

Independent Component Analysis by Product Density Estimation

Trevor Hastie and Robert Tibshirani
Statistics Department
Stanford University

<http://www-stat.stanford.edu/~hastie/Papers/ica.pdf>

ICA Problem

$$X = \mathbf{A}S$$

where

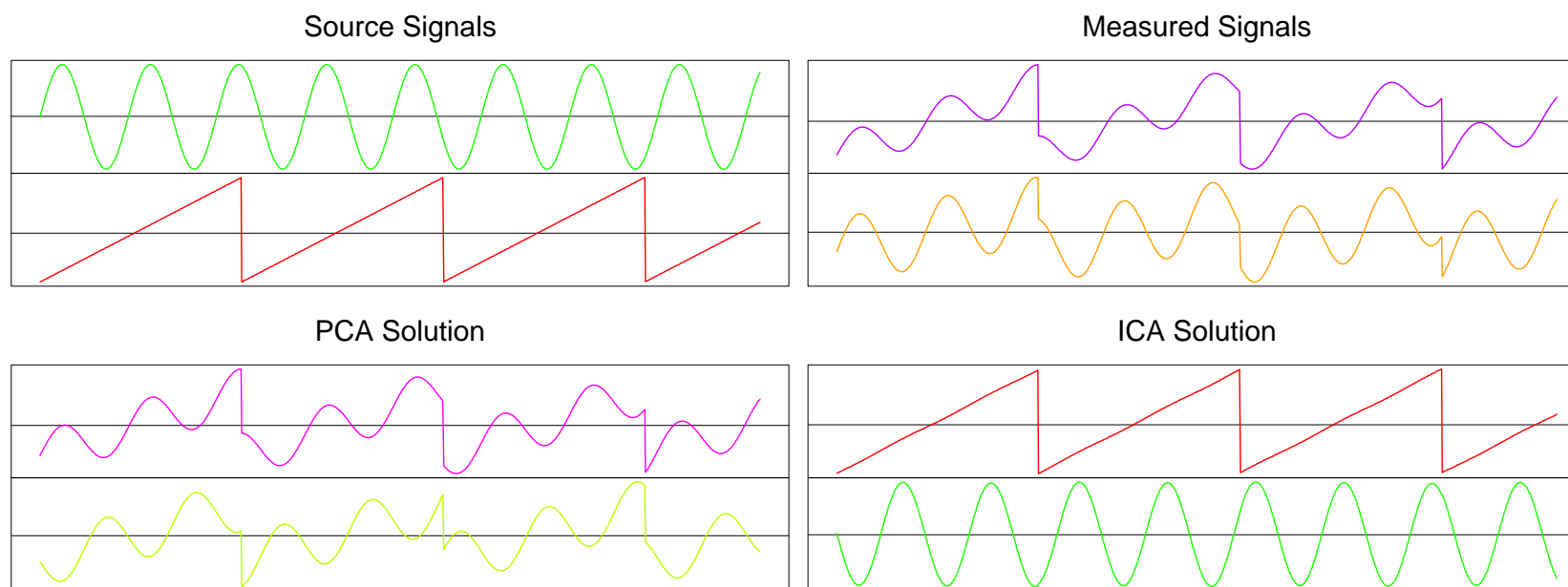
- X is a random p -vector representing multivariate input measurements.
- S is a latent source p -vector whose components are independently distributed random variables.
- \mathbf{A} is $p \times p$ mixing matrix.

Given realizations x_1, x_2, \dots, x_N of X , the goals of ICA are to

- Estimate \mathbf{A}
- Estimate the source distributions $f_{S_j}, j = 1, \dots, p$.

Cocktail Party Problem

In a room there are p independent sources of sound, and p microphones placed around the room hear different mixtures.



Here each of the $x_{ij} = x_j(t_i)$ and recovered sources are a time-series sampled uniformly at times t_i .

Independent vs Uncorrelated

WoLOG can assume that $E(S) = 0$ and $\text{Cov}(S) = \mathbf{I}$, and hence $\text{Cov}(X) = \text{Cov}(\mathbf{A}S) = \mathbf{A}\mathbf{A}^T$.

Can we recover \mathbf{A} using the sample covariance matrix $\widehat{\text{Cov}}(X)$?

No! There are many ways to factor $\widehat{\text{Cov}}(X) = \mathbf{R}\mathbf{R}^T$ (Cholesky, Principal Components, \dots).

But then with $\mathbf{R}^* = \mathbf{R}\mathbf{Q}$, for any $p \times p$ orthonormal \mathbf{Q} (rotation),

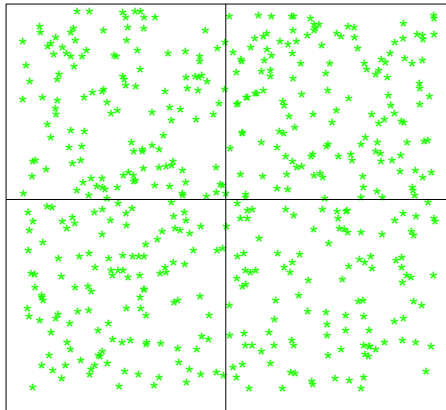
$$\text{Cov}(\mathbf{R}S) = \mathbf{R}\mathbf{R}^T = \mathbf{R}\mathbf{Q}\mathbf{Q}^T\mathbf{R}^T = \mathbf{R}^*\mathbf{R}^{*T} = \text{Cov}(\mathbf{R}^*S)$$

Hence methods based on second order moments, like principal components and Gaussian factor analysis, cannot recover \mathbf{A} .

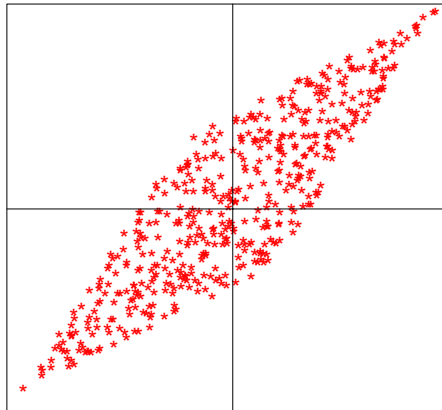
ICA uses [independence](#), and non-Gaussianity of S , to recover \mathbf{A} — e.g. higher order moments.

