Applications of the Hilbert-Schmidt Independence Criterion

Kernel Methods in Machine Learning

Arthur Gretton

Gatsby Computational Neuroscience Unit

Application of HSIC: Feature Selection

HSIC for Microarray feature selection

- Select genes from microarray data for classification
- Different methods choose features optimising different criteria

HSIC for Microarray feature selection

- Select genes from microarray data for classification
- Different methods choose features optimising different criteria
- Several criteria special cases of HSIC: [ICML07a,ISMB07]
  - Pearson's correlation (normalise by standard deviation) [van't Veer et al., 2002, Ein-Dor et al., 2006]
  - Mean difference and variants [Bedo et al., 2006, Hastie et al., 2001]
  - Shrunken centroid [Tibshirani et al., 2002, 2003]
  - (Kernel) ridge regression [Li and Yang, 2005]

HSIC for Microarray feature selection

- Select genes from microarray data for classification
- Different methods choose features optimising different criteria
- Several criteria special cases of HSIC: [ICML07a,ISMB07]
  - Pearson's correlation (normalise by standard deviation) [van't Veer et al., 2002, Ein-Dor et al., 2006]
  - Mean difference and variants [Bedo et al., 2006, Hastie et al., 2001]
  - Shrunken centroid [Tibshirani et al., 2002, 2003]
  - (Kernel) ridge regression [Li and Yang, 2005]
- When are nonlinear feature maps justified?

Feature selection: BAHSIC (1)

- Backwards elimination of irrelevant features to maximise dependence (HSIC). Why backwards?

Feature selection: BAHSIC (1)

- Backwards elimination of irrelevant features to maximise dependence (HSIC). Why backwards?

  Input: The full set of features $\mathcal{S}$

  Output: An ordered set of features $\mathcal{S}^\dagger$

  ```
  $\mathcal{S}^\dagger \leftarrow \varnothing$
  **repeat**
      Adapt kernel parameter $\sigma_0$
      Remove **individual** features to maximize HSIC,
      $\mathcal{I} \leftarrow \arg\max_{\mathcal{I}} \sum_{j\in\mathcal{I}} \mathrm{HSIC}(\sigma_0, \mathcal{S} \setminus \{j\}), \quad \mathcal{I} \subset \mathcal{S}$
      $\mathcal{S} \leftarrow \mathcal{S} \setminus \mathcal{I}$
      $\mathcal{S}^\dagger \leftarrow (\mathcal{S}^\dagger, \mathcal{I})$
  **until** $\mathcal{S} = \varnothing$
  ```

- Application: feature selection in microarrays [ICML07a,ISMB07, JMLR12]

Relation of HSIC to mean difference

- (Biased) empirical HSIC: $\mathrm{HSIC}(X, Y) := \mathsf{Tr}(KHLH)$

Relation of HSIC to mean difference

- (Biased) empirical HSIC: $\text{HSIC}(X, Y) := \mathsf{Tr}(KHLH)$
- HSIC equivalent to difference in means
  - Linear input kernel $K_\ell = x[\ell]\,(x[\ell])^\top$, $K = \sum_\ell K_\ell$ (single feature, HSIC is sum of all feature scores)
  - Linear output kernel, $1/n_+$ for one class, $-1/n_-$ for the other
  - Warning: for nonlinear kernel, features can interact.

$$\mathsf{Tr}(K_\ell HLH) = \left(\frac{1}{n_+} \sum_{i=1}^{n_+} x_i[\ell] - \frac{1}{n_-} \sum_{i=n_++1}^{n} x_i[\ell] \right)^2$$

Relation of HSIC to mean difference

- (Biased) empirical HSIC: $\mathrm{HSIC}(X,Y) := \mathsf{Tr}(KHLH)$
- HSIC equivalent to difference in means
  - Linear input kernel $K_\ell = x[\ell]\,(x[\ell])^\top$, $K = \sum_\ell K_\ell$ (single feature, HSIC is sum of all feature scores)
  - Linear output kernel, $1/n_+$ for one class, $-1/n_-$ for the other
  - Warning: for nonlinear kernel, features can interact.

$$\mathsf{Tr}(K_\ell HLH) = \left(\frac{1}{n_+}\sum_{i=1}^{n_+} x_i[\ell] - \frac{1}{n_-}\sum_{i=n_++1}^{n} x_i[\ell]\right)^2$$

- HSIC equivalent to shrunken centroid
  - Linear kernels, $Y = \begin{pmatrix} \frac{\mathbf{1}_{n_+}}{n_+} - \frac{\mathbf{1}_{n_+}}{n} & -\frac{\mathbf{1}_{n_+}}{n} \\ -\frac{\mathbf{1}_{n_-}}{n} & \frac{\mathbf{1}_{n_-}}{n_-} - \frac{\mathbf{1}_{n_-}}{n} \end{pmatrix}_{n\times 2}.$

$$\mathsf{Tr}(K_\ell HLH) = (\bar{x}_+[\ell] - \bar{x}[\ell])^2 + (\bar{x}_-[\ell] - \bar{x}[\ell])^2$$

Relation of HSIC to ridge regression

- Objective: given $y = [y_1 \ldots y_n]^\top$, minimise

$$R = \|y - Vw\|^2 + \lambda \|w\|^2$$

where

$$V = \begin{pmatrix} k(x_1, \cdot) \\ \vdots \\ k(x_n, \cdot) \end{pmatrix} \quad \text{and} \quad w := \sum_i \alpha_i k(x_i, \cdot)$$

- Solution is:

$$R^* = y^\top y - y^\top (K + \lambda I)^{-1} K y$$

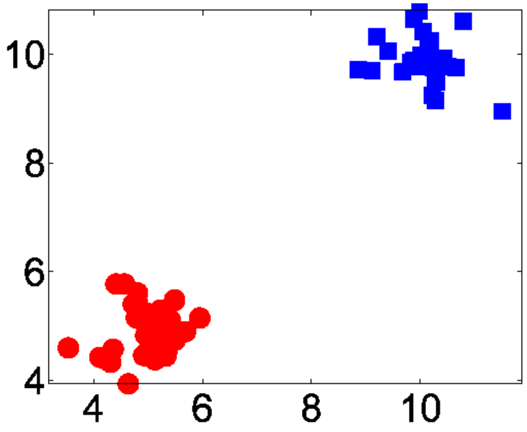
- Features that minimise R^* ⇔ maximise HSIC with kernel

$$\mathfrak{K} = (K + \lambda I)^{-1} K$$

(but take care with centering: either $\sum_i y_i = 0$ or $K = HKH$)

Linear vs nonlinear kernel: idea

- For microarray data (esp. 2 class), difference in means with linear kernel usually works best.



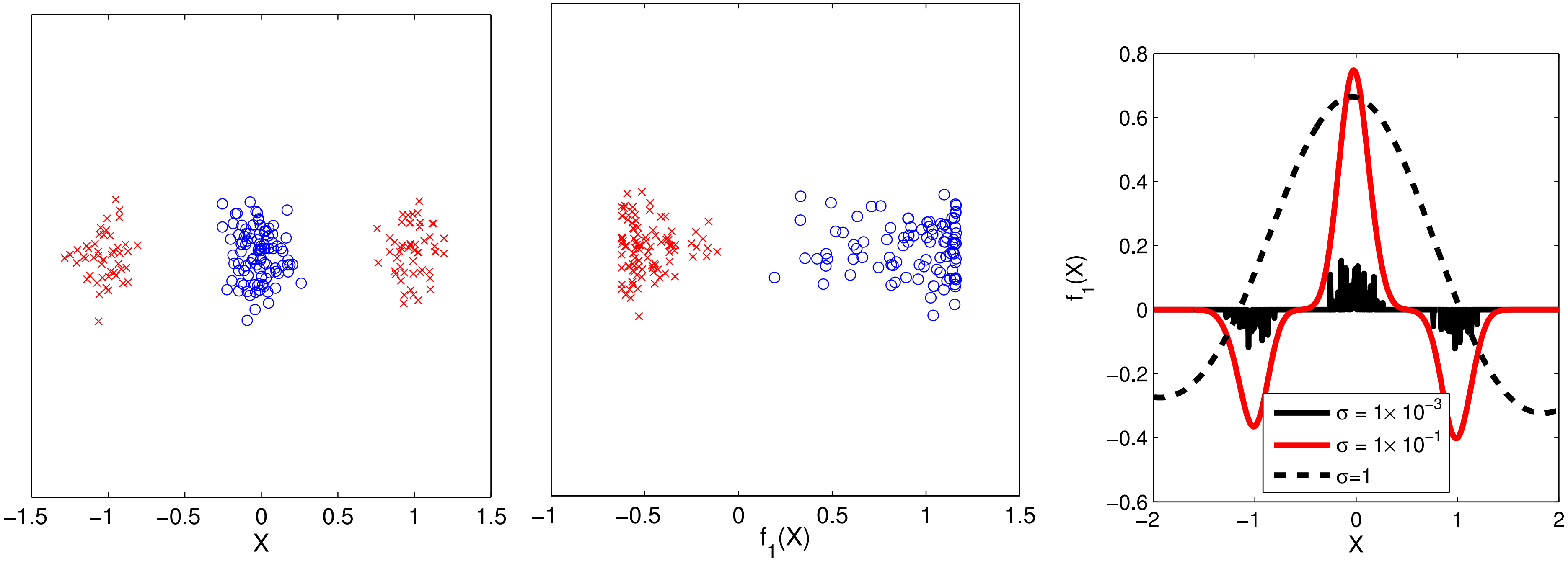
Linear vs nonlinear kernel: idea

- For microarray data (esp. 2 class), difference in means with linear kernel usually works best.
- Exceptions:
  - Nonlinear dependence between features and labels (e.g class with multiple subclasses)
  - Multiple classes, different features serve different purposes

$$L = Y^\top Y = \begin{bmatrix} n_1^{-2} & 0 & \dots & 0 \\ 0 & n_2^{-2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & n_d^{-2} \end{bmatrix}$$

Linear vs nonlinear kernel: application 1

- Two classes, nonlinear relation
- Plot of maximum singular function $f_1(x)$ on $\mathcal{X}$ (as for COCO)

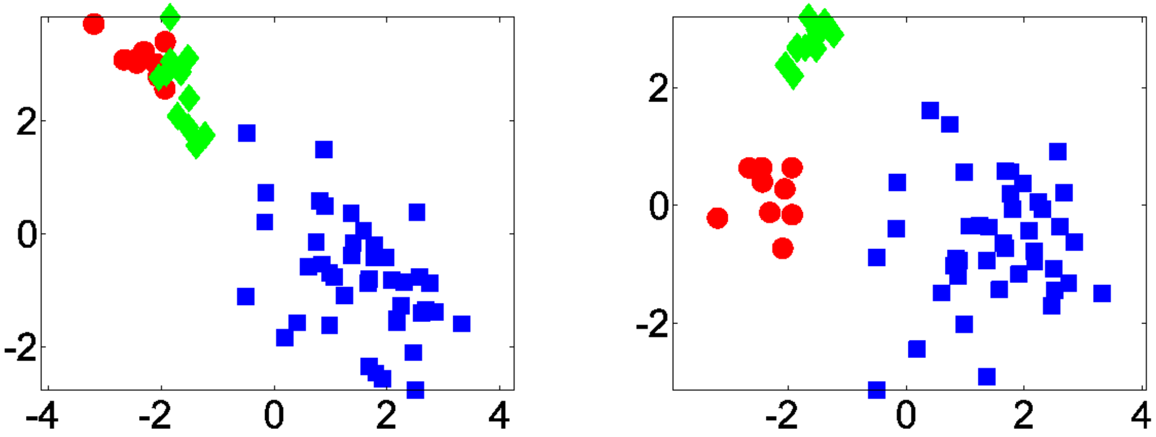


Linear vs nonlinear kernel: application 2

- Three cancer subtypes (diffuse large B-cell lymphoma and leukemia, follicular lymphoma, and chronic lymphocytic leukemia)

Linear

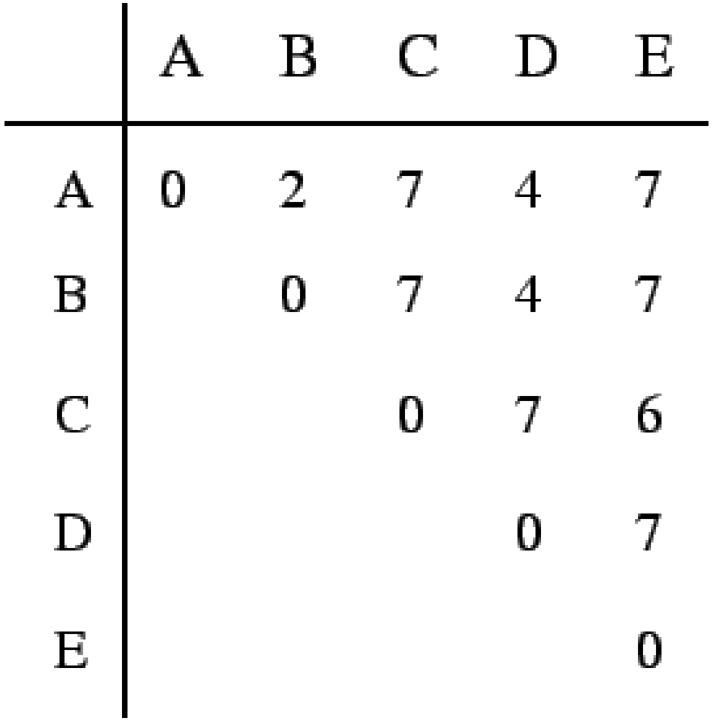
Nonlinear



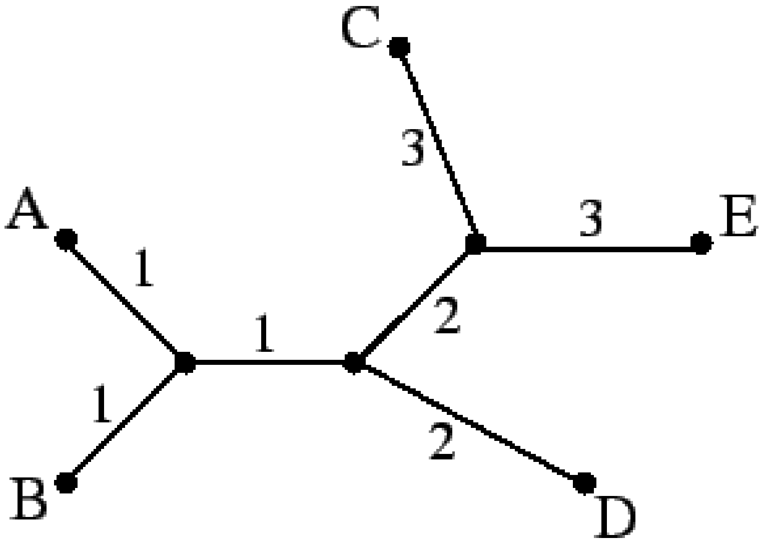
Application 2: Taxonomy Discovery

Overview: HSIC-based taxonomy discovery

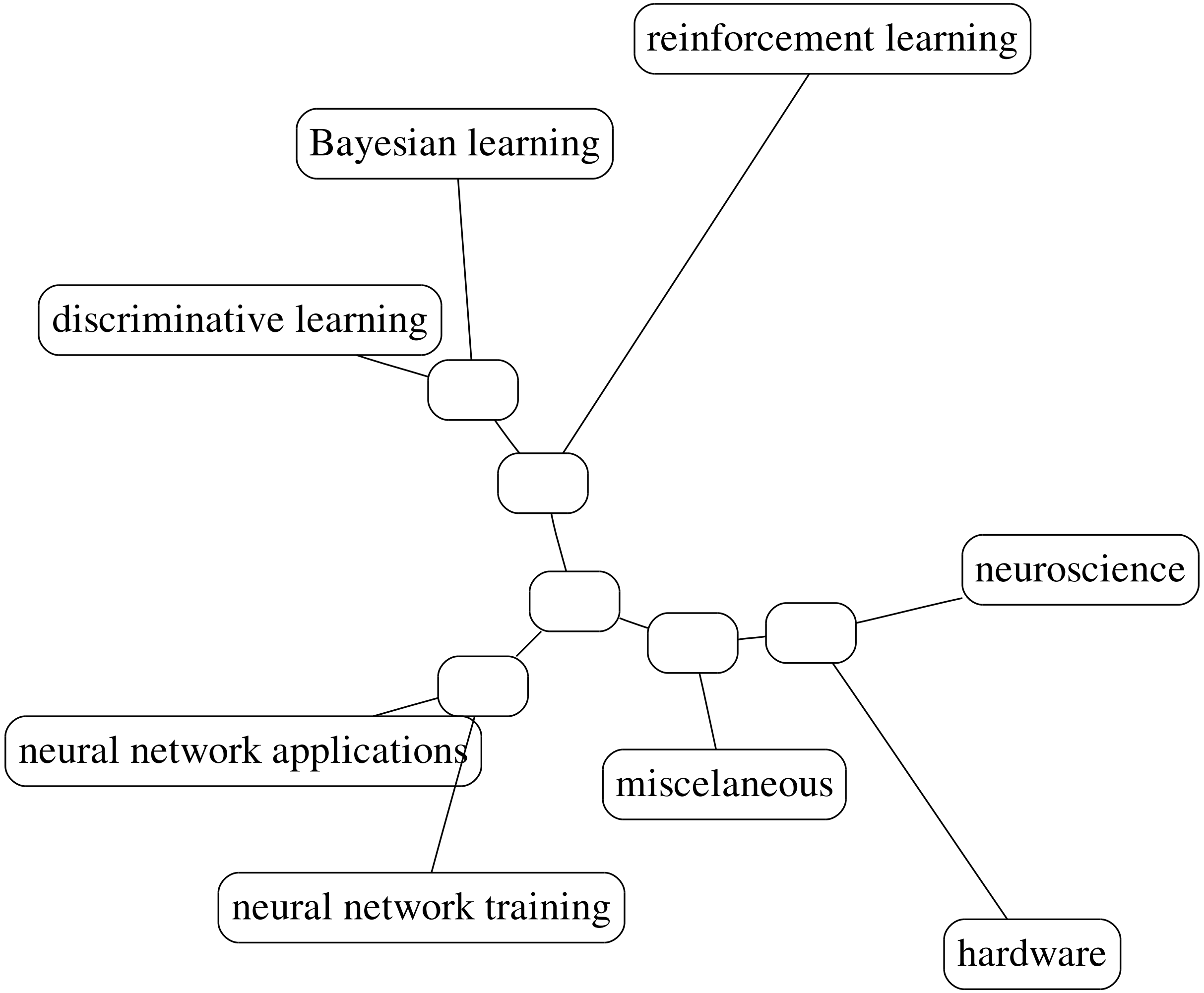
- Simultaneous clustering and taxonomy fitting
  → Numerical Taxonomy Clustering [NIPS08b]
- Maximise dependence (HSIC) between data and clusters



	A	B	C	D	E
A	0	2	7	4	7
B		0	7	4	7
C			0	7	6
D				0	7
E					0



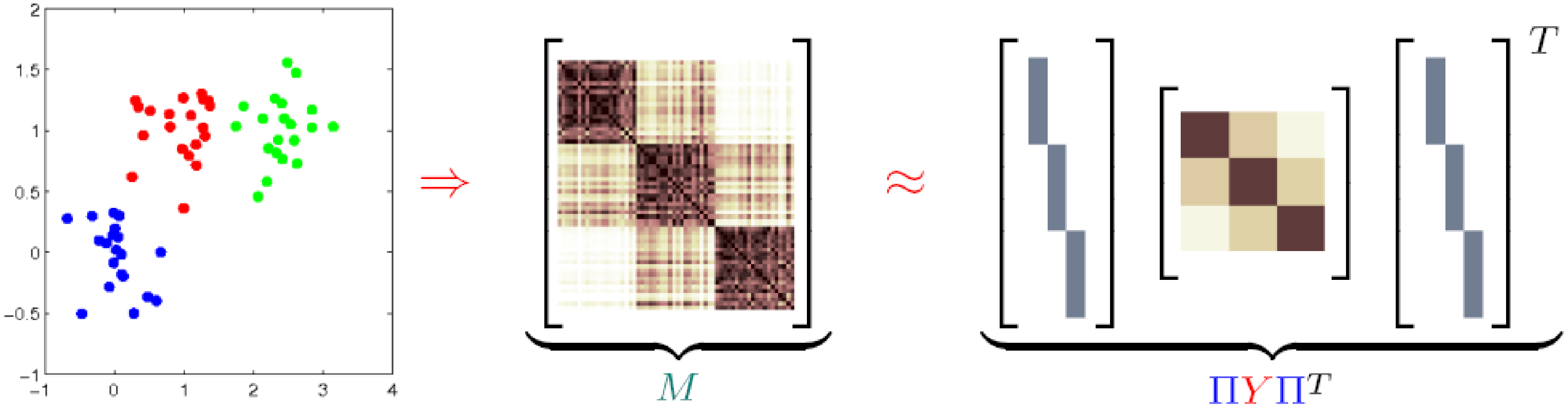
NIPS Articles



The taxonomy discovered for the NIPS dataset.

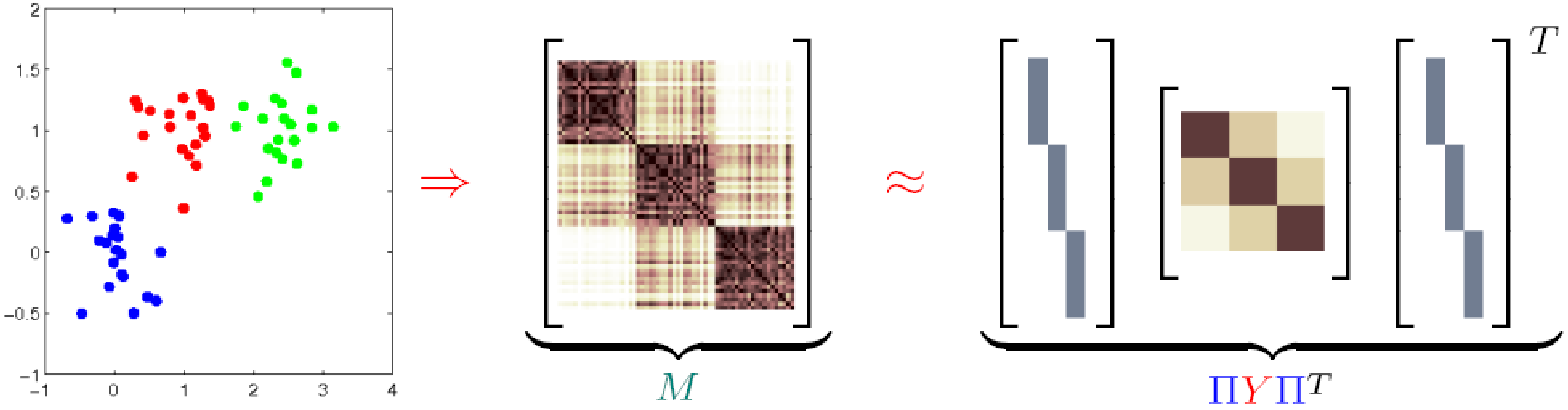
Dependence Maximization

Idea:



Dependence Maximization

Idea:



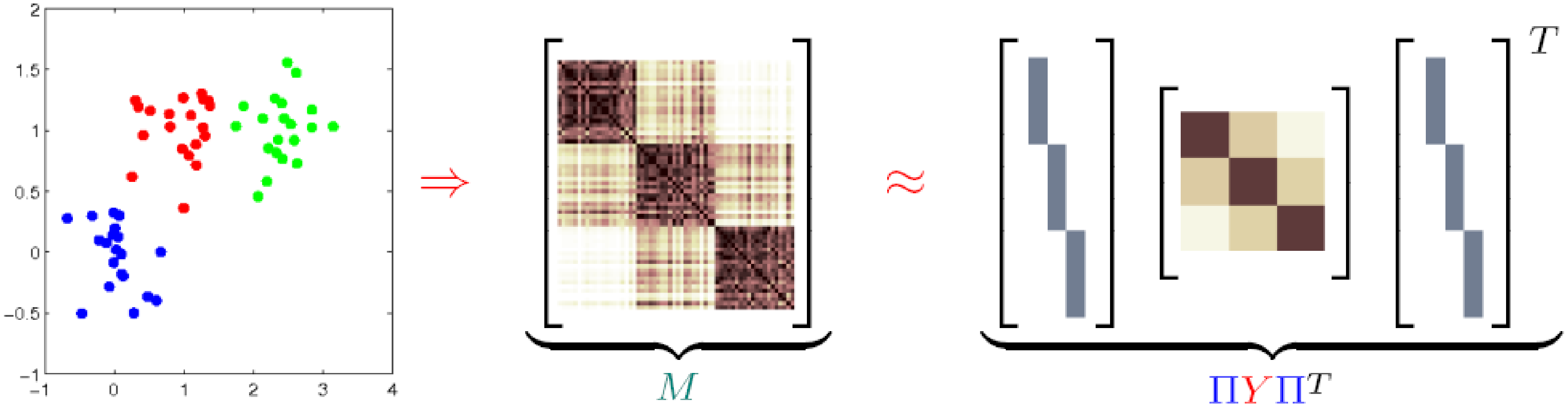
Objective:

$$\max_{Y,\Pi} \frac{\operatorname{Tr}\left[M H \Pi Y \Pi^T H\right]}{\|H \Pi Y \Pi^T H\|_{\mathrm{HS}}}.$$

- Data kernel matrix: M
- Π is $n \times k$ cluster assignment matrix, $\Pi 1 = 1$, $\Pi_{i,j} \in \{0,1\}$.
- $Y \succeq \mathbf{0}$ Gram matrix between clusters

Dependence Maximization

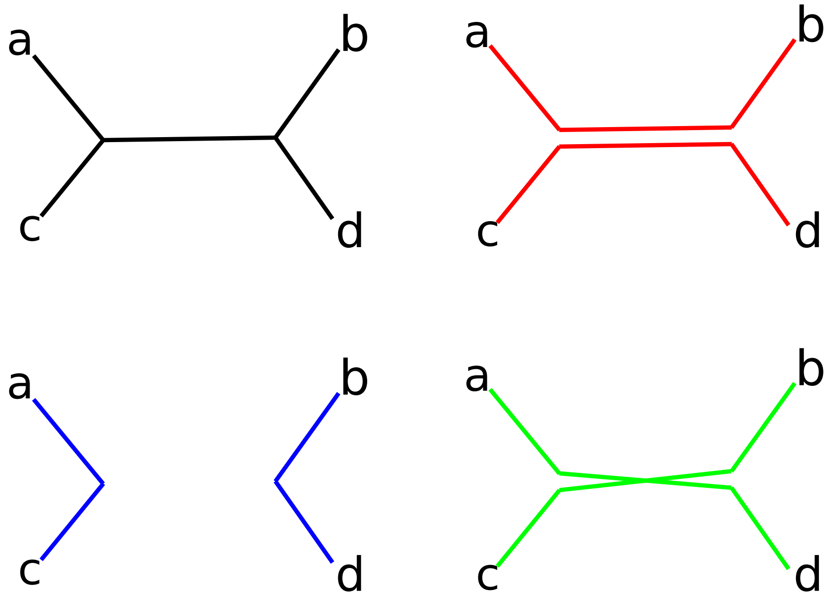
Idea:



Y has no prior structure

- Add constraints to Y
  - Change Y^* → interpretability
  - Change Π^* → improved clustering

Numerical Taxonomy

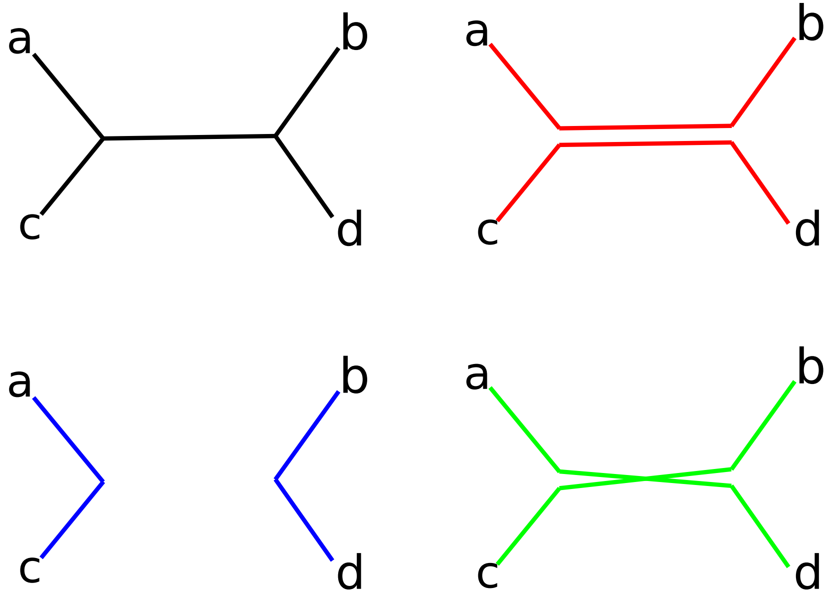


- compute distance matrix, D
- $D_{ij} = \sqrt{Y_{ii} + Y_{jj} - 2Y_{ij}}$

- Four point condition:
- $D_{ab} + D_{cd} \leq \max\left(D_{ac} + D_{bd}, D_{ad} + D_{bc}\right) \quad \forall a, b, c, d$

Numerical Taxonomy

- compute distance matrix, D
- $D_{ij} = \sqrt{Y_{ii} + Y_{jj} - 2Y_{ij}}$



- Four point condition:
- $D_{ab} + D_{cd} \leq \max\left(D_{ac} + D_{bd}, D_{ad} + D_{bc}\right) \quad \forall a, b, c, d$
- Numerical taxonomy objective: $\min_{D_T} \|D - D_T\|^2$ where D_T is subject to the four point condition (NP hard, so approximation only) [Harb et al., 2005]
- From D_T to tree [Waterman et al., 1977]

Numerical Taxonomy Clustering

```
Require: $M \succeq \mathbf{0}$
Ensure: $(\Pi, Y) \approx (\Pi^*, Y^*)$ that max dependence s.t. 4-point condition
  Initialize $Y = I$
  Initialize $\Pi$ using spectral clustering
  while Convergence has not been reached do
    Solve for $Y$ given $\Pi$ using closed form solution
    Construct $D$ such that $D_{ij} = \sqrt{Y_{ii} + Y_{jj} - 2Y_{ij}}$
    Solve for $\min_{D_T} \|D - D_T\|^2$
    Assign $Y = -\frac{1}{2}H(D_T \odot D_T)H$ (Hadamard product, next slide)
    Update $\Pi$ by changing labels to increase score [ICML07b]
  end while
```

Numerical Taxonomy Clustering

Given a matrix of pairwise distances, D_T, we recover a centred kernel matrix,

$$HKH = H\left(D_T \circ D_T\right)H,$$

where $D_T \circ D_T$ denotes the Hadamard (entrywise) product.

