Linear Models: FA, ICA, NFA

Shikui Tu

Department of Computer Science and Engineering, Shanghai Jiao Tong University

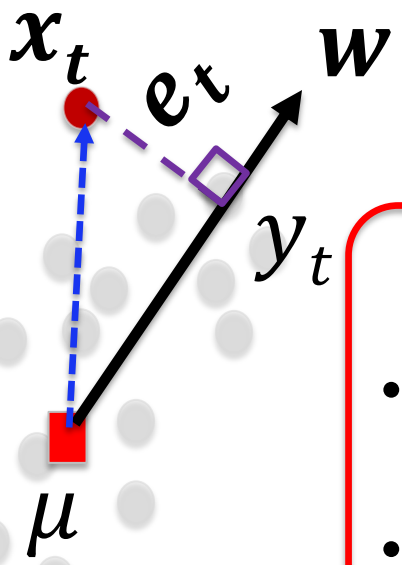
2018-03-29

Outline

- FA and PCA
- Independent Component Analysis (ICA)
- Independent FA (IFA), Non-Gaussian FA (NFA)
- Recent papers related to PCA/ICA/GMM

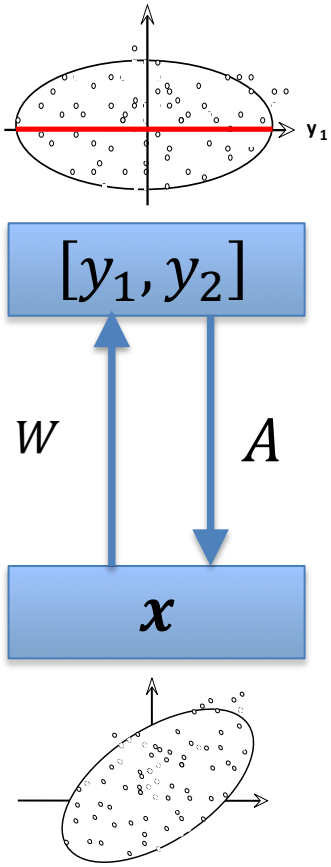
Generative model perspective

Continuous latent variable $\boldsymbol{y}$



For the *t*-th data point:

- Randomly sample a y_t:
  $\boldsymbol{y}_t \sim G(\boldsymbol{y}|\boldsymbol{0}, \Sigma_y)$;
- Randomly generate a noise e_t
  $\boldsymbol{e}_t \sim G(\boldsymbol{e}|0, \sigma^2 I)$
- Generate $\boldsymbol{x}_t$ by:
  $\boldsymbol{x}_t = A\boldsymbol{y}_t + \boldsymbol{\mu} + \boldsymbol{e}_t$

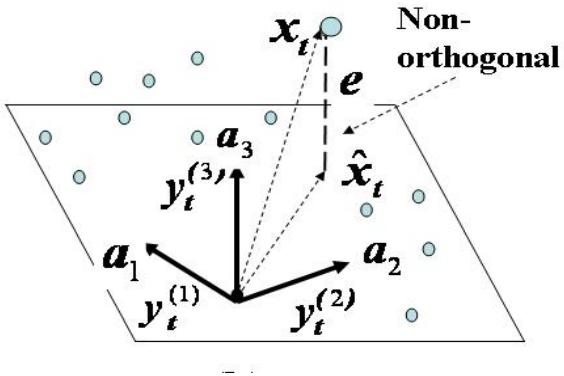


$$||\boldsymbol{w}|| = 1$$

$$y_t = \boldsymbol{x}_t^T \boldsymbol{w}$$

$$\boldsymbol{e}_t = ||\boldsymbol{x}_t - y_t \boldsymbol{w}||^2$$

Factor Analysis (FA) Model



$A^T A = I$ has been removed because it impedes $\sum_t \|e_t\|^2$ to reach its minimum

Indeterminacy

e) a rotation matrix since $A' = \Phi A$ spans the same subspace;

f) a diagonal D with $A' = AD$, $y' = D^{-1} y$.

Assumption

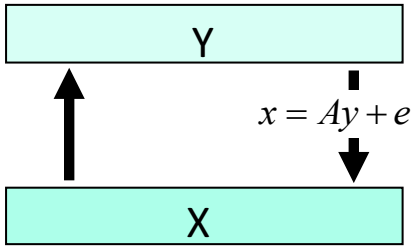
c) $\mathrm{E} e y^T = 0$ (i.e., not correlated) it is not longer a consequenc e;

f) $\mathrm{E} y y^T = \Lambda$ is diagonal !

$$q(y) = \prod_{j=1}^{k} G(y_j | 0,1) = G(y | 0, I)$$

Two Choices

$$q(y) = \begin{cases} G(y|0,I), \text{ choice (a)} \\ G(y|0,\Lambda), \text{ choice (b)} \end{cases}$$



g) a unknown allocation between th e two additive terms $Exx^T = A^T \Lambda A + Eee^T$.

- 样本方差之分割不变性（样本方差在子空间内外可任意分割）
- 超阈维数不变性（高于某维的子空间以零误差描述有限样本集）

EM algorithm for FA

E-Step: $p^{old}(\boldsymbol{y}|\boldsymbol{x}) = \dfrac{G(\boldsymbol{y}|0,I)G(\boldsymbol{x}|A\boldsymbol{y}+\boldsymbol{\mu},\sigma^2 I)}{G(\boldsymbol{x}|\boldsymbol{\mu},AA^T+\sigma^2 I)}$

$$E[\boldsymbol{y}|\boldsymbol{x}] = W\boldsymbol{x} \qquad W = A^T(AA^T+\sigma^2 I)^{-1}$$

$$E[\boldsymbol{y}\boldsymbol{y}^T|\boldsymbol{x}] = I - WA + W\boldsymbol{x}\boldsymbol{x}^T W^T$$

M-Step: max $Q\left(p^{old}(\boldsymbol{y}|\boldsymbol{x}),\Theta\right)$

$$Q = \int p^{old}(\boldsymbol{y}|\boldsymbol{x}) \cdot \ln[G(\boldsymbol{y}|0,I)G(\boldsymbol{x}|A\boldsymbol{y}+\boldsymbol{\mu},\sigma^2 I)]\, d\boldsymbol{y}$$

$$A^{new} = \left(\sum_{t=1}^{N} \boldsymbol{x}_t (E[\boldsymbol{y}|\boldsymbol{x}_t])^T\right)\left(\sum_{t=1}^{N} E[\boldsymbol{y}\boldsymbol{y}^T|\boldsymbol{x}_t]\right)^{-1}$$

$$\sigma^{2^{new}} = \frac{1}{Nd} Tr\left\{\sum_{t=1}^{N} \{\boldsymbol{x}_t\boldsymbol{x}_t^T - A^{new}E[\boldsymbol{y}|\boldsymbol{x}_t]\boldsymbol{x}_t^T\}\right\}$$

Maximum likelihood FA implements PCA

$$p(\boldsymbol{y}) = G(\boldsymbol{y}|\boldsymbol{0}, I), \qquad p(\boldsymbol{x}|\boldsymbol{y}) = G(\boldsymbol{x}|\boldsymbol{A}\boldsymbol{y} + \boldsymbol{\mu}, \Sigma_e),$$

$$p(\boldsymbol{x}|\Theta) = \int p(\boldsymbol{y})p(\boldsymbol{x}|\boldsymbol{y})d\boldsymbol{y} = G\left(\boldsymbol{x}|\boldsymbol{\mu}, \mathrm{AA}^{\mathrm{T}} + \Sigma_e\right),$$

$$\max_{\Theta} \log\left\{\prod_{t=1}^{N=1} p(\boldsymbol{x_t}|\Theta)\right\}$$

Maximum Likelihood

$$\boldsymbol{\Sigma}_e = \sigma_e^2 \mathbf{I}_n$$

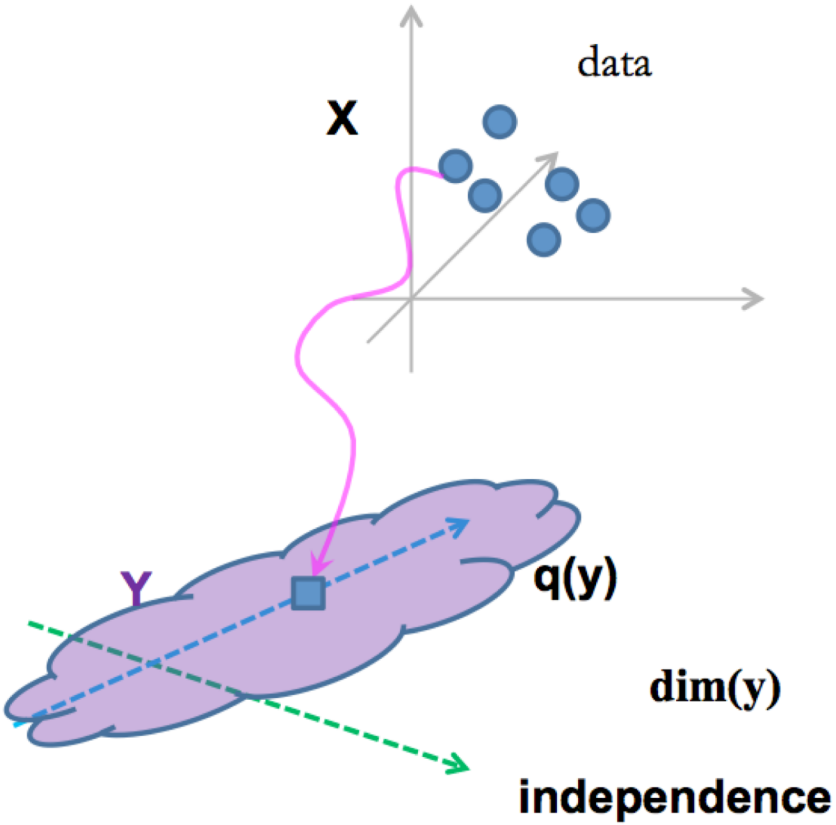
assume $\boldsymbol{\mu} = \mathbf{0}$

PCA

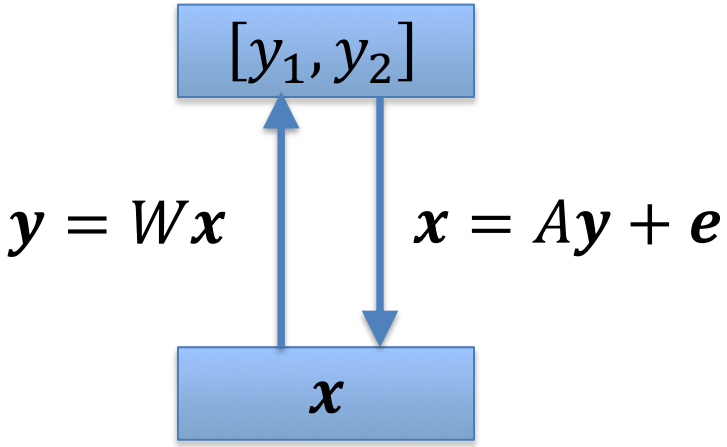
$$\begin{cases} \hat{\mathbf{A}}_{n\times m}^{ML} = \mathbf{U}_{n\times m}(\mathbf{D}_m - \hat{\sigma}_e^2)^{\frac{1}{2}}\mathbf{R}^T, \quad \mathbf{D}_m = diag[s_1, \ldots, s_m] \\ \hat{\sigma}_e^{2,ML} = \frac{1}{n-m}\sum_{i=m+1}^{n} s_i, \end{cases}$$

U is eigenvectors of sample cov.

Dimensionality reduction



$m = \dim(\boldsymbol{y}) = ?$



$n = \dim(\boldsymbol{x}) > m$

Determining *m* is a model selection problem.

