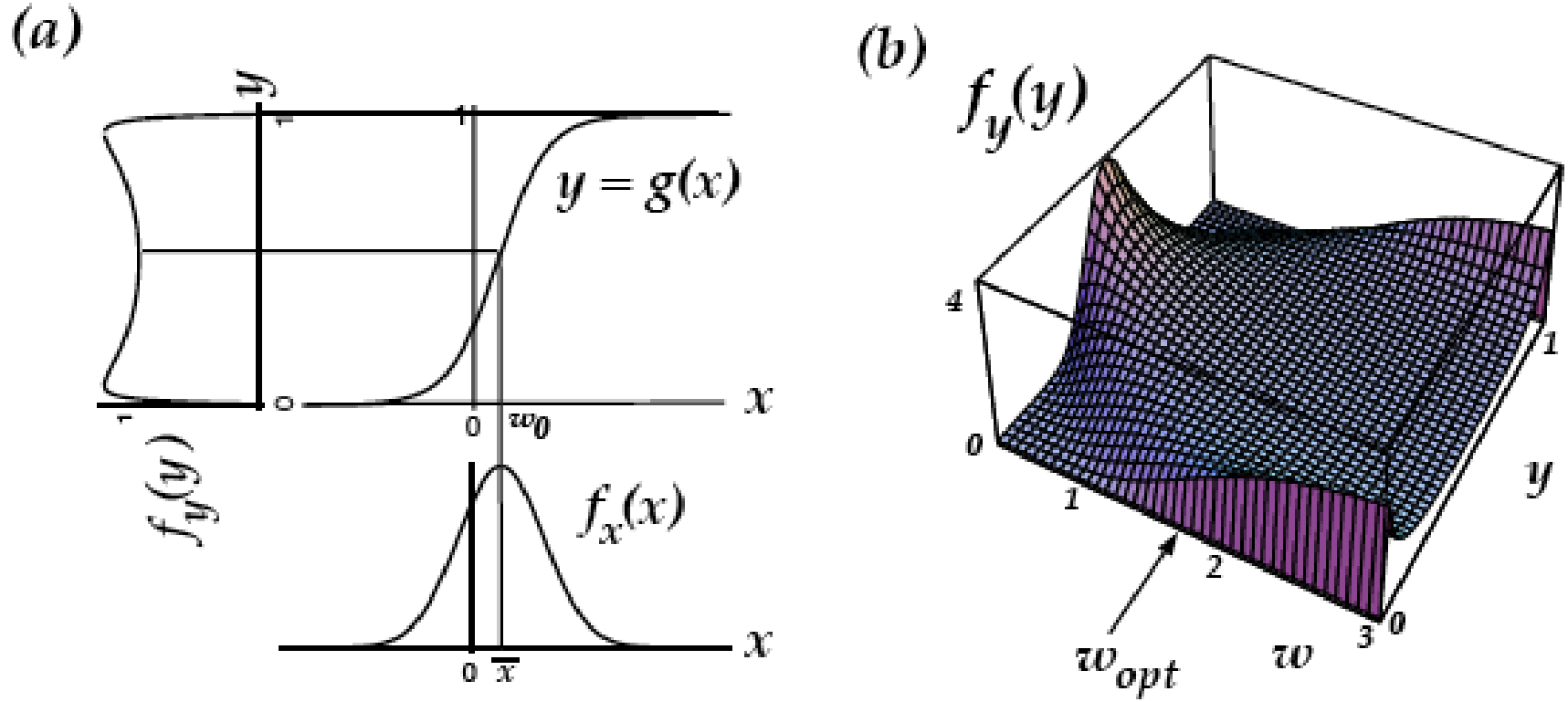
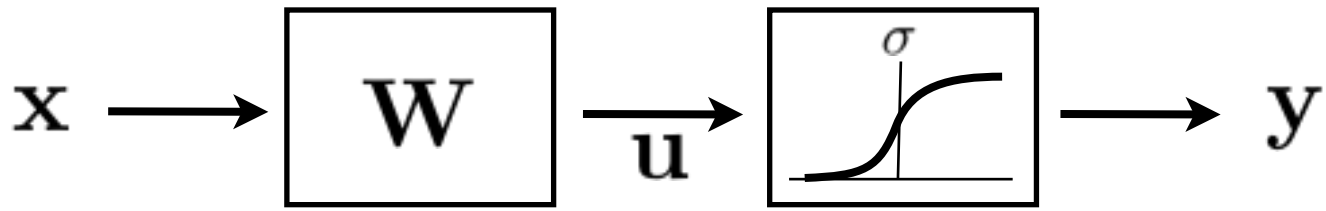
Sparse coding and ‘ICA’