Proof:

$$\begin{aligned} d^2(x_i, x_j) &= \|\phi(x_i) - \phi(x_j)\|^2 \\ &= k(x_i, x_i) + k(x_j, x_j) - 2k(x_i, x_j). \end{aligned}$$

Thus

$$k(x_i, x_j) = \frac{1}{2}\left(k(x_i, x_i) + k(x_j, x_j) - d_T^2(x_i, x_j)\right).$$

Numerical Taxonomy Clustering

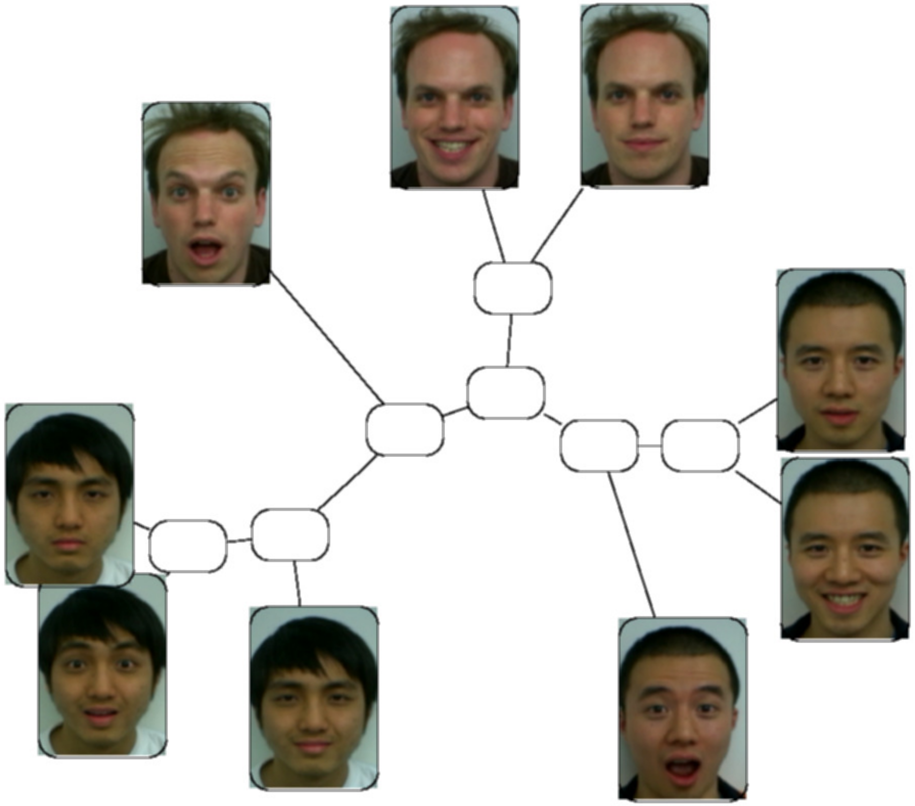
Writing this in matrix form,

$$K = \frac{1}{2}\left(\begin{bmatrix} \dots & k(x_1, x_1) & \dots \\ & \vdots & \\ \dots & k(x_m, x_m) & \dots \end{bmatrix} + \begin{bmatrix} \vdots & & \vdots \\ k(x_1, x_1) & \dots & k(x_m, x_m) \\ \vdots & & \vdots \end{bmatrix} - D_T \circ D_T \right).$$

Next, we use

$$H \begin{bmatrix} \dots & k(x_1, x_1) & \dots \\ & \vdots & \\ \dots & k(x_m, x_m) & \dots \end{bmatrix} = 0, \qquad \begin{bmatrix} \vdots & & \vdots \\ k(x_1, x_1) & \dots & k(x_m, x_m) \\ \vdots & & \vdots \end{bmatrix} H = 0,$$

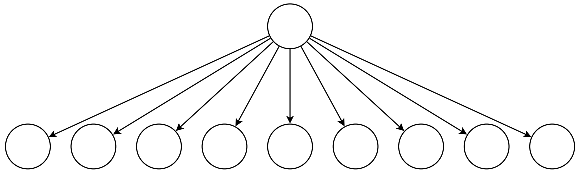
Attractive Scientist Dataset (1)



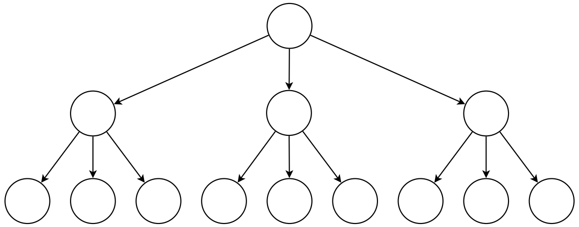
Face dataset and taxonomy discovered by the algorithm

Attractive Scientist Dataset (2)

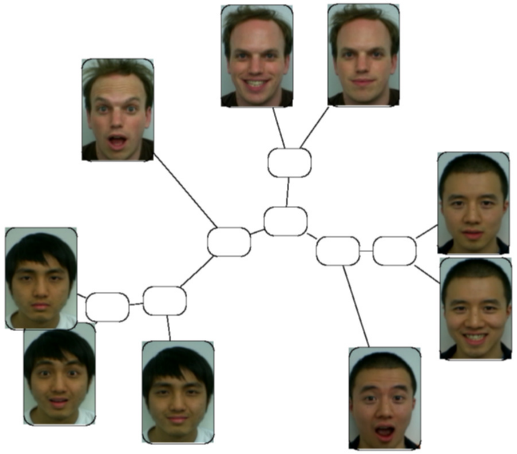
Conditional entropy scores for clusterings using [ICML07b]



flat (0.5180)

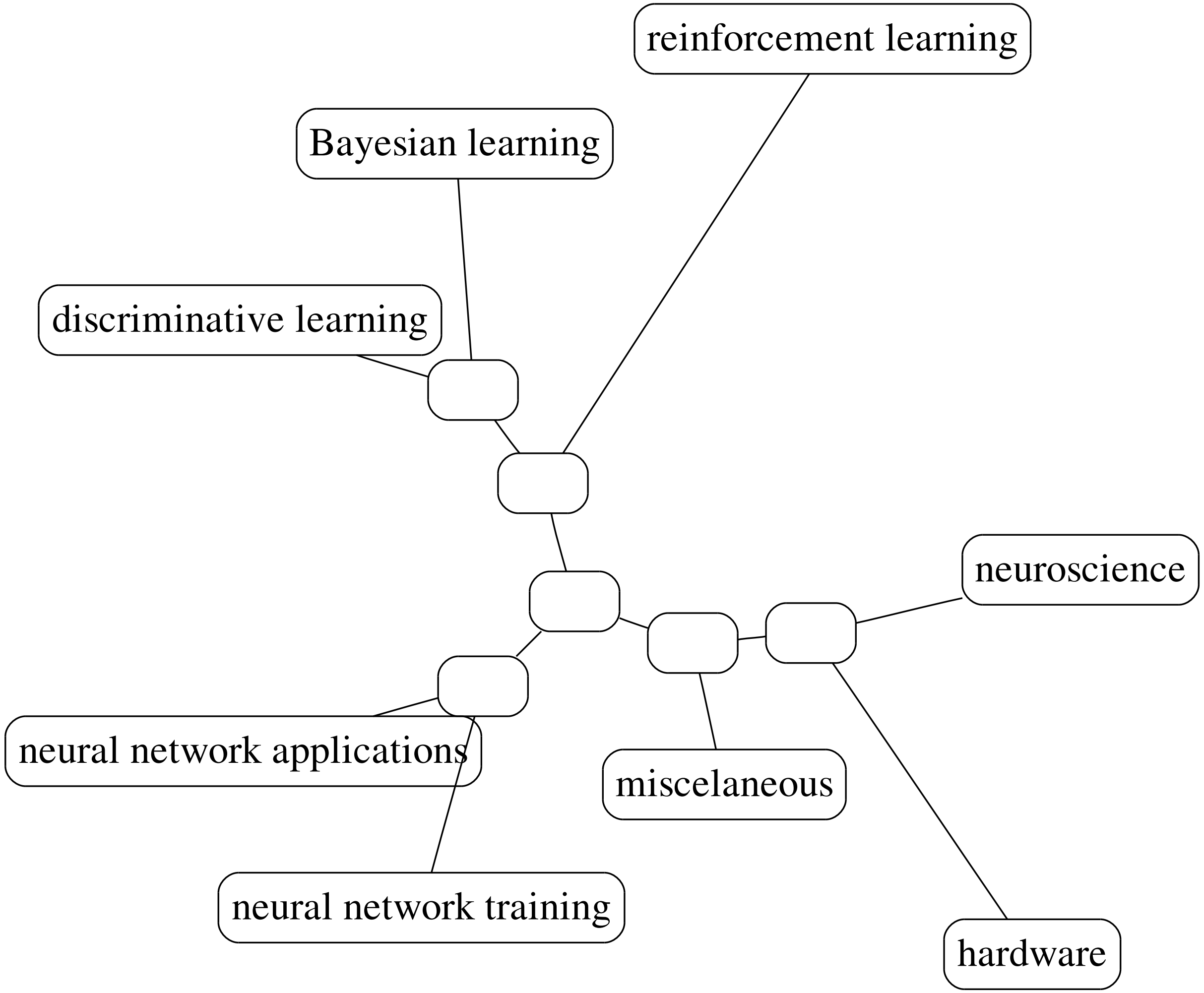


hierarchy (0.4970)



taxonomy (**0.2807**)

NIPS Articles



The taxonomy discovered for the NIPS dataset.

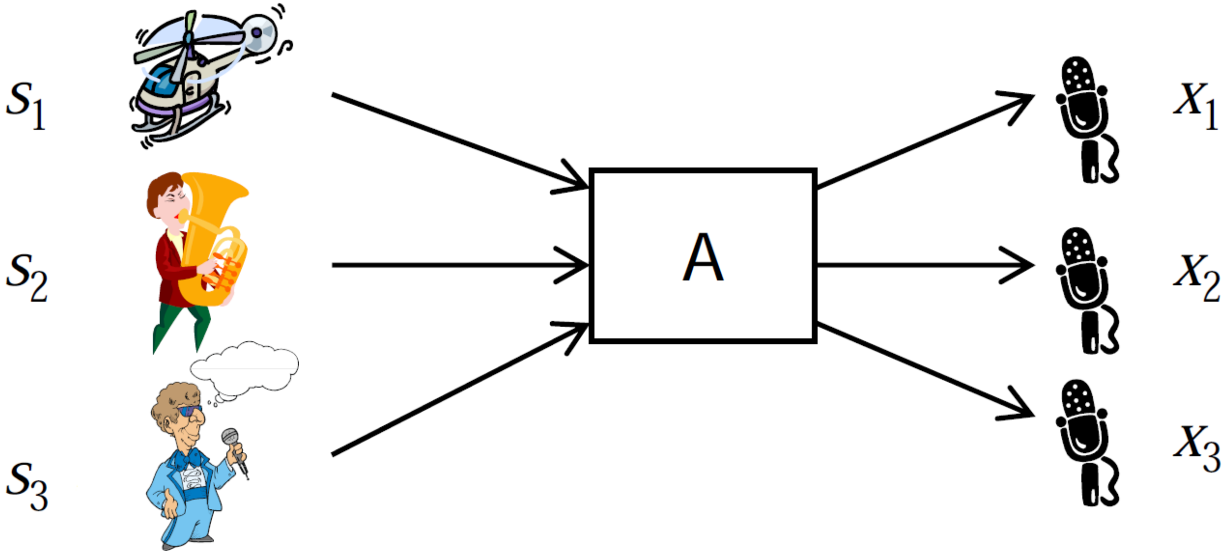
NIPS Articles: Categories

neurosci.	hardware	misc.	train-neural	app.-neural	reinforcement	discriminative	Bayesian
neurons	chip	memory	network	training	state	function	data
cells	circuit	dynamics	units	recognition	learning	error	model
model	analog	image	learning	network	policy	algorithm	models
cell	voltage	neural	hidden	speech	action	functions	distribution
visual	current	hopfield	networks	set	reinforcement	learning	gaussian
neuron	figure	control	input	word	optimal	theorem	likelihood
activity	vlsi	system	training	performance	control	class	parameters
synaptic	neuron	inverse	output	neural	function	linear	algorithm
response	output	energy	unit	networks	time	examples	mixture
firing	circuits	capacity	weights	trained	states	case	em
cortex	synapse	object	error	classification	actions	training	bayesian
stimulus	motion	field	weight	layer	agent	vector	posterior
spike	pulse	motor	neural	input	algorithm	bound	probability
cortical	neural	computational	layer	system	reward	generalization	density
frequency	input	network	recurrent	features	sutton	set	variables
orientation	digital	images	net	test	goal	approximation	prior
motion	gate	subjects	time	classifier	dynamic	bounds	log
direction	cmos	model	back	classifiers	step	loss	approach
spatial	silicon	associative	propagation	feature	programming	algorithms	matrix
excitatory	implementation	attractor	number	image	rl	dimension	estimation

Application 3: ICA

ICA: setting

Independent component analysis:

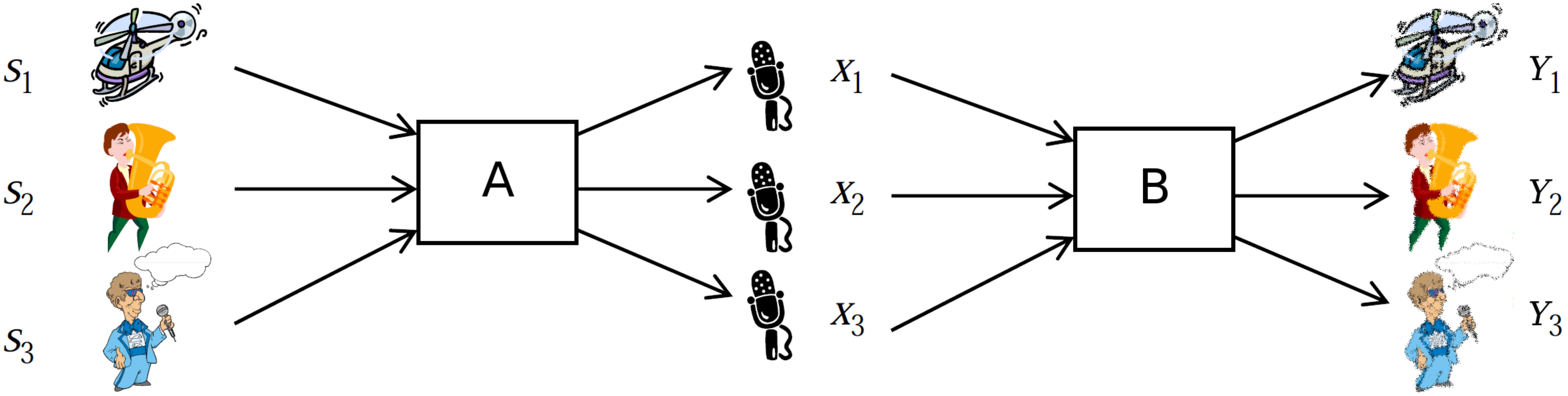


$$S \quad \times A \quad = X$$

- $\mathbf{s}$ a vector of l unknown, independent sources: $\mathsf{P}_{\mathbf{s}} = \prod_{i=1}^{l} \mathsf{P}_{\mathbf{s}_i}$
- $\mathbf{x}$ vector of mixtures
- $\mathbf{A}$ is $l \times l$ mixing matrix (full rank)

ICA: setting

Independent component analysis:

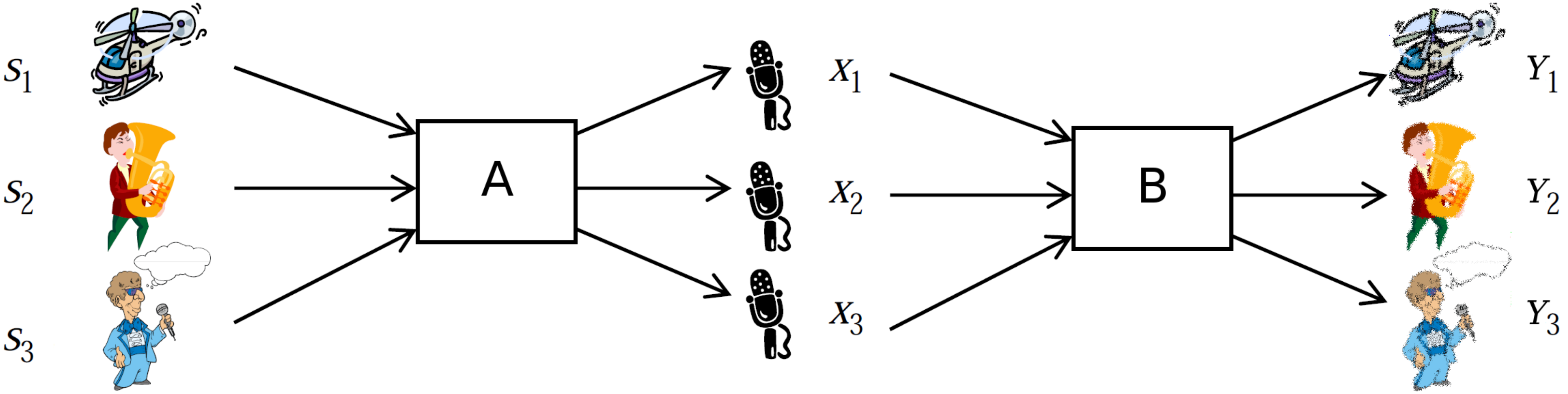


$$X \qquad \times B \quad = Y$$

- $\mathbf{B}$ is estimated $\mathbf{A}^{-1}$, we solve for this
- $\mathbf{y}$ vector of estimated sources

ICA: setting

Independent component analysis:



$$X \qquad \times B \quad = Y$$

- $\mathbf{B}$ is estimated $\mathbf{A}^{-1}$, we solve for this
- $\mathbf{y}$ vector of estimated sources

Neglect time dependence: m i.i.d. mixture observations

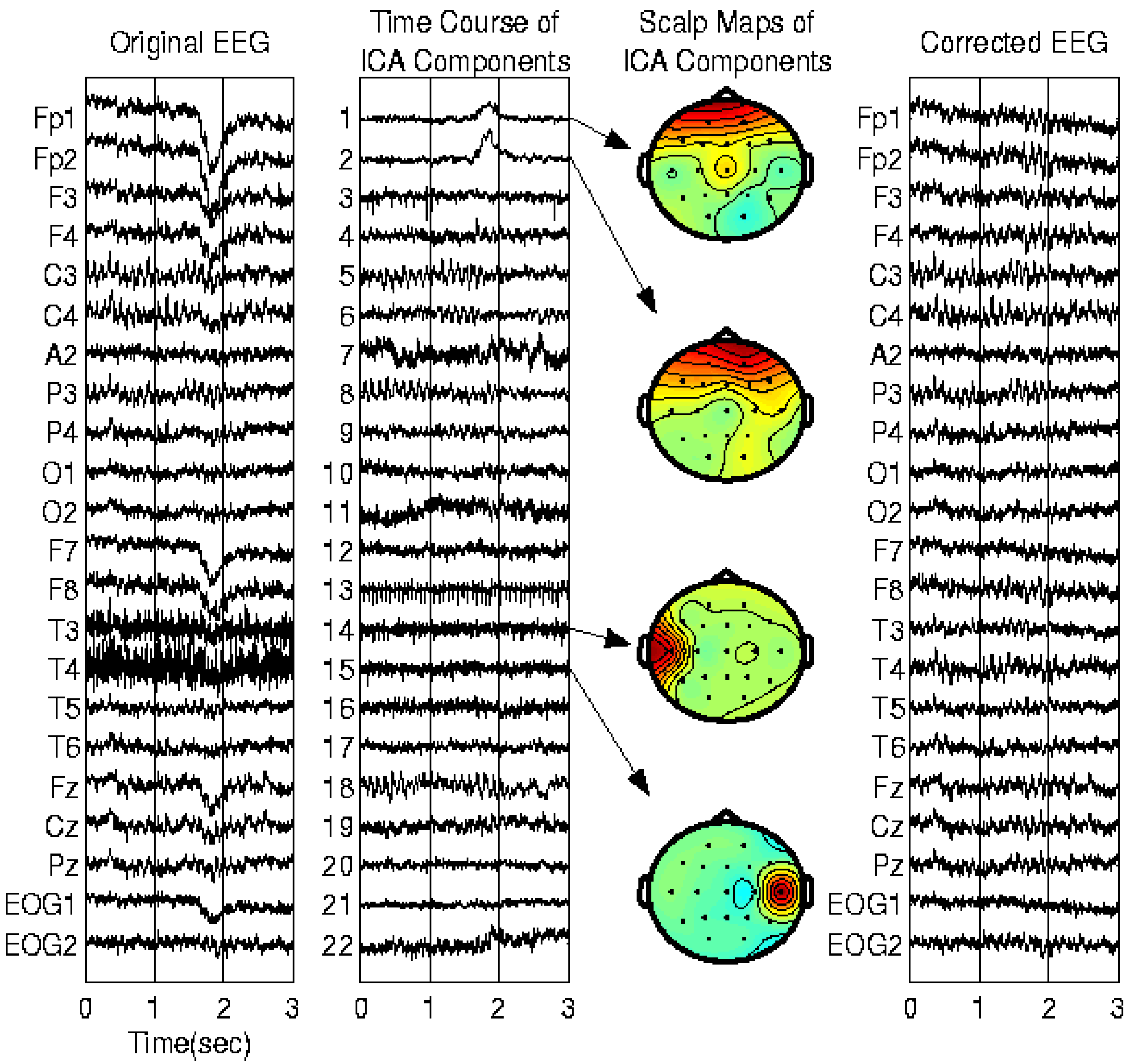
ICA: another example

- Mixtures X are original EEG

  [Jung et al., 2000]

- Estimated sources Y are ICA components

- Scalp map from B



ICA examples

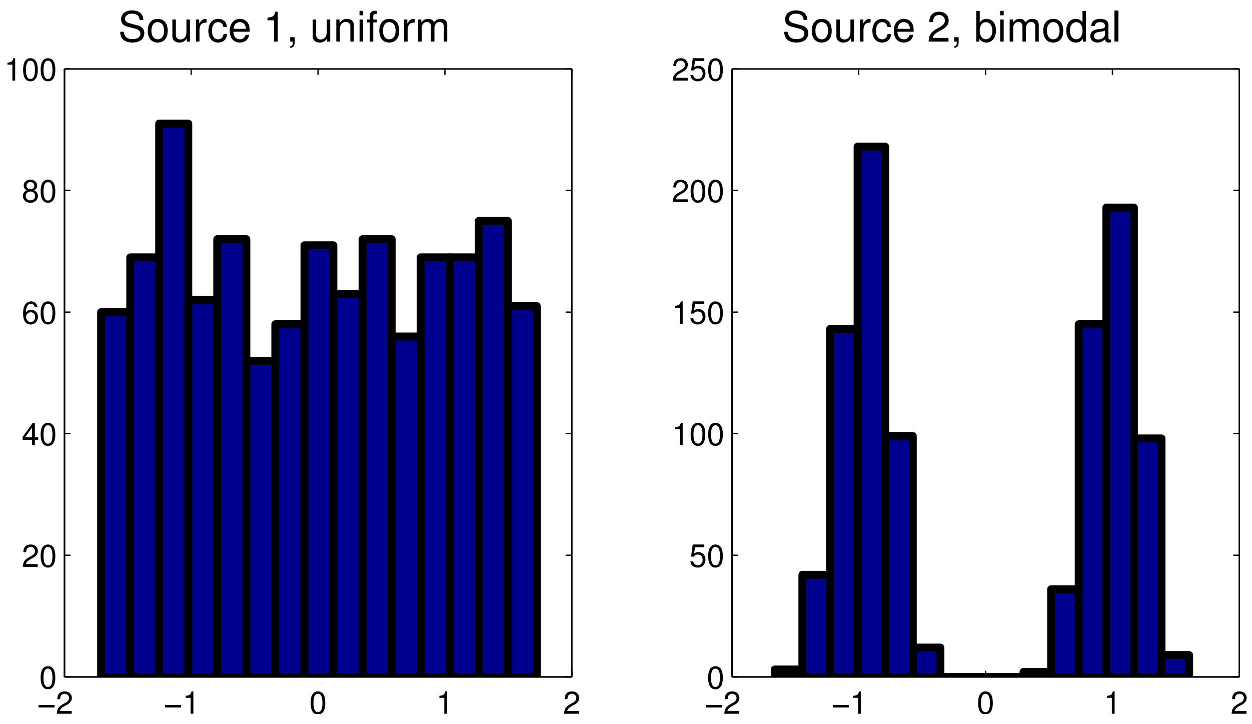
- We've seen:
  - Sounds mixed together ("cocktail party" problem) [Hyvärinen et al., 2001]
  - EEG recordings (brain, fetal heartbeat) [Jung et al., 2000, Stögbauer et al., 2004]

Warning: both the above examples violate the assumptions made in ICA (that the observations at each time are independent and identically distributed).