Independent vs Uncorrelated Demo

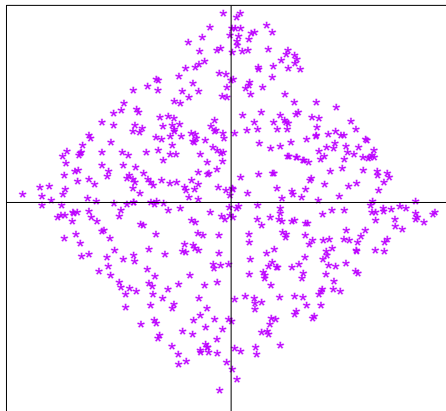
Source S



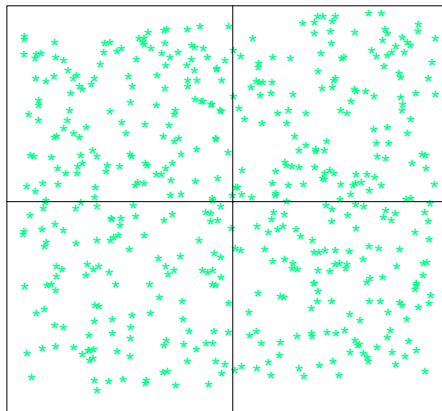
Data X



PCA Solution



ICA Solution

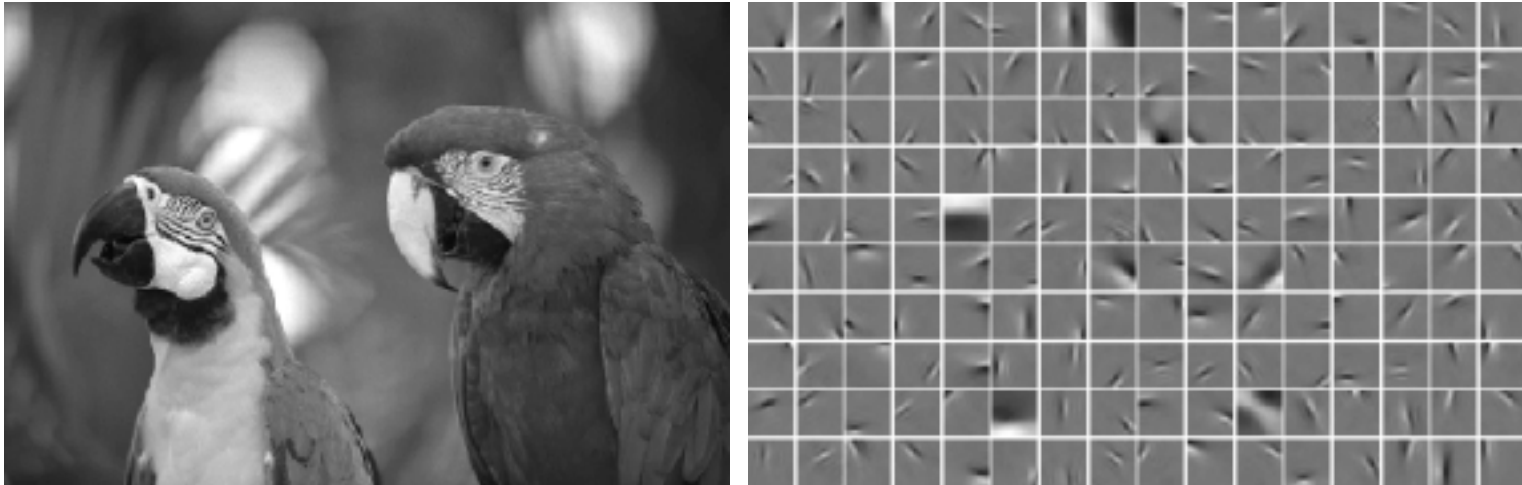


Principal components are uncorrelated linear combinations of X , chosen to successively maximize variance.

Independent components are also uncorrelated linear combinations of X , chosen to be as independent as possible.

Example: ICA representation of Natural Images

Pixel blocks are treated as vectors, and then the collection of such vectors for an image forms an image database. ICA can lead to a sparse coding for the image, using a **natural** basis.

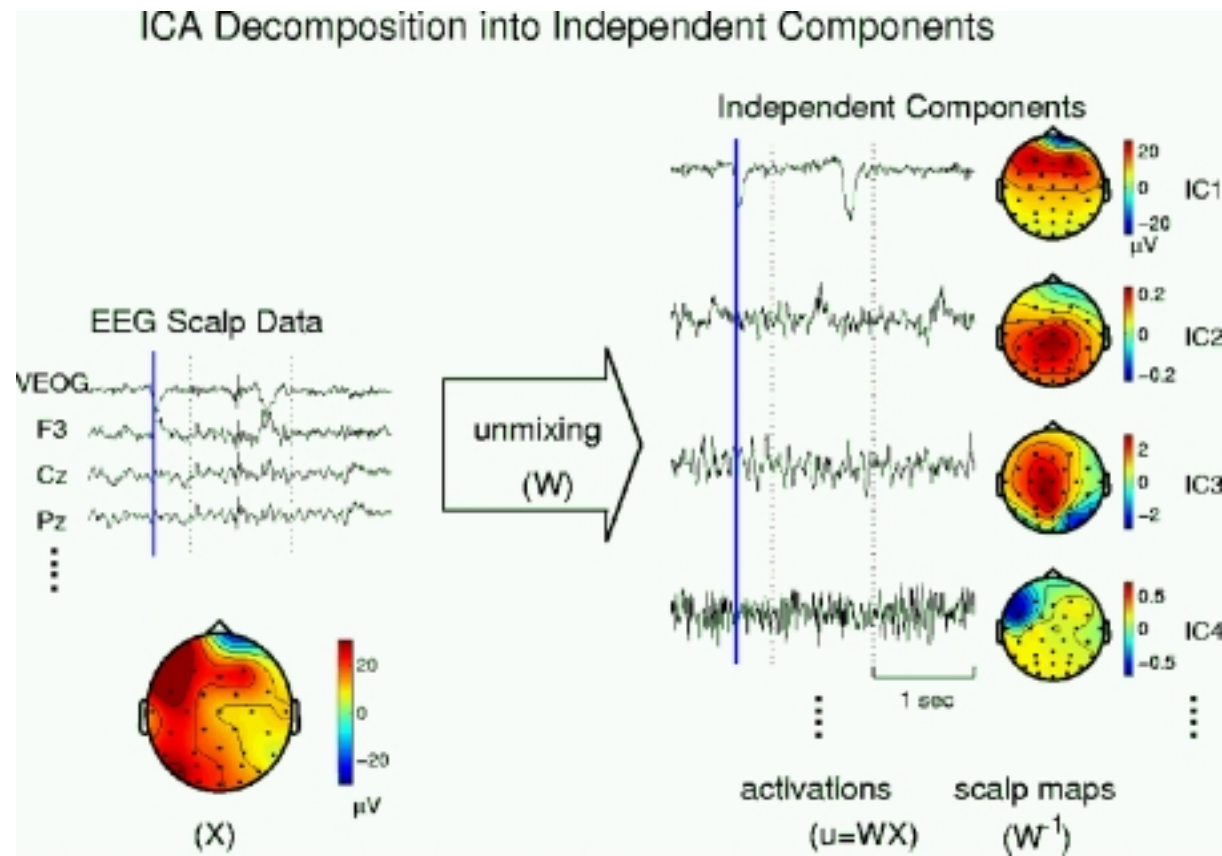


see <http://www.cis.hut.fi/projects/ica/imageica/> (Patrik Hoyer and Aapo Hyvärinen, Helsinki University of Technology)