The effects of dim(y)

$$\boldsymbol{x} = \boldsymbol{a}y_1 + \boldsymbol{a}_2 y_2 + \cdots + \boldsymbol{a}_{m^*} y_{m^*} + \boldsymbol{e}$$

Assume $\boldsymbol{\mu} = \mathbf{0}$

$$\boldsymbol{x} = A\boldsymbol{y} + \boldsymbol{e}$$

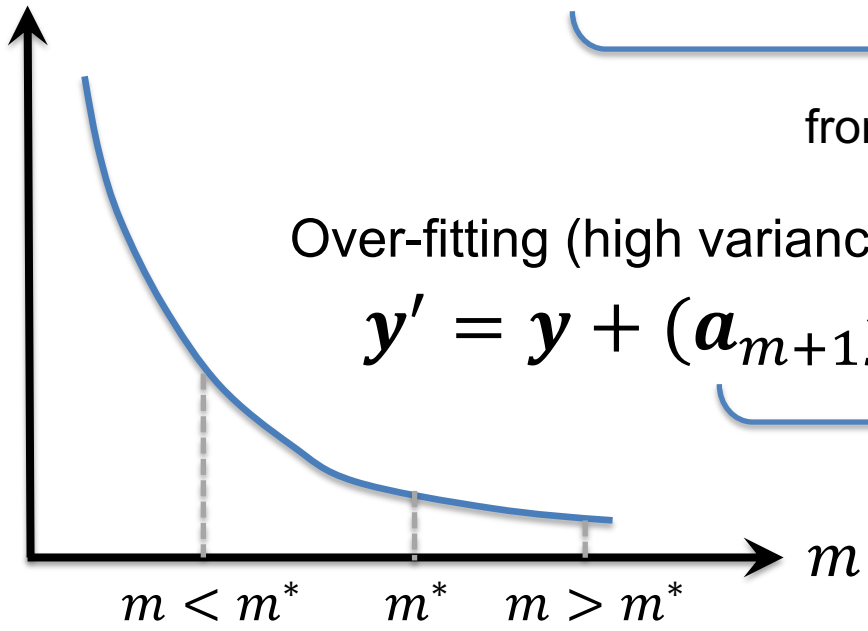
$$\boldsymbol{y} = [y_1, \cdots, y_{m^*}]^T_{m^* \times 1}$$

$$p(\boldsymbol{y}) = G(\boldsymbol{y}|\mathbf{0}, I),$$

Under-fitting (big bias): $m < m^*$

$$\boldsymbol{e}' = \underbrace{\boldsymbol{a}_{m+1} y_{m+1} + \cdots + \boldsymbol{a}_{m^*} y_{m^*}}_{\text{from signal } \boldsymbol{y}} + \boldsymbol{e}$$

Fitting error



Over-fitting (high variance): $m < m^*$

$$\boldsymbol{y}' = \boldsymbol{y} + (\underbrace{\boldsymbol{a}_{m+1} y_{m+1} + \cdots + \boldsymbol{a}_{m^*} y_{m^*}}_{\text{from noise } \boldsymbol{e}})$$

Bias-variance decomposition

We want to find a function $\hat{f}(y)$, that approximates the true function $f(y)$ as well as possible:

$$x = f(y) + \epsilon$$

$$E[\epsilon] = 0$$
$$E[x] = f$$
$$Var[x] = Var[\epsilon]$$

The expected error

$$E\left[\left(x - \hat{f}(y)\right)^2\right] = E\left[x^2 + \hat{f}^2 - 2x\hat{f}\right]$$

$$= Var[x] + (E[x])^2 + Var\left[\hat{f}\right] + \left(E\left[\hat{f}\right]\right)^2 - 2fE[\hat{f}]$$

$$= Var[x] + Var\left[\hat{f}\right] + \left(f - E\left[\hat{f}\right]\right)^2$$

$$= Var[\epsilon] + Var\left[\hat{f}\right] + Bias\left[\hat{f}\right]^2$$

Model selection for FA

Probabilistic model

$$p(X_N \mid \Theta_K)$$

Candidate models:

$$\Theta_1 \subseteq \Theta_2 \subseteq \cdots \subseteq \Theta_K \subseteq \cdots$$

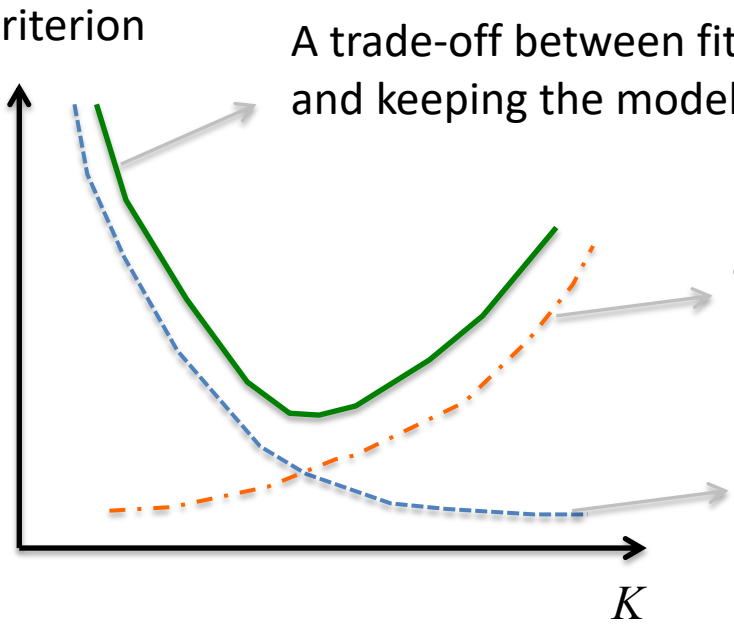
Factor Analysis

$$p(\boldsymbol{y}) = G(\boldsymbol{y}|\boldsymbol{0}, I),$$

$$p(\boldsymbol{x}|\boldsymbol{y}) = G(\boldsymbol{x}|\boldsymbol{A}\boldsymbol{y} + \boldsymbol{\mu}, \Sigma_e),$$

$$p(\boldsymbol{x}|\Theta) = \int p(\boldsymbol{y})p(\boldsymbol{x}|\boldsymbol{y})d\boldsymbol{y} = G\left(\boldsymbol{x}\middle|\boldsymbol{\mu}, \mathrm{AA}^{\mathrm{T}} + \Sigma_e\right),$$

$$K = m = \dim(\boldsymbol{y})$$



A trade-off between fitting the data well and keeping the model simple

Models become more and more complex as K increases.

Negative log-likelihood (or fitting error) to make sure the model fit the data well.

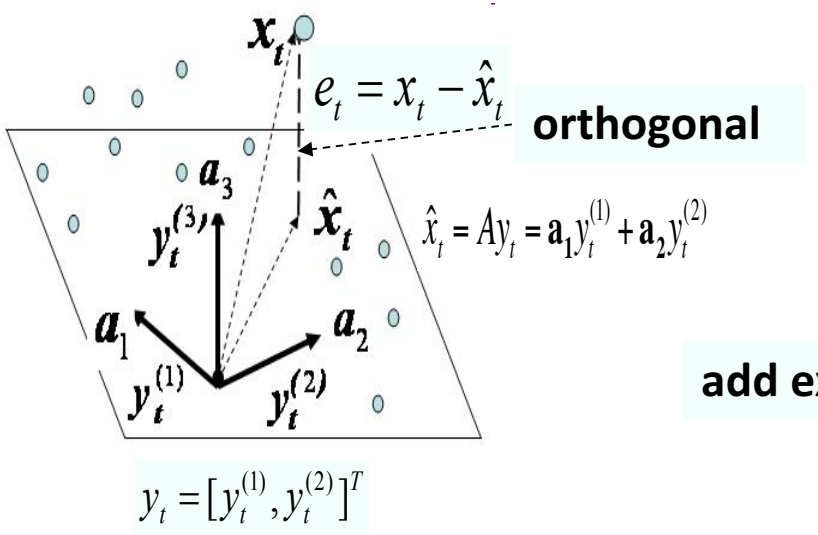
Akaike's Information Criterion (AIC) $\ln p(X_N \mid \hat{\Theta}_K) - d_k$

Bayesian Information Criterion (BIC) $\ln p(X_N \mid \hat{\Theta}_K) - \frac{1}{2} d_k \ln N$

d_k: number of free parameters

N: sample size

Reparameterization of FA



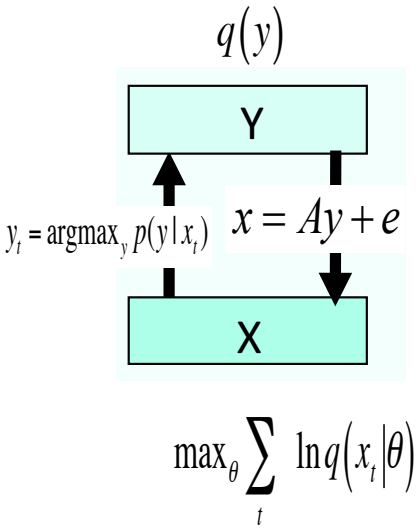
a) *e* and *y* are not correlated

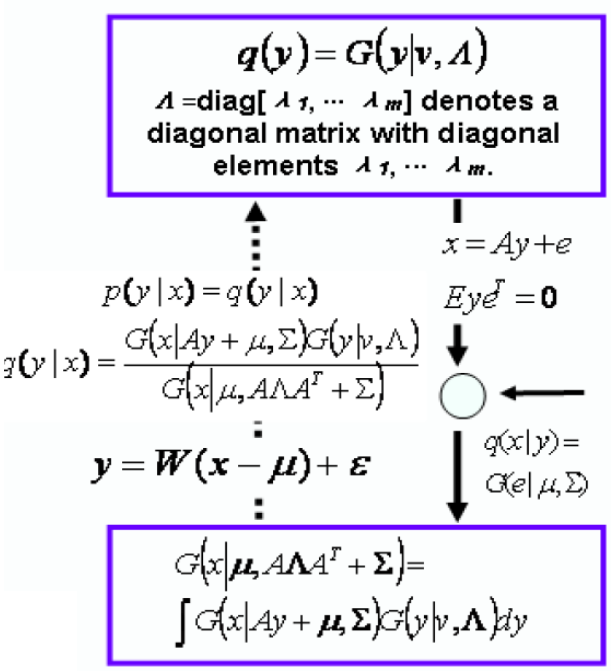
b) $\sum_t \|e_t\|^2$ reaches minimum

c) *e* is a Gaussian noise $G(e|0,\sigma^2 I)$

d) a unique plane but *A* not

add extra constraint I





Two parameterizations of FA

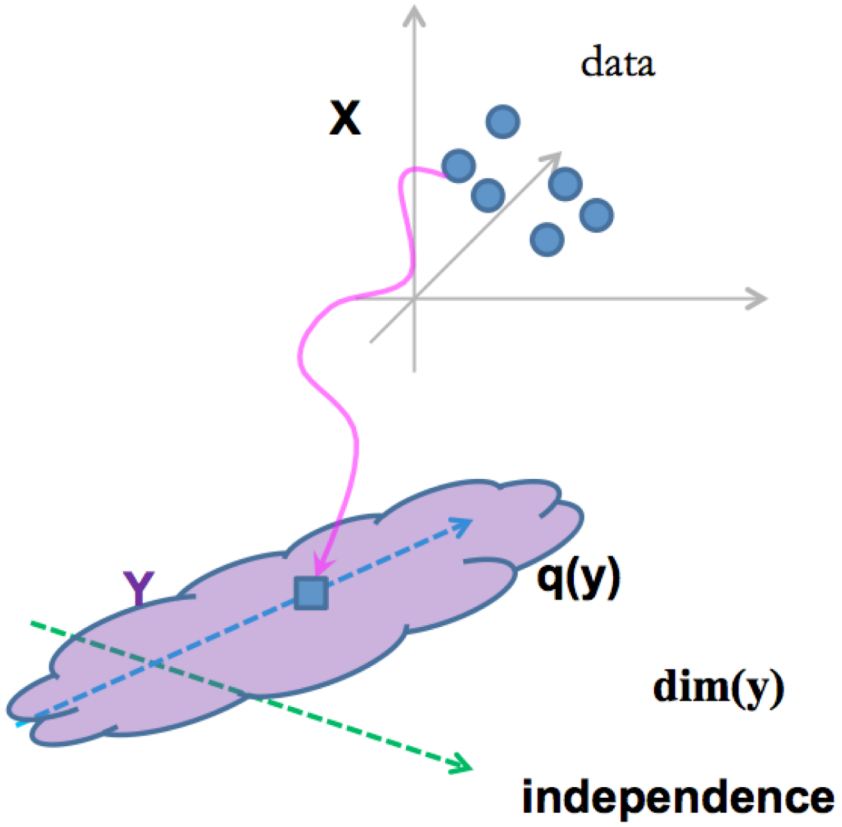
	TYPE-A FA-A: $\Theta_m^a = \{\mathrm{A}, \boldsymbol{\mu}, \Sigma_e\}$	TYPE-B FA-B: $\Theta_m^b = \{\mathrm{U}, \boldsymbol{\mu}, \Lambda, \Sigma_e\}$
$E[\mathrm{ye}^T]$	0 (y and $\mathbf{e}$ uncorrelated)	0 (y and $\mathbf{e}$ uncorrelated)
$q(\mathrm{y}\|\Theta)$	$G(\mathrm{y}\|0, \mathrm{I}_m)$	$G(\mathrm{y}\|0, \Lambda)$, $\Lambda = diag[\lambda_1, \ldots, \lambda_m]$
$\mathbf{A}$	any full column rank matrix	$\mathbf{A} = \mathbf{U}$, $\mathbf{U}^T\mathbf{U} = \mathbf{I}_m$
$q(\mathrm{x}\|\mathrm{y}, \Theta)$	$G(\mathrm{x}\|\mathrm{Ay} + \boldsymbol{\mu}, \Sigma_\mathrm{e})$	$G(\mathrm{x}\|\mathrm{Uy} + \boldsymbol{\mu}, \Sigma_\mathrm{e})$
$q(\mathrm{x}\|\Theta)$	$G(\mathrm{x}\|\boldsymbol{\mu}, \mathrm{AA}^T + \Sigma_e)$	$G(\mathrm{x}\|\boldsymbol{\mu}, \mathrm{U}\Lambda\mathrm{U}^T + \Sigma_\mathrm{e})$

Which one is better?