- Some further examples:
  - Extracting independent activity from fMRI [Calhoun et al., 2003]
  - Financial data [Kiviluoto and Oja, 1998]
  - Linear edge filters for image patch coding? (Possibly not: [Bethge, 2006])

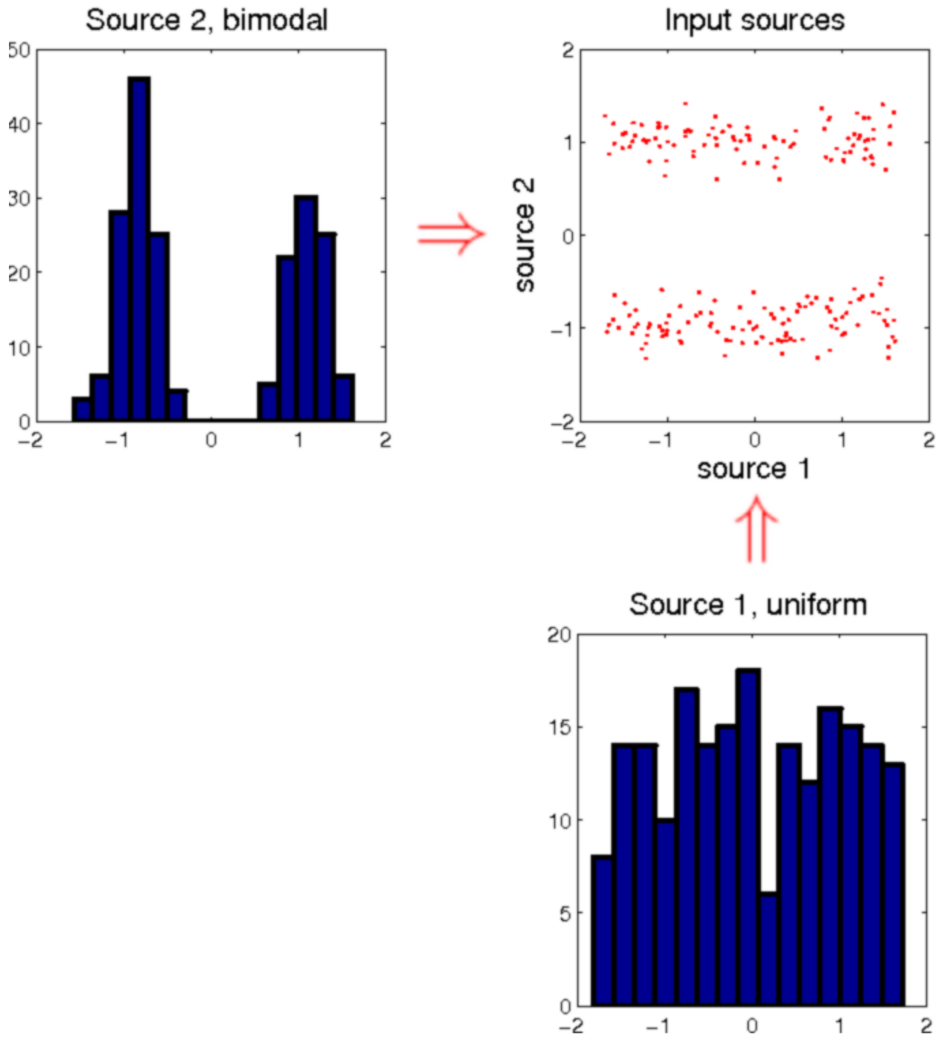
A toy example

- Two distributions: $\mathbf{P}_{\mathbf{s}_1}$ is uniform, $\mathbf{P}_{\mathbf{s}_2}$ is bimodal



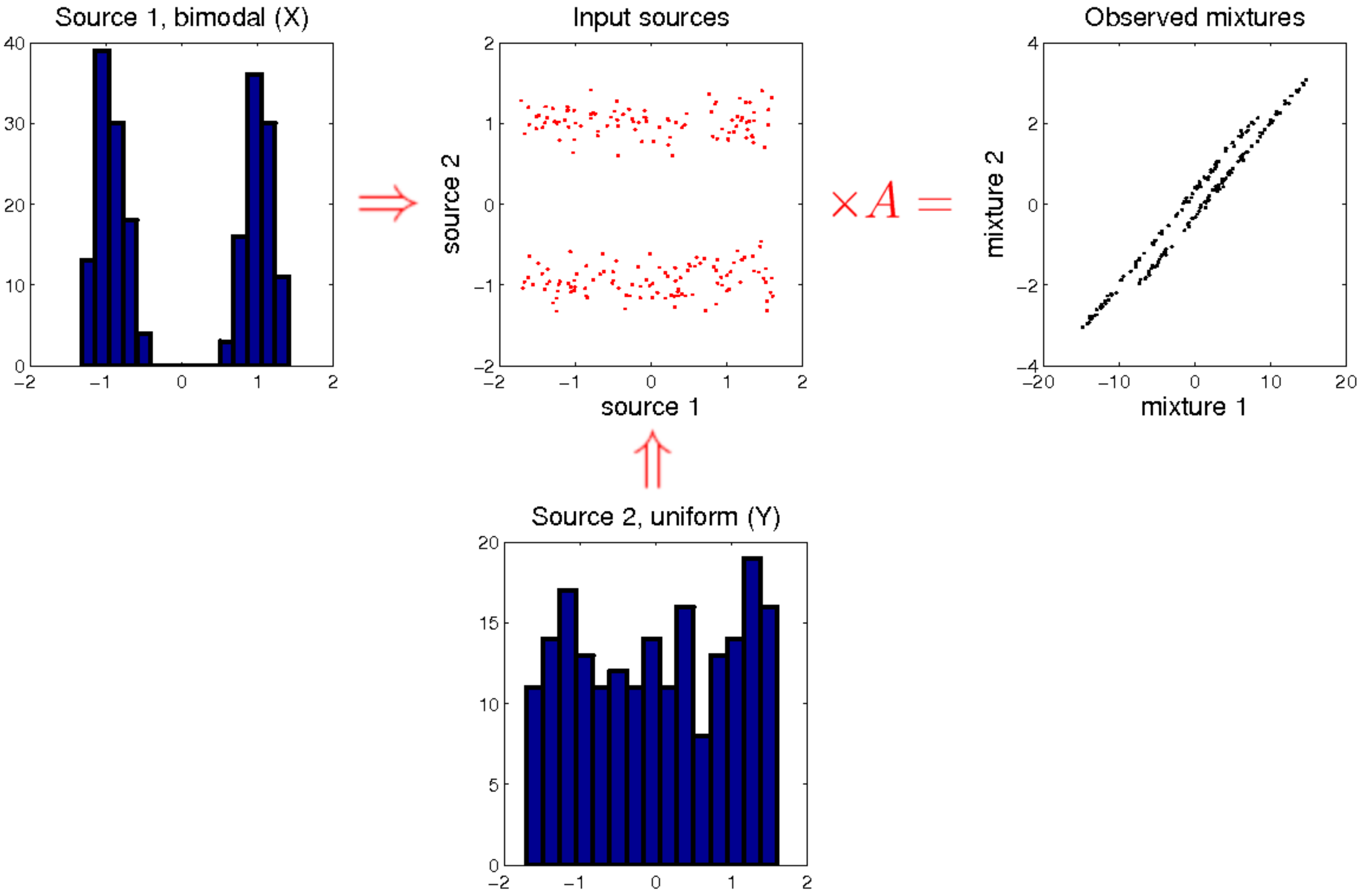
A toy example

- Two distributions: $\mathbf{P}_{\mathbf{s}_1}$ is uniform, $\mathbf{P}_{\mathbf{s}_2}$ is bimodal



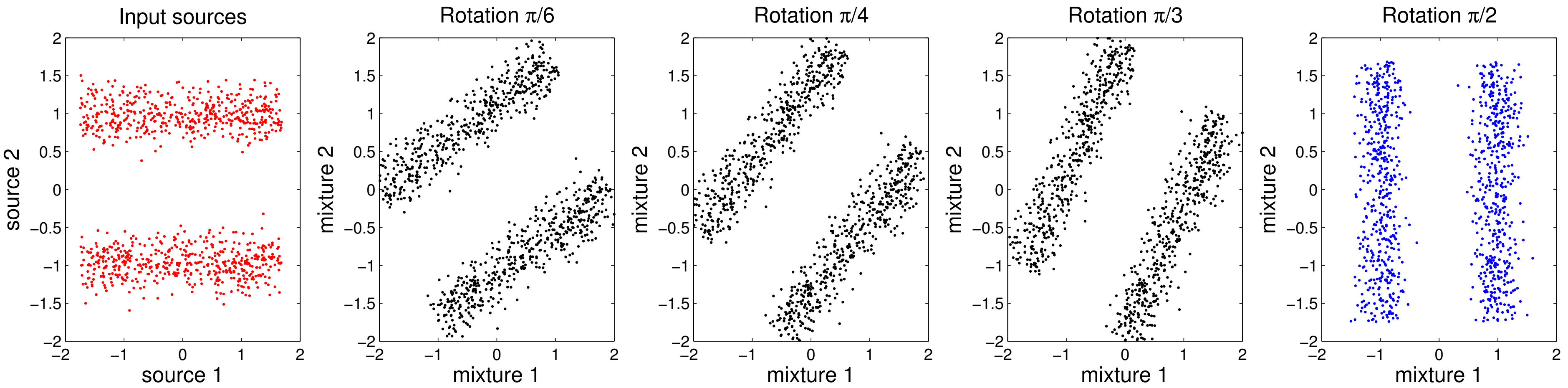
A toy example

- Two distributions: $\mathbf{P}_{\mathbf{s}_1}$ is uniform, $\mathbf{P}_{\mathbf{s}_2}$ is bimodal



First indeterminacy: ordering

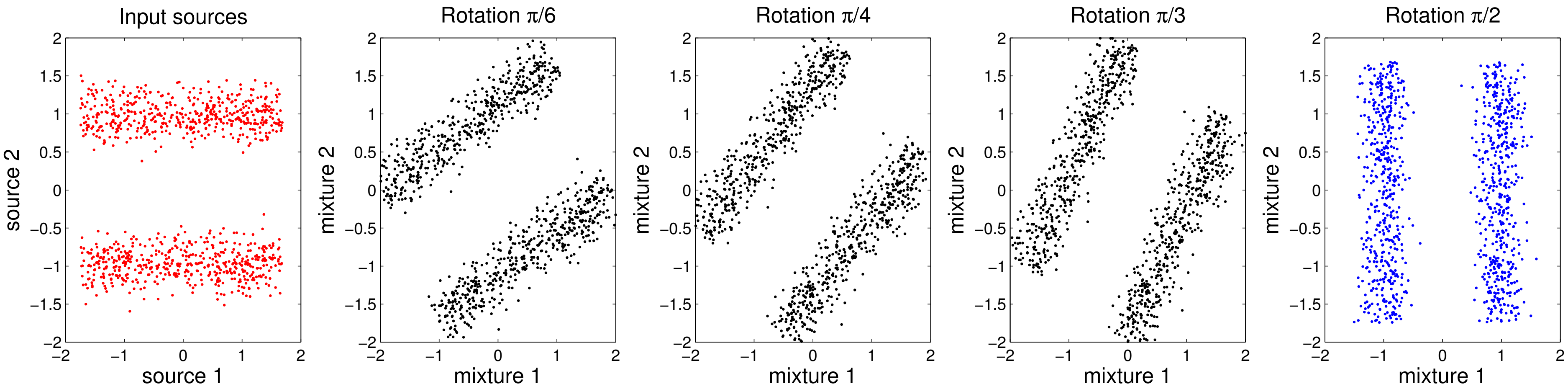
- Initial unmixed RVs in red



- Independent at rotation $\pi/2$

First indeterminacy: ordering

- Initial unmixed RVs in red

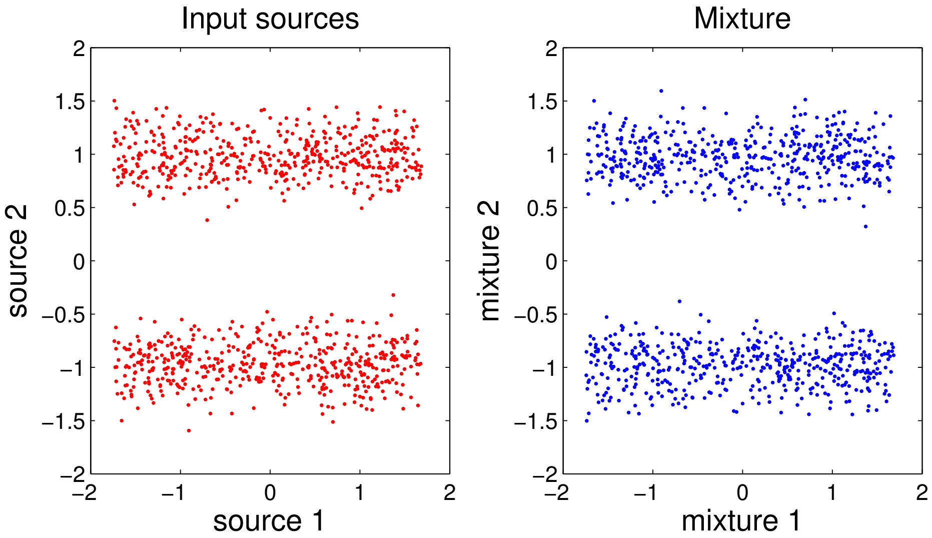


- Independent at rotation $\pi/2$

$$\boxed{\text{Ignore source order}}$$

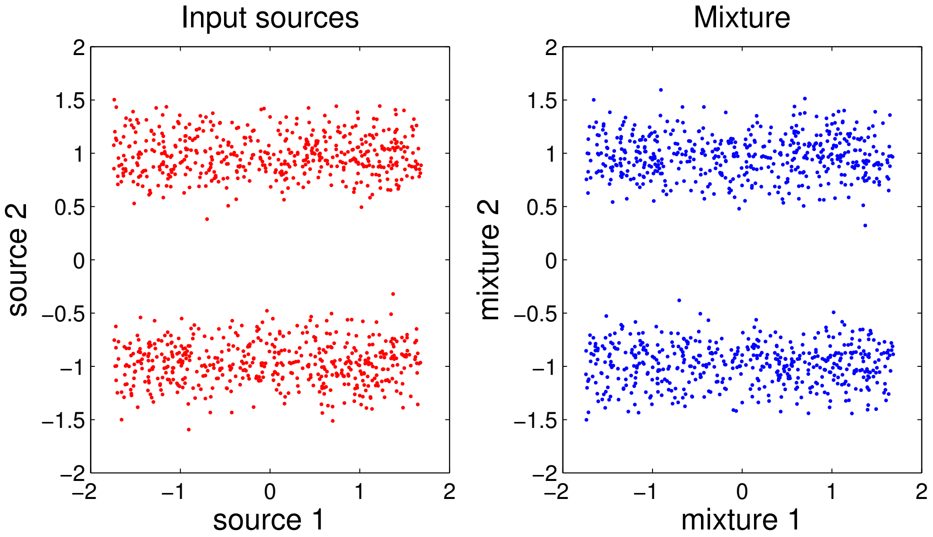
Second indeterminacy: sign

- Initial unmixed RVs in red
- Source 2 sign reversed in blue



Second indeterminacy: sign

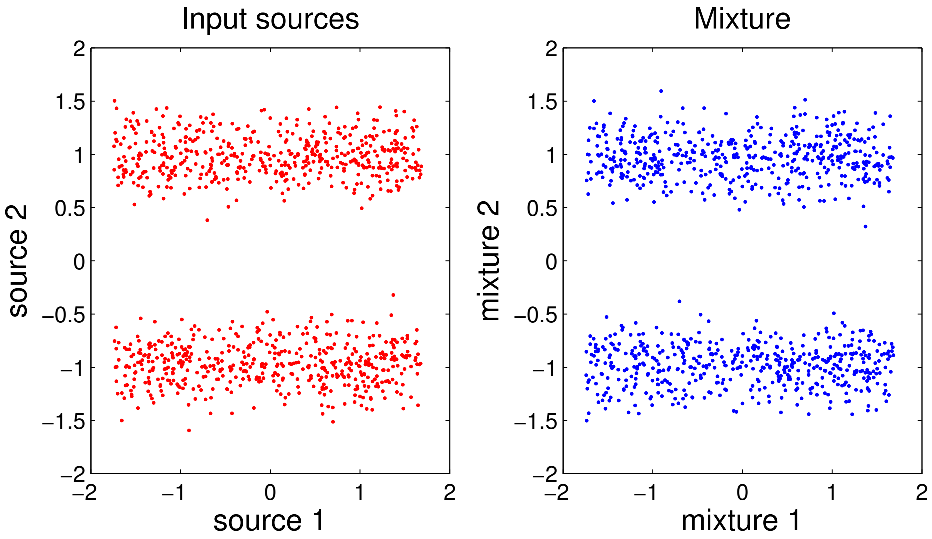
- Initial unmixed RVs in red
- Source 2 sign reversed in blue



Ignore source sign

Second indeterminacy: sign

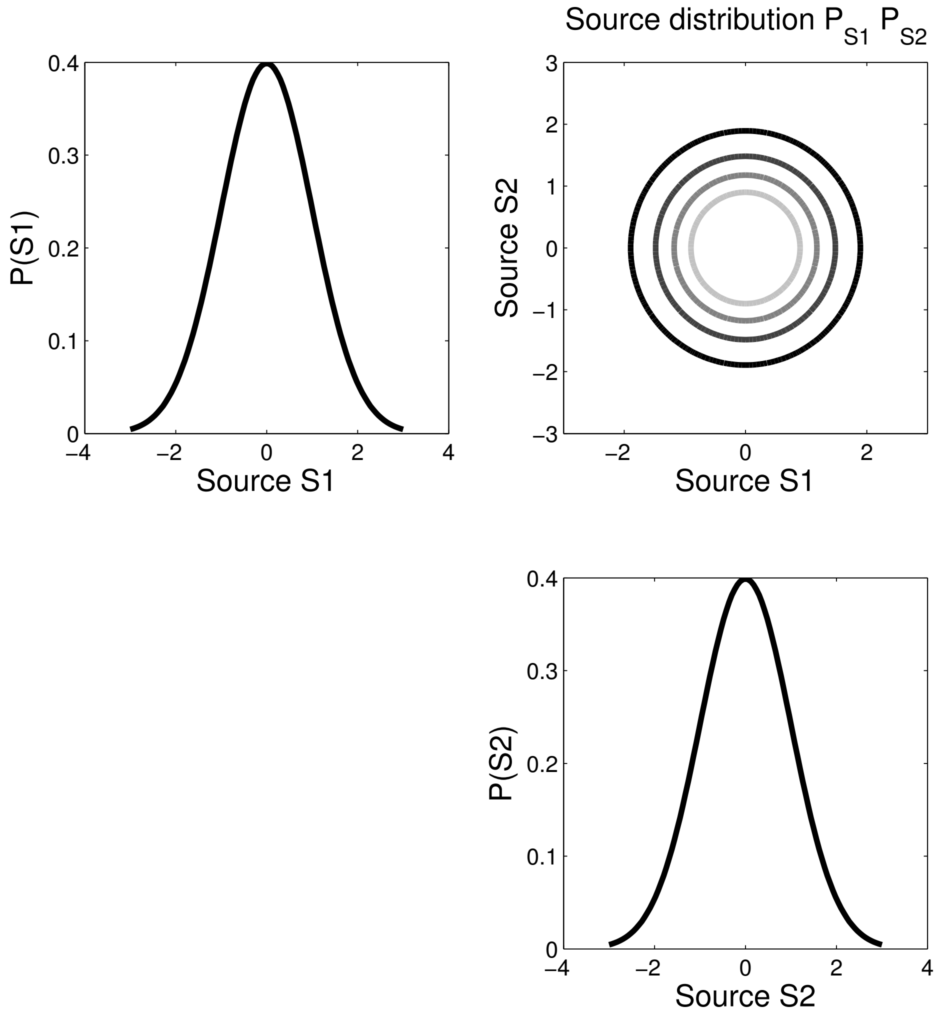
- Initial unmixed RVs in red
- Source 2 sign reversed in blue



- More generally: S_1 and S_2 independent iff aS_1 and S_2 independent for $a \neq 0$
  - Assume sources have unit variance

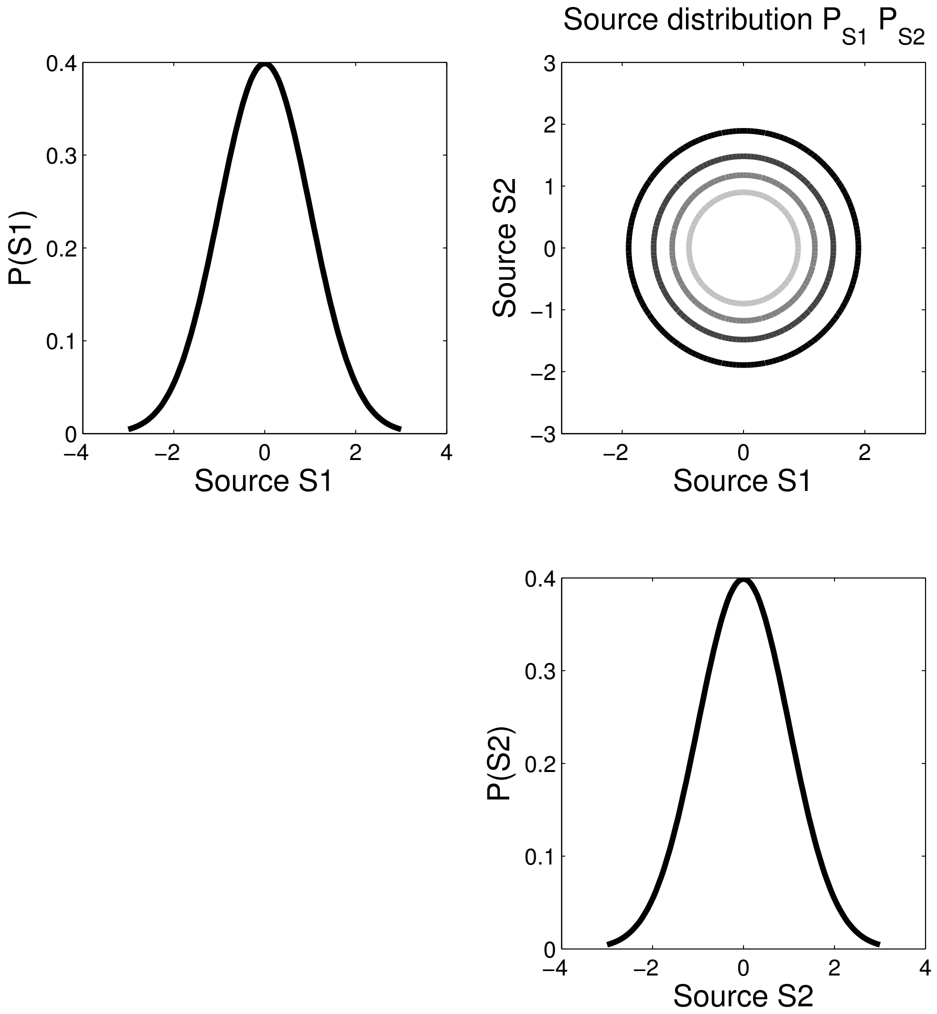
Third indeterminacy: Gaussians

Both sources Gaussian



Third indeterminacy: Gaussians

Both sources Gaussian



Meaningless to “unmix” Gaussians

Things that are impossible for ICA

Using independence alone, we cannot . . .

- recover signal order,
- recover signal sign (or amplitude) ,
- separate multiple Gaussians.

Things that are impossible for ICA

Using independence alone, we cannot …

- recover signal order,
- recover signal sign (or amplitude) ,
- separate multiple Gaussians.

We can recover

$$B^* = PDA^{-1}$$

- P is a permutation matrix
- D diagonal, $d_{ii} \in \{-1, 1\}$

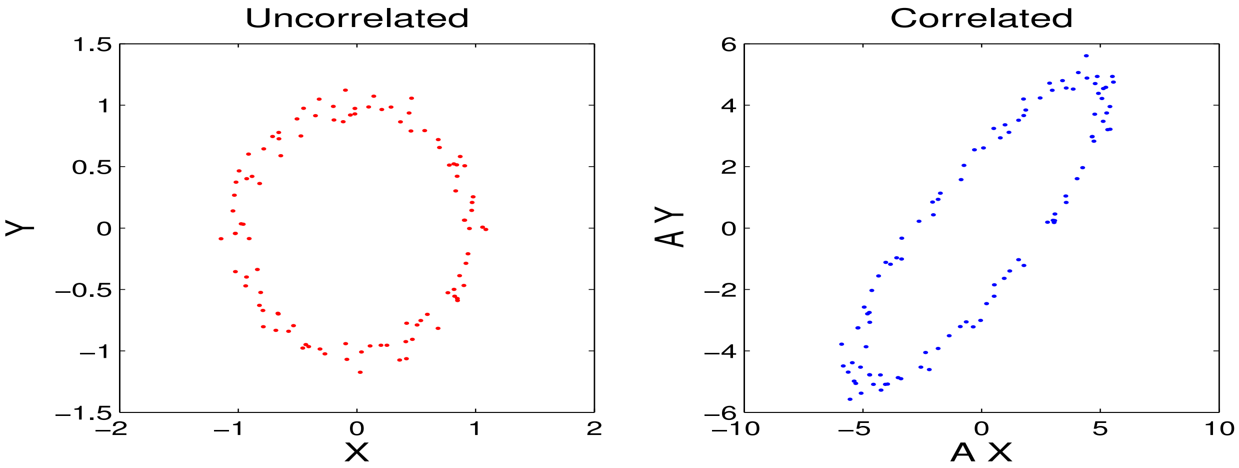
(as long as no more than one Gaussian source)

First step in ICA: decorrelate

- Idea: remove all dependencies of order 2 between mixtures $\mathbf{x}$

First step in ICA: decorrelate

- Idea: remove all dependencies of order 2 between mixtures $\mathbf{x}$



First step in ICA: decorrelate

- Idea: remove all dependencies of order 2 between mixtures $\mathbf{x}$
- New signals have unit covariance:

$$\mathbf{t} = \mathbf{B}_w \mathbf{x} \qquad\qquad \mathbf{C}_t = \mathbf{I}$$

- We thus break up $\mathbf{B}$ as follows:

$$\mathbf{B} = \mathbf{B}_r \mathbf{B}_w$$

  - $\mathbf{B}_w$ is a whitening matrix
  - $\mathbf{B}_r$ is remaining demixing operation
- Use the SVD of mixture covariance $\mathbf{C}_x = \mathbf{U}\Lambda\mathbf{U}^\top$:

$$\mathbf{B}_w = \Lambda^{-1/2}\mathbf{U}^\top$$

First step in ICA: decorrelate

Write C_y (size $l \times l$) as the covariance of $\mathbf{t}$.

$$C_t = m^{-1} T T^\top \qquad \text{where} \qquad T = \mathbf{B}_w X$$

We want to ensure

$$\begin{aligned} I &= C_t \\ &= m^{-1} \mathbf{B}_w X X^\top \mathbf{B}_w{}^\top \\ &= \mathbf{B}_w C_x \mathbf{B}_w{}^\top \end{aligned}$$

First step in ICA: decorrelate

Write C_y (size $l \times l$) as the covariance of **t**.

$$C_t = m^{-1} T T^\top \qquad \text{where} \qquad T = \mathbf{B}_w X$$

We want to ensure

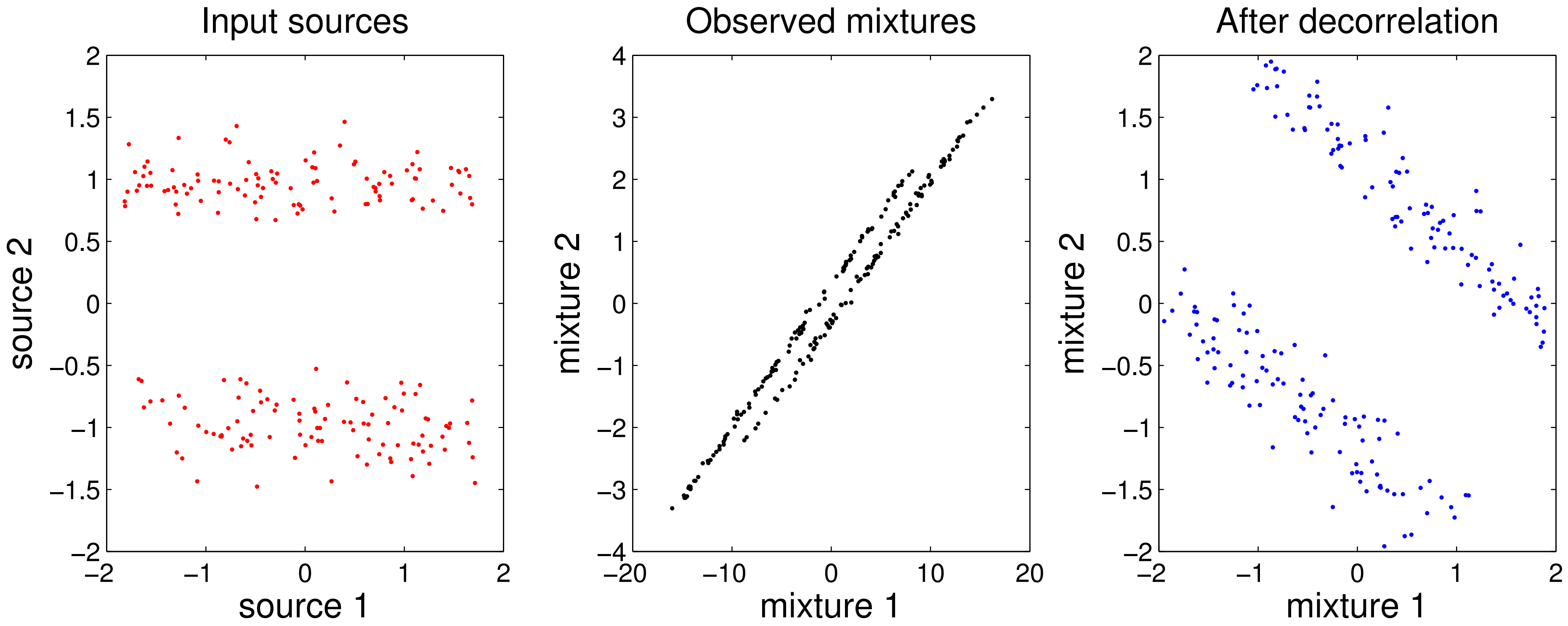
$$\begin{aligned} I &= C_t \\ &= m^{-1} \mathbf{B}_w X X^\top \mathbf{B}_w{}^\top \\ &= \mathbf{B}_w C_x \mathbf{B}_w{}^\top \end{aligned}$$

Write the SVD of $C_x = U \Lambda U^\top$. Write $\mathbf{B}_w = \Lambda^{-1/2} U^\top$. Then

$$\begin{aligned} C_t &= \Lambda^{-1/2} U^\top C_x U \Lambda^{-1/2} \\ &= \Lambda^{-1/2} U^\top U \Lambda U^\top U \Lambda^{-1/2} \\ &= I \end{aligned}$$

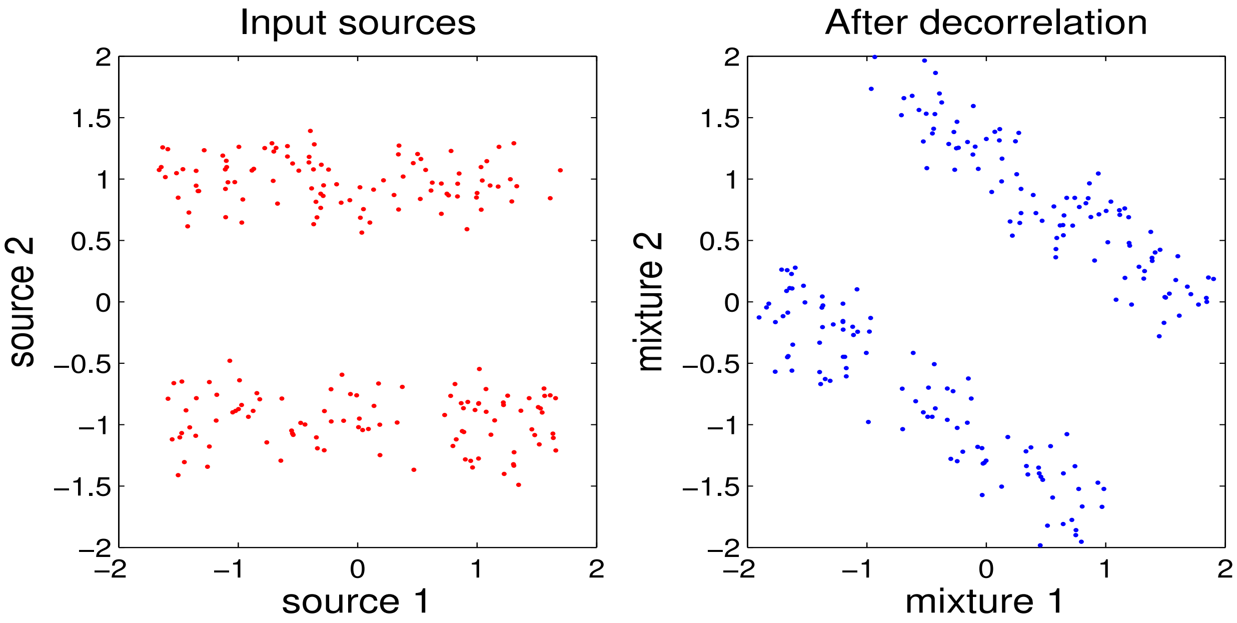
What does decorrelation achieve?