Density estimation via entropy maximization



(from Bell & Sejnowski 1995)

Bell & Sejnowski (1995)



$$\mathbf{y} = \sigma(\mathbf{W}\,\mathbf{x})$$

Objective function: $\max_{\mathbf{W}} I(\mathbf{x}, \mathbf{y})$

$$I(\mathbf{x}, \mathbf{y}) = H(\mathbf{y}) - H(\mathbf{y}|\mathbf{x})$$

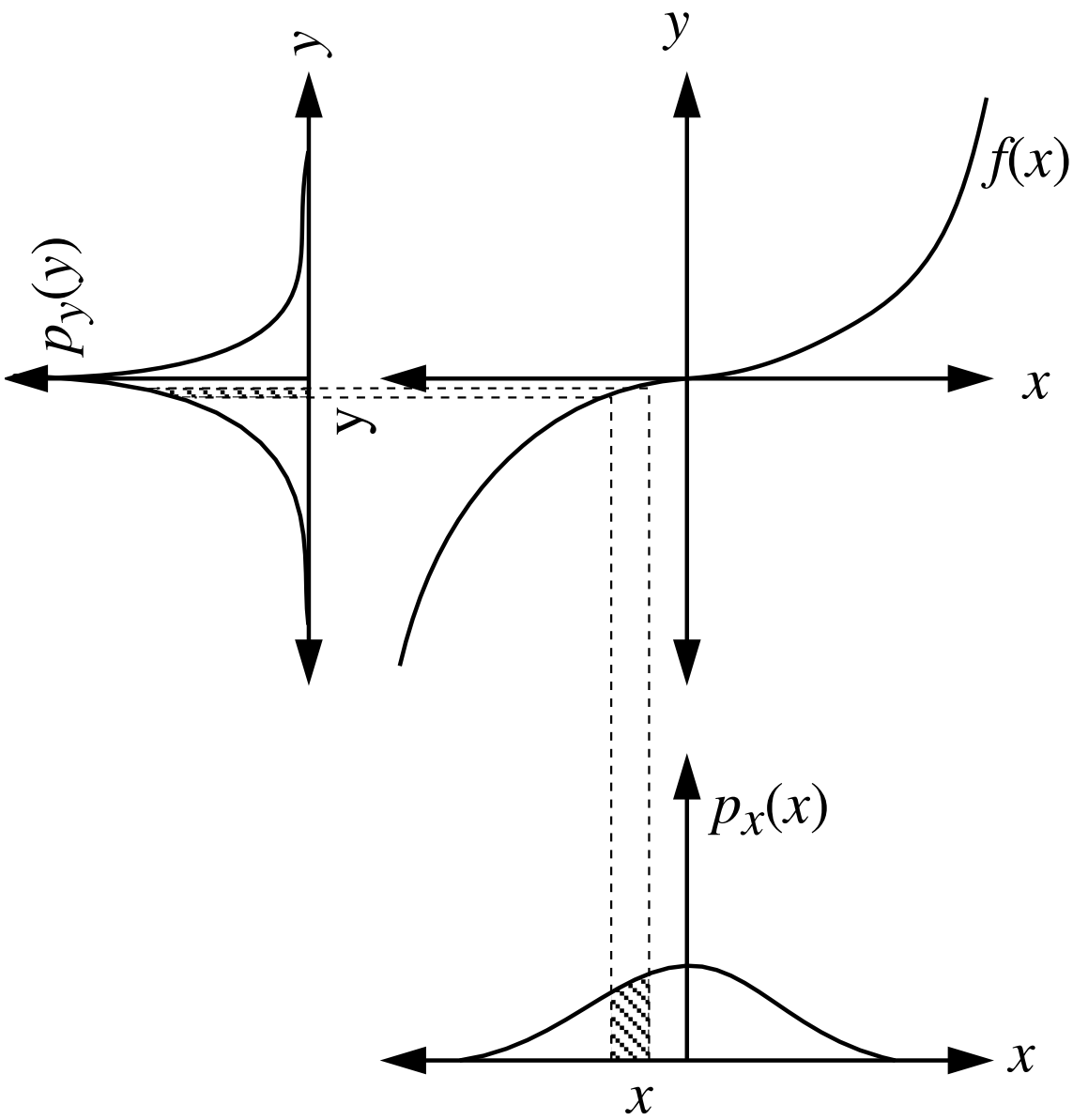
$$
\begin{aligned}
H(\mathbf{y}) &= -\int p(\mathbf{y})\, \log p(\mathbf{y}) d\mathbf{y} \\
&= -\langle \log p(\mathbf{y}) \rangle
\end{aligned}
$$