Outline

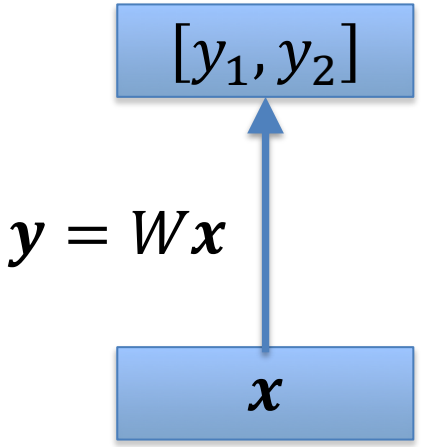
- FA and PCA
- Independent Component Analysis (ICA)
- Independent FA (IFA), Non-Gaussian FA (NFA)
- Recent papers related to PCA/ICA/GMM

Independent Component Analysis (ICA)



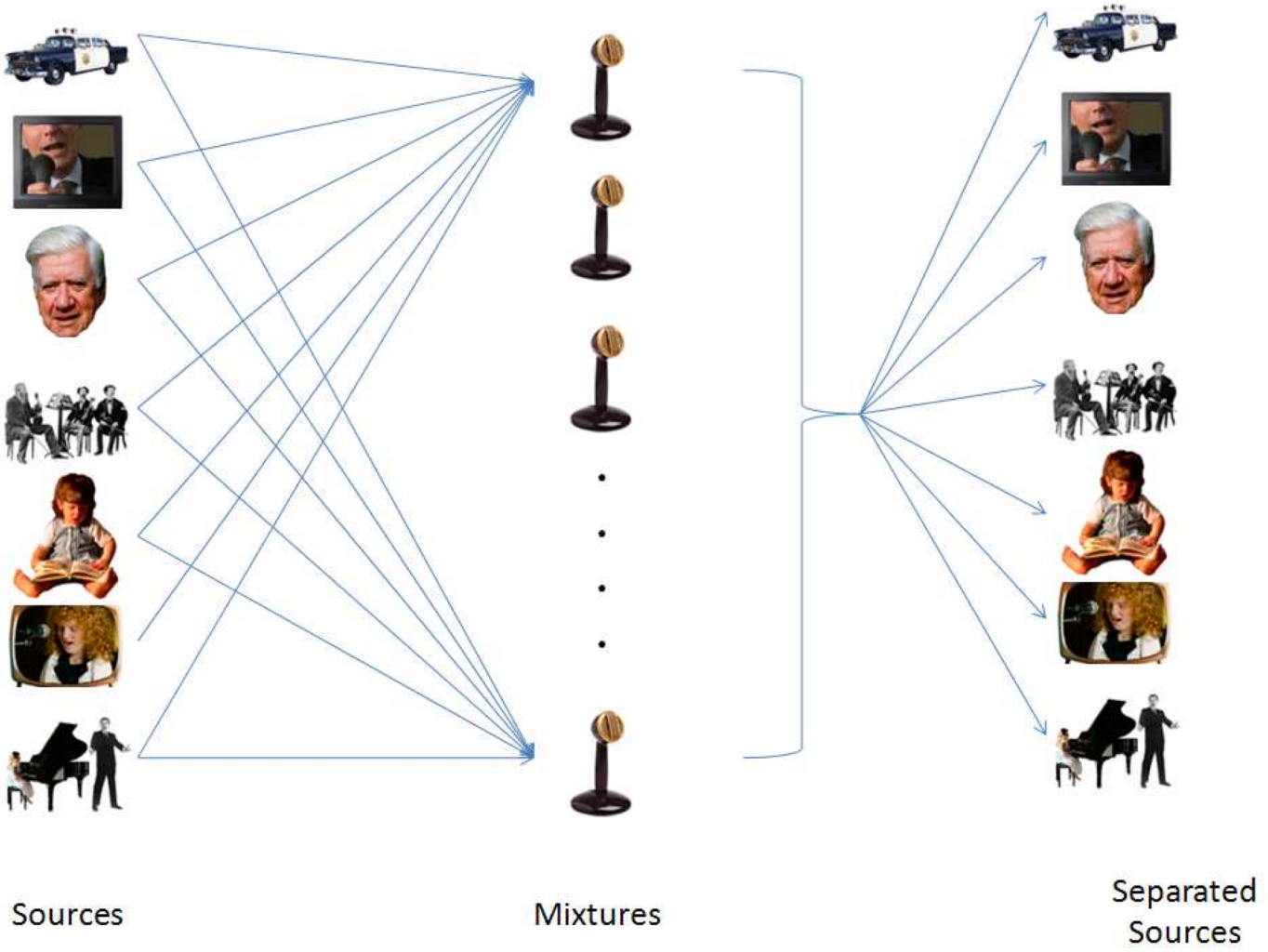
Find W such that

$$p(\boldsymbol{y}) = p(\boldsymbol{y_1}) \cdots p(\boldsymbol{y_m})$$



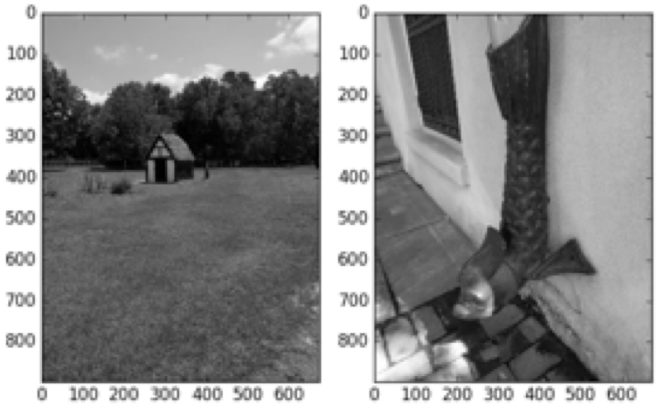
Blind Source Separation (BSS)

Cocktail party problem

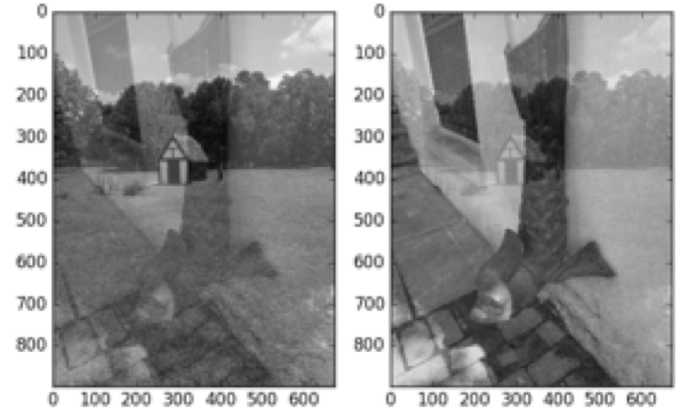


Demo: http://www.kecl.ntt.co.jp/icl/signal/sawada/demo/bss2to4/index.html

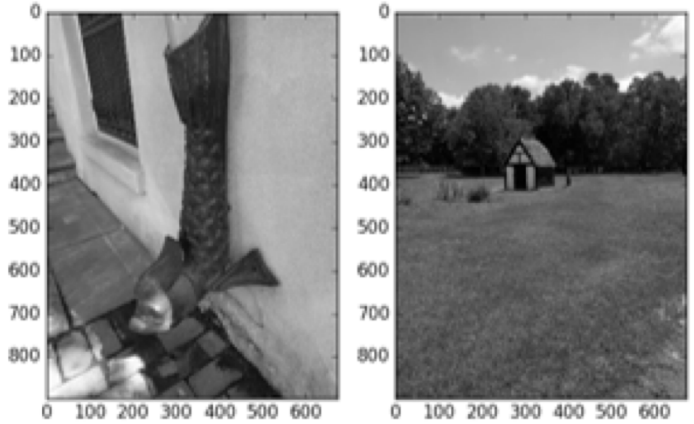
Original Signals



Mixed Signals



Separated signals

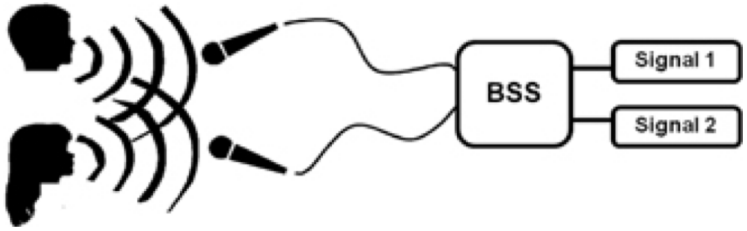


BSS: problem definition

$$\boldsymbol{x} = A\boldsymbol{s}$$

$$x_1(t) = a_{11}s_1 + a_{12}s_2$$

$$x_2(t) = a_{21}s_1 + a_{22}s_2$$



s_1 and s_2

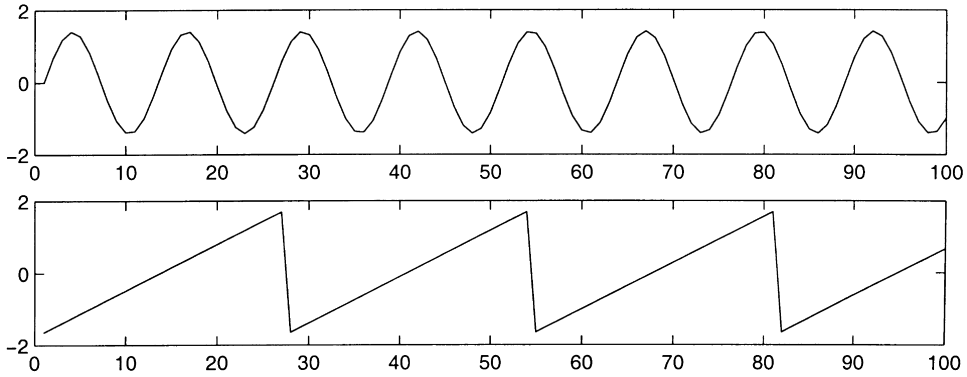


Fig. 1. The original signals.

x_1 and x_2

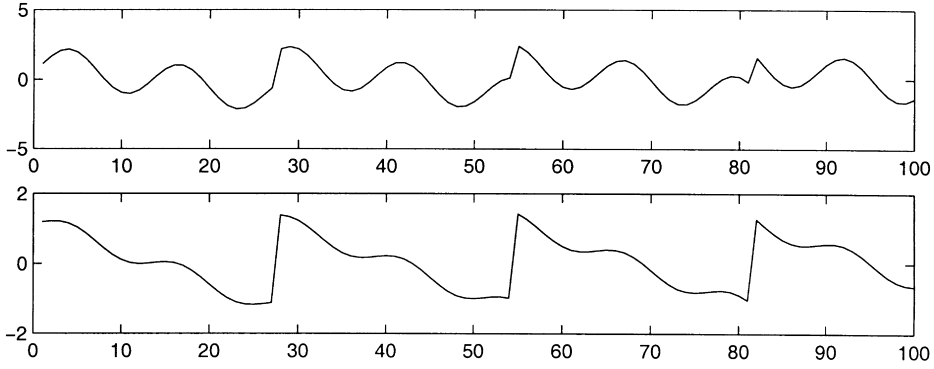


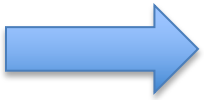
Fig. 2. The observed mixtures of the source signals in Fig. 1.

Task: Find W, and compute $\boldsymbol{s} = W\boldsymbol{x}$

Indeterminacies of ICA

- The variances (energies) of the independent components.

$$\boldsymbol{x} = A\boldsymbol{s} = \sum_{i=1}^{n} \boldsymbol{a} s_i = \sum_{i=1}^{n} (\boldsymbol{a} \cdot \lambda_i^{-1})(\lambda_i s_i)$$



Assume $E[s_i^2] = 1$

- The order of the independent components

$$\boldsymbol{x} = A\boldsymbol{s} = (AP^{-1})(P\boldsymbol{s})$$

P is a permutation matrix

What is independence?

- The variables y_1 and y_2 are said to be independent, if information on the value of y_1 dose not give any information on the value of y_2, and vice versa.