ICA image compression of coauthor



Example: ICA and EEG data



See http://www.cnl.salk.edu/~tewon/ica_cnl.html (Scott Makeig, Salk Institute)

Approaches to ICA

ICA literature is HUGE. Recent book by [Hyvärinen, Karhunen & Oja \(Wiley, 2001\)](#) is a great source for learning about ICA, and some good computational tricks.



- Mutual Information and Entropy, maximizing non-Gaussianity — [FastICA](#) (HKO 2001), [Infomax](#) (Bell and Sejnowski, 1995)
- Likelihood methods — [today's talk](#), in literature, e.g. Pham and Garat (1993), but considered difficult.
- Nonlinear decorrelation — Y_1 independent Y_2 iff $\max_{g,f} \text{Corr}[f(Y_1), g(Y_2)] = 0$ (Hérault-Jutten, 1984), [KernelICA](#) (Bach and Jordan, 2001)
- Tensorial moment methods

ICA Density Model

Density of S :

$$f_S(s) = \prod_{j=1}^p f_j(s_j)$$

with each f_j a univariate density, with mean 0 and variance 1.

Density of $X = \mathbf{A}S$:

$$f_X(x) = |\mathbf{B}| \prod_{j=1}^p f_j(b_j^T x)$$

with $\mathbf{B} = \mathbf{A}^{-1}$.

Log-likelihood, given x_1, x_2, \dots, x_N :

$$\ell(\mathbf{B}, \{f_j\}_1^p) = \log |\mathbf{B}| + \sum_{i=1}^N \sum_{j=1}^p \log f_j(b_j^T x_i)$$

Simplifications

- Let $\Sigma = \text{Cov}(X)$, and let $\mathbf{B} = \mathbf{W}\Sigma^{-\frac{1}{2}}$ for some nonsingular \mathbf{W} . Then $S = \mathbf{B}X$ and $\text{Cov}(S) = \mathbf{I} \implies \mathbf{W}$ is orthonormal. With $\tilde{X} = \Sigma^{-\frac{1}{2}}X$, we seek $S = \mathbf{W}\tilde{X}$.

- Let

$$f_j(s_j) = \phi(s_j)e^{g_j(s_j)},$$

a **tilted** Gaussian density. Here ϕ is the standard Gaussian density, and g_j satisfies the normalization conditions.

Then

$$\ell(\mathbf{W}, \Sigma, \{g_j\}_1^p) = -\frac{1}{2} \log |\Sigma| + \sum_{i=1}^N \sum_{j=1}^p [\log \phi_j(w_j^T \tilde{x}_i) + g_j(w_j^T \tilde{x}_i)]$$

- We estimate $\hat{\Sigma}$ by the sample covariance of the x_i , and ignore it thereafter — **pre-whitening** in the ICA literature.

Restrictions on g_j

Our model is still over-parameterized. We maximize instead a **penalized** log-likelihood:

$$\sum_{j=1}^p \left[\frac{1}{N} \sum_{i=1}^N [\log \phi(w_j^T x_i) + g_j(w_j^T x_i)] - \lambda_j \int \{g_j'''(t)\}^2(t) dt \right]$$

w.r.t. $\mathbf{W}^T = (w_1, w_2, \dots, w_p)$ and $g_j, j = 1, \dots, p$

subject to

- $\mathbf{W}^T \mathbf{W} = \mathbf{I}$
- each $f_j(s) = \phi(s)e^{g_j(s)}$ is a density, with mean 0 and variance 1.

ProDenICA: Product Density ICA algorithm

1. Initialize \mathbf{W} (random Gaussian matrix followed by orthogonalization).
2. Alternate until convergence of \mathbf{W} , using the Amari metric.
 - (a) Given \mathbf{W} , optimize the penalized log-likelihood w.r.t. g_j (separately for each j), using the penalized density estimation algorithm.
 - (b) Given g_j , $j = 1, \dots, p$, perform one step of the fixed point algorithm towards finding the optimal \mathbf{W} .

Penalized Density Estimation — Lagrange trick

For fixed \mathbf{W} , let $s_{ij} = w_j^T x_i$. Note that s_{ij} has sample mean 0 and variance 1 (since x_i are pre-whitened).

$$\max_g \frac{1}{N} \sum_{i=1}^N [\log \phi(s_i) + g(s_i)] - \int \phi(t) e^{g(t)} dt - \lambda \int \{g'''(t)\}^2(t) dt$$

Theorem

- Solution \hat{g} is a quartic smoothing spline
- $\hat{f}(x) = \phi(x) e^{\hat{g}(x)}$ integrates to 1.
- $\hat{f}(x)$ has mean and variance equal to the sample mean and variance (0,1) of the data s_i .

Proof: based on results in Silverman (1986) and Efron and Tibshirani (1996), and null space of quadratic penalty.

Penalized Density Estimation — Poisson trick

Discretize the integral $\int \phi(t)e^{g(t)} dt$. Define fine grid of L values s_ℓ^* with increments Δ covering observed s_i , and define

$$y_\ell^* = \frac{\#s_i \in (s_\ell^* - \Delta/2, s_\ell^* + \Delta/2)}{N}$$

Approximate log-likelihood by

$$\sum_{\ell=1}^L \left\{ y_i^* [\log(\phi(s_\ell^*)) + g(s_\ell^*)] - \Delta \phi(s_\ell^*) e^{g(s_\ell^*)} \right\} - \lambda \int g''^2(s) ds.$$

This is a penalized Poisson log-likelihood with response y_ℓ^*/Δ , mean $\mu(s) = \phi(s)e^{g(s)}$, and penalty λ/Δ .