$$
\begin{aligned}
p(\mathbf{y}) &= \frac{p(\mathbf{x})}{|\mathbf{J}|} \\
\mathbf{J} &= d\mathbf{y}/d\mathbf{x} \\
&= \mathbf{W}\, \mathrm{diag}(\sigma'(\mathbf{u}))
\end{aligned}
$$

Thus:

$$
\max_{\mathbf{W}} \langle \log |\mathbf{J}| \rangle = \max_{\mathbf{W}} \left[\log \det \mathbf{W} + \sum_i \log \sigma'(u_i) \right]
$$

Aside: how to compute the probability of a *function* of a random variable?



$$y = f(x)$$

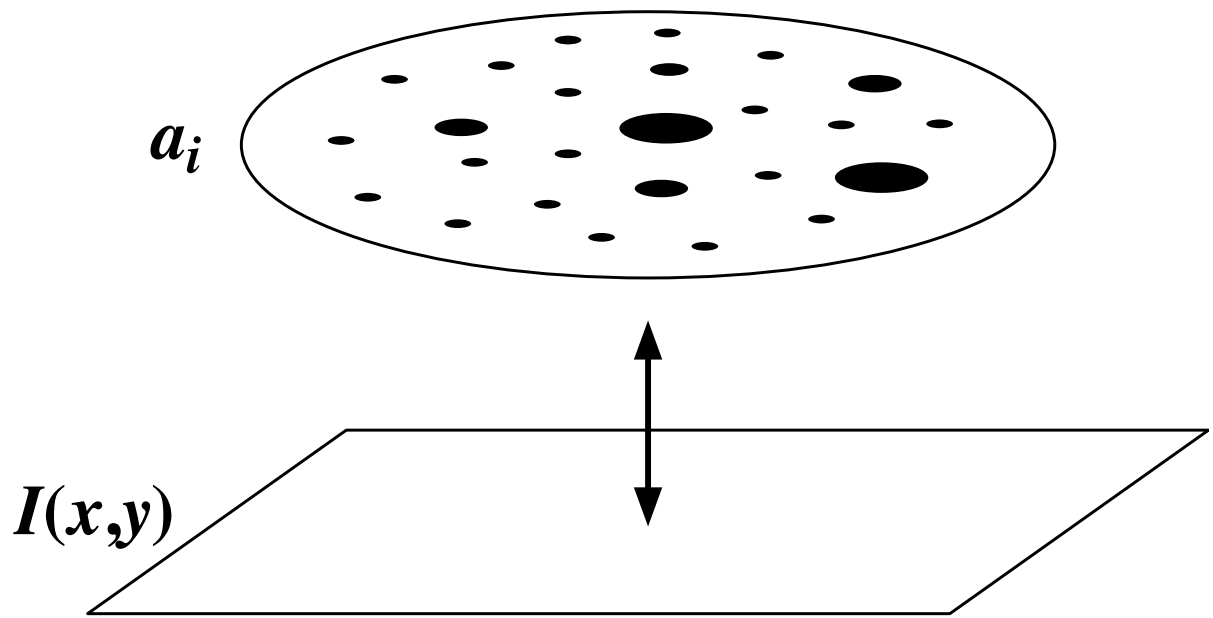
$$p_x(x_0)\delta x = p_y(f(x_0))\delta f(x_0)$$