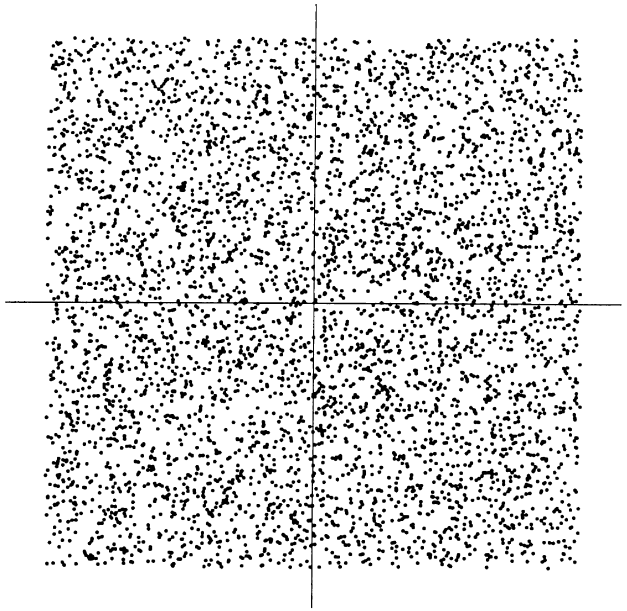


Fig. 5. The joint distribution of the independent components s_1 and s_2 with uniform distributions. Horizontal axis: s_1, vertical axis: s_2.

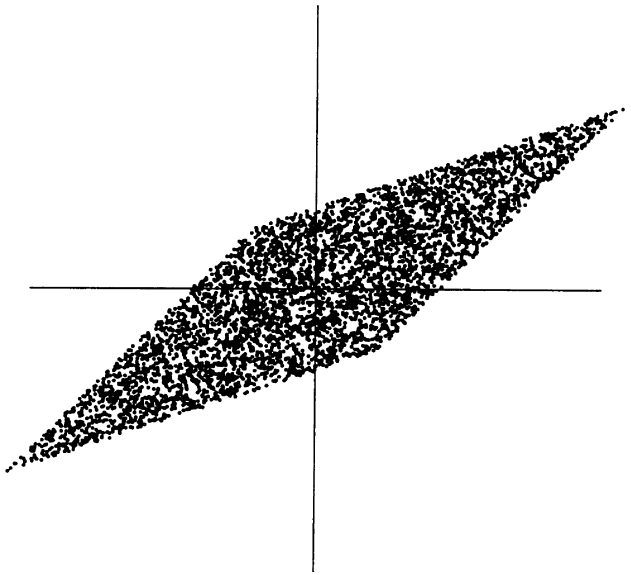


Fig. 6. The joint distribution of the observed mixtures x_1 and x_2. Horizontal axis: x_1, vertical axis: x_2.

What is independence?

Joint distribution is factorizable:

$$p(y_1, y_2) = p_1(y_1)p_2(y_2).$$

Which, for any functions h_1, h_2, implies

$$E\{h_1(y_1)h_2(y_2)\} = E\{h_1(y_1)\}E\{h_2(y_2)\}$$

$$E\{h_1(y_1)h_2(y_2)\} = \int\int h_1(y_1)h_2(y_2)p(y_1, y_2)\mathrm{d}y_1\ \mathrm{d}y_2$$

$$= \int\int h_1(y_1)p_1(y_1)h_2(y_2)p_2(y_2)\mathrm{d}y_1\ \mathrm{d}y_2$$

$$= \int h_1(y_1)p_1(y_1)\mathrm{d}y_1 \int h_2(y_2)p_2(y_2)\mathrm{d}y_2$$

$$= E\{h_1(y_1)\}E\{h_2(y_2)\}.$$

Uncorrelated variables are only partly independent

Uncorrelation: $E\{y_1 y_2\} - E\{y_1\}E\{y_2\} = 0$

Uncorrelation does not imply independence.

$$P\big((y_1, y_2) = (0, +1)\big) = \frac{1}{4}$$

$$P\big((y_1, y_2) = (0, -1)\big) = \frac{1}{4}$$

$$P\big((y_1, y_2) = (+1, 0)\big) = \frac{1}{4}$$

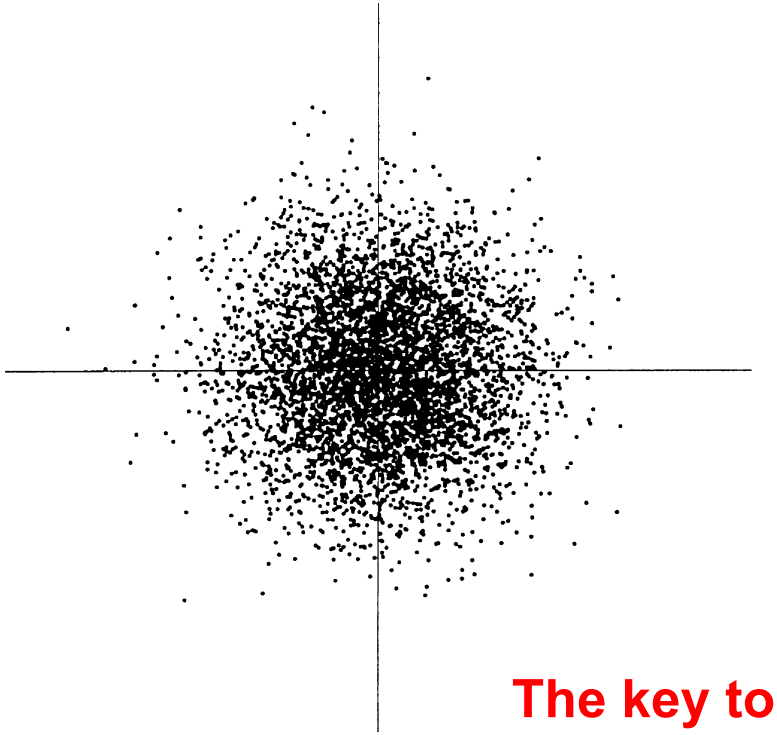
$$P\big((y_1, y_2) = (+1, 0)\big) = \frac{1}{4}$$

不相关只在二维

$$E\{y_1^2 y_2^2\} = 0 \neq \frac{1}{4} = E\{y_1^2\}E\{y_2^2\}$$

At most one Gaussian variable is allowed in ICA

$$p(x_1, x_2) = \frac{1}{2\pi} \exp\left(-\frac{x_1^2 + x_2^2}{2}\right)$$



- Completely **symmetric**, no information on the directions of the columns of $\boldsymbol{A}$

最多只有一个高斯

- Any **orthogonal transformation** of Gaussian (x_1, x_2) has exactly the same distribution as (x_1, x_2).

The key to estimating ICA is non-Gaussianity.

Fig. 7. The multivariate distribution of two independent Gaussian variables.

Principles of ICA estimation

- "Non-Gaussian is independent"
  - The Central Limit Theorem, a classical result in probability theory, tells that the distribution of a sum of independent random variables tends toward a Gaussian distribution, under certain conditions.
  - Thus, a sum of two independent random variables usually has a distribution that is closer to Gaussian than any of the two original random variables.

$$y = \boldsymbol{w}^T \boldsymbol{x} = \boldsymbol{w}^T A \boldsymbol{s} = (\boldsymbol{w}^T A)\boldsymbol{s} = \boldsymbol{z}^T \boldsymbol{s}$$

$\boldsymbol{z}^T \boldsymbol{s}$ is more Gaussian than any of the s_i (Assume s_i is i.i.d.)

Therefore, we could take w that maximizes the non-Gaussianity.

Measures of non-Gaussian

- Kurtosis or the fourth order cumulant

$$\text{kurt}(y) = E\{y^4\} - 3(E\{y^2\})^2$$

$$= E\{y^4\} - 3, \text{ if } E\{y^2\} = 1 \text{ for unit variance.}$$

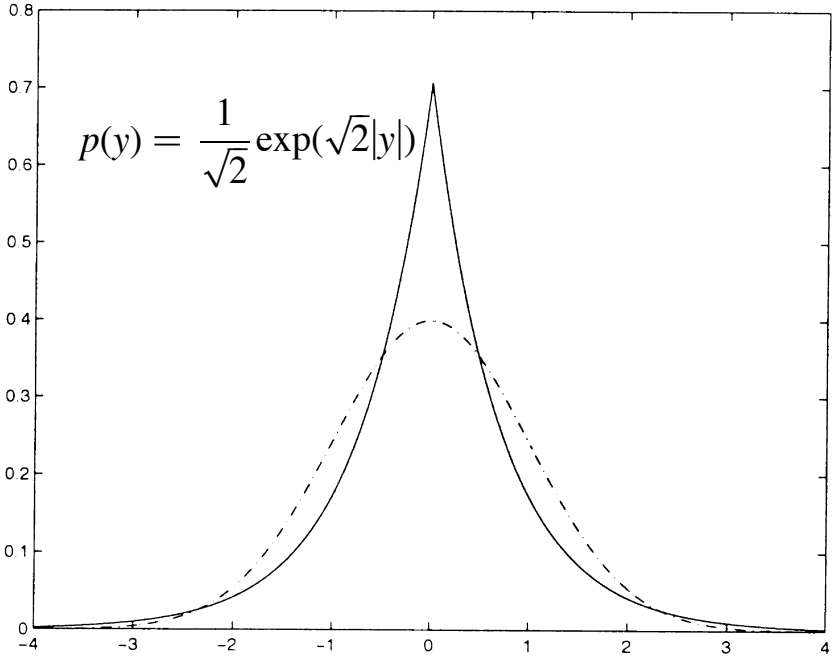


Fig. 8. The density function of the Laplace distribution, which is a typical super-Gaussian distribution. For comparison, the Gaussian density is given by a dashed line. Both densities are normalized to unit variance.

$\text{kurt}(y) < 0$, sub-Gaussian

$\text{kurt}(y) > 0$, super-Gaussian

Measures of non-Gaussian

A Gaussian variable has the largest entropy among all random variables of equal variance.

- Negentropy $J(\mathbf{y}) = H(\mathbf{y}_{\text{gauss}}) - H(\mathbf{y})$

Where $H(\boldsymbol{y})$ is entropy defined by

$$H(Y) = -\sum_i P(Y = a_i)\log P(Y = a_i)$$ discrete

$$H(\mathbf{y}) = -\int f(\mathbf{y})\log f(\mathbf{y})\mathrm{d}\mathbf{y}$$ continuous

The more "random", i.e. unpredictable and unstructured the variable is, the larger its entropy.

Negentropy is in some sense the optimal estimator of non- Gaussianity, as far as statistical properties are concerned. The problem in using negentropy is, however, that it is computationally very difficult due to estimation of pdf.

Measures of non-Gaussian

- Mutual information

$$I(y_1, y_2, \ldots, y_m) = \sum_{i=1}^{m} H(y_i) - H(y)$$

Actually it is KL divergence between $P(y_1, \ldots, y_m)$ and $P(y_1) \cdots P(y_m)$

There is a fundamental relation between mutual information and negentropy:

$$I(y_1, y_2, \ldots, y_n) = C - \sum_i J(y_i).$$

FastICA algorithm [Hyvarinen 1999]

- FastICA learning finds a unit vector $\boldsymbol{w}$ such that the projection $y = \boldsymbol{w}^T \boldsymbol{x}$ maximize non-Gaussian via an approximation of negentropy:

$$J(y) \propto [E\{G(y)\} - E\{G(\nu)\}]^2$$

ν is a zero-mean unit-variance Gaussian

$$G_1(u) = \frac{1}{a_1} \log \cosh a_1 u, \qquad G_2(u) = -\exp(-u^2/2) \qquad 1 \le a_1 \le 2$$

Denote by g the derivative of the above non-quadratic function G:

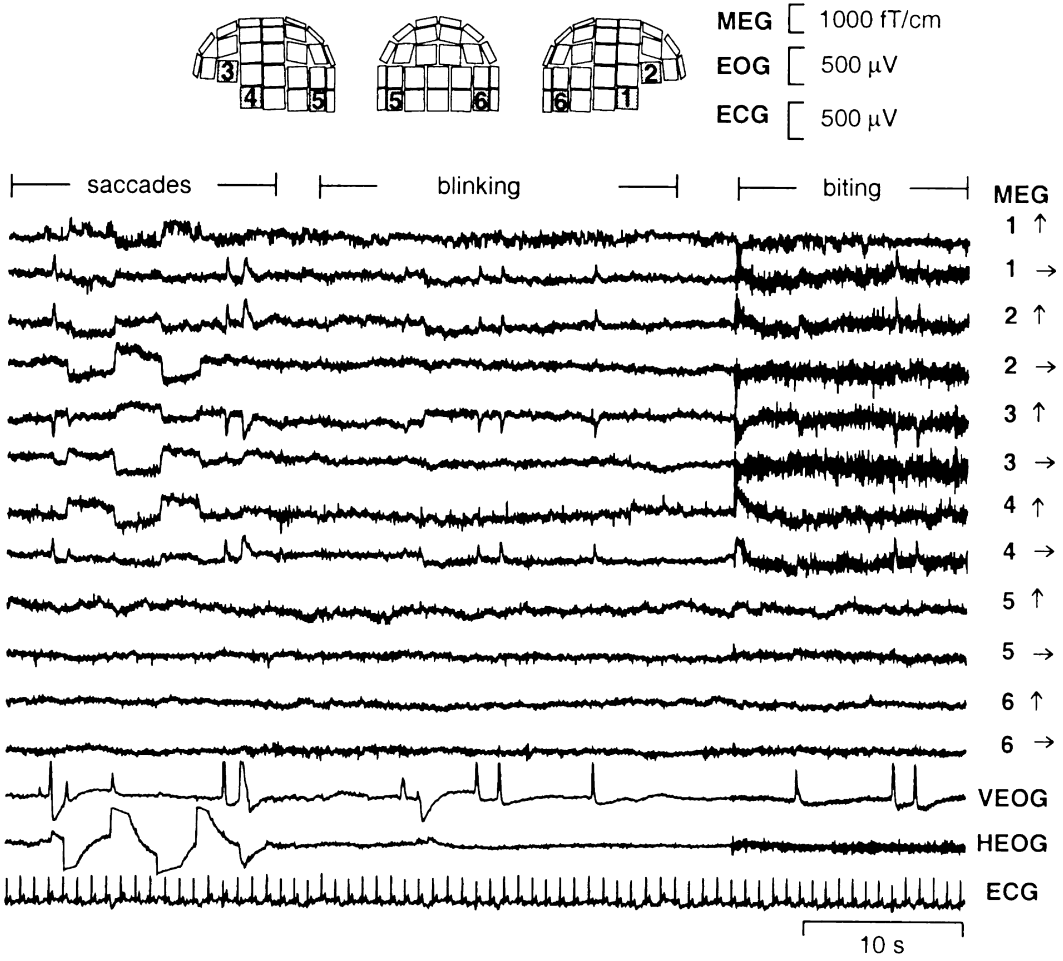
$$g_1(u) = \tanh(a_1 u), \qquad g_2(u) = u \exp(-u^2/2)$$

The basic form of FastICA algorithm:

1. Choose an initial (e.g. random) weight vector $\mathbf{w}$.
2. Let $\mathbf{w}^+ = E\{\mathbf{x} g(\mathbf{w}^{\mathrm{T}}\mathbf{x})\} - E(g'(\mathbf{w}^{\mathrm{T}}\mathbf{x})\}\mathbf{w}$
3. Let $\mathbf{w} = \mathbf{w}^+ / \|\mathbf{w}^+\|$
4. If not converged, go back to 2.

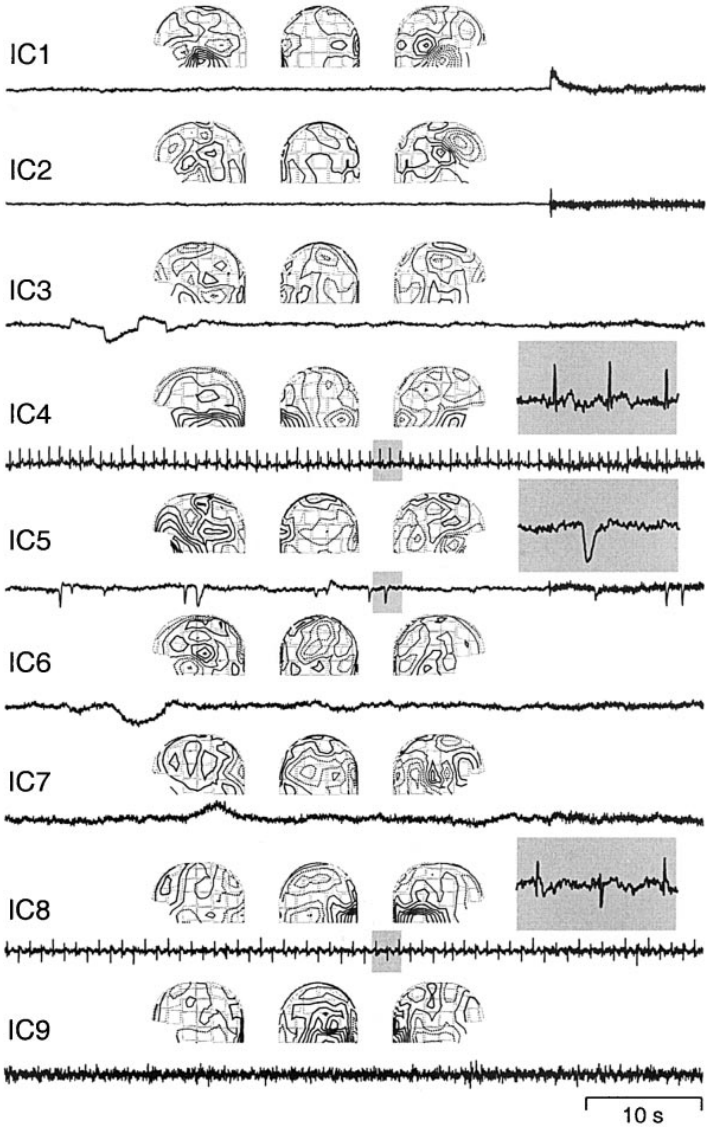
$$\cosh x = \frac{e^x + e^{-x}}{2} = \frac{e^{2x}+1}{2e^x} = \frac{1+e^{-2x}}{2e^{-x}}. \quad \tanh x = \frac{\sinh x}{\cosh x} = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{e^{2x}-1}{e^{2x}+1} = \frac{1-e^{-2x}}{1+e^{-2x}}.$$

Separation of artifacts in Magnetoencephalography(MEG) data



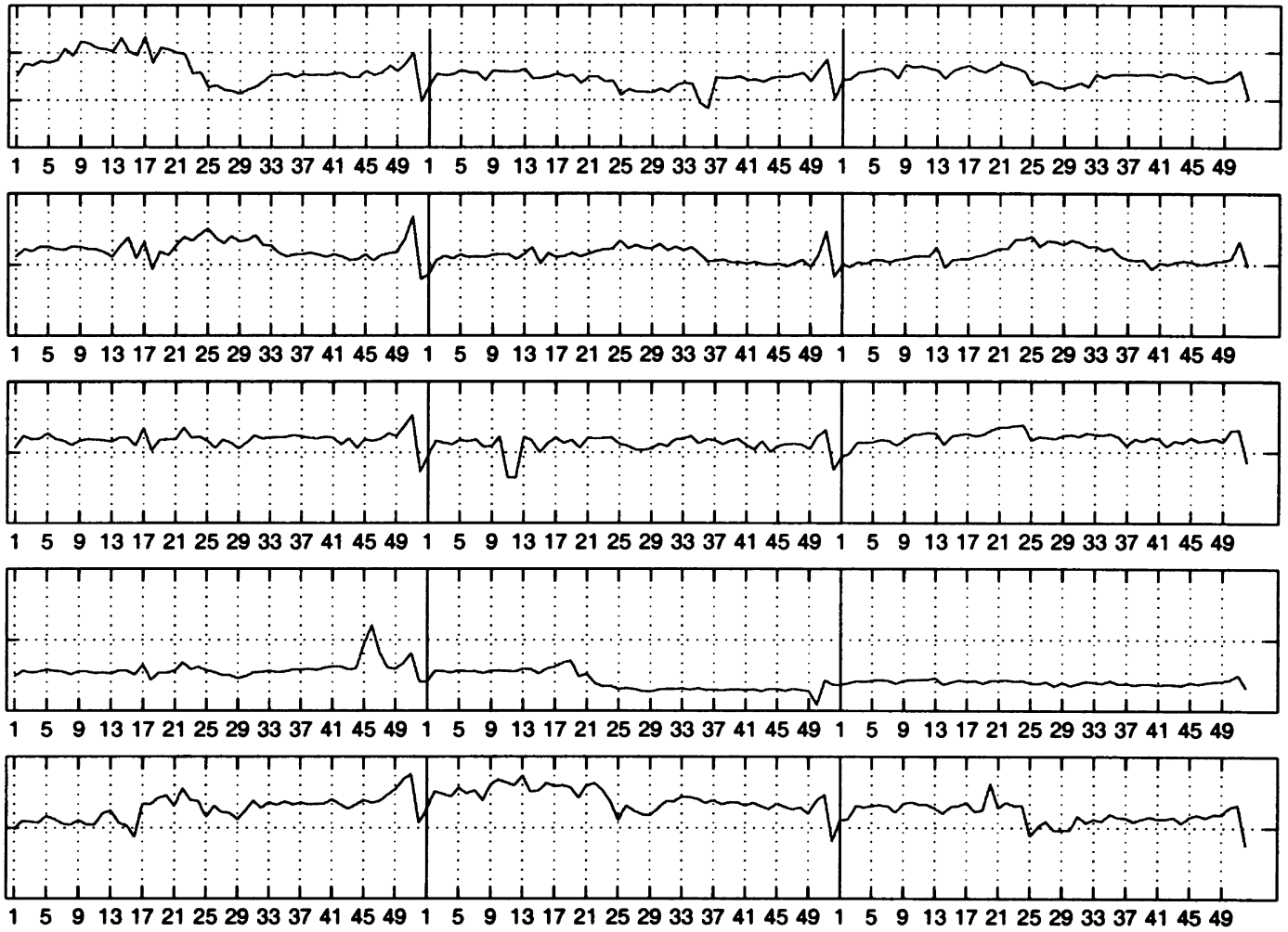
Samples of MEG signals, showing artifacts produced by blinking, saccades, biting and cardiac cycle. For each of the six positions shown, the two orthogonal directions of the sensors are plotted.

Separation of artifacts in Magnetoencephalography(MEG) data



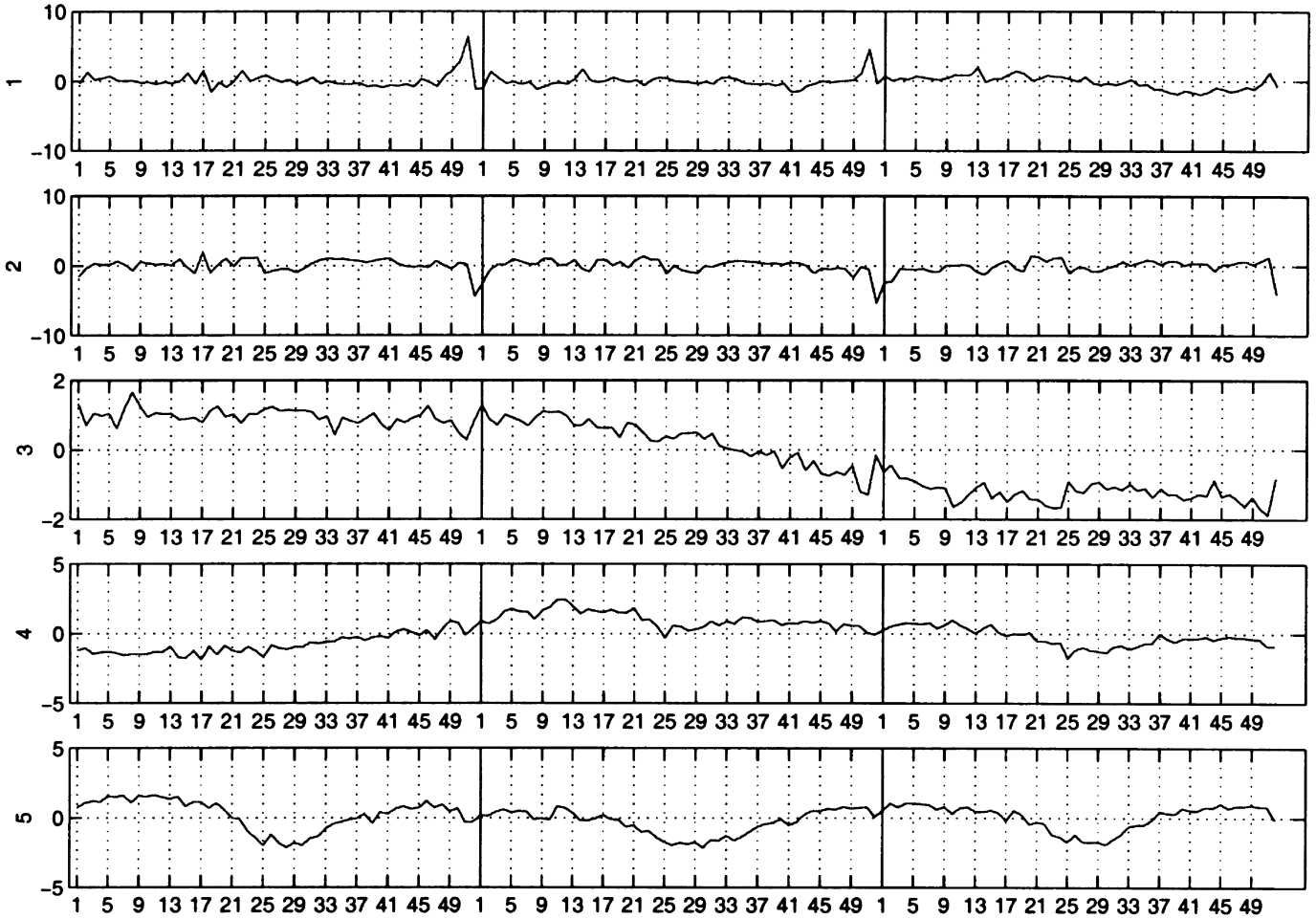
Nine independent components found from the MEG data. For each component the left, back and right views of the field patterns generated by these components are shown—full line stands for magnetic flux coming out from the head, and dotted line the flux inwards.