- Two distributions: $\mathbf{P}_{\mathbf{s}_1}$ is uniform, $\mathbf{P}_{\mathbf{s}_2}$ is bimodal



Problem remaining: *rotation*

- Assume correlation has already been removed
- To recover original signal, need to rotate



- In remainder: unmixing matrix $\mathbf{B}$ is rotation,

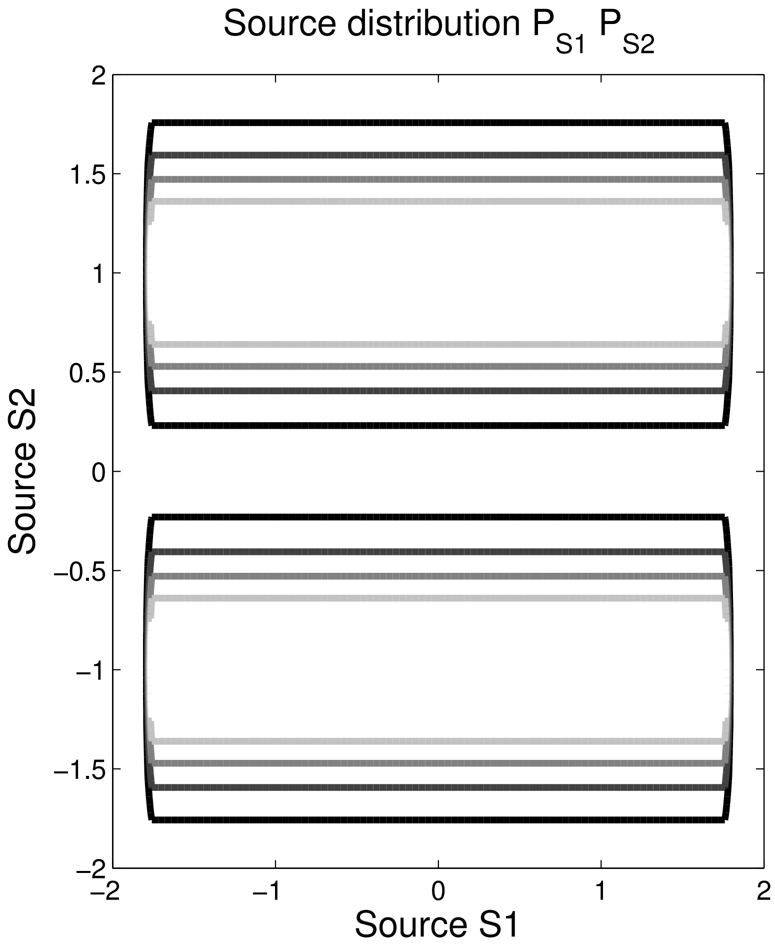
$$\mathbf{B}^\top \mathbf{B} = \mathbf{I}$$

ICA: maximum likelihood

- “ICA” using model parametrised by $(\mathbf{B}, \hat{\mathsf{P}}_{\mathsf{s}})$
- Interpretation: assume we are given the source densities $\hat{\mathsf{P}}_{\mathsf{s}}$, so we only need to find $\mathbf{B}$.

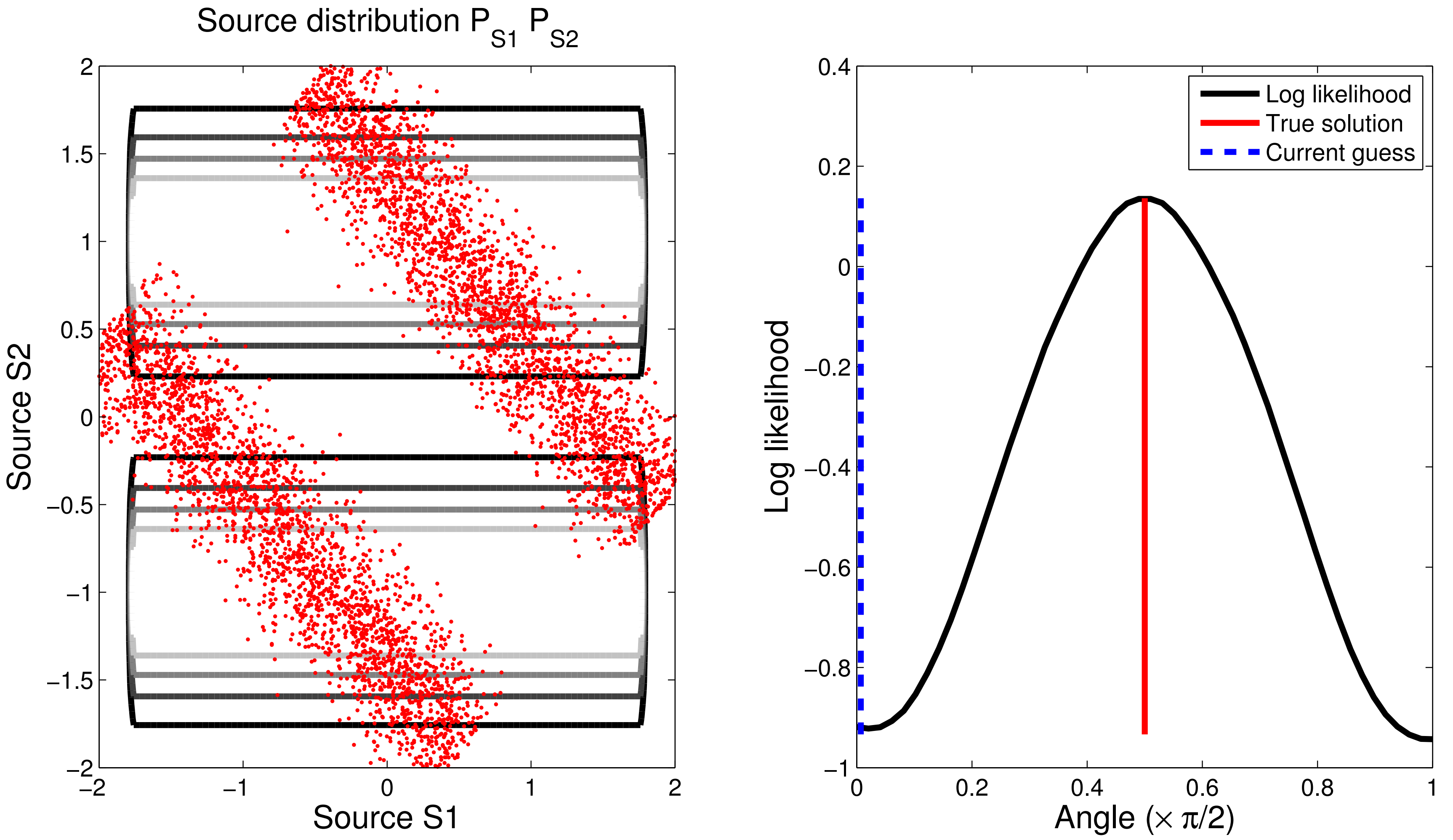
ICA: maximum likelihood

- "ICA" using model parametrised by $(\mathbf{B}, \hat{\mathbf{P}}_{\mathbf{s}})$



ICA: maximum likelihood

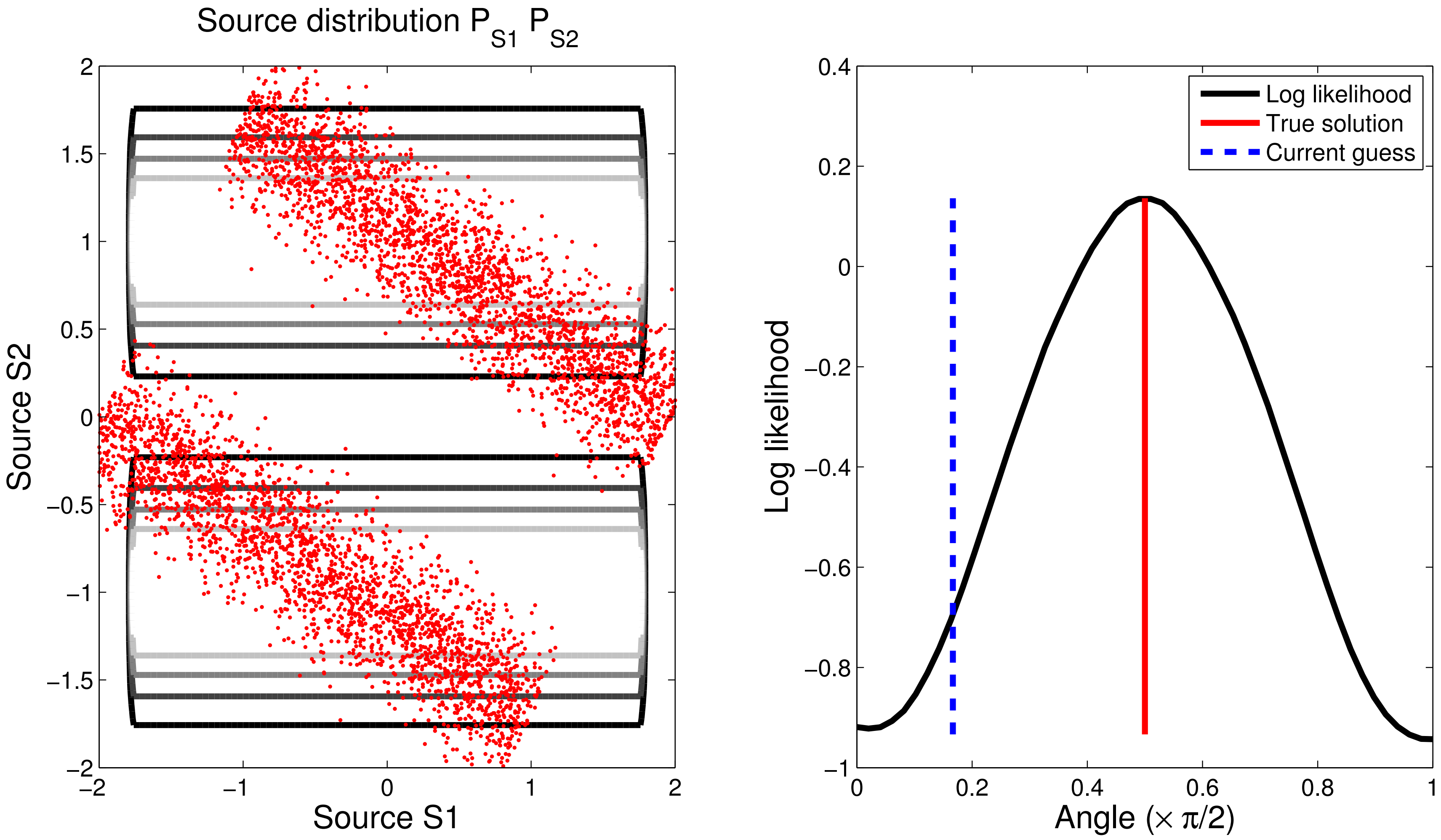
- "ICA" using model parametrised by $(\mathbf{B}, \hat{\mathbf{P}}_\mathbf{s})$



Unmixing angle for B: 0

ICA: maximum likelihood

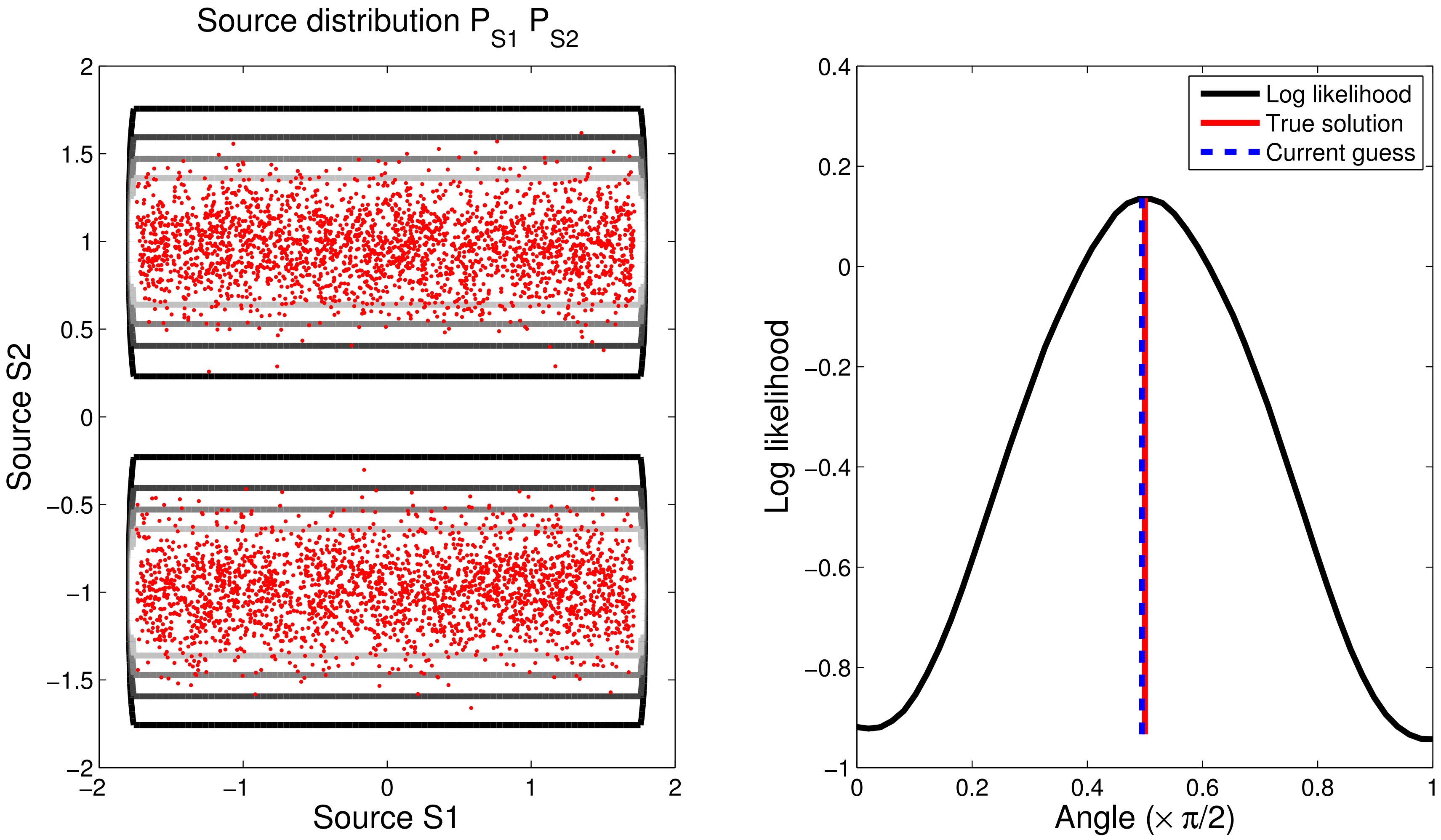
- "ICA" using model parametrised by $(\mathbf{B}, \hat{\mathbf{P}}_{\mathbf{s}})$



Unmixing angle for B: $\pi/12$

ICA: maximum likelihood

- “ICA” using model parametrised by $(\mathbf{B}, \hat{\mathbf{P}}_{\mathbf{s}})$



Unmixing angle for B: $\pi/4$

ICA: maximum likelihood

- We have a model for the observations, parametrised by $(\mathbf{B}, \hat{\mathsf{P}}_{\mathsf{s}})$
  - Model must have $\hat{\mathsf{P}}_{\mathsf{s}} = \prod_{i=1}^{l} \hat{\mathsf{P}}_{\mathsf{s}_i}$

ICA: maximum likelihood

- We have a model for the observations, parametrised by $(\mathbf{B}, \hat{\mathsf{P}}_{\mathsf{s}})$
  - Model must have $\hat{\mathsf{P}}_{\mathsf{s}} = \prod_{i=1}^{l} \hat{\mathsf{P}}_{\mathsf{s}_i}$
- We use the relation:

$$\begin{aligned} \mathbf{x} &= A\mathbf{s} \\ \mathsf{P}_{\mathsf{x}}(\mathbf{x}) &= \det(A^{-1})\mathsf{P}_{\mathsf{s}}(A^{-1}\mathbf{x}) \qquad (1) \end{aligned}$$

- Thus our **estimated** density of observations is

$$\hat{\mathsf{P}}_{\mathsf{x}} = \det(\mathbf{B})\, \hat{\mathsf{P}}_{\mathsf{s}}(\mathbf{B}\mathsf{x})$$

ICA: maximum likelihood

- We have a model for the observations, parametrised by $(\mathbf{B}, \hat{\mathsf{P}}_{\mathsf{s}})$
  - Model must have $\hat{\mathsf{P}}_{\mathsf{s}} = \prod_{i=1}^{l} \hat{\mathsf{P}}_{\mathsf{s}_i}$
- We use the relation:

$$\begin{aligned}
\mathbf{x} &= A\mathbf{s} \\
\mathsf{P}_{\mathsf{x}}(\mathbf{x}) &= \det(A^{-1})\mathsf{P}_{\mathsf{s}}(A^{-1}\mathbf{x})
\end{aligned}$$

- Thus, our **estimated** density of observations is

$$\hat{\mathsf{P}}_{\mathsf{x}} = \cancel{\det(\mathbf{B})}\, \hat{\mathsf{P}}_{\mathsf{s}}(\mathbf{B}\mathsf{x})$$

ICA: maximum likelihood

- We have a model for the observations, parametrised by $(\mathbf{B}, \hat{\mathsf{P}}_{\mathsf{s}})$
  - Model must have $\hat{\mathsf{P}}_{\mathsf{s}} = \prod_{i=1}^{l} \hat{\mathsf{P}}_{\mathsf{s}_i}$
- Our **estimated** density of observations is

$$\hat{\mathsf{P}}_{\mathsf{x}} = \hat{\mathsf{P}}_{\mathsf{s}}(\mathbf{B}\mathsf{x})$$

- Maximise the expected log likelihood, ($\mathbf{B}_{i,:}$ is ith row)

$$L := \mathbf{E}_{\mathsf{x}}\left[\log \hat{\mathsf{P}}_{\mathsf{x}}\right] = \sum_{i=1}^{l} \mathbf{E}_{\mathsf{x}} \log \hat{\mathsf{P}}_{\mathsf{s}_i}(\mathbf{B}_{i,:}\mathsf{x})$$

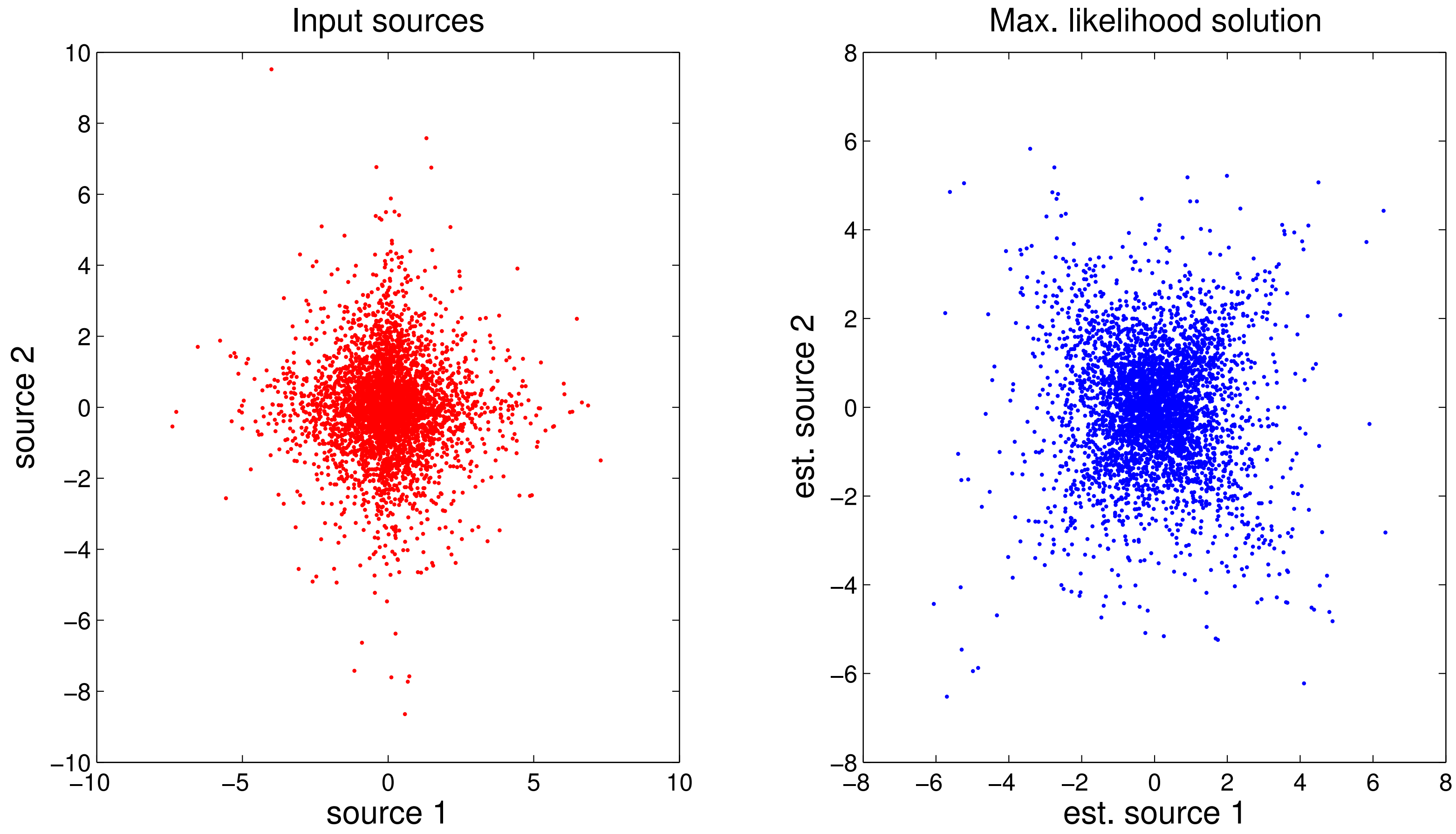
- Finite sample version:

$$L_{\text{emp}} = \frac{1}{m} \sum_{j=1}^{m} \sum_{i=1}^{l} \log \hat{\mathsf{P}}_{\mathsf{s}_i}(\mathbf{B}_{i,:} X_{:,j})$$

Notation: $X_{:,j}$ is jth column.

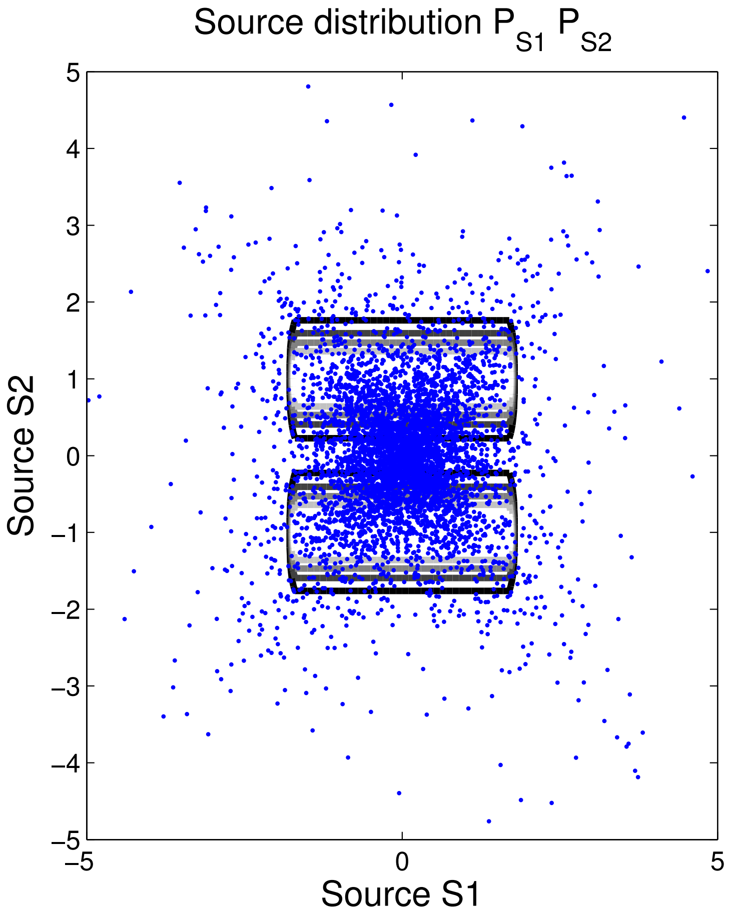
Maximum likelihood: where it fails

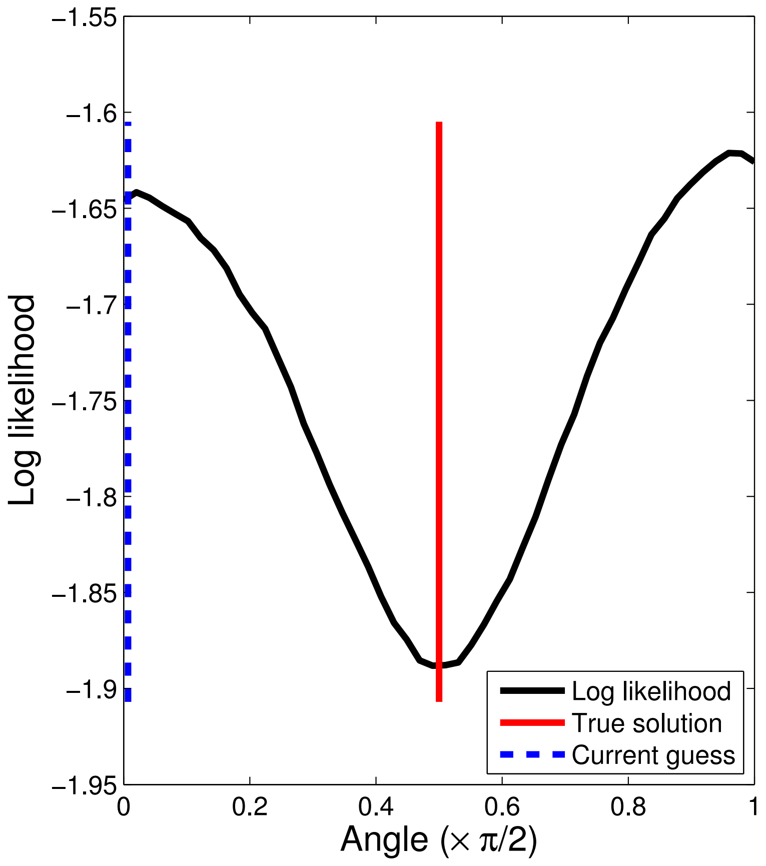
- Model as before, but true source densities are Laplace.
- Why is this wrong?



Maximum likelihood: where it fails

- Model as before, but true source densities are Laplace.
- Why is this wrong?





Another failure mode: Gaussians revisited

Setting:

- $\mathbf{s}$ are two independent, unit variance Gaussians.
- Unmixing matrix B is orthogonal

The density of the mixture $\mathbf{x}$ is proportional to

$$\hat{\mathbf{P}}_{\mathbf{x}} = \mathbf{P}_{\mathbf{s}}(B\mathbf{x}) \propto \exp\left(-\mathbf{x}^\top B^\top C_s^{-1} B\mathbf{x}\right).$$

- C_s is diagonal with equal entries, hence B commutes with C_s^{-1}.
- $B^\top B = I$
- Hence: $\hat{\mathbf{P}}_{\mathbf{x}}$ constant wrt B

We cannot recover independent Gaussians when they are mixed with a rotation matrix.

Back to original setting: independence

- A model-free approach to ICA: use an objective function (contrast function) $\phi(\mathbf{y})$ which measures “closeness to independence”.