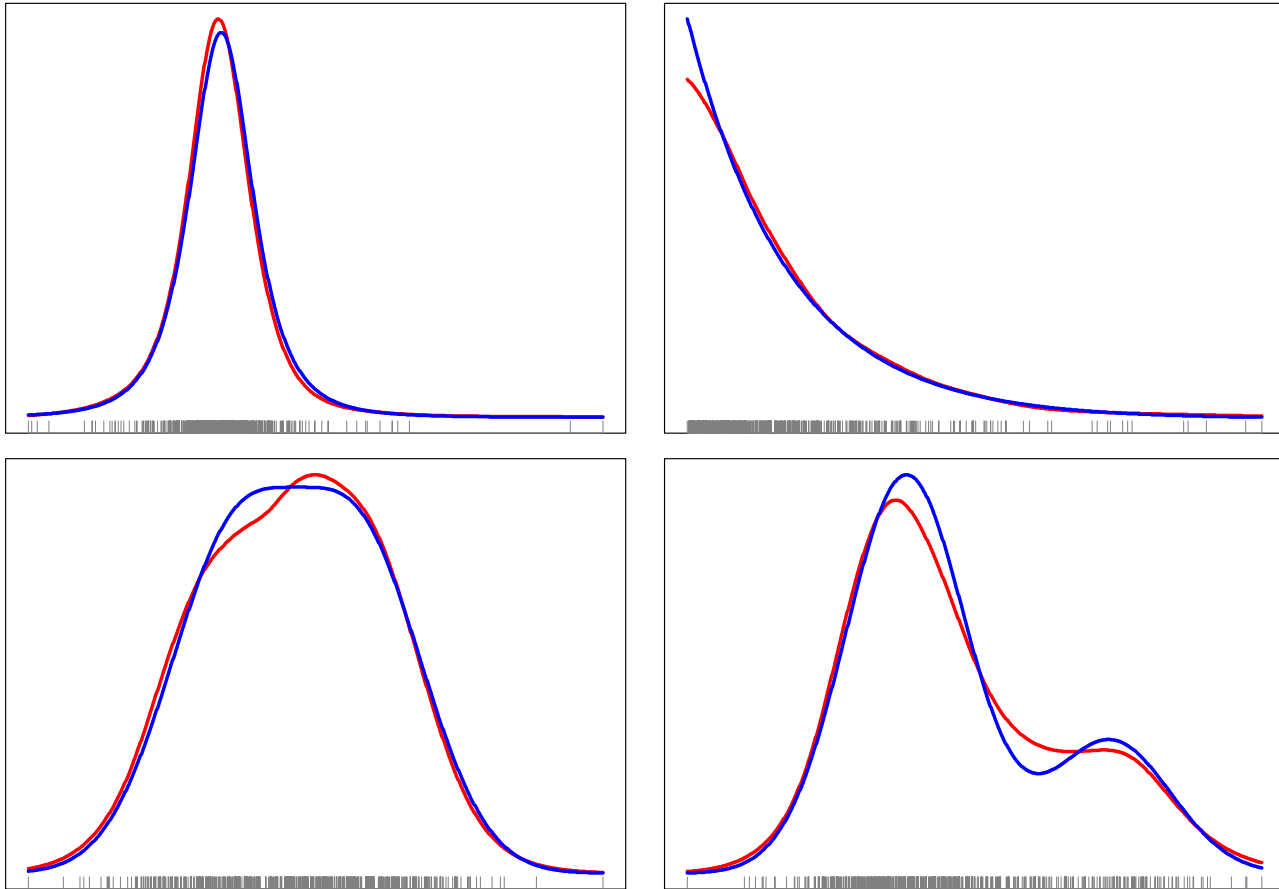
A.k.a. a **generalized additive model** (Poisson error, log link) with an **offset** $\log \phi(s)$.

GAMs and computational issues

- Newton algorithm in $O(L)$ computations
- Splus `gam()` function fits cubic, not quartic splines. Can instead use $g(s) = \beta s^2 + h(s)$, and use $\int \{h''(t)\}^2 dt$. Solution will have \hat{h} a cubic spline, and \hat{f} will still have unit variance.
- In practice, we ignore the βs^2 term, since quadratics are almost unpenalized in $\int \{h''(t)\}^2 dt$.
- Algorithm delivers first and second derivatives of \hat{g} for free, needed in optimization of \mathbf{W} .
- Can specify λ via **effective degrees of freedom** — trace of smoother matrix.
- Fit a Poisson GAM to each $\{w_j^T x\}_1^p$, delivering $\{\hat{f}_j\}_1^p$.