Finding hidden factors in financial data



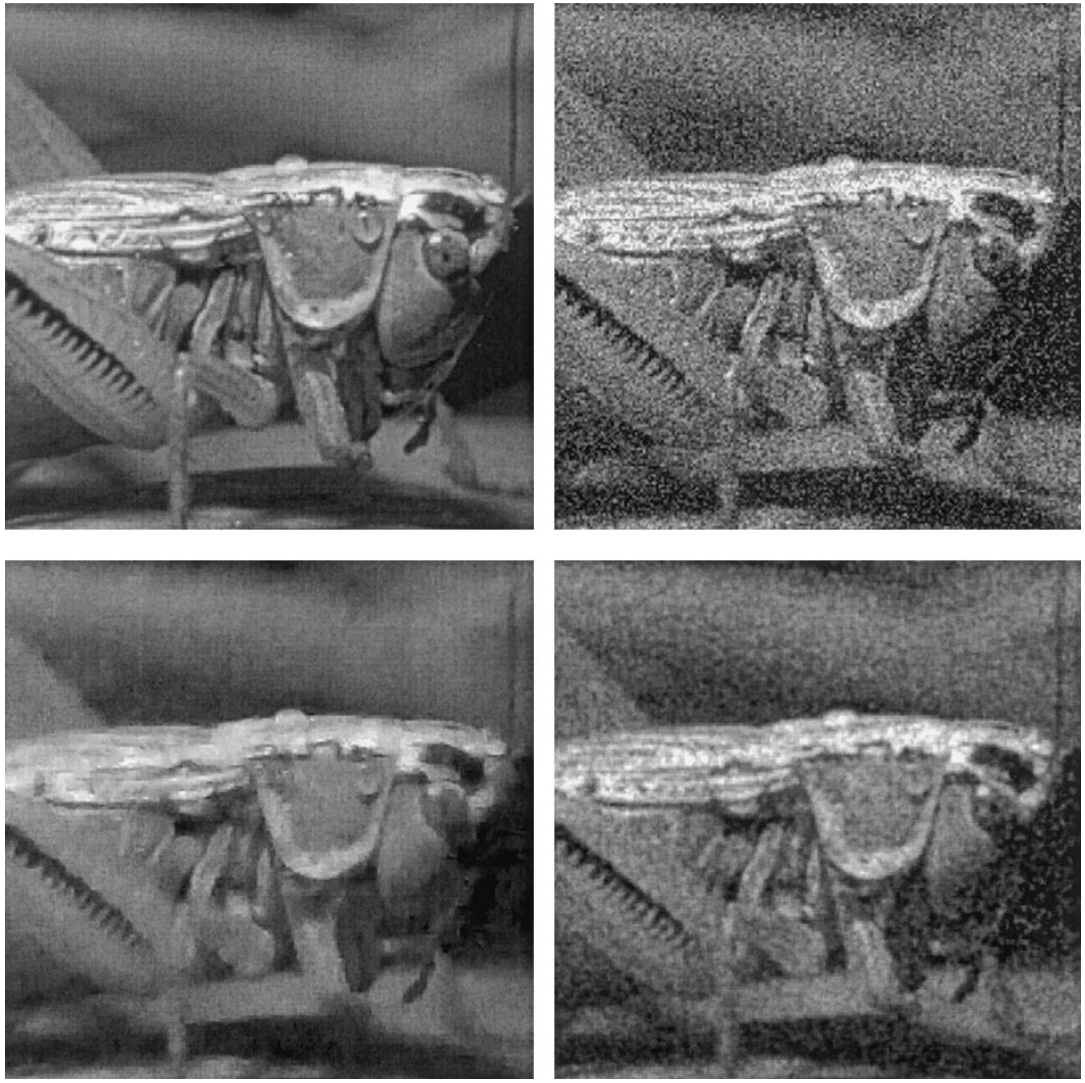
Five samples of **the original cashflow time series** (mean removed, normalized to unit standard deviation).

Finding hidden factors in financial data



Five independent components or fundamental factors found from the cashflow data.

Reducing noise in natural images



An experiment in denoising. Upper left: original image. **Upper right: original image** corrupted with noise; the noise level is 50%. **Lower left: the recovered image** after applying sparse code shrinkage. Lower right: for comparison, a wiener filtered image.

Outline

- FA and PCA
- Independent Component Analysis (ICA)
- Independent FA (IFA), Non-Gaussian FA (NFA)
- Recent papers related to PCA/ICA/GMM

Generative model for ICA with noise

- Generate independent, non-Gaussian source signals $\boldsymbol{y} = [y_1, \dots, y_k]$;
- Generate $\boldsymbol{x} = \Lambda \boldsymbol{y} + \boldsymbol{e}$, where $\boldsymbol{e} \sim G(e|0, \Psi)$ is Gaussian noise.

How to represent a non-Gaussian distribution?

$$\boldsymbol{x} = \Lambda \boldsymbol{y} + \boldsymbol{e}$$

$$x_j = \sum_{i=1}^{k} \lambda_{ji} y_i + e_j$$

$$p(\boldsymbol{y}) = \prod_{i=1}^{k} f(y_i)$$

$$p(\boldsymbol{x}|\boldsymbol{y}) = \boldsymbol{G}(\boldsymbol{x}|\Lambda \boldsymbol{y}, \Psi)$$

Use GMM!

$$f(y_i) = \sum_{q_i=1}^{m_i} w_{i,q_i} \mathcal{N}(\mu_{i,q_i}, \nu_{i,q_i})$$

Latent variables:

- Factors $\boldsymbol{y} = [y_1, \dots, y_k]$;
- Allocation variable $\boldsymbol{z} = [0, \dots, 0, 1, 0, \dots, 0]$

Equivalently as a GMM model

The distribution of $\boldsymbol{x}$:

Θ denotes the set of parameters.

$$
\begin{aligned}
f(\mathbf{x}|\Theta) &= \sum_{\mathbf{z}} \int f(\mathbf{x},\mathbf{y},\mathbf{z}|\Theta)d\mathbf{y} \\
&= \sum_{\mathbf{z}} \int f(\mathbf{z}|\Theta) f(\mathbf{y}|\mathbf{z},\Theta) f(\mathbf{x}|\mathbf{y},\mathbf{z},\Theta)d\mathbf{y} \\
&= \sum_{\mathbf{z}} f(\mathbf{z}|\Theta) f(\mathbf{x}|\mathbf{z},\Theta)
\end{aligned}
$$

$$
\begin{aligned}
f(\mathbf{x}|\mathbf{y},\mathbf{z},\Theta) &= \mathcal{N}(\Lambda\mathbf{y},\Psi) \\
f(\mathbf{y}|\mathbf{z},\Theta) &= \mathcal{N}(\boldsymbol{\mu}_{\mathbf{z}},\mathbf{V}_{\mathbf{z}})
\end{aligned}
$$

$$
\boldsymbol{\mu}_{\mathbf{z}} = \left[\prod_{q_1=1}^{m_1} \mu_{1,q_1}^{z_{1,q_1}}, \ldots, \prod_{q_k=1}^{m_k} \mu_{k,q_k}^{z_{k,q_k}}\right]
$$

$$
\mathbf{V}_{\mathbf{z}} = \operatorname{diag}\left[\prod_{q_1=1}^{m_1} \nu_{1,q_1}^{z_{1,q_1}}, \ldots, \prod_{q_k=1}^{m_k} \nu_{k,q_k}^{z_{k,q_k}}\right]
$$

$$
f(\mathbf{x}|\mathbf{z},\Theta) = \mathcal{N}\left(\Lambda\boldsymbol{\mu}_{\mathbf{z}}, \Lambda\mathbf{V}_{\mathbf{z}}\Lambda^T + \Psi\right)
$$

Then, $f(x|\Theta)$ is a GMM with the number of Gaussians: $m = \prod_{i=1}^{k} m_i$

EM for maximum likelihood

Likelihood function: $$f(\mathbf{x}|\Theta) = \sum_{\mathbf{z}} \int f(\mathbf{z}, \mathbf{y}, \mathbf{x}|\Theta)\, d\mathbf{y},$$

$$f(\mathbf{x}, \mathbf{y}, \mathbf{z}|\Theta) = f(\mathbf{x}|\mathbf{y}, \Theta)\, f(\mathbf{y}|\mathbf{z}, \Theta)\, f(\mathbf{z}|\Theta)$$

E-Step: Compute the posterior $f(\boldsymbol{z}, \boldsymbol{y}|\boldsymbol{x}, \Theta) = f(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z}, \Theta) f(\boldsymbol{z}|\boldsymbol{x}, \Theta)$

$$f(\mathbf{y}|\mathbf{x}, \mathbf{z}) = \mathcal{N}(\rho_{\mathbf{z}}(\mathbf{x}), \Sigma_{\mathbf{z}})$$

$$\Sigma_{\mathbf{z}} = (\Lambda^{\top}\Psi^{-1}\Lambda + \mathbf{V}_{\mathbf{z}}^{-1})^{-1}$$

$$f(\boldsymbol{z}|\boldsymbol{x}, \Theta) \propto f(\boldsymbol{x}|\boldsymbol{z}, \Theta) f(\boldsymbol{z}|\Theta)$$

$$\rho_{\mathbf{z}}(\mathbf{x}) = \Sigma_{\mathbf{z}}(\Lambda^{\top}\Psi^{-1}\mathbf{x} + \mathbf{V}_{\mathbf{z}}^{-1}\mu_{\mathbf{z}}^{-1})^{-1}$$

$$\boldsymbol{E}[\boldsymbol{y}^{\boldsymbol{T}}|\boldsymbol{x}], \qquad \boldsymbol{E}[\boldsymbol{y}\boldsymbol{y}^{\boldsymbol{T}}|\boldsymbol{x}]$$

M-Step: $$\arg\max_{\Theta} E_{\mathbf{z},\mathbf{y}|\mathbf{x},\Theta'}\left[\ln f(\mathbf{z}, \mathbf{y}, \mathbf{x}|\Theta)\right]$$

$$\hat{\Lambda} = \mathbf{x}E[\mathbf{y}^T|\mathbf{x}]E\left[\mathbf{y}\mathbf{y}^T|\mathbf{x}\right]^{-1} \qquad \hat{\Psi} = \mathbf{x}\mathbf{x}^T - \mathbf{x}E[\mathbf{y}^T|\mathbf{x}]\Lambda^T$$

$$\hat{\mu}_{i,q_i} = \frac{f(z_i|\mathbf{x})E[y_i|z_i, \mathbf{x}]}{f(z_i|\mathbf{x})} \qquad \hat{\nu}_{i,q_i} = \frac{f(z_i|\mathbf{x})E[y_i^2|z_i, \mathbf{x}]}{f(z_i|\mathbf{x})} - \mu_{i,q_i}^2$$

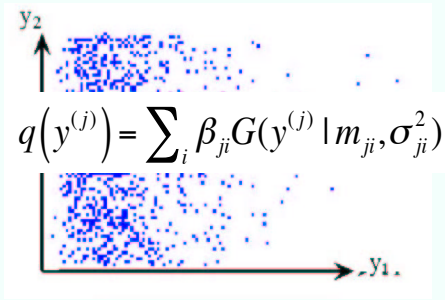
$\boldsymbol{z_{i,q_i}}$

$$\hat{w}_{i,q_i} = f(z_i|\mathbf{x})$$

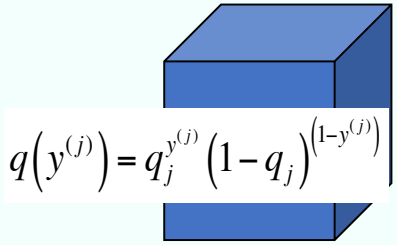
Summation w.r.t. $\boldsymbol{x}$ was ignored for simplicity.