Back to original setting: independence

- A model-free approach to ICA: use an objective function (contrast function) $\phi(\mathbf{y})$ which measures “closeness to independence”.
- Ideally: contrast $\phi(\mathbf{y}) = 0$ if and only if all components of $\mathbf{y}$ mutually independent:

$$\mathbf{P}_{\mathbf{y}} = \prod_{i=1}^{l} \mathbf{P}_{\mathsf{y}_i}.$$

  - Under our mixing assumptions: $\mathbf{y}$ are original sources $\mathbf{s}$ besides permutations, sign swaps

Back to original setting: independence

- A model-free approach to ICA: use an objective function (contrast function) $\phi(\mathbf{y})$ which measures "closeness to independence".
- Ideally: contrast $\phi(\mathbf{y}) = 0$ if and only if all components of $\mathbf{y}$ mutually independent:

$$\mathsf{P}_{\mathsf{y}} = \prod_{i=1}^{l} \mathsf{P}_{\mathsf{y}_i}.$$

  – Under our mixing assumptions: $\mathbf{y}$ are original sources $\mathbf{s}$ besides permutations, sign swaps

- How it's *really* used: contrast should be "smallest" when random variables are "most independent"

Mutual information

- A widely used contrast function: The mutual information,

$$I(\mathbf{y}) = D_{\mathrm{KL}}\left(\mathbf{P}_{\mathbf{y}} \middle\| \prod_{i=1}^{l} \mathbf{P}_{\mathrm{y}_i}\right) = \int \log\left(\frac{\mathbf{P}_{\mathbf{y}}}{\prod_{i=1}^{l} \mathbf{P}_{\mathrm{y}_i}}\right) d\mathbf{P}_{\mathbf{y}}$$

- $D_{\mathrm{KL}} \geq 0$ with equality iff $\mathbf{P}_{\mathbf{y}} = \prod_{i=1}^{l} \mathbf{P}_{\mathrm{y}_i}$

Mutual information

- A widely used contrast function: The mutual information,

$$I(\mathbf{y}) = D_{\mathrm{KL}}\left(\mathsf{P}_{\mathbf{y}} \middle\| \prod_{i=1}^{l} \mathsf{P}_{\mathrm{y}_i}\right) = \int \log\left(\frac{\mathsf{P}_{\mathbf{y}}}{\prod_{i=1}^{l} \mathsf{P}_{\mathrm{y}_i}}\right) d\mathsf{P}_{\mathbf{y}}$$

- $D_{\mathrm{KL}} \geq 0$ with equality iff $\mathsf{P}_{\mathbf{y}} = \prod_{i=1}^{l} \mathsf{P}_{\mathrm{y}_i}$
- Simplification: when $\mathbf{B}$ is a rotation,

$$D_{\mathrm{KL}}\left(\mathsf{P}_{\mathbf{y}} \middle\| \prod_{i=1}^{l} \mathsf{P}_{\mathrm{y}_i}\right) = \sum_{i=1}^{l} h\left(\mathrm{y}_i\right) - h\left(\mathbf{x}\right) - \log\det\mathbf{B}.$$

where $h(\mathrm{y}) = -\mathbf{E}_{\mathrm{y}} \log(\mathsf{P}_{\mathrm{y}}(y))$

Proof: Given $\mathbf{y} = \mathbf{Bx}$

$$\mathsf{P}_{\mathbf{y}}(\mathbf{y}) = \det(\mathbf{B}^{-1})\mathsf{P}_{\mathbf{x}}(\mathbf{B}^{-1}\mathbf{y}) = \det(\mathbf{B}^{-1})\mathsf{P}_{\mathbf{x}}(\mathbf{x})$$

and $\det(\mathbf{B}^{-1}) = (\det(\mathbf{B}))^{-1}$

Mutual information

- A widely used contrast function: The mutual information,

$$I(\mathbf{y}) = D_{\mathrm{KL}}\left(\mathbf{P}_{\mathbf{y}} \middle\| \prod_{i=1}^{l} \mathbf{P}_{\mathrm{y}_i}\right) = \int \log\left(\frac{\mathbf{P}_{\mathbf{y}}}{\prod_{i=1}^{l} \mathbf{P}_{\mathrm{y}_i}}\right) d\mathbf{P}_{\mathbf{y}}$$

- $D_{\mathrm{KL}} \geq 0$ with equality iff $\mathbf{P}_{\mathbf{y}} = \prod_{i=1}^{l} \mathbf{P}_{\mathrm{y}_i}$
- Simplification: when $\mathbf{B}$ is a rotation,

$$D_{\mathrm{KL}}\left(\mathbf{P}_{\mathbf{y}} \middle\| \prod_{i=1}^{l} \mathbf{P}_{\mathrm{y}_i}\right) = \sum_{i=1}^{l} h\left(\mathrm{y}_i\right) - \underbrace{h\left(\mathbf{x}\right) - \log\det\mathbf{B}.}_{\text{constant}}$$

where $h(\mathrm{y}) = -\mathbf{E}_{\mathrm{y}} \log(\mathbf{P}_{\mathrm{y}}(y))$

Mutual information

- A widely used contrast function: The mutual information,

$$I(\mathbf{y}) = D_{\mathrm{KL}} \left(\mathsf{P}_{\mathbf{y}} \middle\| \prod_{i=1}^{l} \mathsf{P}_{\mathrm{y}_i} \right) = \int \log \left(\frac{\mathsf{P}_{\mathbf{y}}}{\prod_{i=1}^{l} \mathsf{P}_{\mathrm{y}_i}} \right) d\mathsf{P}_{\mathbf{y}}$$

- $D_{\mathrm{KL}} \geq 0$ with equality iff $\mathsf{P}_{\mathbf{y}} = \prod_{i=1}^{l} \mathsf{P}_{\mathrm{y}_i}$
- Simplification: when $\mathbf{B}$ is a rotation,

$$D_{\mathrm{KL}} \left(\mathsf{P}_{\mathbf{y}} \middle\| \prod_{i=1}^{l} \mathsf{P}_{\mathrm{y}_i} \right) = \sum_{i=1}^{l} h\left(\mathrm{y}_i\right) - \underbrace{h\left(\mathbf{x}\right) - \log\det \mathbf{B}.}_{\text{constant}}$$

where $h(\mathrm{y}) = -\mathbf{E}_{\mathrm{y}} \log(\mathsf{P}_{\mathrm{y}}(y))$

Contrast: $\phi_{KL}(\mathbf{y}) := \sum_{i=1}^{l} h\left(\mathrm{y}_i\right)$

Maximum likelihood revisited

- Mutual information contrast: minimize

$$\phi_{KL}(\mathbf{y}) := \sum_{i=1}^{l} -\mathbf{E}_{\mathsf{y}_i} \log(\mathbf{P}_{\mathsf{y}_i}(y_i))$$

- Maximum likelihood: maximize

$$\begin{aligned} L &:= \sum_{i=1}^{l} \mathbf{E}_{\mathbf{x}} \log \hat{\mathbf{P}}_{\mathsf{s}_i}(\mathbf{B}_{i,:}\mathbf{x}) \\ &= \sum_{i=1}^{l} \mathbf{E}_{\mathsf{y}_i} \log(\mathbf{P}_{\mathsf{y}_i}(y_i)) \end{aligned}$$

- Same thing! The difference is in approach:
  - For max. likelihood we assumed a model $\hat{\mathbf{P}}_{\mathbf{s}}$
  - Now we (ideally...) assume no model for $\mathbf{P}_{\mathbf{y}}$

Contrast functions with fixed nonlinearities

- Entropies hard to compute/optimize: replace with

$$\phi_f(\mathbf{y}) = \sum_{i=1}^{l} \mathbf{E}_{\mathsf{y}_i}(f(y_i))$$

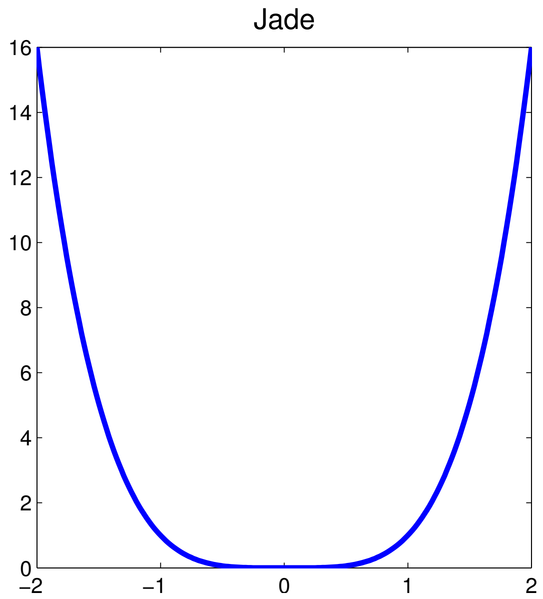
for some other nonlinear $f(y)$

Contrast functions with fixed nonlinearities

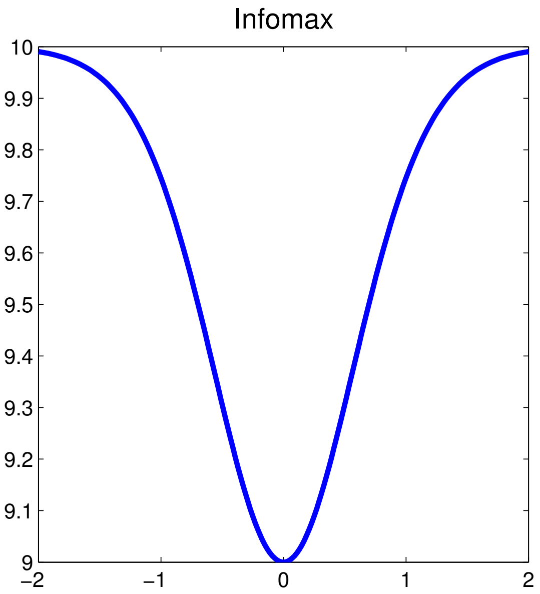
- Entropies hard to compute/optimize: replace with

$$\phi_f(\mathbf{y}) = \sum_{i=1}^{l} \mathbf{E}_{\mathrm{y}_i}(f(y_i))$$

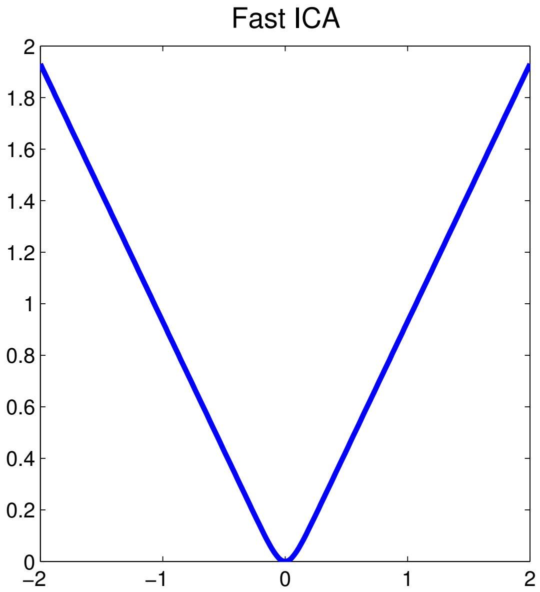
for some other nonlinear $f(y)$



$f(y) = y^4$



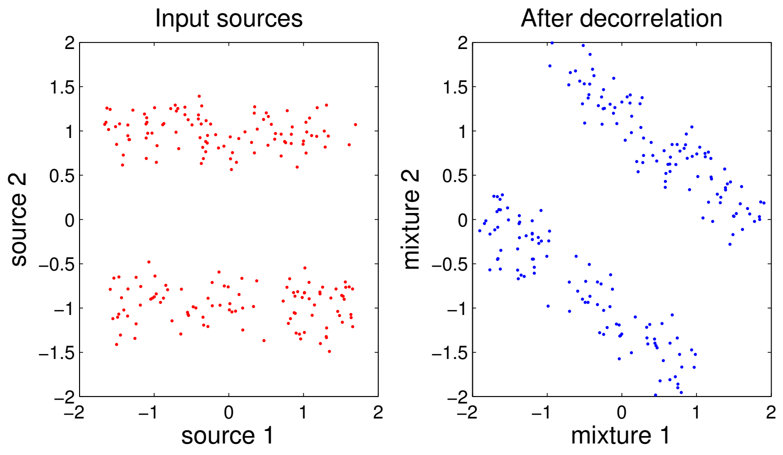
$f(y) = a - exp(-y^2/2)\mathrm{sech}^2(y)$

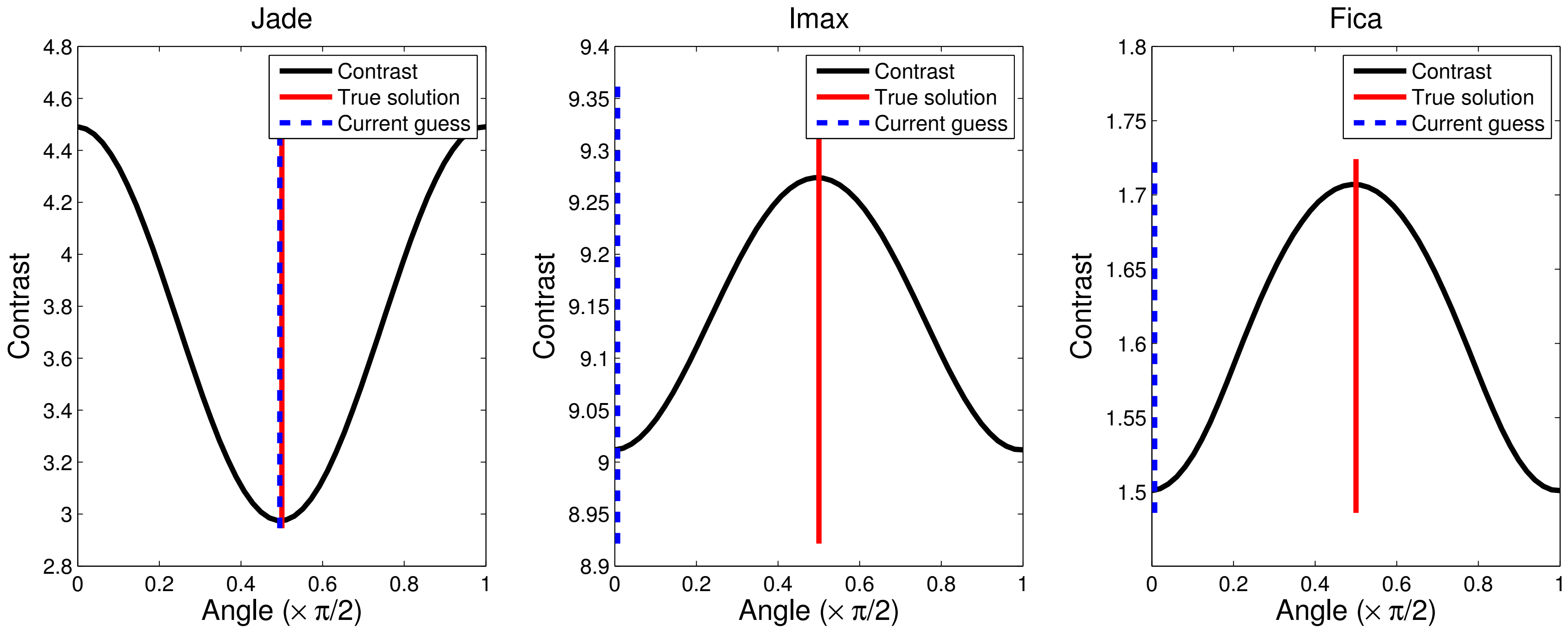


$f(y) = \frac{1}{a} \log \cosh(ay),$

Our example again

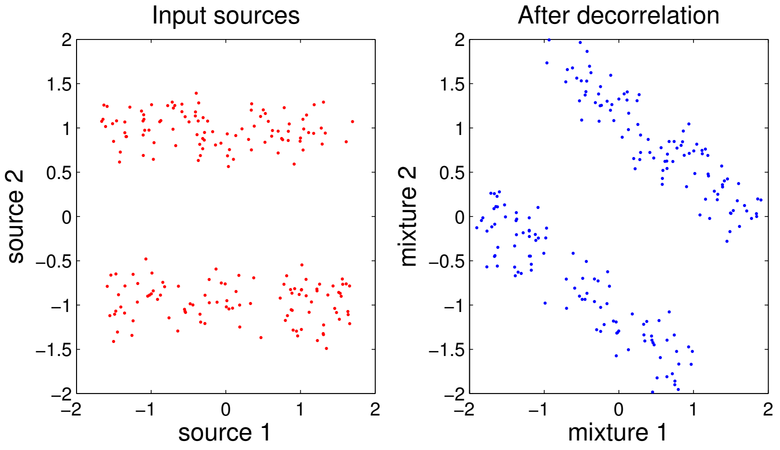
Recall: minimize contrast.





Our example again

Recall: minimize contrast.



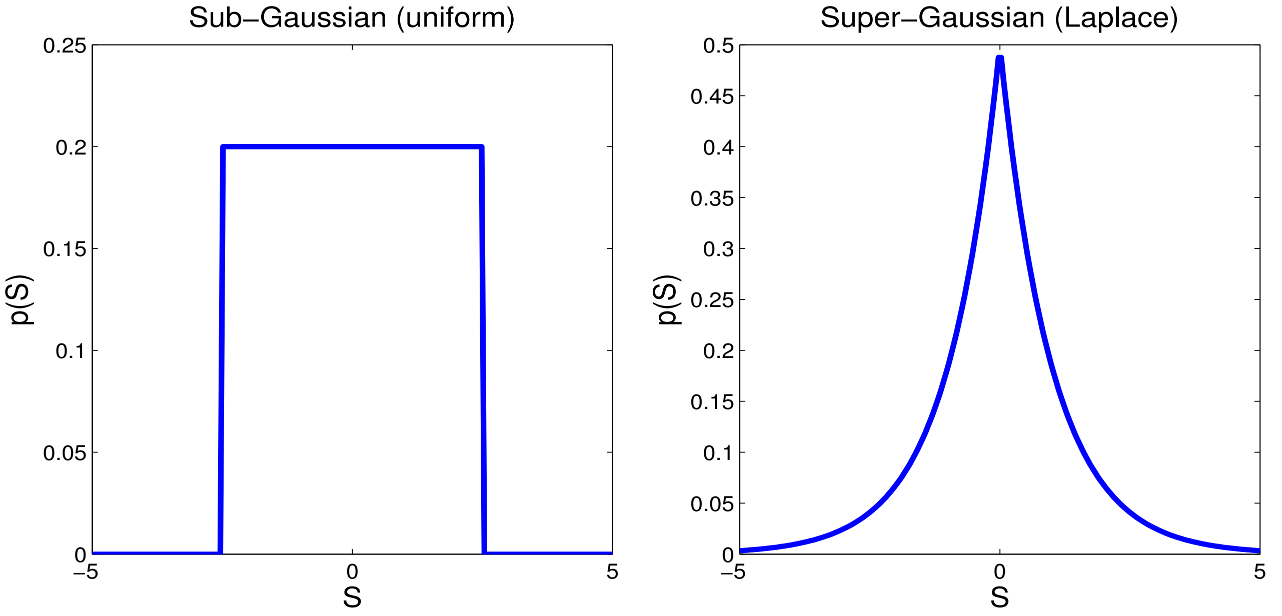
What went wrong?

Kurtosis: an important concept

- Kurtosis definition: when mean is zero,

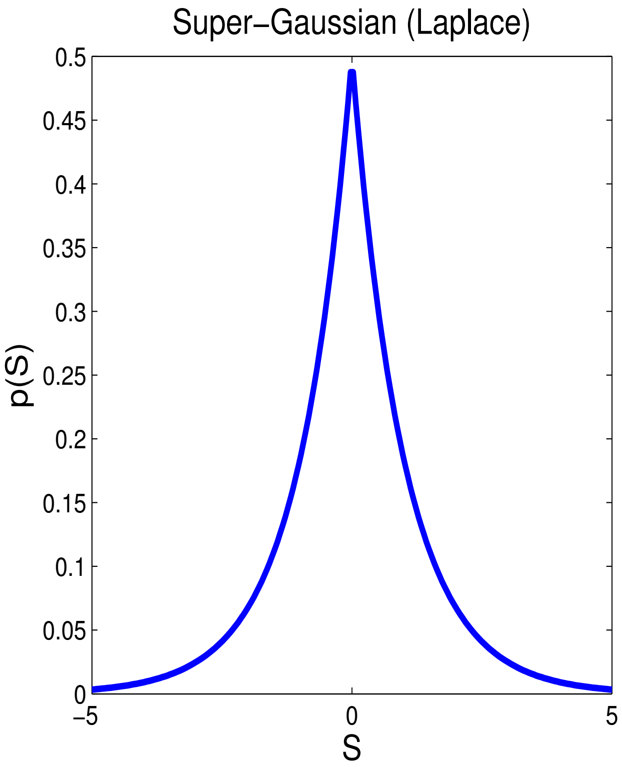
$$\kappa_4 = \mathbf{E}\left(\mathrm{x}^4\right) - 3\left(\mathbf{E}\left(\mathrm{x}^2\right)\right)^2 .$$

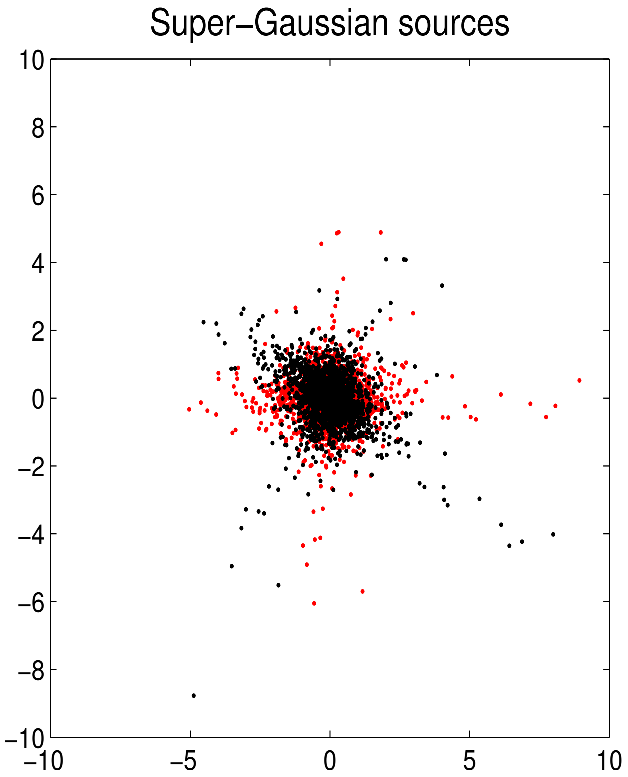
- Source densities can be super-Gaussian (positive kurtosis) or sub-Gaussian (negative kurtosis)
- Zero kurtosis does not mean Gaussian!



Demo: contrasts with fixed nonlinearities

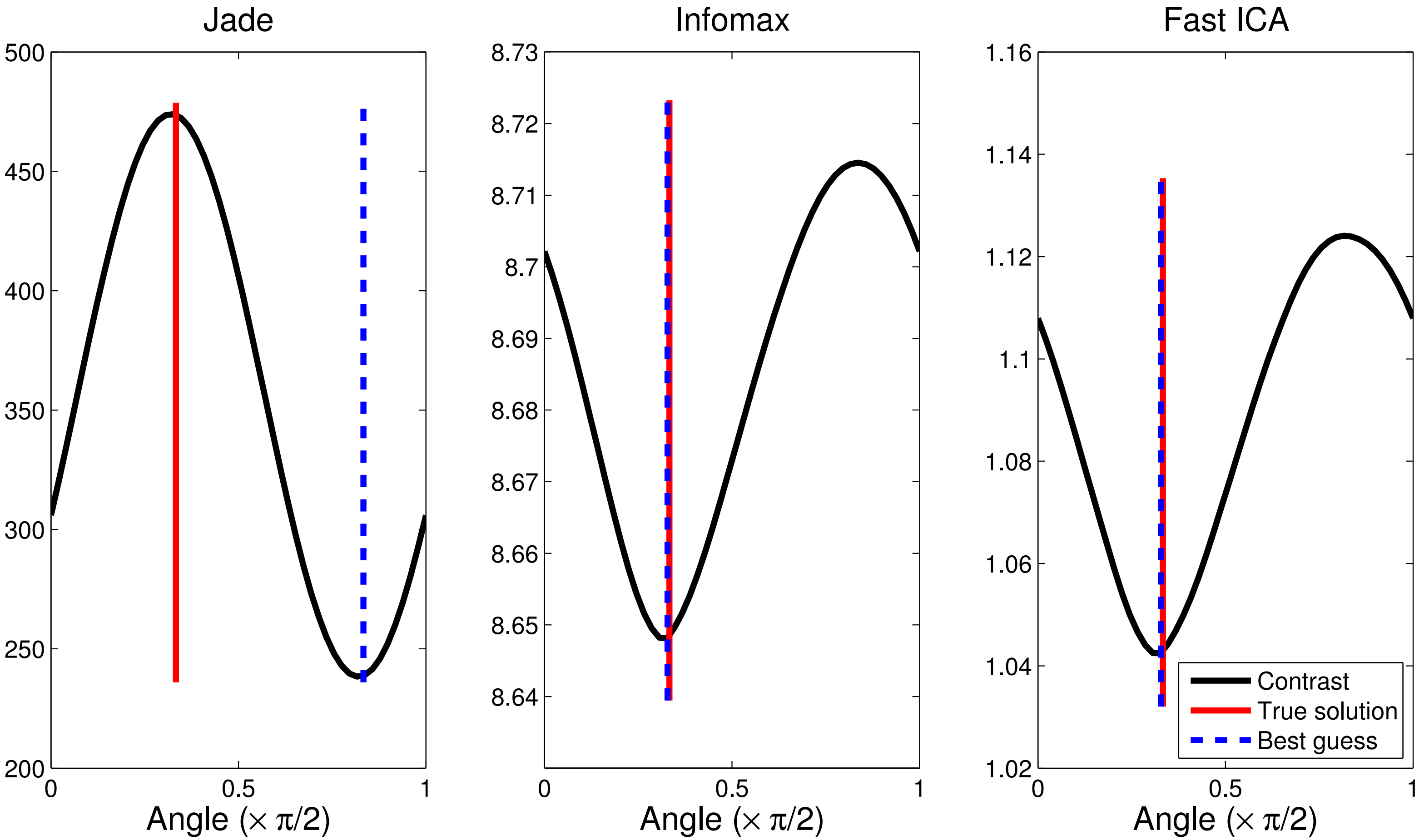
- Super-Gaussian (Laplace) sources
- Unmixed sources in red
- Mixture (angle $\pi/6$) in black





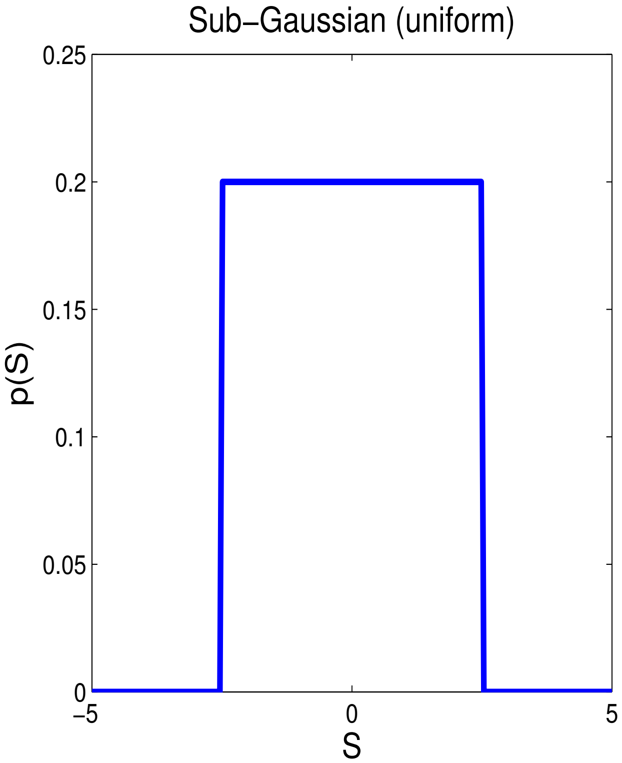
Demo: contrasts with fixed nonlinearities

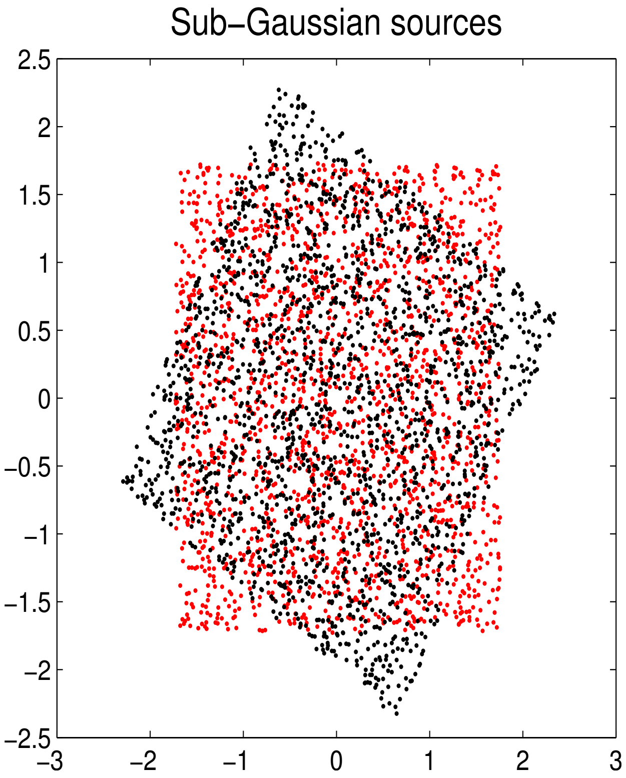
- Super-Gaussian results for Jade, Infomax, and Fast ICA