Examples: Density Estimates \hat{f}

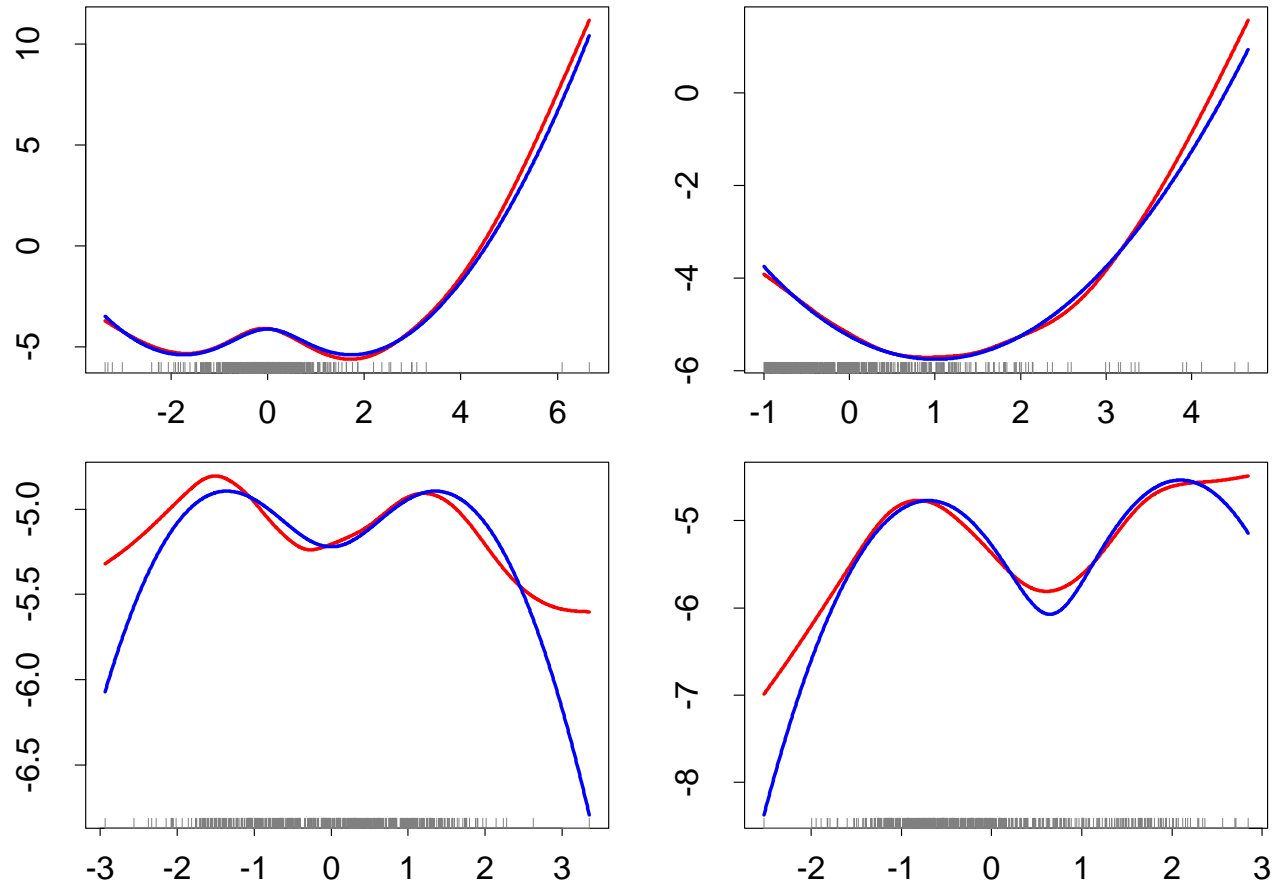
N=500 df=6 L=1000



True Estimated

Examples: Tilt Estimates \hat{g}

N=500 df=6 L=1000



True Estimated

ProDenICA: Product Density ICA algorithm

1. Initialize \mathbf{W} (random Gaussian matrix followed by orthogonalization).
2. Alternate until convergence of \mathbf{W} , using the Amari metric
 - (a) Given \mathbf{W} , optimize the penalized log-likelihood w.r.t. g_j (separately for each j), by fitting p separate **Poisson GAM** models.
 - (b) Given g_j , $j = 1, \dots, p$, perform one step of the fixed point algorithm towards finding the optimal \mathbf{W} .

Updating \mathbf{W} for fixed \hat{f}_j

$$\sum_{j=1}^p \left[\frac{1}{N} \sum_{i=1}^N [\log \phi(w_j^T x_i) + g_j(w_j^T x_i)] - \lambda_j \int \{g_j'''(t)\}^2(t) dt \right]$$

- Last term does not depend on \mathbf{W} ,
- $\sum_{j=1}^p \log \phi(w_j^T x_i) = c - x_i^T \mathbf{W}^T \mathbf{W} x_i / 2 = c - x_i^T x_i / 2$ — does not depend on \mathbf{W} either

Suffices to optimize

$$C(\mathbf{W}) = \frac{1}{N} \sum_{j=1}^p \sum_{i=1}^N g_j(w_j^T x_i) = \sum_{j=1}^p C_j(w_j)$$

$C(\mathbf{W})$ is log likelihood-ratio between fitted density and Gaussian, (an estimate of **negentropy**), and each of \hat{g}_j are **contrast** functions.

Algorithm: Fixed Point update for \mathbf{W}

1. For $j = 1, \dots, p$:

$$w_j \leftarrow \mathbf{E} \left\{ X g'_j(w_j^T X) - \mathbf{E} \{ g''_j(w_j^T X) \} w_j \right\},$$

where \mathbf{E} represents expectation w.r.t. the sample x_i , and w_j is the j th column of \mathbf{W} .

2. Orthogonalize \mathbf{W} : Compute its SVD, $\mathbf{W} = \mathbf{U}\mathbf{D}\mathbf{V}^T$, and replace $\mathbf{W} \leftarrow \mathbf{U}\mathbf{V}^T$.

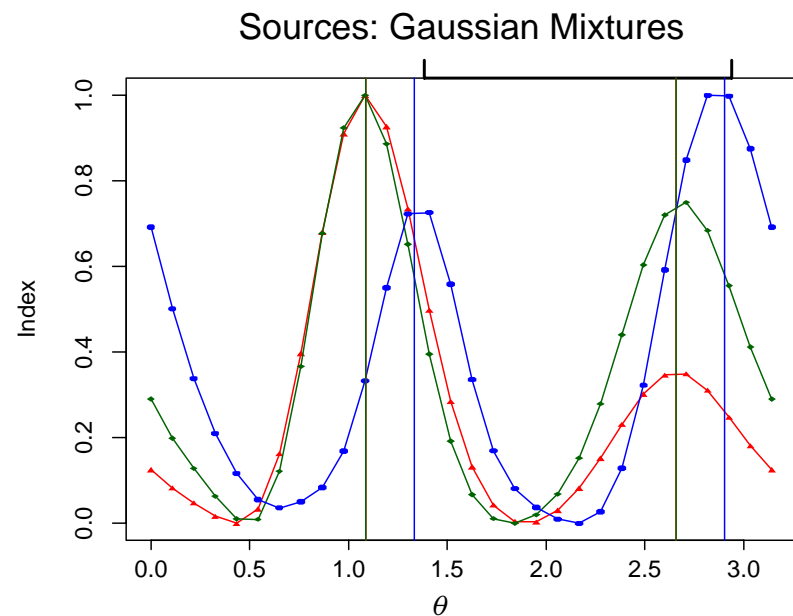
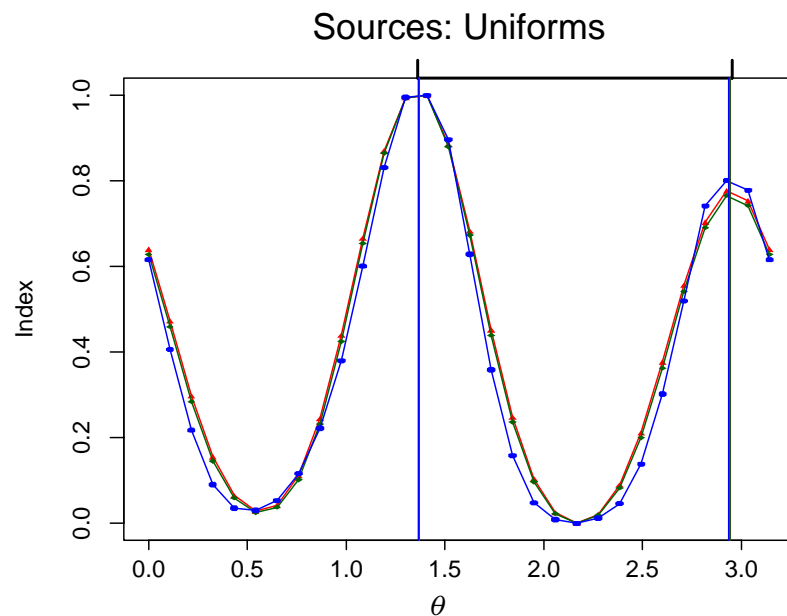
- See ICA book by Hyvärinen, Karhunen & Oja (2001) — modified Newton algorithm.
- since \hat{g}_j are cubic splines, \hat{g}'_j and \hat{g}''_j are readily available.

