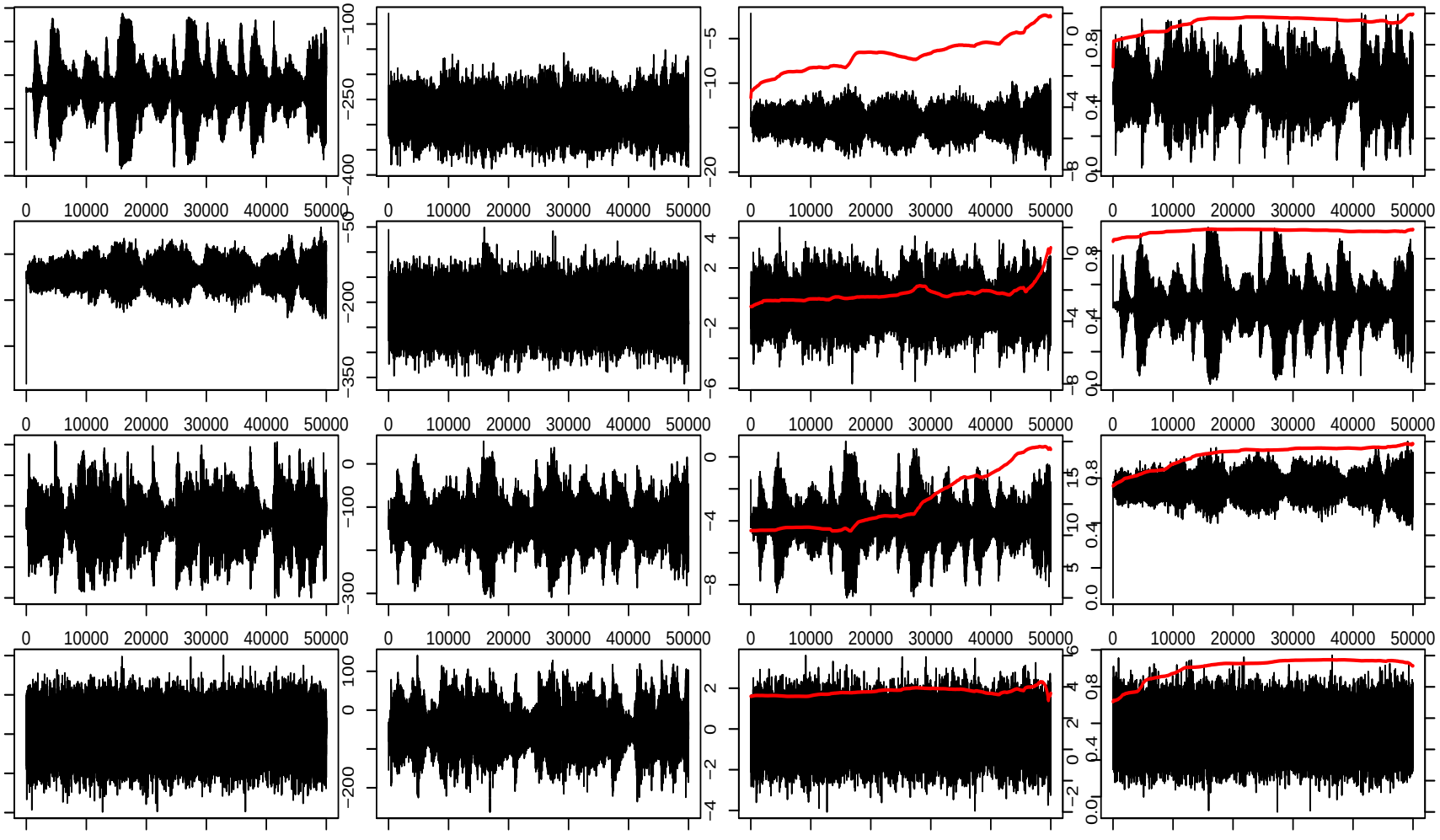
# Auto-dependence a Hidden Blessing?

- Positive auto-correlation means a higher proportion of consecutive stream pairs are close to one another
- Non-constant variance common in sound signals also has periods of high density suitable for faster learning with consecutive pairs

# With Independent Sampling



# Actual Ordering (the realistic case)



# Conclusions

- ICA has diverse applications with remarkable success
- Don't be afraid of kernel smoothing computational costs
- `FastICA` is very fast and often as good as more accurate methods, but sometimes fails poorly
- `ProDenICA` is pretty accurate, but comparatively slower
- Use my `R` package `FKSUM`

# References


- Hofmeyr, D.P. (2021) Fast Exact Evaluation of Univariate Kernel Sums, *IEEE TPAMI*, 43(2):447-458
- Hofmeyr, D.P. (2022) Fast Kernel Smoothing in R with applications to Projection Pursuit, *JSS*, 101(3):1–33.
- Comon, P. (1994): Independent Component Analysis: a new concept?, *Signal Processing*, 36(3):287–314
- Hyvärinen, A.; Oja, E. (2000) Independent Component Analysis: Algorithms and Application, *Neural Networks*, 13(4-5):411-430
- De Lathauwer, L.; De Moor, B.; Vandewalle J (1995) Fetal Electrocardiogram Extraction by Source Subspace Separation. *IEEE SP/Athos Workshop on Higher-Order Statistics*, pp. 134–138.

# References


- Farid, H.; Adelson, E. H. (1999) Separating reflections and lighting using independent components analysis, *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, pp. 262–267
- Bach, F.R. and Jordan, M.I. (2002) Kernel independent component analysis. *JMLR*, 3(Jul):1–48.
- Li, C.J. and Jordan, M.I. (2020) Stochastic Approximation for Online Tensorial Independent Component Analysis, *arXiv:2012.14415v1*
- Hastie, T. and Tibshirani, R. (2003) Independent components analysis through product density estimation. *NeurIPS*, pp 665–672.
- Hofmeyr, D.P. (202?) Maximum Wasserstein Non-Gaussianity for Independent Components Analysis, *In preparation*.
- Hofmeyr, D.P. (202?) Online Minmisation of Collision Entropy for Source Separation, *In preparation*