Demo: contrasts with fixed nonlinearities

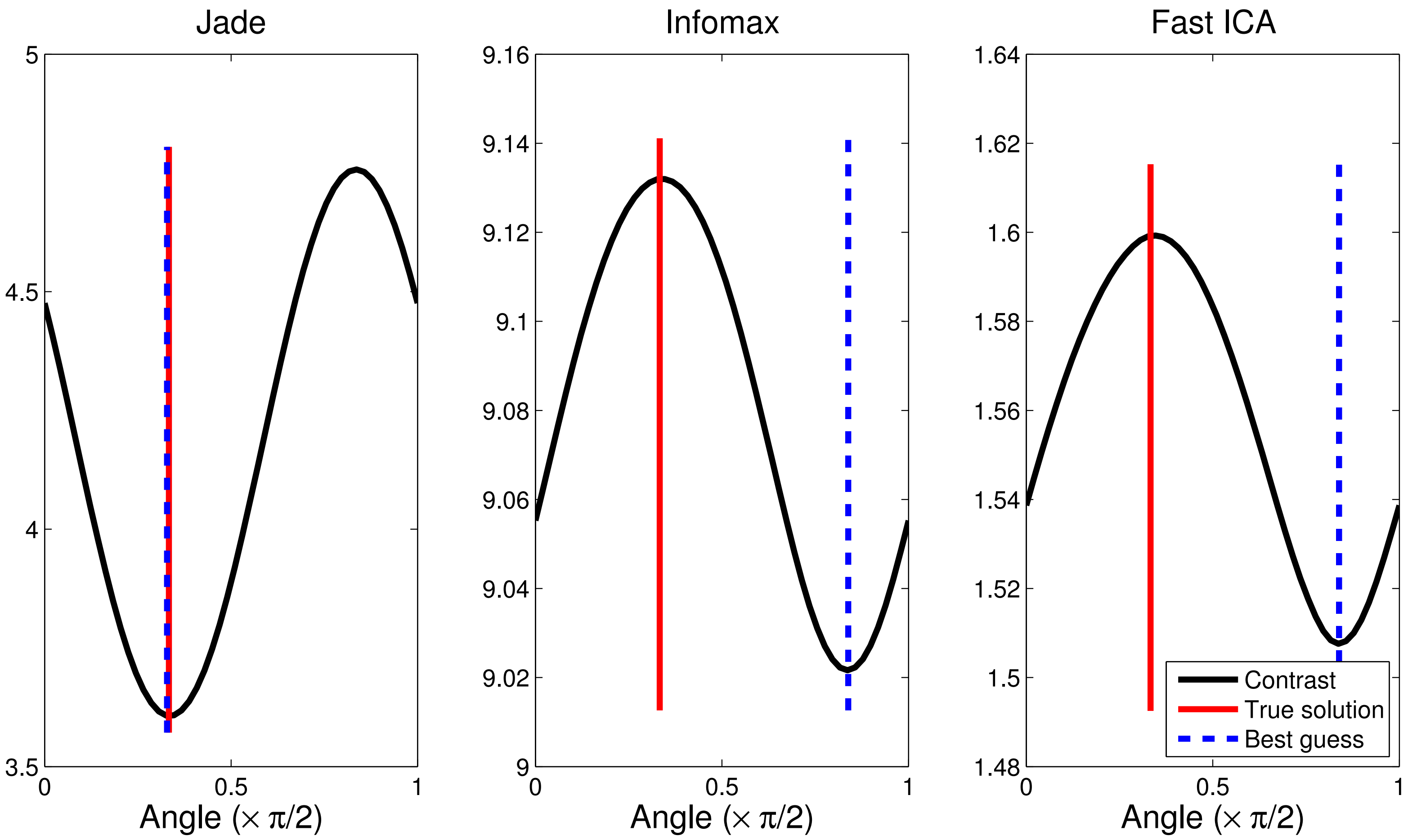
- Sub-Gaussian (Uniform) sources
- Unmixed sources in red
- Mixture (angle $\pi/6$) in black





Demo: contrasts with fixed nonlinearities

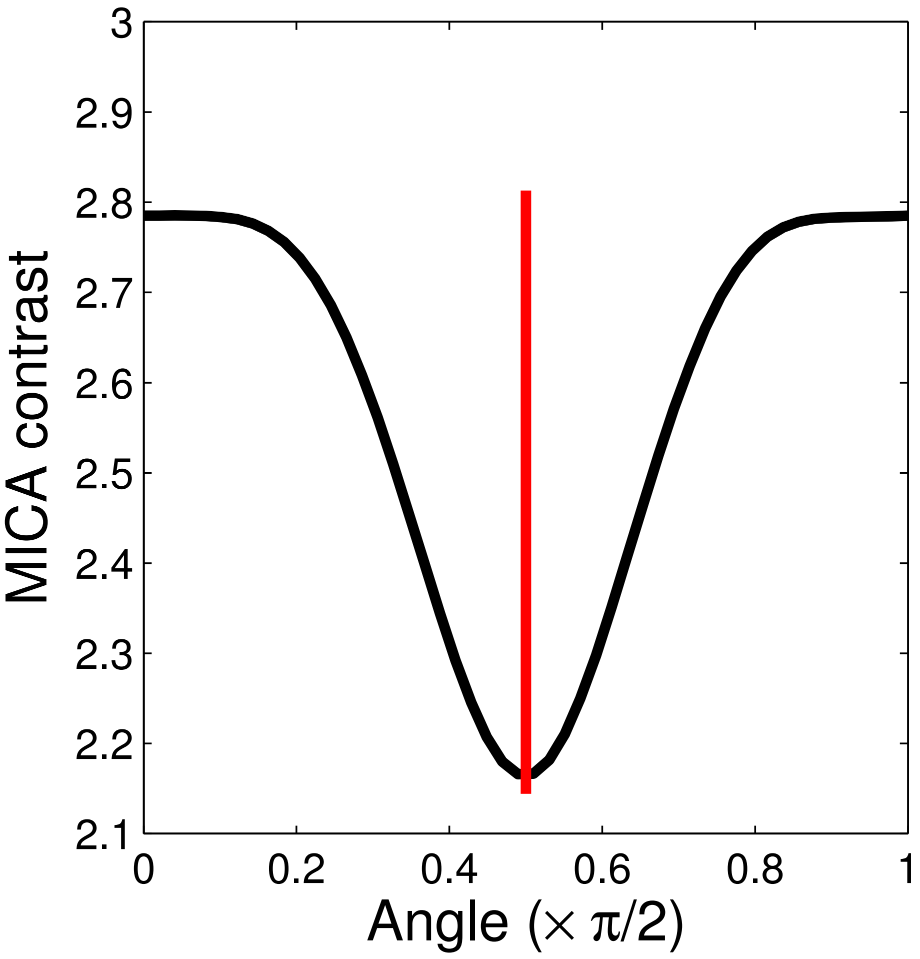
- Sub-Gaussian results for Jade, Infomax, and Fast ICA

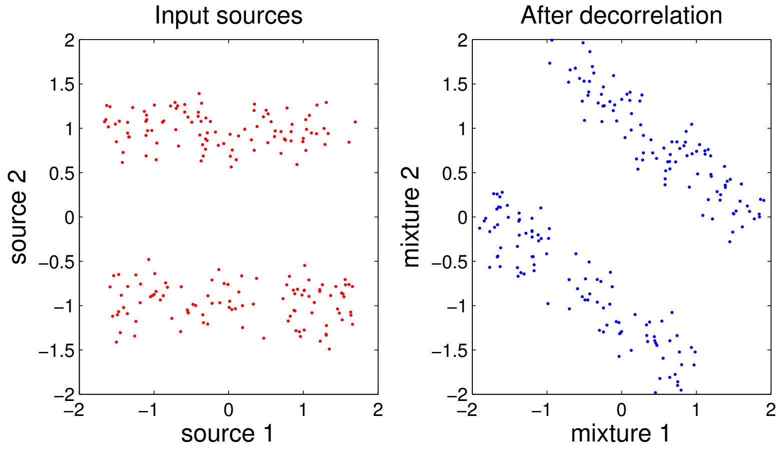


Care needed when using fixed contrasts!

Contrast functions using entropy estimates

- Simplest option: convolve with spline kernel, then compute discrete entropy via space partition [Pham, 2004]

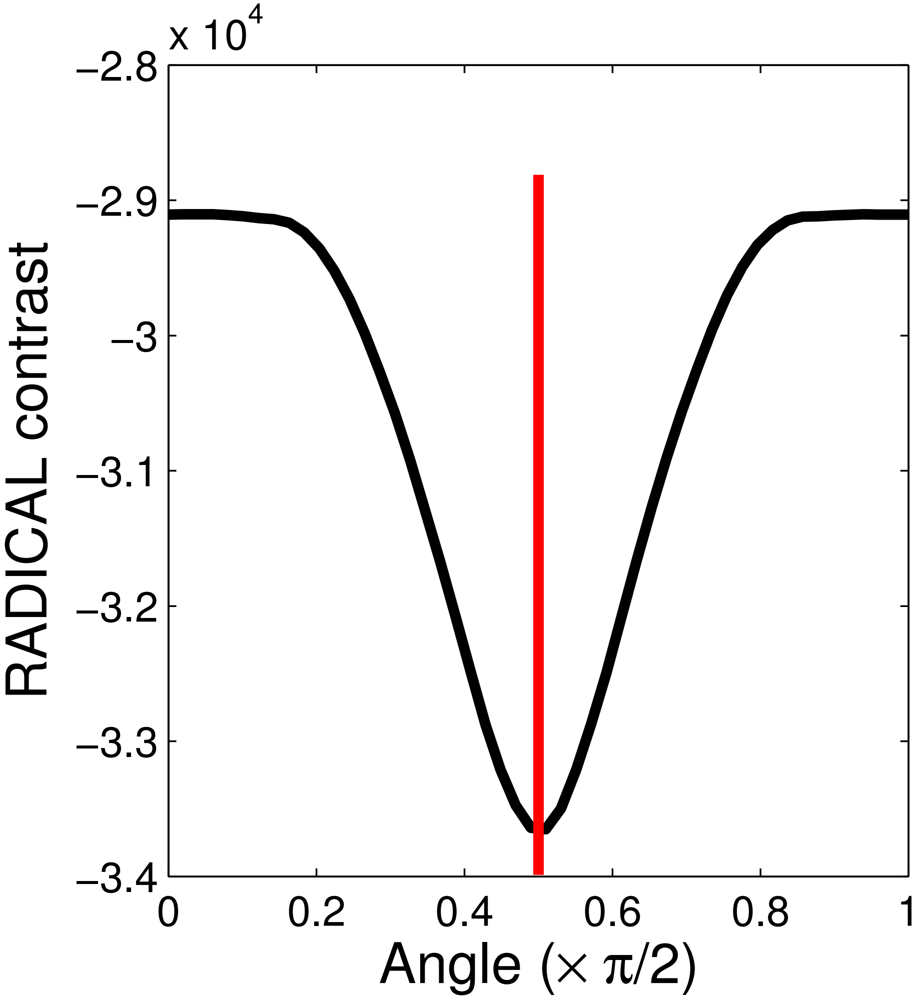


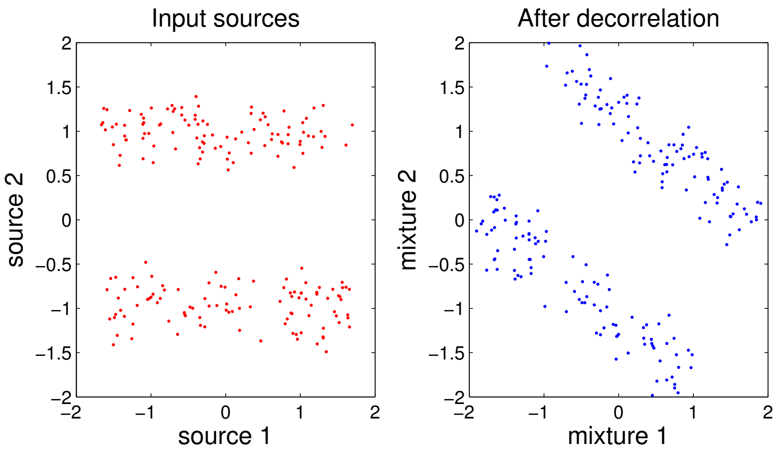


Contrast functions using spacings entropy estimate

- More sophisticated option: spacings estimate of entropy

[Learned-Miller and Fisher III, 2003]



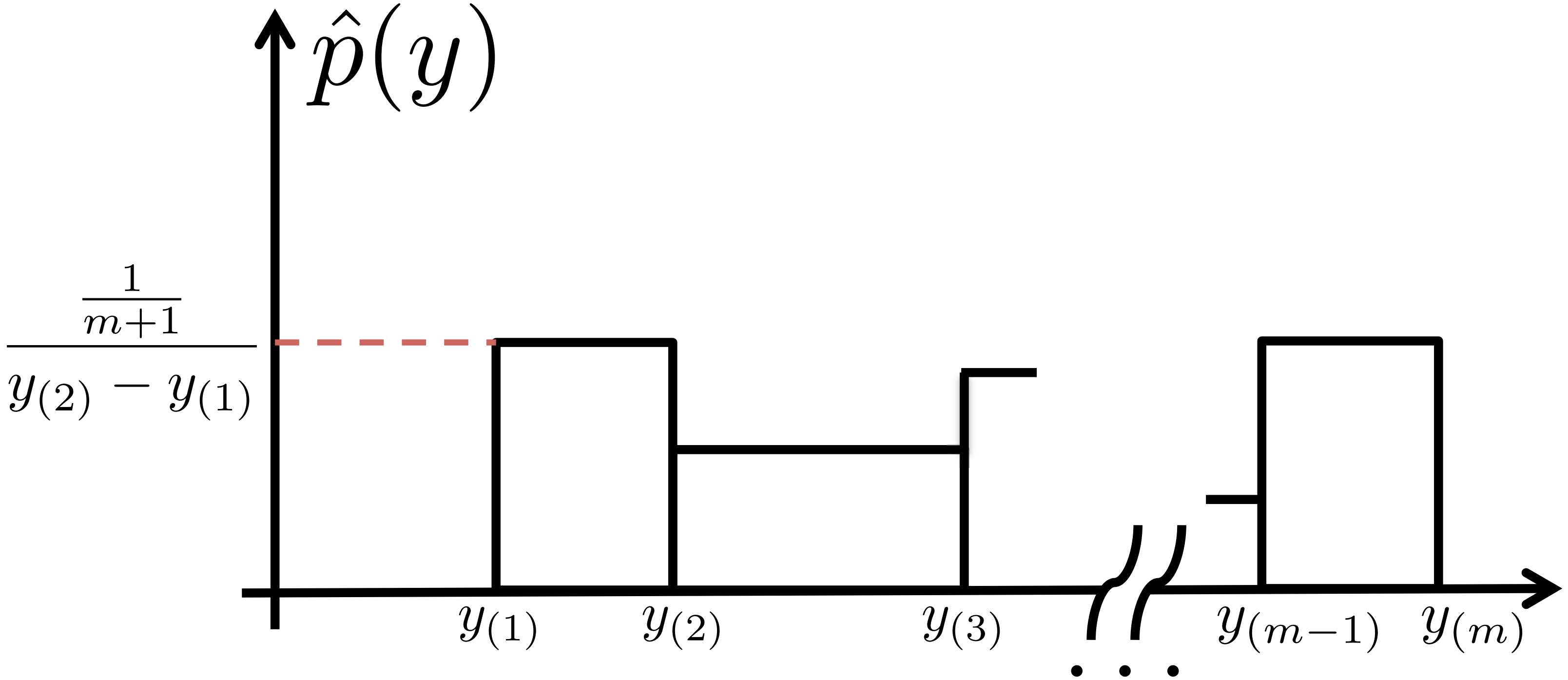


Contrast functions using spacings entropy estimate

- More sophisticated option: spacings estimate of entropy

  [Learned-Miller and Fisher III, 2003]

- Sort sample $Y_1, \ldots, Y_m$ in increasing order: $Y_{(i)} \leq Y_{(i+1)}$
- Prob. density estimate based on spacings
- Idea: prob. mass between adjacent samples $y_{(i)}, y_{(i+1)}$ is $\approx (m+1)^{-1}$



Contrast functions using spacings entropy estimate

- More sophisticated option: spacings estimate of entropy

  [Learned-Miller and Fisher III, 2003]

- Sort sample $Y_1, \ldots, Y_m$ in increasing order: $Y_{(i)} \leq Y_{(i+1)}$
- Prob. density estimate based on spacings

$$\hat{\mathsf{P}}(y; Y_1, \ldots, Y_m) = \frac{1}{(m+1)(Y_{(i+1)} - Y_{(i)})}, \qquad Y_{(i)} \leq y < Y_{(i+1)}$$

- Entropy estimate based on spacings

$$\hat{h}(Y) = \frac{1}{m-1} \sum_{i=1}^{m-1} \log(m+1)(Y_{(i+1)} - Y_{(i)})$$

Contrast functions using spacings entropy estimate

Proof:

$$
\begin{aligned}
H(Y) &= -\int_{-\infty}^{\infty} p(y)\log p(y)dy \\
&\approx -\sum_{i=0}^{m}\int_{y_{(i)}}^{y_{(i+1)}} \hat{p}(y)\log\hat{p}(y)dy \\
&= -\sum_{i=0}^{m}\int_{y_{(i)}}^{y_{(i+1)}} \frac{(m+1)^{-1}}{y_{(i+1)}-y_{(i)}}\log\frac{(m+1)^{-1}}{y_{(i+1)}-y_{(i)}}dy \\
&= -\sum_{i=1}^{m-1}(m+1)^{-1}\log\frac{(m+1)^{-1}}{y_{(i+1)}-y_{(i)}} \\
&\approx -\sum_{i=1}^{m-1}(m-1)^{-1}\log\frac{(m+1)^{-1}}{y_{(i+1)}-y_{(i)}} \\
&= \sum_{i=1}^{m-1}(m-1)^{-1}\log\left[(m+1)\left(y_{(i+1)}-y_{(i)}\right)\right]
\end{aligned}
$$

Contrast functions using spacings entropy estimate

- More sophisticated option: spacings estimate of entropy

  [Learned-Miller and Fisher III, 2003]

- Sort sample $Y_1, \ldots, Y_m$ in increasing order: $Y_{(i)} \leq Y_{(i+1)}$
- Prob. density estimate based on spacings

$$\hat{\mathbf{P}}(y; Y_1, \ldots, Y_m) = \frac{1}{(m+1)(Y_{(i+1)} - Y_{(i)})}, \qquad Y_{(i)} \leq y < Y_{(i+1)}$$

- Entropy estimate based on spacings

$$\hat{h}(Y) = \frac{1}{m-1} \sum_{i=1}^{m-1} \log(m+1)(Y_{(i+1)} - Y_{(i)})$$

- Smoothing: add "extra" mixture points (noisy copies of original mixtures)
- Hard to optimize

Other independence measures as contrasts

- Why mutual information?
  - Same as maximum likelihood (good if model is correct)
  - Contrast function is sum of entropies: fast
- Other independence measures?

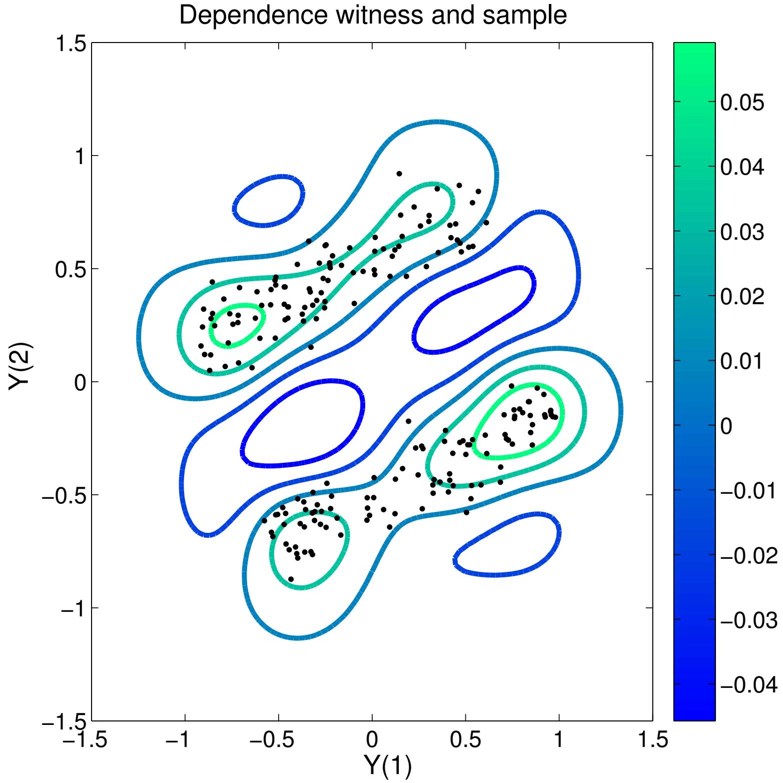
Other independence measures as contrasts

- Why mutual information?
  - Same as maximum likelihood (good if model is correct)
  - Contrast function is sum of entropies: fast
- Other independence measures?
- Most common: kernel/characteristic function-based
  - Characteristic function-based ICA [Eriksson and Koivunen, 2003, Chen and Bickel, 2005]
  - Kernel ICA (covariance): COCO, KMI, HSIC [Gretton et al., 2005, Shen et al., 2007, 2009]
  - Kernel ICA (correlation): KCCA, KGV [Bach and Jordan, 2002]
- HSIC same as characteristic function-based (for the purposes of ICA) [Shen et al., 2009]

Kernel contrast function: HSIC

- Dependence measure:

$$\mathrm{HSIC}(\mathbf{P}_{UV}, F) := \left(\sup_{f \in F} \left[\mathbf{E}_{UV} f - \mathbf{E}_U \mathbf{E}_V f \right] \right)^2$$



HSIC: empirical expression

- Empirical HSIC:

$$\mathrm{HSIC} := \frac{1}{m^2}\mathrm{tr}(KHLH)$$

  - K Gram matrix for $(u_1, \ldots, u_m)$
  - L Gram matrix for $(v_1, \ldots, v_m)$
  - Centering $H = I - \frac{1}{m}\mathbf{1}_m\mathbf{1}_m^\top$

Contrast functions: a small selection

Contrast function summary

- Sum of expectations of a fixed nonlinearity
  - Fast ICA, Infomax, Jade
- Sum of entropies/mutual information. . .
  - . . . using fast, smoothed entropy estimates
  - . . . using spacings/k-nn entropy estimates
- Kernel/characteristic function dependence measures

Contrast functions: a small selection

Contrast function summary

- Sum of expectations of a fixed nonlinearity
  - Fast ICA, Infomax, Jade
- Sum of entropies/mutual information...
  - ... using fast, smoothed entropy estimates
  - ... using spacings/k-nn entropy estimates
- Kernel/characteristic function dependence measures

How do we optimize?

Optimization (Jacobi)

- For two signals, the rotation is expressed

$$\mathbf{B} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

- Higher dimensions, eg for $l = 3$,

$$\mathbf{B} := \begin{bmatrix} \cos(\theta_z) & -\sin(\theta_z) & 0 \\ \sin(\theta_z) & \cos(\theta_z) & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \cos(\theta_y) & 0 & -\sin(\theta_y) \\ 0 & 1 & 0 \\ \sin(\theta_y) & 0 & \cos(\theta_y) \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_x) & -\sin(\theta_x) \\ 0 & \sin(\theta_x) & \cos(\theta_x) \end{bmatrix}$$

- Coordinate descent, exhaustive search, etc...

Optimization (Newton)

- Unmixing matrix B satisfies $B^\top B = I$
- Local parameterisation Ω about B: at iteration k,

$$B_{k+1} = B_k \exp(\Omega) \qquad \Omega = -\Omega^\top$$

- How to choose direction and size of Ω?

Optimization (Newton)

- Unmixing matrix B satisfies $B^\top B = I$
- Local parameterisation Ω about B: at iteration k,

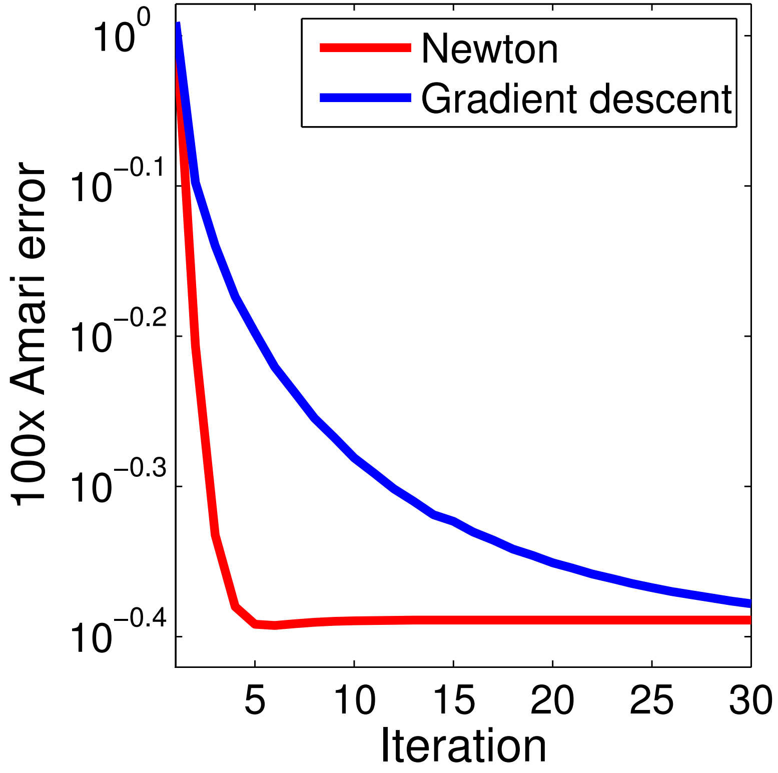
$$B_{k+1} = B_k \exp(\Omega) \qquad \Omega = -\Omega^\top$$

- How to choose direction and size of Ω?
- Write $\widetilde{\Omega} \in \mathbb{R}^{l(l-1)/2}$ the unique entries of Ω
- Newton-like method: solve the linear system for $\widetilde{\Omega} \in \mathbb{R}^{l(l-1)/2}$

$$\mathcal{H}_{B_k}(\phi)\widetilde{\Omega} = -\nabla_{B_k}(\phi)$$

  - $\nabla_{B_k}(\phi)$ is gradient of ϕ wrt $\widetilde{\Omega}$
  - $\mathcal{H}_{B_k}(\phi)$ is Hessian of ϕ wrt $\widetilde{\Omega}$
- Approximate Hessian as diagonal: FastICA [Shen and Hüper, 2006]

Gradient descent vs Newton



What if we have time dependence?

- We can get extra information from sources not being i.i.d.
- Mixture $\mathbf{x}(t)$ now stationary random process, depends on $\mathbf{x}(t-\tau)$
- Define mixture covariances

$$\mathbf{C}_0 = \mathbf{E}(\mathbf{x}(t)\mathbf{x}(t)), \qquad \mathbf{C}_\tau = \mathbf{E}(\mathbf{x}(t)\mathbf{x}(t-\tau)),$$

  - $\mathbf{C}_\tau$ independent of t (stationarity)

What if we have time dependence?

- We can get extra information from sources not being i.i.d.
- Mixture $\mathbf{x}(t)$ now stationary random process, depends on $\mathbf{x}(t-\tau)$
- Define mixture covariances

$$\mathbf{C}_0 = \mathbf{E}(\mathbf{x}(t)\mathbf{x}(t)), \qquad \mathbf{C}_\tau = \mathbf{E}(\mathbf{x}(t)\mathbf{x}(t-\tau)),$$

  - $\mathbf{C}_\tau$ independent of t (stationarity)

- Decorrelate:

$$\mathbf{B}\mathbf{C}_0\mathbf{B}^\top = \Lambda \qquad \mathbf{B}\mathbf{C}_\tau\mathbf{B}^\top = \widetilde{\Lambda}$$

  - Λ and $\widetilde{\Lambda}$ diagonal

- Combining both requirements:

$$\mathbf{B}\mathbf{C}_0\mathbf{C}_\tau^{-1} = \left(\Lambda\widetilde{\Lambda}^{-1}\right)\mathbf{B}$$

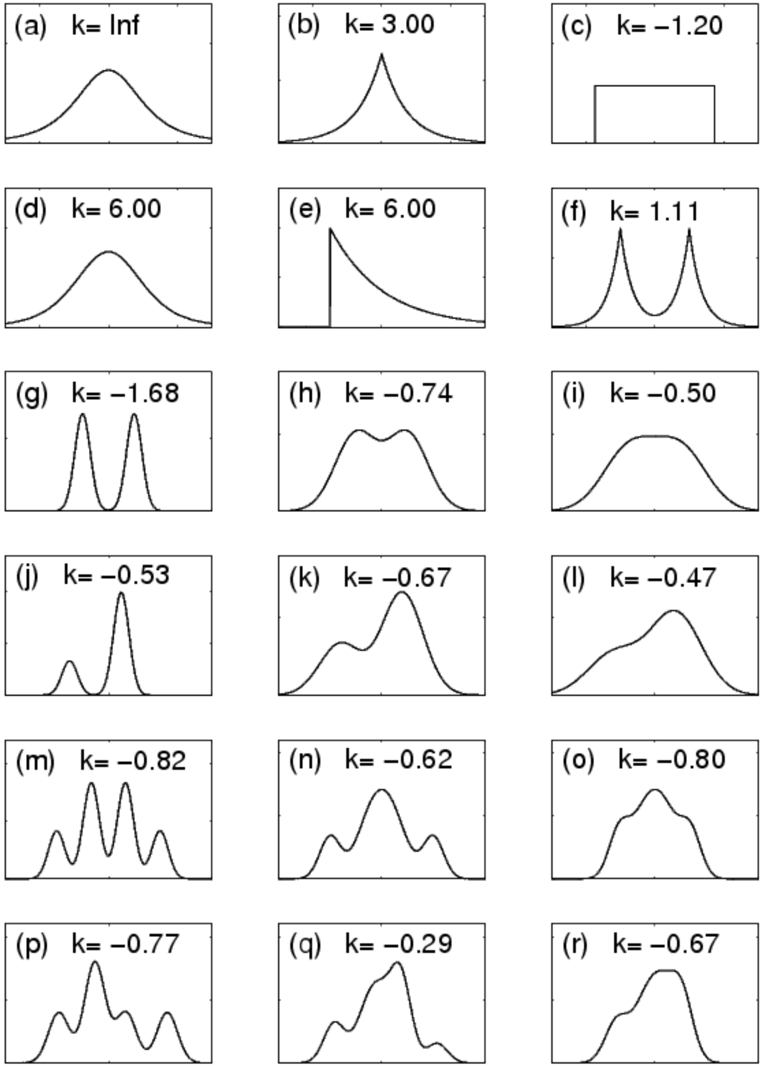
- Greater number of delays: joint diagonalisation

What’s the best method?