ProDenICA: Product Density ICA algorithm

1. Initialize \mathbf{W} (random Gaussian matrix followed by orthogonalization).
2. Alternate until convergence of \mathbf{W} , using the Amari metric
 - (a) Given \mathbf{W} , optimize the penalized log-likelihood w.r.t. g_j (separately for each j), by fitting p separate **Poisson GAM** models.
 - (b) Given g_j , $j = 1, \dots, p$, perform one step of the **fixed point algorithm** towards finding the optimal \mathbf{W} .

Two 2-D examples

We compare our ProDenICA to FastICA (using two popular contrast functions) in 2-D. The orthogonal frame \mathbf{W} is indexed by the angle θ .



FastICA G_1 FastICA G_2 ProDenICA 6df spline

Entropy and Mutual Information

Entropy: $H(Y) = - \int f(y) \log f(y) dy$ — maximized by $f(y) = \phi(y)$, the Gaussian (for fixed variance).

Mutual Information: $I(Y) = \sum_{j=1}^p H(Y_j) - H(Y)$

- Y is a random vector with joint density $f(y)$ and entropy $H(Y)$
- $H(Y_j)$ is the (marginal) entropy of component Y_j , with marginal density $f_j(y_j)$.
- $I(Y)$ is the **Kullback-Leibler** divergence between $f(Y)$ and its **independence version** $\prod_1^p f_j(y_j)$ (which is the KL closest of all independence densities to $f(y)$)
- Hence $I(Y)$ is a measure of dependence between the components of a random vector Y .

Entropy, Mutual Information, and ICA

If $\text{Cov}(X) = I$ and \mathbf{W} is orthogonal then simple calculations show

$$I(\mathbf{W}X) = \sum_{j=1}^p H(w_j^T X) - H(X)$$

Hence

$$\begin{aligned} \min_{\mathbf{W}} I(\mathbf{W}X) &\iff \min_{\mathbf{W}} \{\text{dependence between } w_j^T X\} \\ &\iff \min_{\mathbf{W}} \{\text{the sum of the entropies of the } w_j^T X\} \\ &\iff \max_{\mathbf{W}} \{\text{departures from Gaussianity of the } w_j^T X\} \end{aligned}$$

- Many methods for ICA look for low-entropy or non-gaussian projections (Hyvärinen, Karhunen & Oja, 2001)
- Strong similarities with projection pursuit (Friedman and Tukey, 1974)

Negentropy and *FastICA*

Negentropy: $J(Y_j) = H(Y_j) - H(Z_j)$, where Z_j is a Gaussian RV with same variance as Y_j . Measures the departure from Gaussianity.

FastICA uses simple approximations to negentropy

$$J(w_j^T X) \approx [EG(w_j^T X) - EG(Z_j)]^2,$$

and with data replace the first expectation by a sample averages.

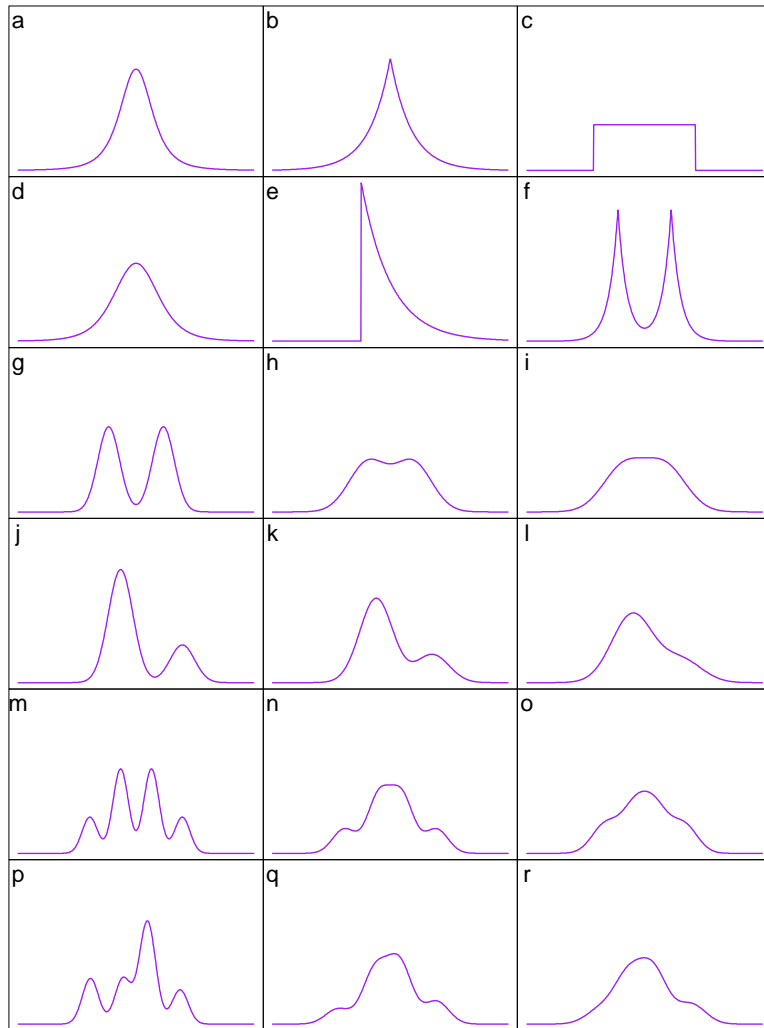
$$G_0(y) = y^4, \quad G_1(y) = \frac{1}{a} \log \cosh ay, \quad G_2(y) = -\exp(-y^2)$$

Poisson GAM deviance from Gaussian

$$C_j(w_j) = \frac{1}{N} \sum_{i=1}^N g_j(w_j^T x_i)$$

can be viewed as a **negentropy** measure, with contrast function g_j

Simulations



Taken from Bach and Jordan (2001)

- Each distribution used to generate 2-dim S and X (with random \mathbf{A}) — 30 reps each
- 300 reps with 4-dim S — distributions of S_j picked at random.