$$p_y(y) = p_x(x)\left[\frac{dy}{dx}\right]^{-1}$$

ICA learning rule

$$
\begin{aligned}
\Delta W &\propto \frac{\partial}{\partial \mathbf{W}} \langle \log |\mathbf{J}| \rangle \\
&= \frac{\partial}{\partial \mathbf{W}} \left[\log \det \mathbf{W} + \sum_i \log \sigma'(u_i) \right] \\
&= \left[\mathbf{W}^T \right]^{-1} + (1 - 2\mathbf{y})\, \mathbf{x}^T
\end{aligned}
$$

Sparse, distributed representations



Sparse coding energy function

$$E = \frac{1}{2}|\mathbf{I} - \Phi\,\mathbf{a}|^2 + \lambda \sum_i C(a_i)$$

$$-\log P(\mathbf{a}|\mathbf{I}) = -\log P(\mathbf{I}|\mathbf{a}) + -\log P(\mathbf{a}) + K$$

$$P(\mathbf{a}) \propto \Pi_i\, e^{-\lambda C(a_i)}$$

Sparse coding model

$$\mathbf{x} = \mathbf{A}\,\mathbf{s} + \mathbf{n}$$

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{s})\, p_s(\mathbf{s})\, d\mathbf{s}$$

$$p(\mathbf{x}|\mathbf{s}) \propto e^{-\frac{|\mathbf{x}-\mathbf{A}\,\mathbf{s}|^2}{2\,\sigma_n^2}}$$

$$p_s(\mathbf{s}) \propto e^{-\sum_i C(s_i)}$$

Objective for learning

$$\langle \log p(\mathbf{x}) \rangle$$

Gradient ascent yields:

$$\begin{aligned} \Delta \mathbf{A} &\propto \frac{\partial}{\partial \mathbf{A}} \langle \log p(\mathbf{x}) \rangle \\ &= \left\langle \int [\mathbf{x} - \mathbf{A}\,\mathbf{s}]\, \mathbf{s}^T\, p(\mathbf{s}|\mathbf{x})\, d\mathbf{s} \right\rangle \end{aligned}$$

Inference

$$
\begin{aligned}
\hat{\mathbf{s}} &= \arg\max_{\mathbf{s}} \; p(\mathbf{s}|\mathbf{x}) \\
&= \arg\min_{\mathbf{s}} \; -\log p(\mathbf{s}|\mathbf{x}) \\
&= \arg\min_{\mathbf{s}} \left[\frac{\lambda_n}{2} |\mathbf{x} - \mathbf{A}\,\mathbf{s}|^2 + \sum_i C(s_i) \right]
\end{aligned}
$$

Gradient descent yields:

$$\dot{\mathbf{s}} \propto \lambda_n \left[\mathbf{b} - \mathbf{G}\,\mathbf{s}\right] - \mathbf{z}(\mathbf{s})$$

where $\mathbf{b} = \mathbf{A}^T\,\mathbf{x}, \quad \mathbf{G} = \mathbf{A}^T\,\mathbf{A}, \quad z_i = C'(s_i)$