Non-linear, non-Gaussian FA

$$q(y) = \prod_{j=1}^{k} q(y^{(j)})$$

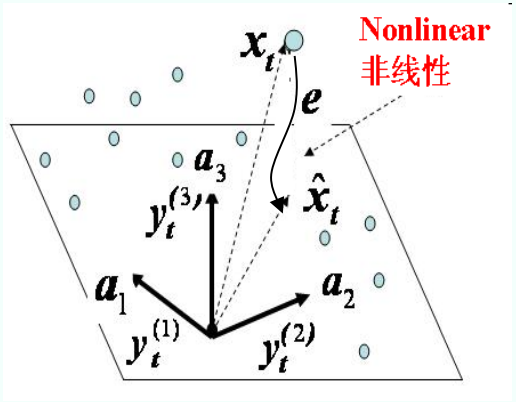


$$q\left(y^{(j)}\right) = \sum_i \beta_{ji} G(y^{(j)} \mid m_{ji}, \sigma_{ji}^2)$$

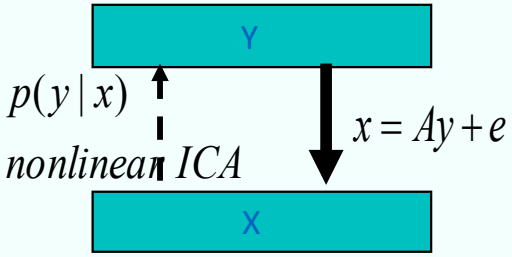


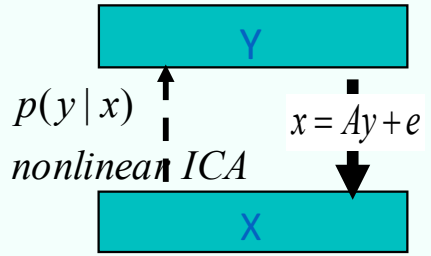
$$q\left(y^{(j)}\right) = q_j^{y^{(j)}} \left(1 - q_j\right)^{\left(1-y^{(j)}\right)}$$

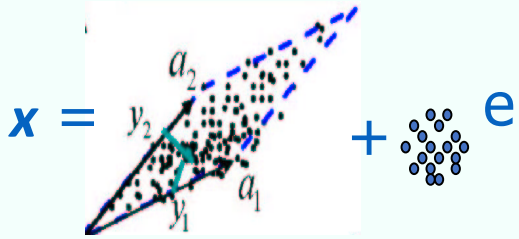
$$y_j = 1 \text{ or } 0$$



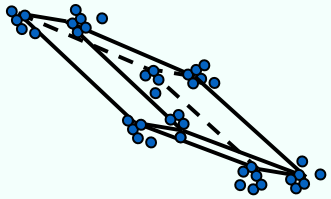
y_1 y_2 y_3 are real







NFA



BFA

Outline

- FA and PCA
- Independent Component Analysis (ICA)
- Independent FA (IFA), Non-Gaussian FA (NFA)
- Recent papers related to PCA/ICA/GMM

ICML 2012

Deep Mixtures of Factor Analysers

Yichuan Tang TANG@CS.TORONTO.EDU
Ruslan Salakhutdinov RSALAKHU@CS.TORONTO.EDU
Geoffrey Hinton HINTON@CS.TORONTO.EDU
Department of Computer Science, University of Toronto, Toronto, Ontario, CANADA

Abstract

An efficient way to learn deep density models that have many layers of latent variables is to learn one layer at a time using a model that has only one layer of latent variables. After learning each layer, samples from the posterior distributions for that layer are used as training data for learning the next layer. This approach is commonly used with Restricted Boltzmann Machines, which are *undirected* graphical models with a single hidden layer, but it can also be used with Mixtures of Factor Analysers (MFAs) which are *directed* graphical models. In this paper, we present a greedy layer-wise learning algorithm for Deep Mixtures of Factor Analysers (DMFAs). Even though a DMFA can be converted to an equivalent shallow MFA by multiplying together the factor loading matrices at different levels, learning and inference are much more efficient in a DMFA and the sharing of each lower-level factor loading matrix by many different higher level MFAs prevents overfitting. We demonstrate empirically that DMFAs learn better density models than both MFAs and two types of Restricted Boltzmann Machine on a wide variety of datasets.

Let $\mathbf{x} \in \mathbb{R}^D$ denote the D-dimensional data, $\{\mathbf{z} \in \mathbb{R}^d : d \leq D\}$ denote the d-dimensional latent variable, and $c \in \{1, \ldots, C\}$ denote the component indicator variable of C total components. The MFA is a directed generative model, defined as follows:

$$p(c) = \pi_c, \quad \sum_{c=1}^{C} \pi_c = 1, \tag{1}$$

$$p(\mathbf{z}|c) = p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I}), \tag{2}$$

$$p(\mathbf{x}|\mathbf{z}, c) = \mathcal{N}(\mathbf{x}; \mathbf{W}_c\mathbf{z} + \boldsymbol{\mu}_c, \boldsymbol{\Psi}_c), \tag{3}$$

$$p(\mathbf{x}|c) = \int_{\mathbf{z}} p(\mathbf{x}|\mathbf{z}, c)p(\mathbf{z}|c)d\mathbf{z} = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_c, \Gamma_c) \tag{4}$$

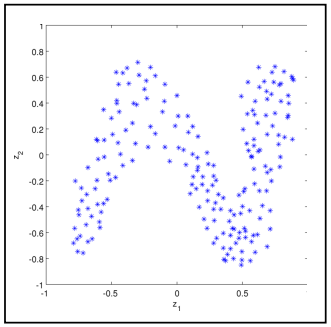
$$\Gamma_c = \mathbf{W}_c\mathbf{W}_c^{\mathsf{T}} + \boldsymbol{\Psi}_c$$

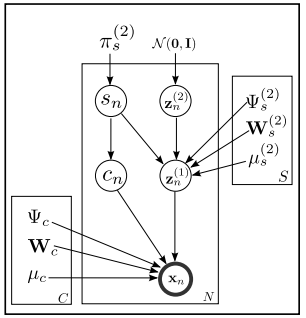
$$p(\mathbf{x}) = \sum_{c=1}^{C} \pi_c \, \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_c, \Gamma_c). \tag{5}$$

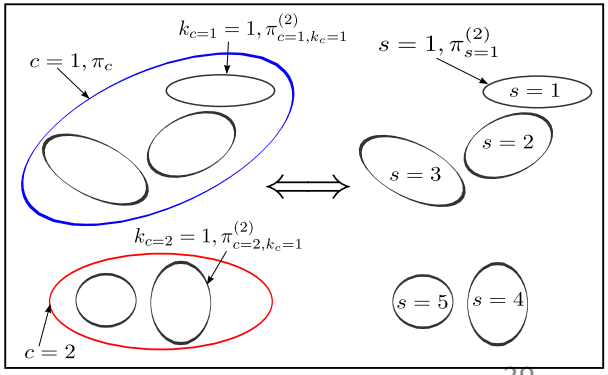
$$p(\mathbf{z}|\mathbf{x}, c) = \mathcal{N}(\mathbf{z}; \mathbf{m}_c, \mathbf{V}_c^{-1}), \tag{8}$$

where

$$\mathbf{V}_c = \mathbf{I} + \mathbf{W}_c^{\mathsf{T}}\boldsymbol{\Psi}_c^{-1}\mathbf{W}_c,$$
$$\mathbf{m}_c = \mathbf{V}_c^{-1}\mathbf{W}_c^{\mathsf{T}}\boldsymbol{\Psi}_c^{-1}(\mathbf{x} - \boldsymbol{\mu}).$$







NIPS 2015

Rethinking LDA: Moment Matching for Discrete ICA

Anastasia Podosinnikova **Francis Bach** **Simon Lacoste-Julien**
INRIA - École normale supérieure Paris

Abstract

We consider moment matching techniques for estimation in latent Dirichlet allocation (LDA). By drawing explicit links between LDA and discrete versions of independent component analysis (ICA), we first derive a new set of cumulant-based tensors, with an improved sample complexity. Moreover, we reuse standard ICA techniques such as joint diagonalization of tensors to improve over existing methods based on the tensor power method. In an extensive set of experiments on both synthetic and real datasets, we show that our new combination of tensors and orthogonal joint diagonalization techniques outperforms existing moment matching methods.

Michael I. Jordan

FOLLOW

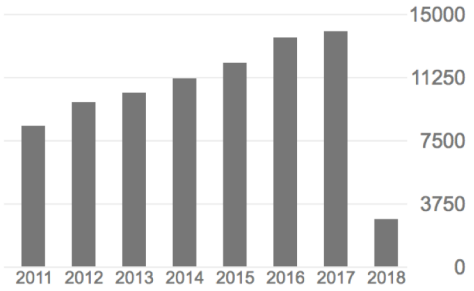
Professor of EECS and Professor of Statistics, University of California, Berkeley
Verified email at cs.berkeley.edu - Homepage

machine learning statistics computational biology artificial intelligence optimization

TITLE	CITED BY	YEAR
Latent dirichlet allocation DM Blei, AY Ng, MI Jordan Journal of machine Learning research 3 (Jan), 993-1022	22234	2003
On spectral clustering: Analysis and an algorithm AY Ng, MI Jordan, Y Weiss Advances in neural information processing systems, 849-856	6489	2002
Adaptive mixtures of local experts	3683	1991

Cited by VIEW ALL

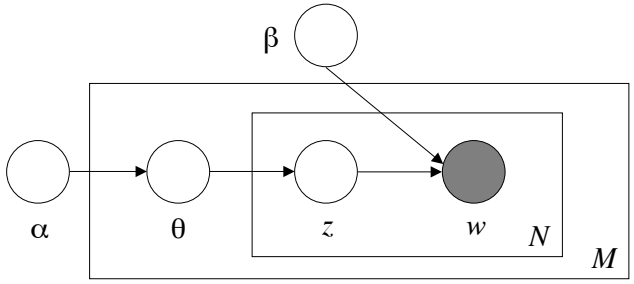
	All	Since 2013
Citations	132283	64236
h-index	144	98
i10-index	456	358



Latent Dirichlet allocation (LDA) is a generative probabilistic model of a corpus. The basic idea is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words.[1]

LDA assumes the following generative process for each document $\mathbf{w}$ in a corpus D:

1. Choose $N \sim \text{Poisson}(\xi)$.
2. Choose $\theta \sim \text{Dir}(\alpha)$.
3. For each of the N words w_n:
   (a) Choose a topic $z_n \sim \text{Multinomial}(\theta)$.
   (b) Choose a word w_n from $p(w_n \mid z_n, \beta)$, a multinomial probability conditioned on the topic z_n.



IEEE TRANS. ON IMAGE PROC. 2015

PCANet: A Simple Deep Learning Baseline for Image Classification?

Tsung-Han Chan, *Member, IEEE*, Kui Jia, Shenghua Gao, Jiwen Lu, *Senior Member, IEEE*, Zinan Zeng, and Yi Ma, *Fellow, IEEE*

***Abstract*— In this paper, we propose a very simple deep learning network for image classification that is based on very basic data processing components: 1) cascaded principal component analysis (PCA); 2) binary hashing; and 3) blockwise histograms. In the proposed architecture, the PCA is employed to learn multistage filter banks. This is followed by simple binary hashing and block histograms for indexing and pooling. This architecture is thus called the PCA network (PCANet) and can be extremely easily and efficiently designed and learned. For comparison and to provide a better understanding, we also introduce and study two simple variations of PCANet: 1) RandNet and 2) LDANet. They share the same topology as PCANet, but their cascaded filters are either randomly selected or learned from linear discriminant analysis. We have extensively tested these basic networks on many benchmark visual data sets for different tasks, including Labeled Faces in the Wild (LFW) for face verification; the MultiPIE, Extended Yale B, AR, Facial Recognition Technology (FERET) data sets for face recognition; and MNIST for hand-written digit recognition. Surprisingly, for all tasks, such a seemingly naive PCANet model is on par with the state-of-the-art features either prefixed, highly hand-crafted, or carefully learned [by deep neural networks (DNNs)]. Even more surprisingly, the model sets new records for many classification tasks on the Extended Yale B, AR, and FERET data sets and on MNIST variations. Additional experiments on other public data sets also demonstrate the potential of PCANet to serve as a simple but highly competitive baseline for texture classification and object recognition.**

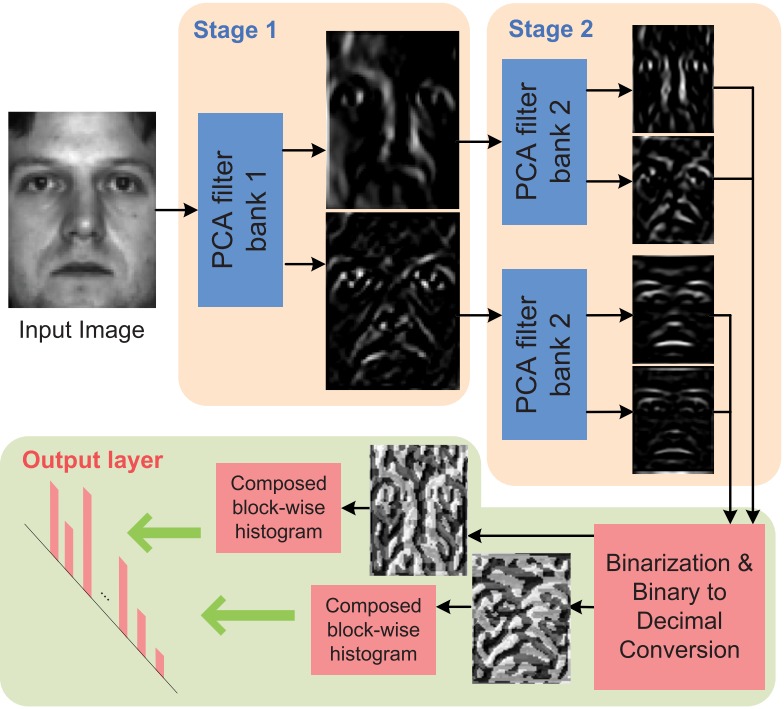


Fig. 1. Illustration of how the proposed PCANet extracts features from an image through the three simplest processing components: PCA filters, binary hashing, and histograms.