Compared [FastICA](#) (homegrown Splus version), [KernelICA](#) (Francis Bach's matlab version), and [ProDenICA](#) (Splus).

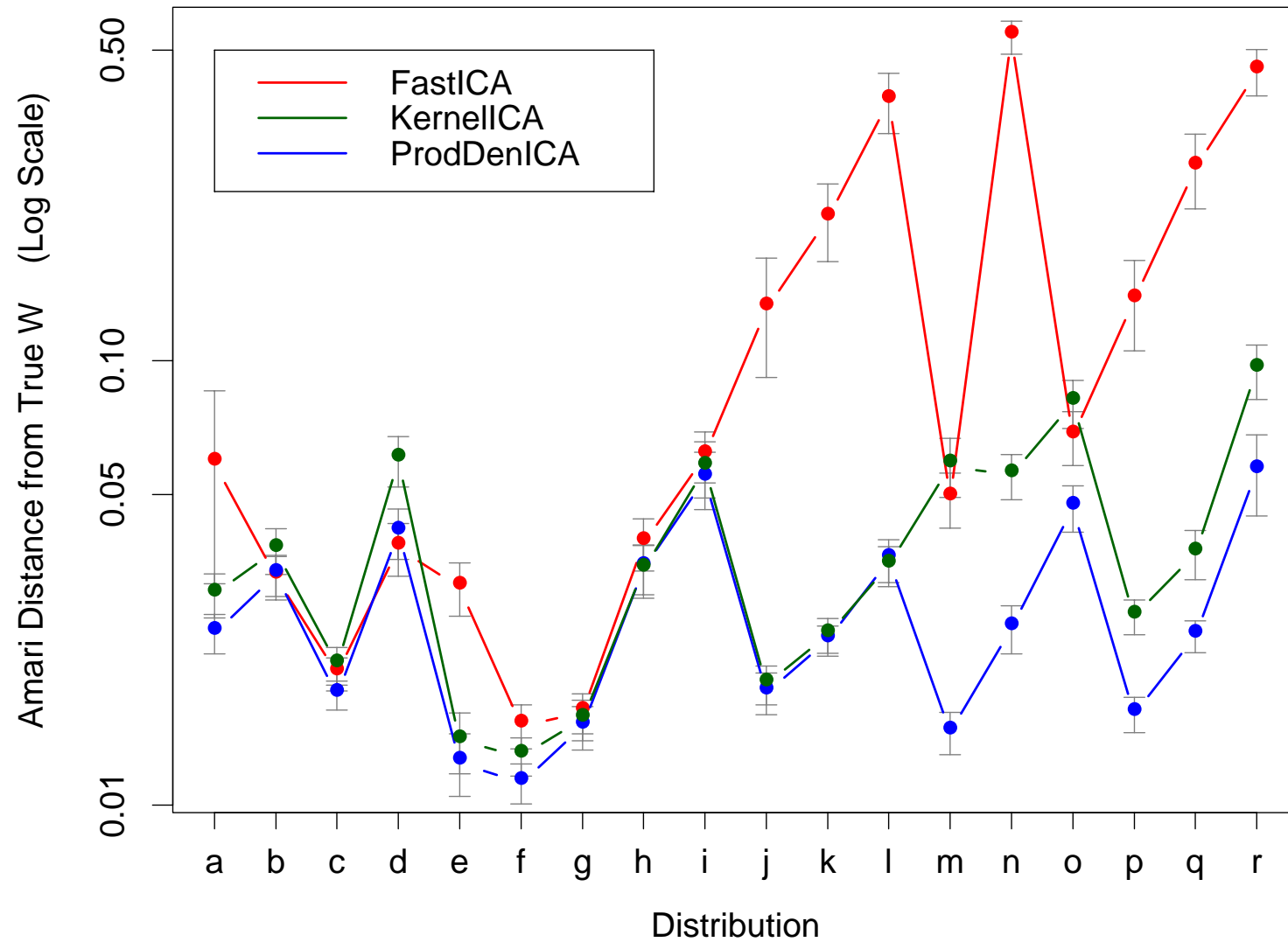
Amari Metric

We evaluate solutions by comparing their estimated \mathbf{W} with the true \mathbf{W}_0 , using the Amari metric (HKO 2001, Bach and Jordan 2001):

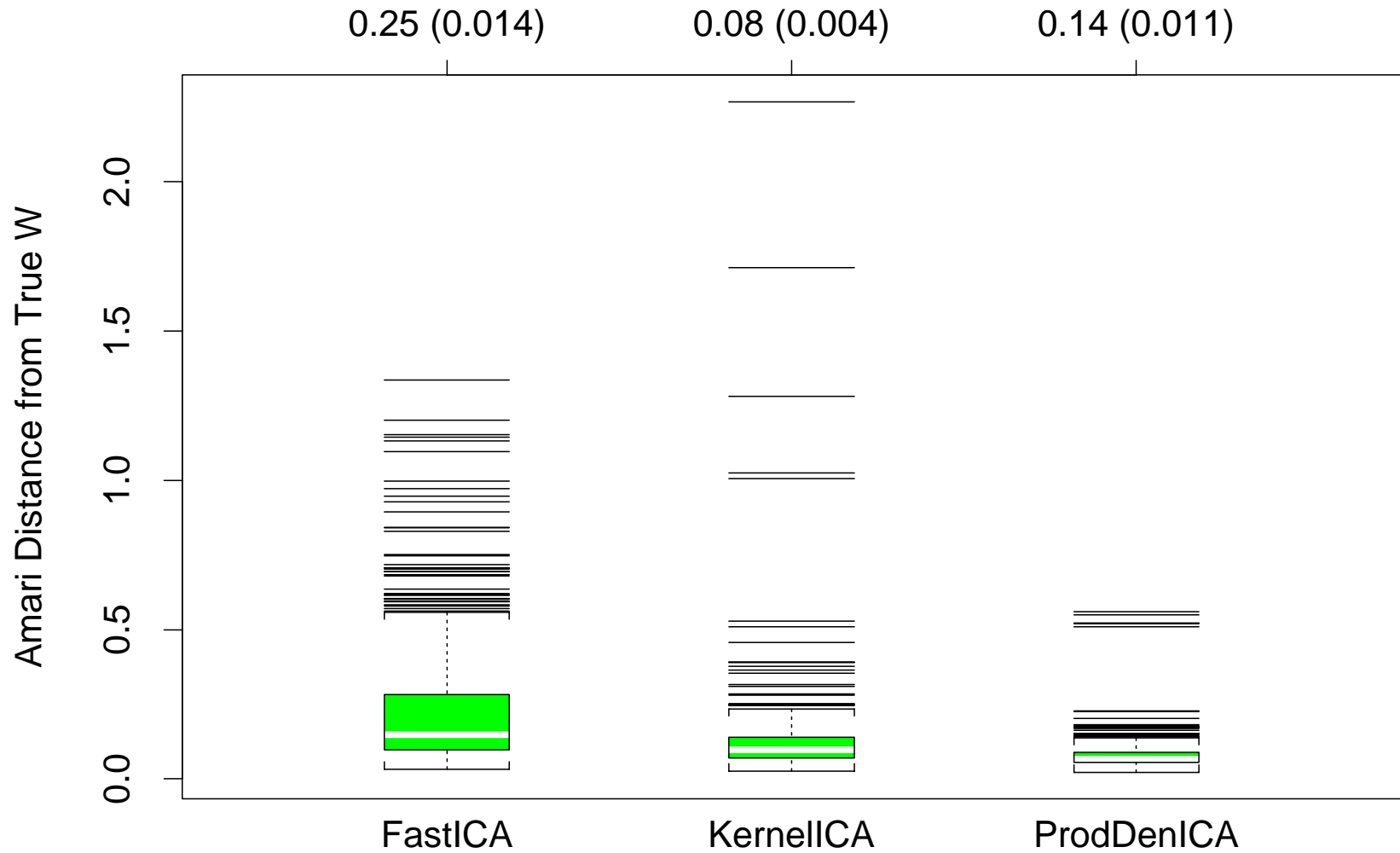
$$d(\mathbf{W}_0, \mathbf{W}) = \frac{1}{2p} \sum_{i=1}^p \left(\frac{\sum_{j=1}^p |r_{ij}|}{\max_j |r_{ij}|} - 1 \right) + \frac{1}{2p} \sum_{j=1}^p \left(\frac{\sum_{i=1}^p |r_{ij}|}{\max_i |r_{ij}|} - 1 \right)$$