A basic benchmark

- $l = 8$ sources
- $m = 40,000$ samples
- Benchmark data from

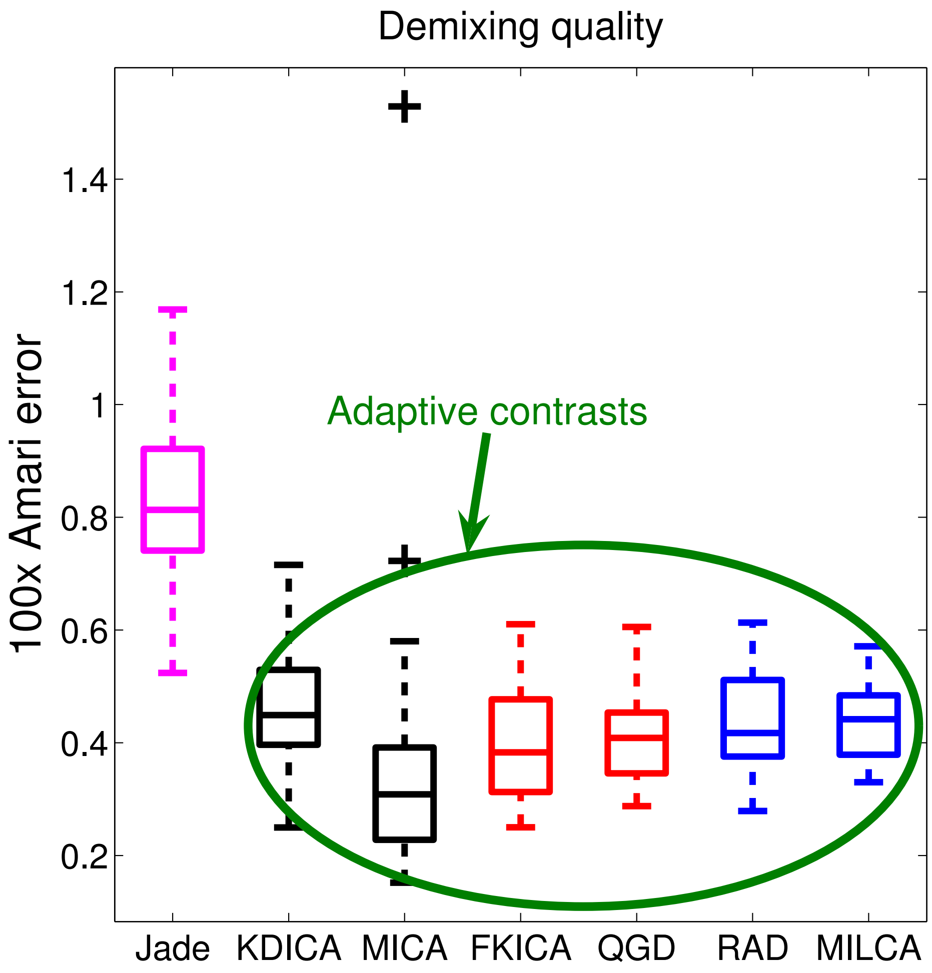
  [Bach and Jordan, 2002]
- Average over 24 repetitions



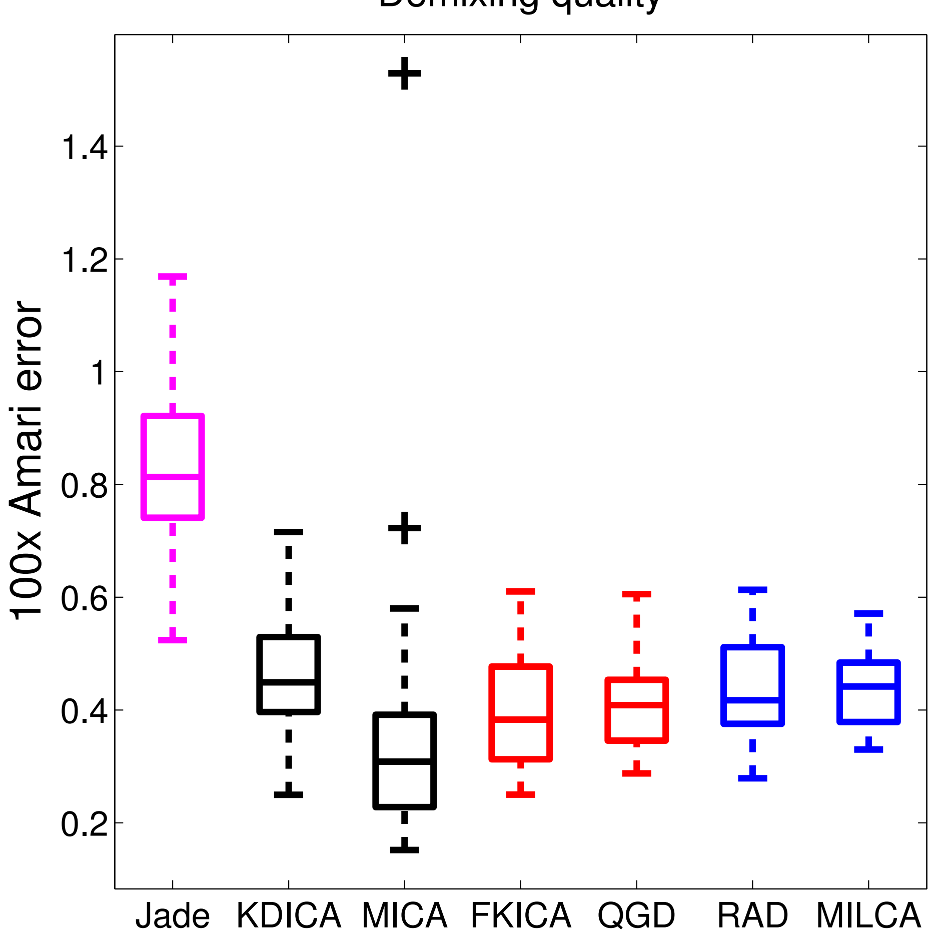
A basic benchmark: results

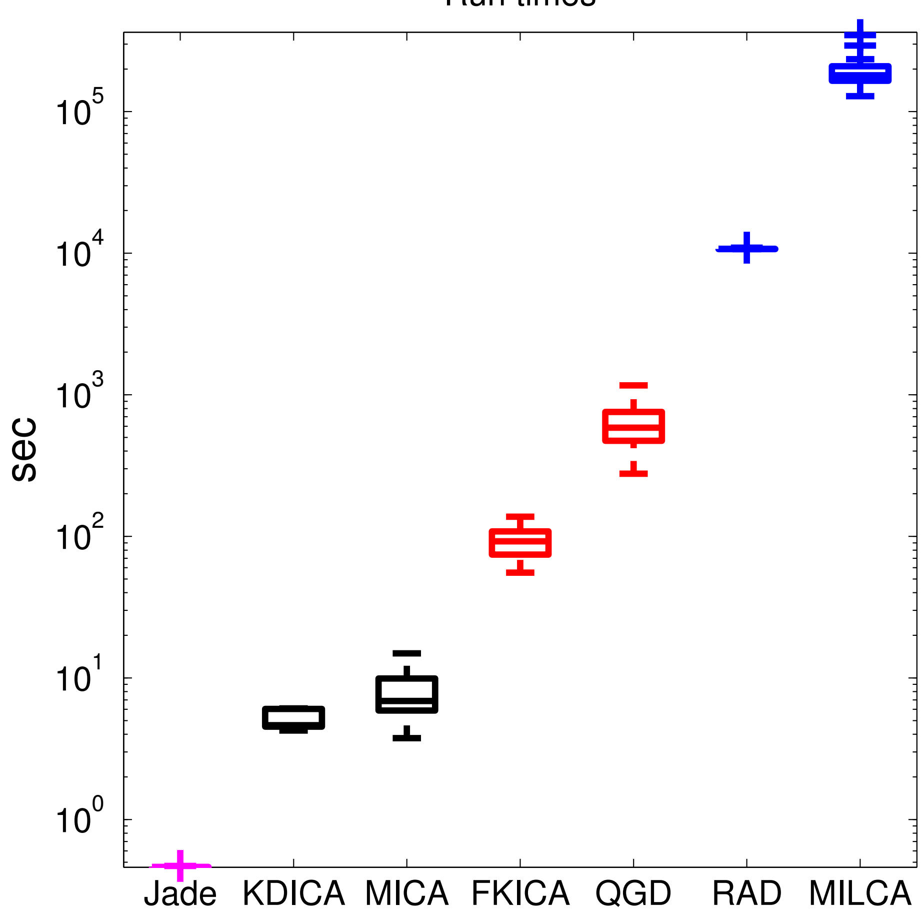
A basic benchmark: results

Adaptive contrasts outperform fixed nonlinearities



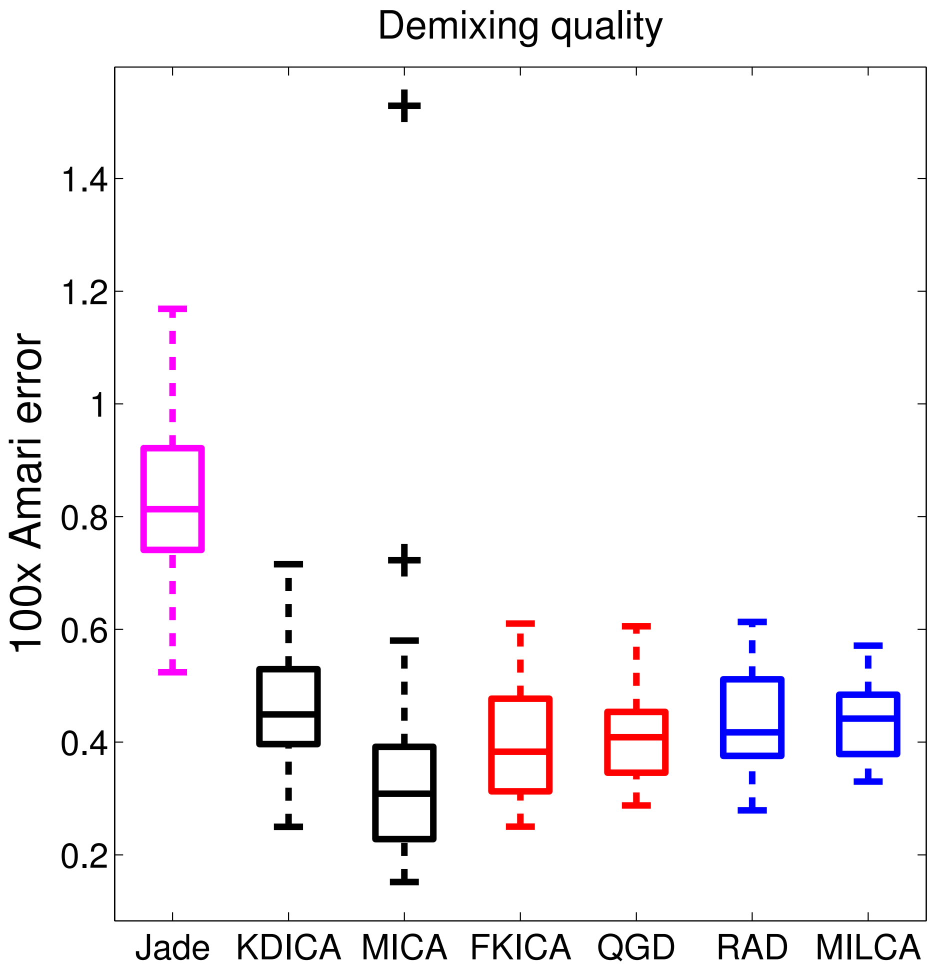
A basic benchmark: computational cost

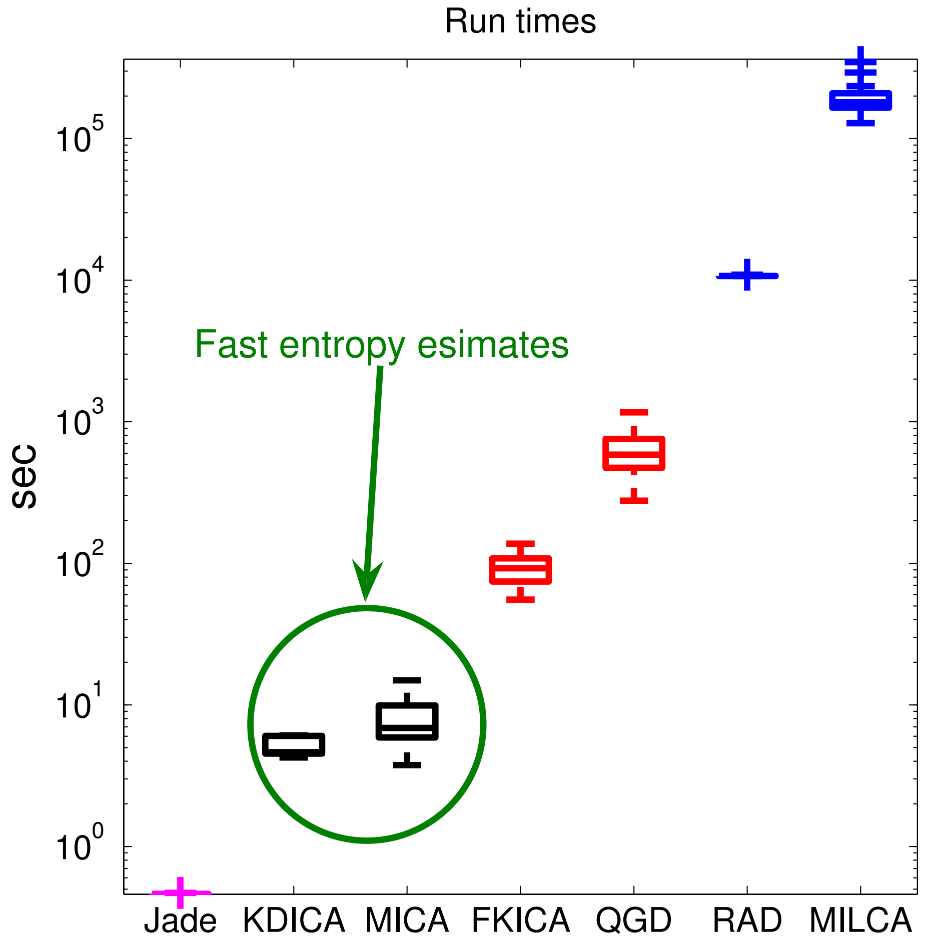

Demixing quality


Run times

A basic benchmark: computational cost

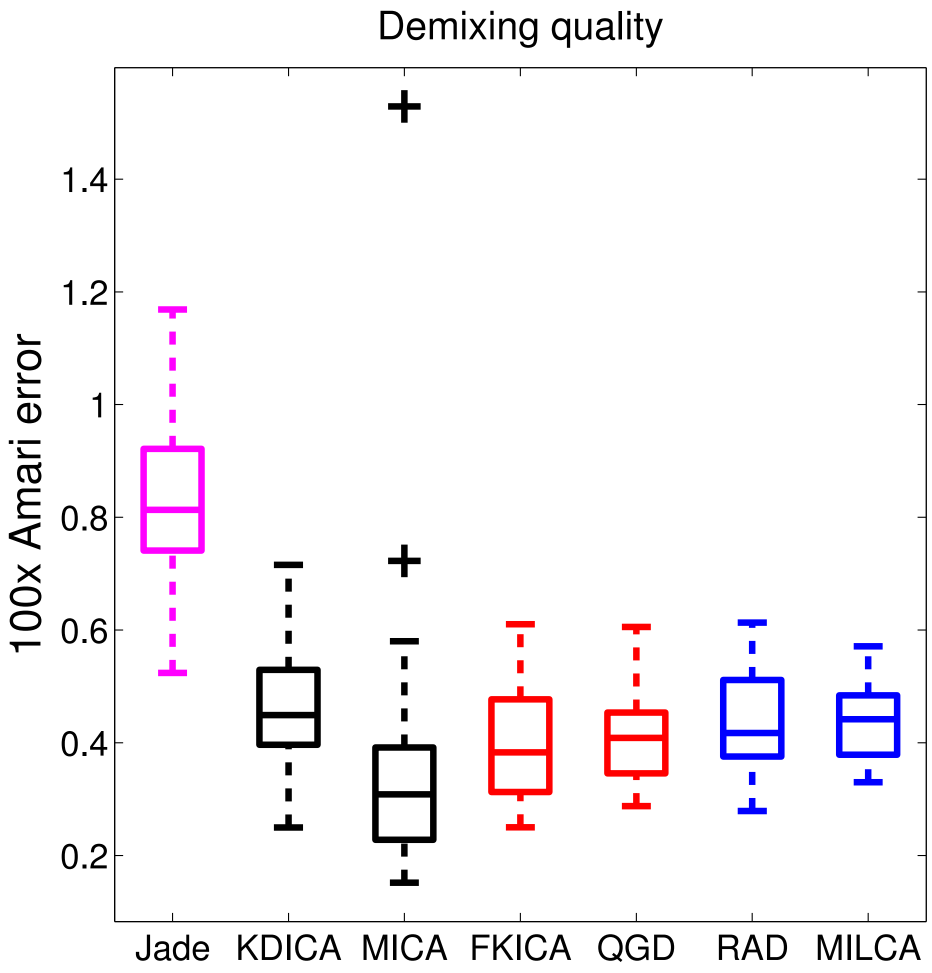
Best runtime (adaptive): fast entropy estimates

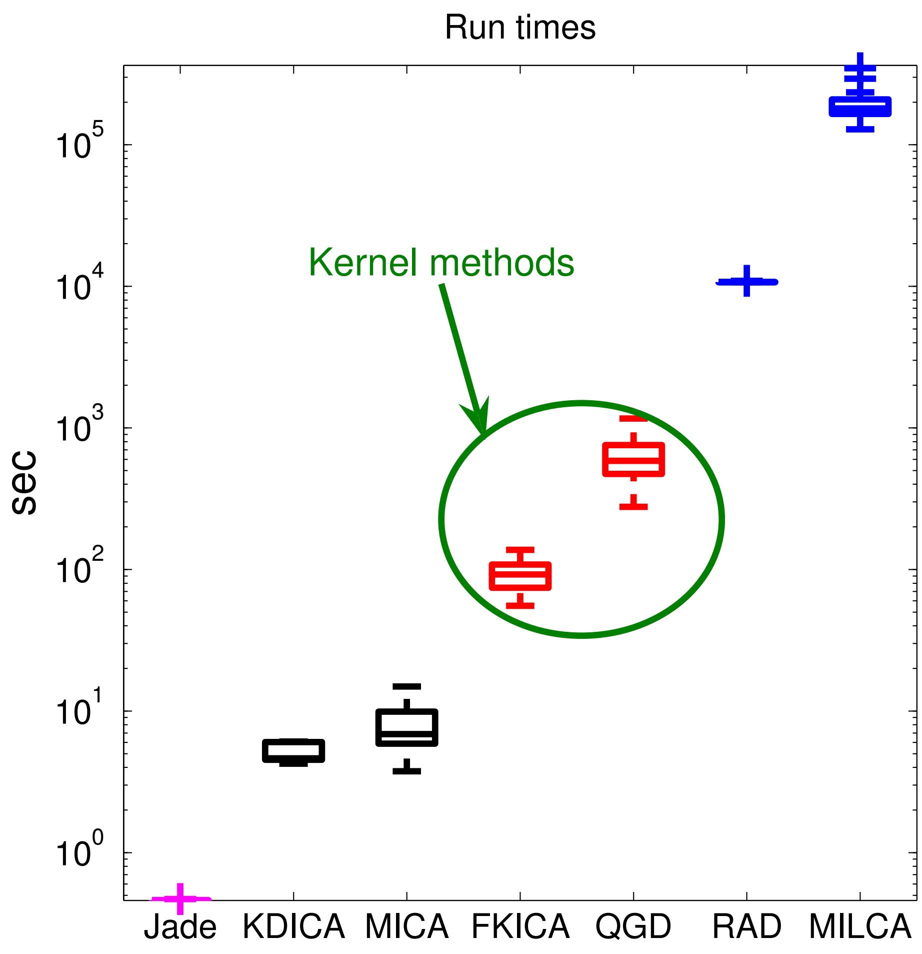




A basic benchmark: computational cost

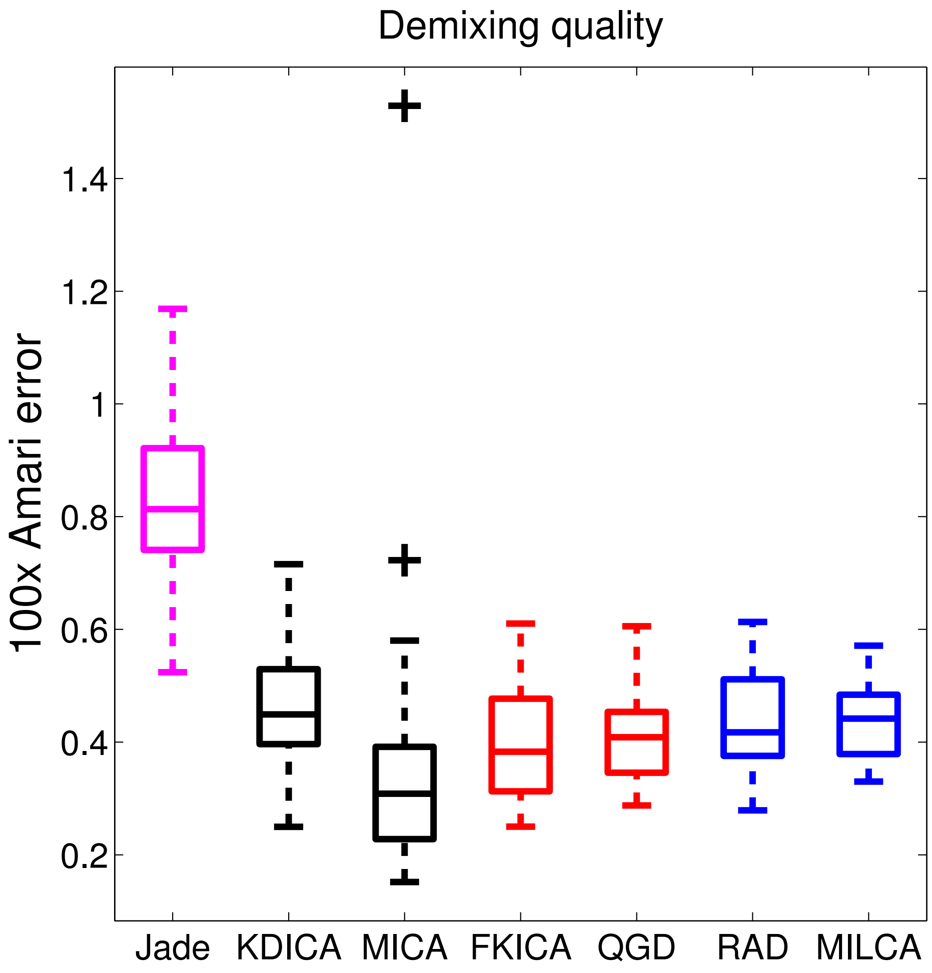
Kernel methods: Newton outperforms Gradient Descent

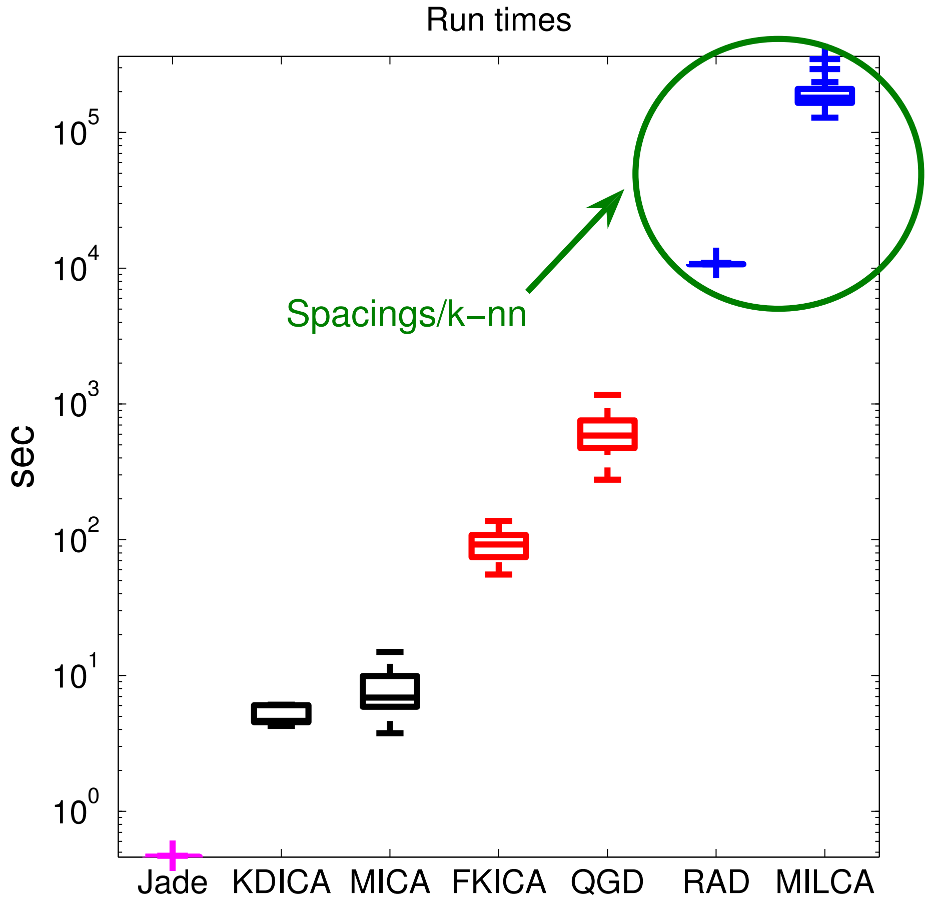




A basic benchmark: computational cost

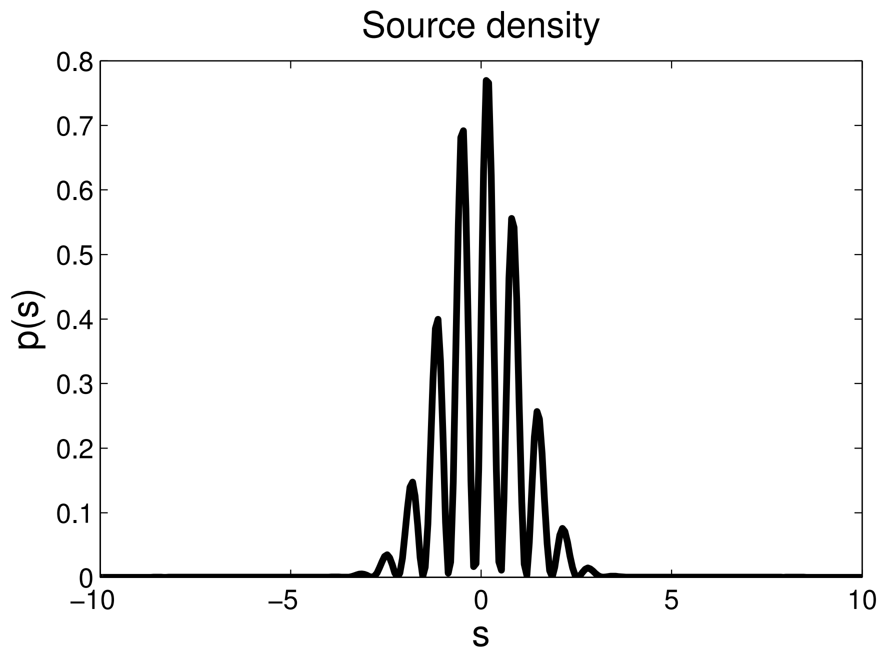
Spacings/*k*-nn entropy contrasts slowest