NIPS 2016

Global Analysis of Expectation Maximization for Mixtures of Two Gaussians

Ji Xu
Columbia University
jixu@cs.columbia.edu

Daniel Hsu
Columbia University
djhsu@cs.columbia.edu

Arian Maleki
Columbia University
arian@stat.columbia.edu

Abstract

Expectation Maximization (EM) is among the most popular algorithms for estimating parameters of statistical models. However, EM, which is an iterative algorithm based on the maximum likelihood principle, is generally only guaranteed to find stationary points of the likelihood objective, and these points may be far from any maximizer. This article addresses this disconnect between the statistical principles behind EM and its algorithmic properties. Specifically, it provides a global analysis of EM for specific models in which the observations comprise an i.i.d. sample from a mixture of two Gaussians. This is achieved by (i) studying the sequence of parameters from idealized execution of EM in the infinite sample limit, and fully characterizing the limit points of the sequence in terms of the initial parameters; and then (ii) based on this convergence analysis, establishing statistical consistency (or lack thereof) for the actual sequence of parameters produced by EM.

1.1 Expectation Maximization

Among the algorithms mentioned above, Expectation Maximization (EM) has attracted more attention for the simplicity of its iterations, and its good performance in practice (Dempster et al., 1977; Redner and Walker, 1984). EM is an iterative algorithm for climbing the likelihood objective starting from an initial setting of the parameters $\hat{\boldsymbol{\eta}}^{\langle 0 \rangle}$. In iteration t, EM performs the following steps:

$$\text{E-step:} \qquad \hat{Q}(\boldsymbol{\eta} \mid \hat{\boldsymbol{\eta}}^{\langle t \rangle}) \triangleq \sum_{\boldsymbol{z}} f(\boldsymbol{z} \mid \mathcal{Y}; \hat{\boldsymbol{\eta}}^{\langle t \rangle}) \log f(\mathcal{Y}, \boldsymbol{z}; \boldsymbol{\eta}) \,, \tag{1}$$

$$\text{M-step:} \qquad \hat{\boldsymbol{\eta}}^{\langle t+1 \rangle} \triangleq \arg\max_{\boldsymbol{\eta}} \hat{Q}(\boldsymbol{\eta} \mid \hat{\boldsymbol{\eta}}^{\langle t \rangle}) \,, \tag{2}$$

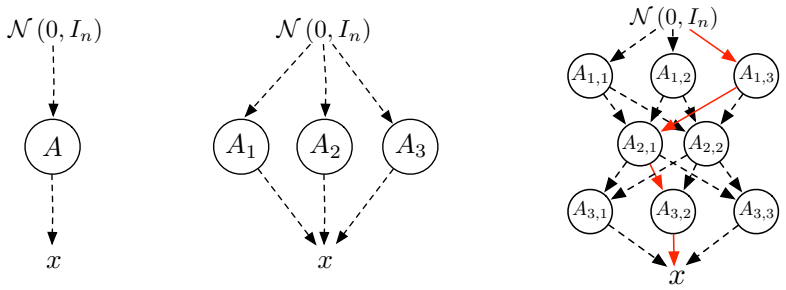
NIPS 2016

Factoring Variations in Natural Images with Deep Gaussian Mixture Models

Aäron van den Oord, Benjamin Schrauwen
Electronics and Information Systems department (ELIS), Ghent University
{aaron.vandenoord, benjamin.schrauwen}@ugent.be

Abstract

Generative models can be seen as the swiss army knives of machine learning, as many problems can be written probabilistically in terms of the distribution of the data, including prediction, reconstruction, imputation and simulation. One of the most promising directions for unsupervised learning may lie in Deep Learning methods, given their success in supervised learning. However, one of the current problems with deep unsupervised learning methods, is that they often are harder to scale. As a result there are some easier, more scalable shallow methods, such as the Gaussian Mixture Model and the Student-t Mixture Model, that remain surprisingly competitive. In this paper we propose a new *scalable* deep generative model for images, called the Deep Gaussian Mixture Model, that is a straightforward but powerful generalization of GMMs to multiple layers. The parametrization of a Deep GMM allows it to efficiently capture products of variations in natural images. We propose a new EM-based algorithm that scales well to large datasets, and we show that both the Expectation and the Maximization steps can easily be distributed over multiple machines. In our density estimation experiments we show that deeper GMM architectures generalize better than more shallow ones, with results in the same ballpark as the state of the art.



(a) Gaussian (b) GMM (c) Deep GMM

NIPS 2016

Sparse PCA via Bipartite Matchings

Megasthenis Asteris
The University of Texas at Austin
megas@utexas.edu

Dimitris Papailiopoulos
University of California, Berkeley
dimitrisp@berkeley.edu

Anastasios Kyrillidis
The University of Texas at Austin
anastasios@utexas.edu

Alexandros G. Dimakis
The University of Texas at Austin
dimakis@austin.utexas.edu

Abstract

We consider the following multi-component sparse PCA problem: given a set of data points, we seek to extract a small number of sparse components with *disjoint* supports that jointly capture the maximum possible variance. Such components can be computed one by one, repeatedly solving the single-component problem and deflating the input data matrix, but this greedy procedure is suboptimal. We present a novel algorithm for sparse PCA that jointly optimizes multiple disjoint components. The extracted features capture variance that lies within a multiplicative factor arbitrarily close to 1 from the optimal. Our algorithm is combinatorial and computes the desired components by solving multiple instances of the bipartite maximum weight matching problem. Its complexity grows as a low order polynomial in the ambient dimension of the input data, but exponentially in its rank. However, it can be effectively applied on a low-dimensional sketch of the input data. We evaluate our algorithm on real datasets and empirically demonstrate that in many cases it outperforms existing, deflation-based approaches.

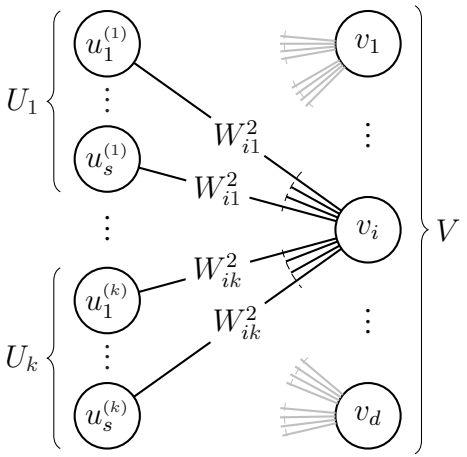


Figure 1: The graph G generated by Alg. 2. It is used to determine the support of the solution $\widehat{\mathbf{X}}$ in (6).

ICML 2017

SPLICE: Fully Tractable Hierarchical Extension of ICA with Pooling

Jun-ichiro Hirayama [1 2] **Aapo Hyvärinen** [3 4] **Motoaki Kawanabe** [2 1]

Abstract

We present a novel probabilistic framework for a hierarchical extension of independent component analysis (ICA), with a particular motivation in neuroscientific data analysis and modeling. The framework incorporates a general subspace pooling with linear ICA-like layers stacked recursively. Unlike related previous models, our generative model is fully tractable: both the likelihood and the posterior estimates of latent variables can readily be computed with analytically simple formulae. The model is particularly simple in the case of complex-valued data since the pooling can be reduced to taking the modulus of complex numbers. Experiments on electroencephalography (EEG) and natural images demonstrate the validity of the method.

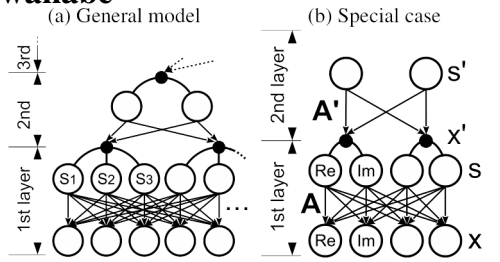


Figure 1: Generative model of SPLICE: (a) A higher layer directly gives the squared L_2-norms of lower sources s within each subspace. (b) An important special case having one complex source s per subspace.

2. Proposed Method

2.1. First-Layer Model

We begin with formulating the generative model for our SPLICE. Denote by $\mathbf{x}_t$ observed data vectors ($t = 1, 2, \ldots, n$), either real- or complex-valued, consisting of d entries x_{it}. Each of the d entries is given by a linear combination of the same number of unknown (first-layer) components or *sources*, collectively denoted as source vector $\mathbf{s}_t$. Here, we consider the fundamental case where $\mathbf{x}_t$ and $\mathbf{s}_t$ are independently and identically distributed (i.i.d.). Omitting sample index t for notational simplicity, we write

$$\mathbf{x} = \mathbf{A}\mathbf{s}, \tag{1}$$

where the coefficient matrix $\mathbf{A}$, called mixing matrix, is square and assumed to be invertible; the inverse $\mathbf{W} := \mathbf{A}^{-1}$ is called demixing matrix. For convenience, we as-

ICLR 2018

Learning Independent Features with Adversarial Nets for Non-linear ICA

Philemon Brakel, Yoshua Bengio

16 Feb 2018 ICLR 2018 Conference Blind Submission readers: everyone Show Bibtex Revisions

Abstract: Reliable measures of statistical dependence could potentially be useful tools for learning independent features and performing tasks like source separation using Independent Component Analysis (ICA). Unfortunately, many of such measures, like the mutual information, are hard to estimate and optimize directly. We propose to learn independent features with adversarial objectives (Goodfellow et al. 2014, Arjovsky et al. 2017) which optimize such measures implicitly. These objectives compare samples from the joint distribution and the product of the marginals without the need to compute any probability densities. We also propose two methods for obtaining samples from the product of the marginals using either a simple resampling trick or a separate parametric distribution. Our experiments show that this strategy can easily be applied to different types of model architectures and solve both linear and non-linear ICA problems.

Keywords: adversarial networks, ica, unsupervised, independence

ICLR 2018 Conference Acceptance Decision

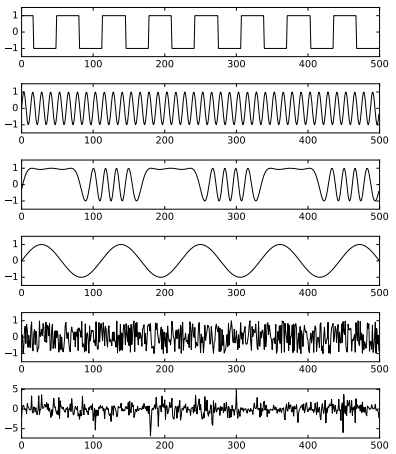
ICLR 2018 Conference Program Chairs

30 Jan 2018 ICLR 2018 Conference Acceptance Decision readers: everyone

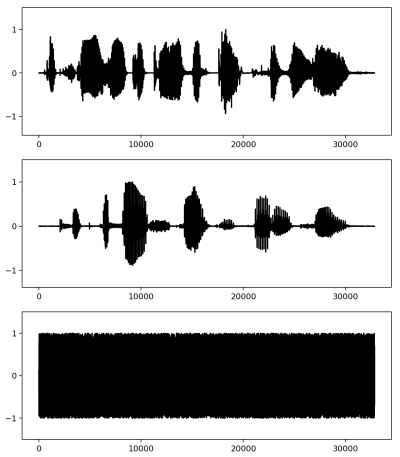
Decision: Reject

Comment: The paper proposes the use of GANs to match the joint distribution of features to the product
totally plausible but reviewers have complaints about lack of rigor and analysis in terms of (i) mixing condi
approach will work, given that ICA is ill-posed for general nonlinear mixing (ii) comparison with prior work

Further, in most scenarios where GANs are used, one of the distributions is fixed (say, the real distribution
trying to come close to the fixed distribution during optimization. In the proposed method, the discriminat
product of marginals which are both dynamic during the learning. It might be useful to comment whether
instability of training, etc.



(a) Synthetic source signals.



(b) Audio source signals.

Figure 3: Source signals used in the experiments.

https://openreview.net/forum?id=ryykVe-0W

Thank you!