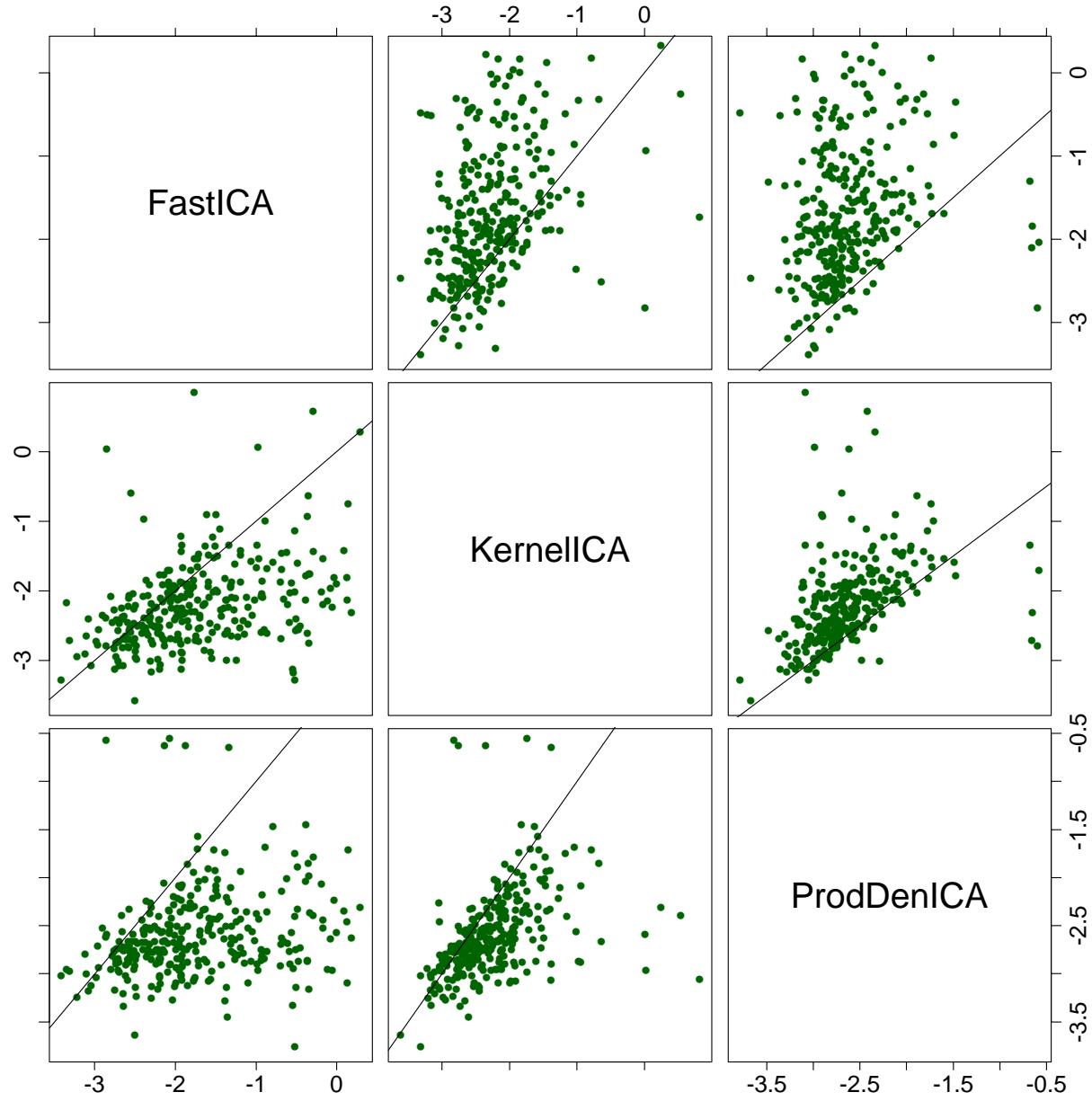
where $r_{ij} = (\mathbf{W}_0 \mathbf{W}^{-1})_{ij}$.

2-Dim examples



4-dim examples





Future Directions

- Try some real examples!
- Extend **noiseless** $X = \mathbf{A}S$ model to **noisy** model (a.k.a. **factor analysis**)

$$X = \mathbf{A}S + E$$

- Simple adaptations allow us to model $q < p$ components
 - Reduce via PCA to q -dim subspace, and then run algorithm
 - Can leave dimension at p , and then leave $p - q$ densities as Gaussian (hybrid factor analysis model).
- Study asymptotic distributions and rates of convergence of **ProDenICA** solutions (kidding!!)