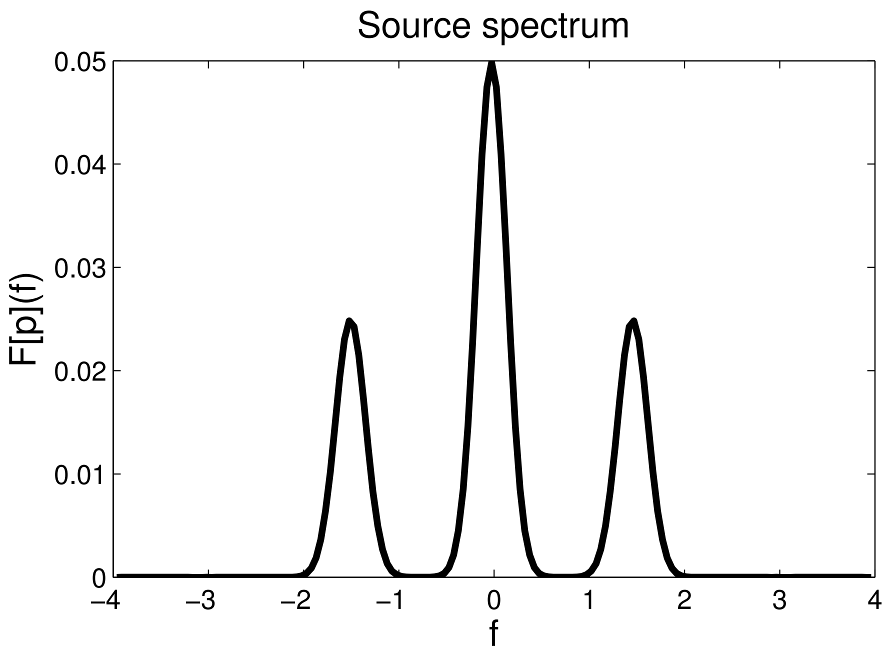




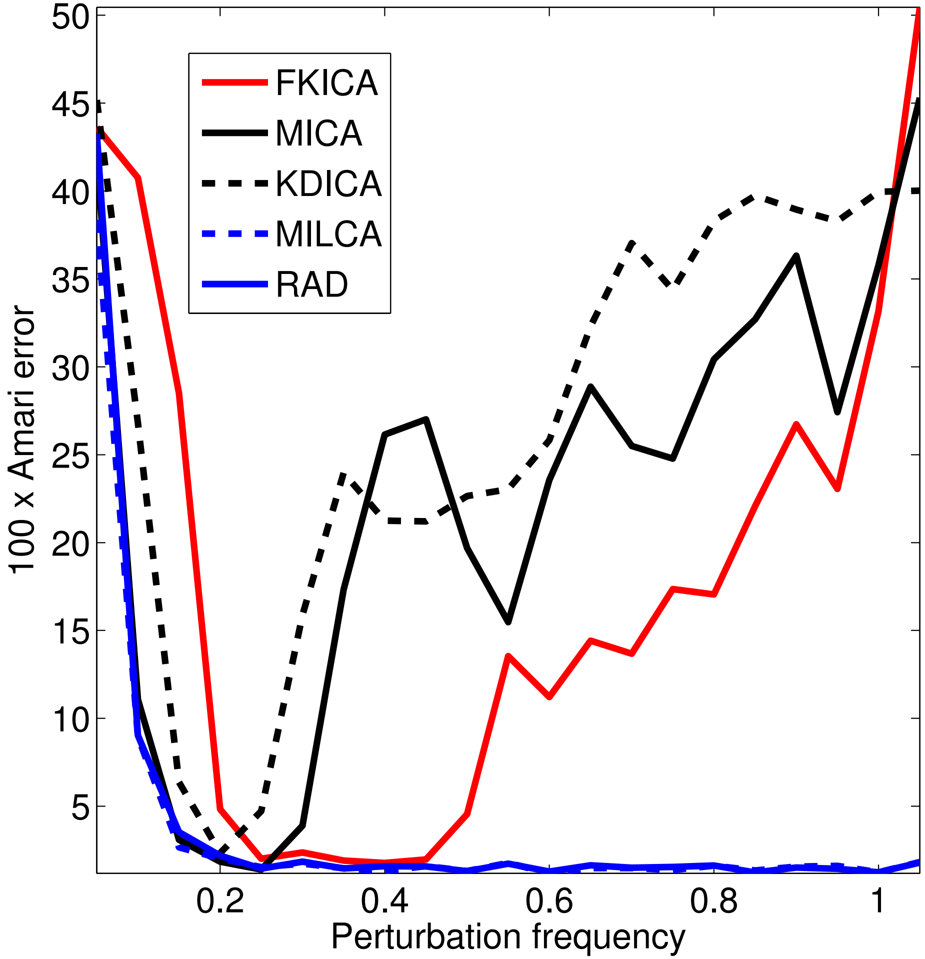
High frequency perturbations

- Two sources, sinusoidal perturbations to Gaussian
- Random mixing angle.
- Results averaged over 25 datasets, $m = 1000$





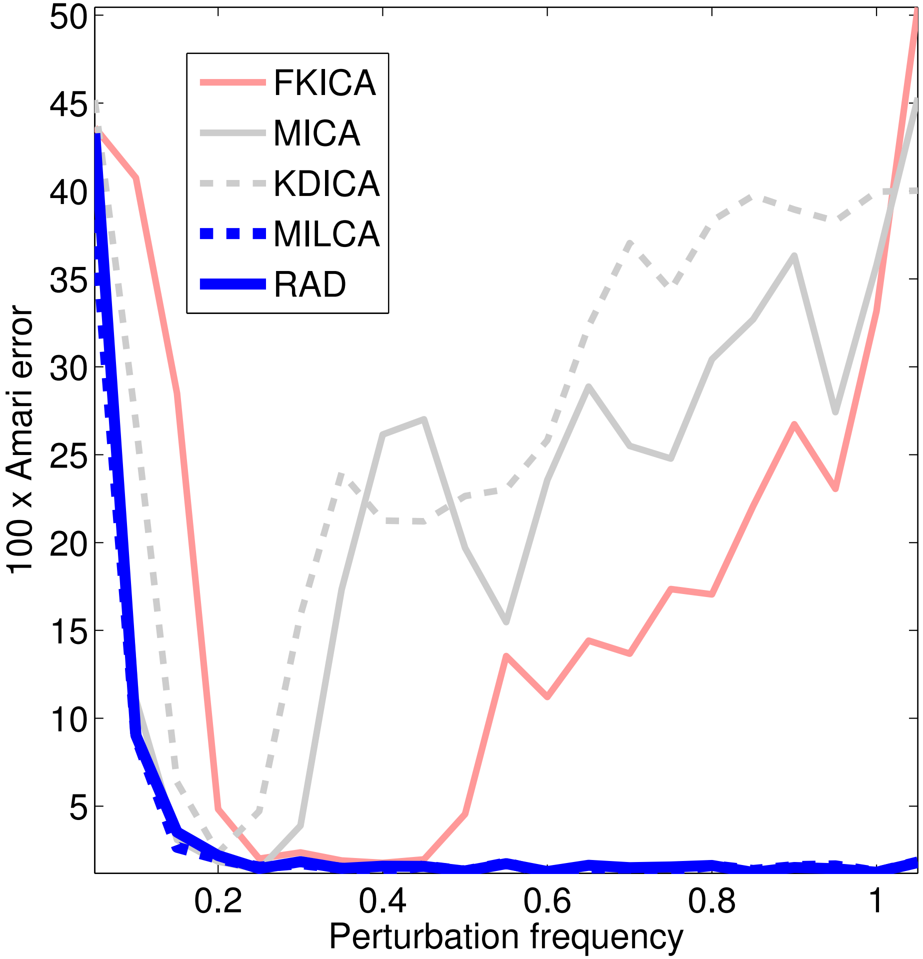
High frequency perturbations



High frequency perturbations

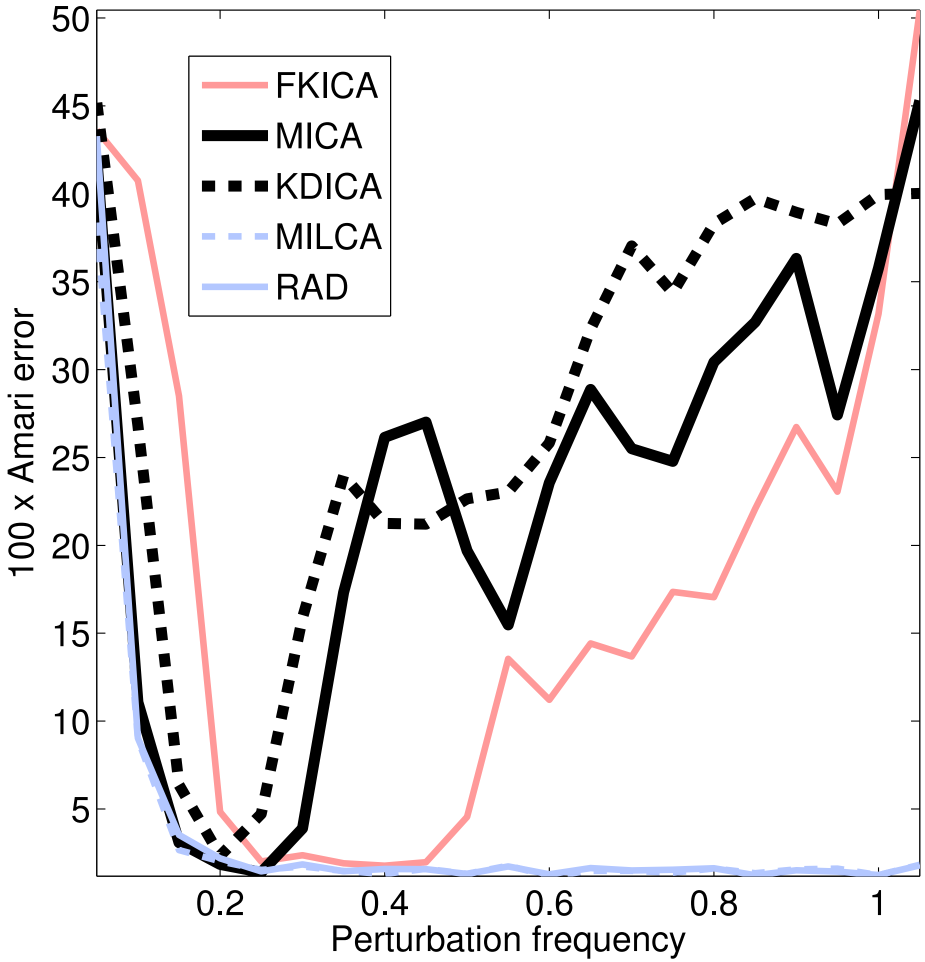
Spacings/k-nn methods perform best

(but slow)



High frequency perturbations

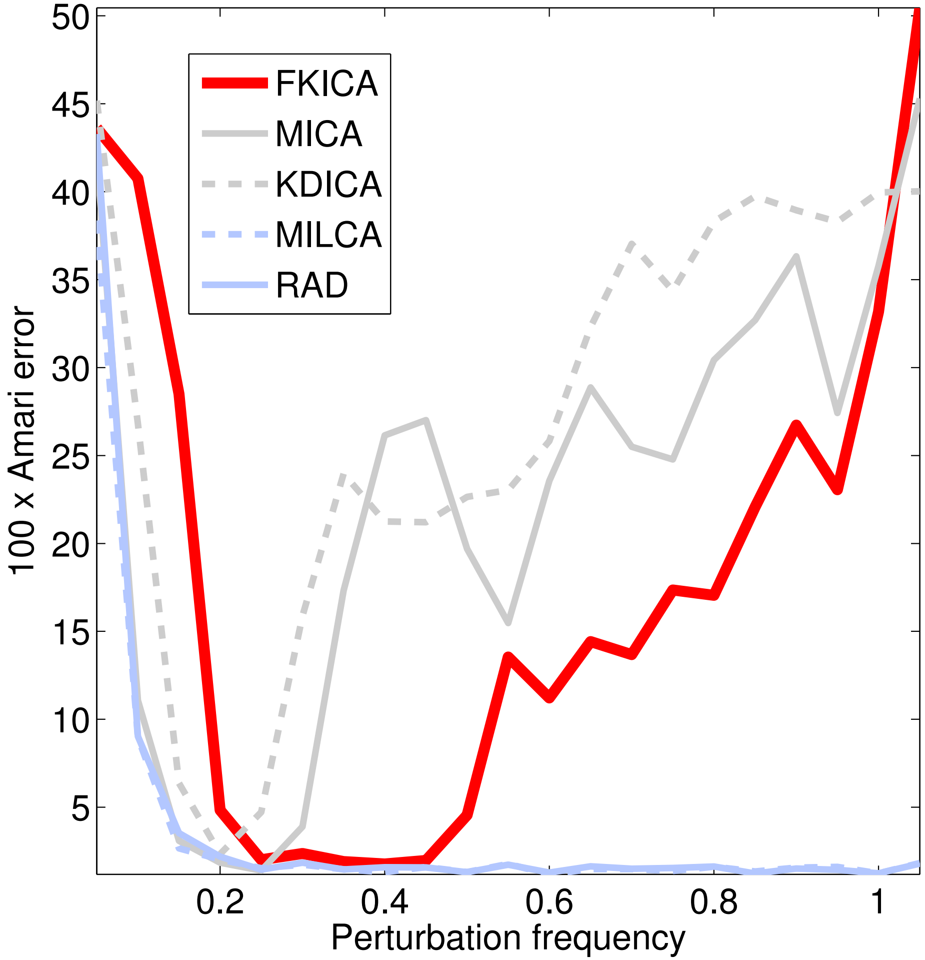
Fast entropy estimates: narrowest range



High frequency perturbations

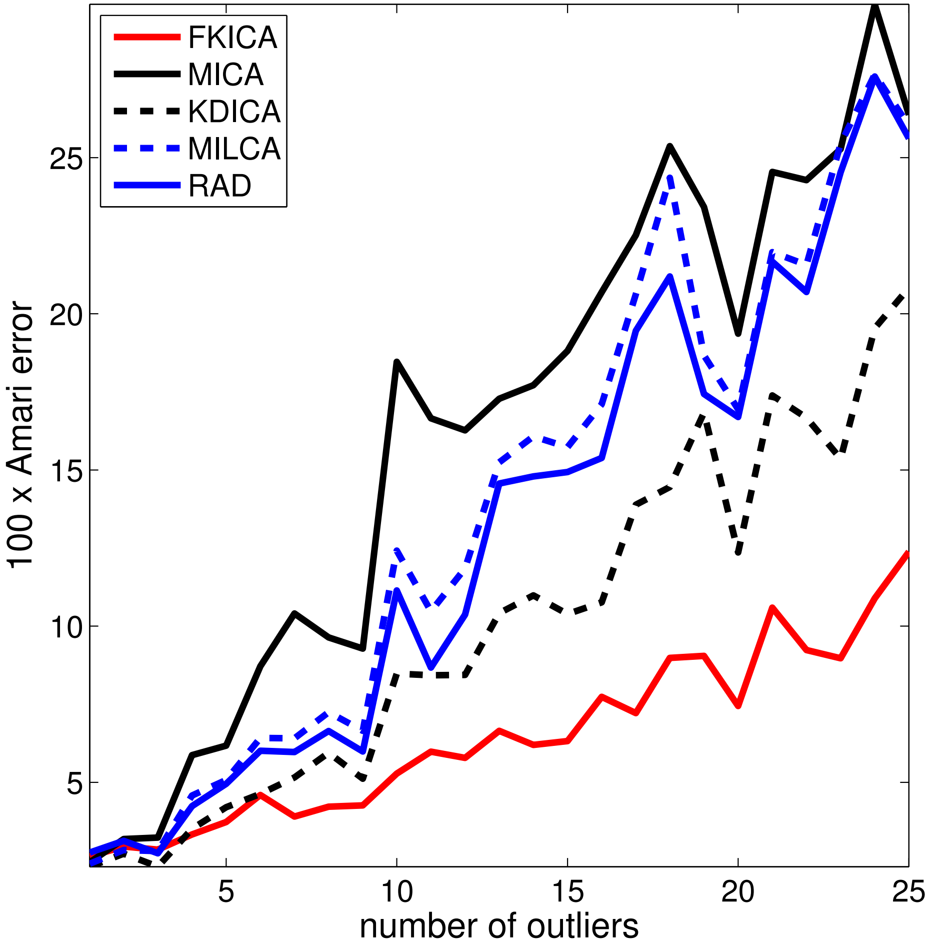
Fast Kernel ICA: peforms in between

(good performance/runtime tradeoff)



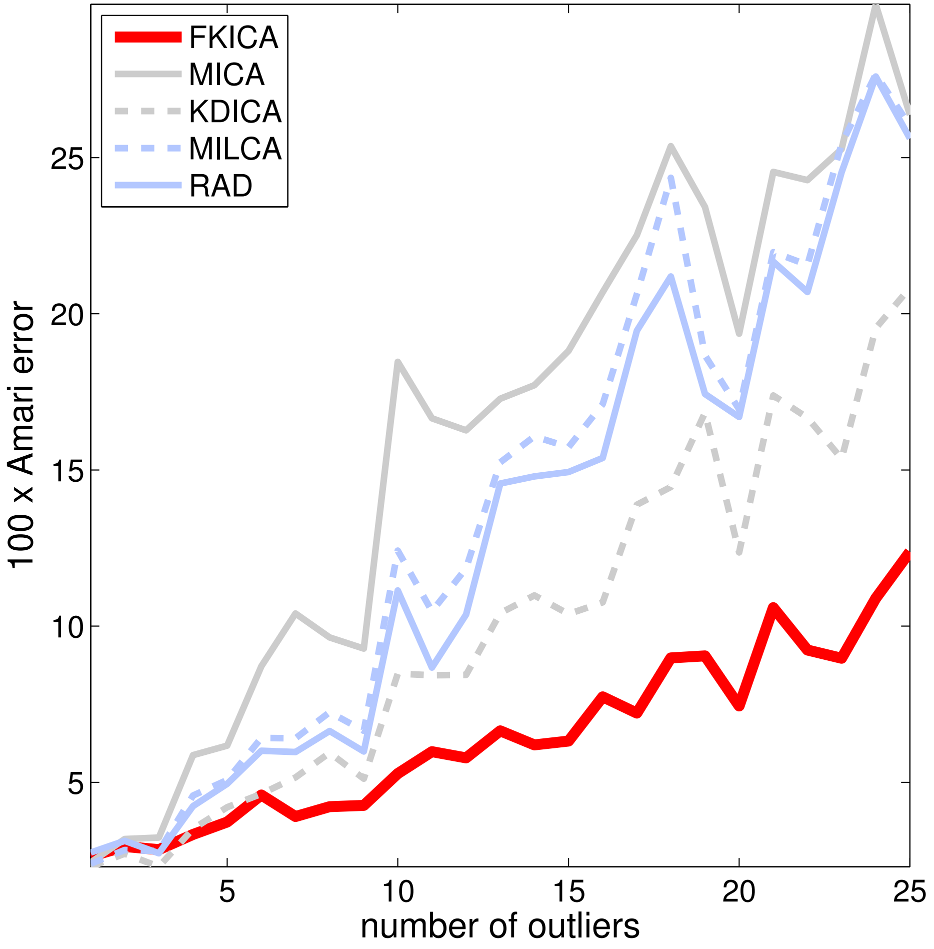
Outlier resistance

Two sources, outliers added to both *mixtures*



Outlier resistance

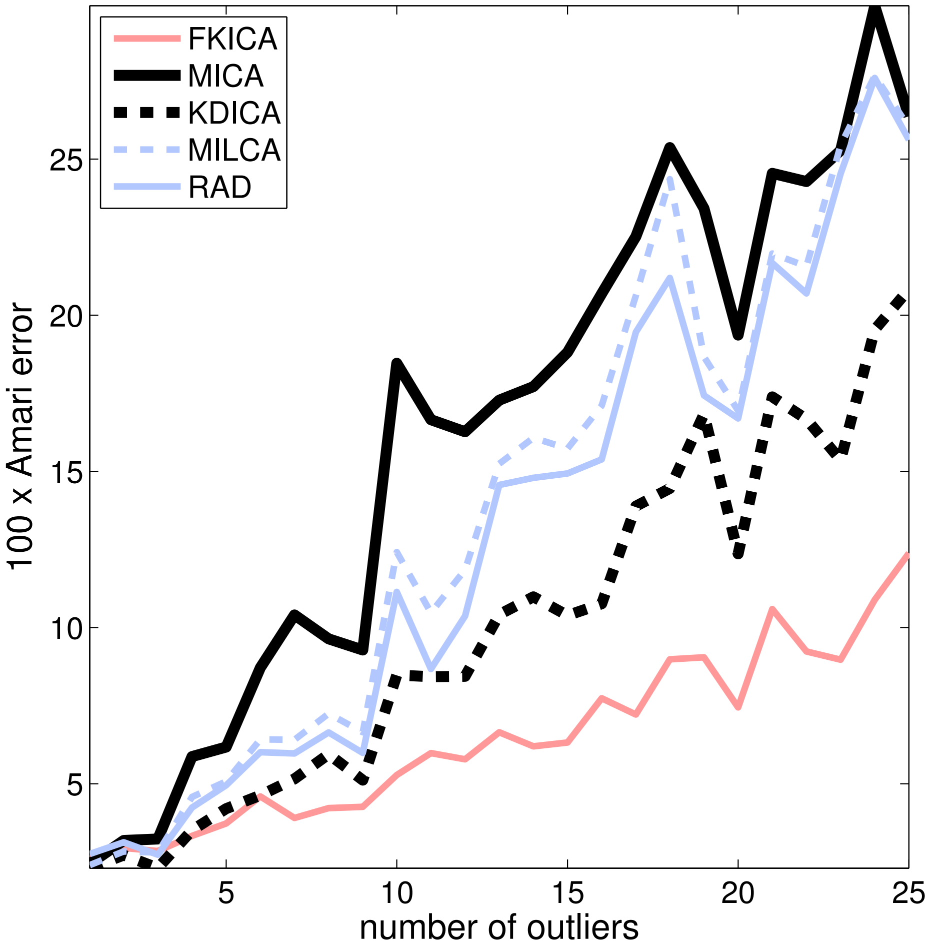
Kernel ICA performs best



Outlier resistance

Fast entropy estimates: less good

KDICA initialized with kernel ICA solution!



ICA algorithm choice

- Choosing kernel ICA approach
  - Fastest (by far): Fast ICA [Hyvärinen et al., 2001], Jade [Cardoso, 1998]
  - Good tradeoff between speed and performance: MICA [Pham, 2004]
  - Tricky cases (outliers, non-smooth sources): Fast KICA [Shen et al., 2007, 2009]
  - Small sample size: KGV very good [Bach and Jordan, 2002]

ICA algorithm choice

- Choosing kernel ICA approach
  - Fastest (by far): Fast ICA [Hyvärinen et al., 2001], Jade [Cardoso, 1998]
  - Good tradeoff between speed and performance: MICA [Pham, 2004]
  - Tricky cases (outliers, non-smooth sources): Fast KICA [Shen et al., 2007, 2009]
  - Small sample size: KGV very good [Bach and Jordan, 2002]
- Some further hints:
  - Use multiple restarts (non-convex)
  - Independence test to check answer

ICA algorithm choice

- Choosing kernel ICA approach
  - Fastest (by far): Fast ICA [Hyvärinen et al., 2001], Jade [Cardoso, 1998]
  - Good tradeoff between speed and performance: MICA [Pham, 2004]
  - Tricky cases (outliers, non-smooth sources): Fast KICA [Shen et al., 2007, 2009]
  - Small sample size: KGV very good [Bach and Jordan, 2002]
- Some further hints:
  - Use multiple restarts (non-convex)
  - Independence test to check answer
- Comparing (usually fixed contrast) algorithms:
  - One approach "better" than another?
  - Example: sources l very large, samples m small (wrt l), e.g. microarray data [Lee and Batzoglou, 2003]

Selected ICA references

- Start with Cardoso's excellent introduction [Cardoso, 1998], and the book by Hyvärninen *et al.* [Hyvärinen et al., 2001]
- Fast kernel ICA is described in [Shen et al., 2007, 2009]. Characteristic function-based ICA is described in [Eriksson and Koivunen, 2003, Chen and Bickel, 2005]. For earlier kernel ICA methods, see [Bach and Jordan, 2002, Gretton et al., 2005]
- Mutual information/entropy based: [Pham, 2004, Learned-Miller and Fisher III, 2003, Stögbauer et al., 2004, Chen, 2006]
- Classic algorithms for *time series* separation with second order methods (not covered much in this talk): [Molgedey and Schuster, 1994, Belouchrani et al., 1997]
- An important paper for optimising over orthogonal matrices: [Edelman et al., 1998]. The Newton-like method: [Hüper and Trumpf, 2004].

Conclusion

- With RKHS distribution embeddings, compare distributions in high dimensions and on structured objects
  - Easier than density estimation

Conclusion

- With RKHS distribution embeddings, compare distributions in high dimensions and on structured objects
  - Easier than density estimation
- It is easy to check whether distribution embeddings are unique
  - Characteristic kernel: check Fourier transform
  - Any difference in distributions detectable

Conclusion

- With RKHS distribution embeddings, compare distributions in high dimensions and on structured objects
  - Easier than density estimation
- It is easy to check whether distribution embeddings are unique
  - Characteristic kernel: check Fourier transform
  - Any difference in distributions detectable
- Can use HSIC dependence measure for feature relevance
  - Feature selection
  - Taxonomy fitting
- More: conditional dependence tests, independent component analysis, covariate shift correction,. . .

References from my publications

- MMD a distance between distributions [ISMB06, NIPS06a, JMLR10, JMLR12a]
  - high dimensionality
  - non-euclidean data (strings, graphs)
  - Nonparametric hypothesis tests
- Measure and test independence [ALT05, NIPS07a, NIPS07b, ALT08, JMLR10, JMLR12a]
- Characteristic RKHS: MMD a metric [NIPS07b, COLT08, NIPS08a]
  - Easy to check: does spectrum cover $\mathbb{R}^d$
- Applications:
  - Feature selection [ISMB07, ICML07a, JMLR12b]
  - Clustering and taxonomy discovery [ICML07b, NIPS08b]
  - Covariate shift correction [NIPS06b, Book Ch. 08] , testing conditional dependence [NIPS07b] , independent component analysis [JMLR05, Book Ch. 07, AISTATS07, IEEE TSP 09] , . . .

References

F. R. Bach and M. I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.

J. Bedo, C. Sanderson, and A. Kowalczyk. An efficient alternative to SVM based recursive feature elimination with applications in natural language processing and bioinformatics. In *Artificial Intelligence*, 2006.

A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines. A blind source separation technique using second order statistics. *IEEE Transactions on Signal Processing*, 45(2):434–444, 1997.

M. Bethge. Factorial coding of natural images: how effective are linear models in removing higher-order dependencies? *Journal of the Optical Society of America A*, 23(6):1253–1268, 2006.

V. Calhoun, T. Adali, L. Hansen, J. Larsen, and J. Pekar. Ica of functional mri data: An overview. In *Proc. 4th Int. Symp. on Independent Component Analysis and Blind Signal Separation (ICA2003)*, pages 281–288, 2003.

J.-F. Cardoso. Blind signal separation: Statistical principles. *Proceedings of the IEEE, Special Issue on Blind Identification and Estimation*, 86(10):2009–2025, 1998.

A. Chen. Fast kernel density independent component analysis. In *Sixth International Conference on ICA and BSS*, volume 3889, pages 24–31, Berlin/Heidelberg, 2006. Springer-Verlag.

A. Chen and P. J. Bickel. Consistent independent component analysis and prewhitening. *IEEE Transactions on Signal Processing*, 53(10):3625–3632, 2005.

A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, 1998.

L. Ein-Dor, O. Zuk, and E. Domany. Thousands of samples are needed to generate a robust gene list for predicting outcome in cancer. *Proc. Natl. Acad. Sci. USA*, 103(15):5923–5928, Apr 2006.

J. Eriksson and K. Koivunen. Characteristic-function based independent component analysis. *Signal Processing*, 83(10):2195–2208, 2003.

A. Gretton, R. Herbrich, A. J. Smola, O. Bousquet, and B. Schölkopf. Kernel methods for measuring independence. *Journal of Machine Learning Research*, 6:2075–2129, 2005.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, New York, 2001.

K. Hüper and J. Trumpf. Newton-like methods for numerical optimisation on manifolds. In *Proceedings of Thirty-eighth Asilomar Conference on Signals, Systems and Computers*, pages 136–139, 2004.

A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley, New York, 2001.

T.-P. Jung, S. Makeig, C. Humphries, T.-W. Lee, M. McKeown, V. Iragui, and T. Sejnowski. Removing electroencephalographic artifacts by blind source separation. *Psychophysiology*, 37:163–178, 2000.

K. Kiviluoto and E. Oja. Independent component analysis for parallel financial time series. In *In Proc. Int. Conf. on Neural Information Processing (ICONIP)*, volume 2, pages 895–898, 1998.

E. G. Learned-Miller and J. W. Fisher III. ICA using spacings estimates of entropy. *Journal of Machine Learning Research*, 4:1271–1295, 2003.

S.I. Lee and S. Batzoglou. Application of independent component analysis to microarrays. *Genome Biology*, 4(11):R76, 2003.

F. Li and Y. Yang. Analysis of recursive gene selection approaches from microarray data. *Bioinformatics*, 21(19):3741–3747, Oct 2005.

L. Molgedey and H. Schuster. Separation of a mixture of independent signals using time delayed correlation. *Physical Review Letters*, 72(23):3634–3637, 1994.

D.-T. Pham. Fast algorithms for mutual information based independent component analysis. *IEEE Transactions on Signal Processing*, 52(10):2690–2700, 2004.

H. Shen and K. Hüper. Newton-like methods for parallel independent component analysis. In *MLSP 16*, pages 283–288, Maynooth, Ireland, 2006.

H. Shen, S. Jegelka, and A. Gretton. Fast kernel ICA using an approximate Newton method. In *AISTATS 11*, pages 476–483. Microtome, 2007.

H. Shen, S. Jegelka, and A. Gretton. Fast kernel-based independent component analysis. *IEEE Transactions on Signal Processing*, 57:3498 – 3511, 2009.

H. Stögbauer, A. Kraskov, S. Astakhov, and P. Grassberger. Least dependent component analysis based on mutual information. *Phys. Rev. E*, 70(6): 066123, 2004.

R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu. Diagnosis of multiple cancer types by shrunken centroids of gene expression. In *National Academy of Sciences*, volume 99, pages 6567–6572, 2002.

R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu. Class prediction by nearest shrunken centroids, with applicaitons to dna microarrays. *Stat Sci*, 18:104–117, 2003.

L. J. van't Veer, H. Dai, M. J. van de Vijver, Y. D. He, A. A. M. Hart, et al. Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415:530–536, 2002.