Approximate learning rule

Instead of

$$\Delta \mathbf{A} \propto \left\langle \int [\mathbf{x} - \mathbf{A}\,\mathbf{s}]\, \mathbf{s}^T\, p(\mathbf{s}|\mathbf{x})\, d\mathbf{s} \right\rangle$$

Use

$$\Delta \mathbf{A} \propto \left\langle [\mathbf{x} - \mathbf{A}\,\hat{\mathbf{s}}]\, \hat{\mathbf{s}}^T \right\rangle$$

Special case

- No noise

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

- Invertible **A** matrix

$$\mathbf{s} = \mathbf{A}^{-1}\mathbf{x}$$

Special case

Thus $$p(\mathbf{x}|\mathbf{s}) = \delta(\mathbf{x} - \mathbf{A}\,\mathbf{s})$$

$$\begin{aligned} p(\mathbf{x}) &= \int \delta(\mathbf{x} - \mathbf{A}\,\mathbf{s})\, p_s(\mathbf{s})\, d\mathbf{s} \\ &= p_s(\mathbf{A}^{-1}\mathbf{x})/|\det \mathbf{A}| \end{aligned}$$

$$\log p(\mathbf{x}) = -\sum_i C(s_i) - \log\det \mathbf{A}$$

Special case

$$\begin{aligned}
\Delta \mathbf{A} &\propto \frac{\partial}{\partial \mathbf{A}} \langle \log p(\mathbf{x}) \rangle \\
&= \frac{\partial}{\partial \mathbf{A}} \left[-\sum_i C(s_i) - \log \det \mathbf{A} \right] \\
&= \boxed{\langle [\mathbf{A}^T]^{-1} \mathbf{z}(\mathbf{s}) \, \mathbf{s}^T - [\mathbf{A}^T]^{-1} \rangle}
\end{aligned}$$

Its the ICA learning rule!

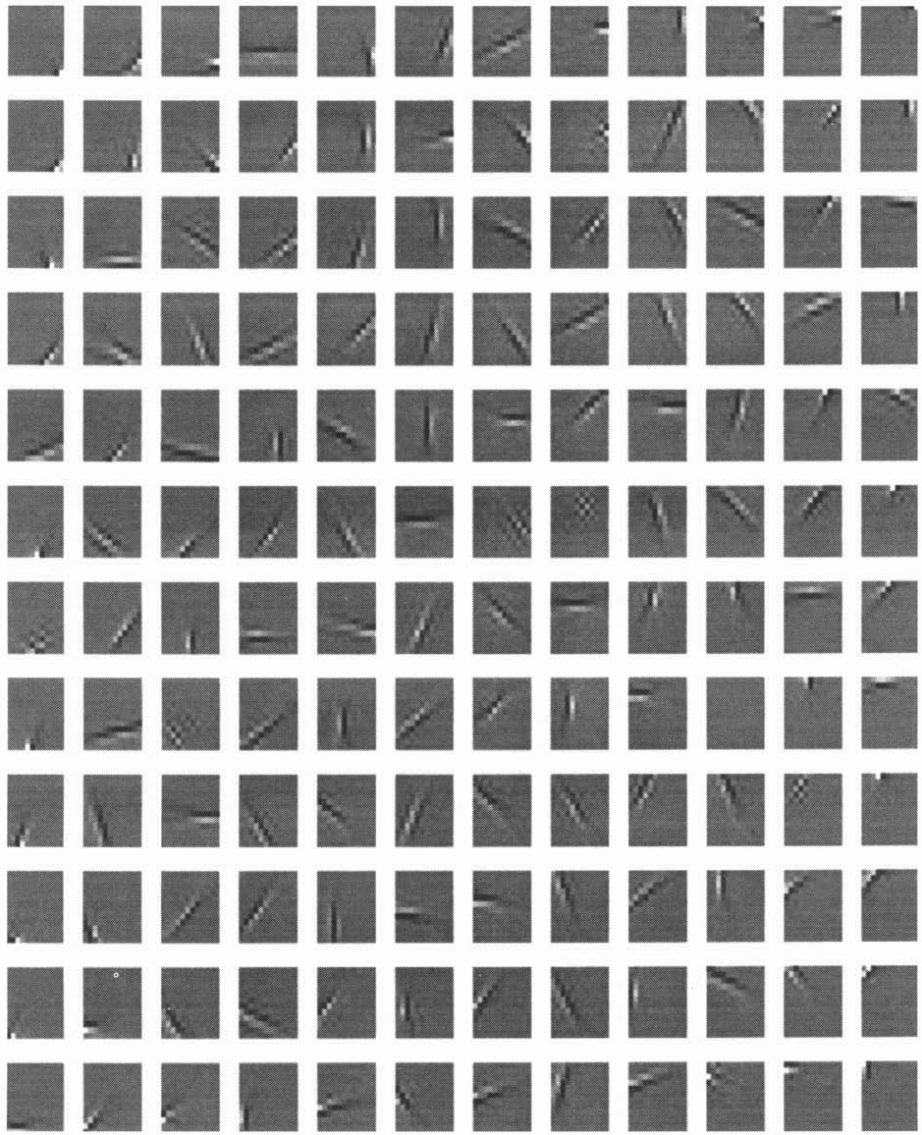
Pre-multiplying by $A\,A^T$ (natural gradient) yields:

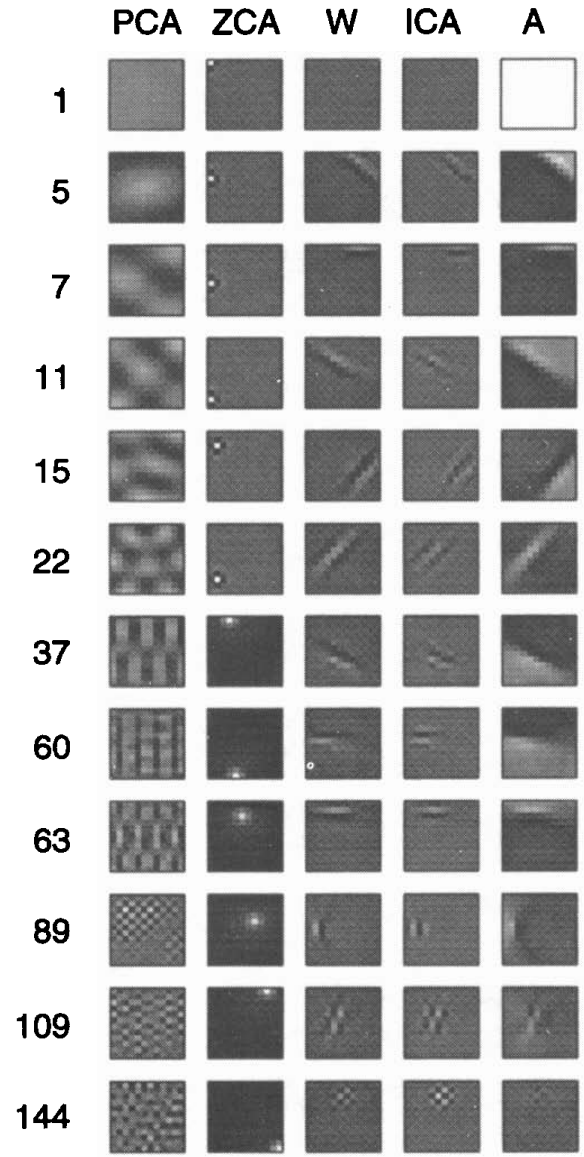
$$\begin{aligned}
\Delta \mathbf{A} &\propto \langle \mathbf{A} \, \mathbf{z} \, \mathbf{s}^T - \mathbf{A} \rangle \\
&= \langle [\mathbf{x} - \mathbf{A}(\mathbf{s} - \mathbf{z})] \, \mathbf{s}^T - \mathbf{A} \rangle
\end{aligned}$$

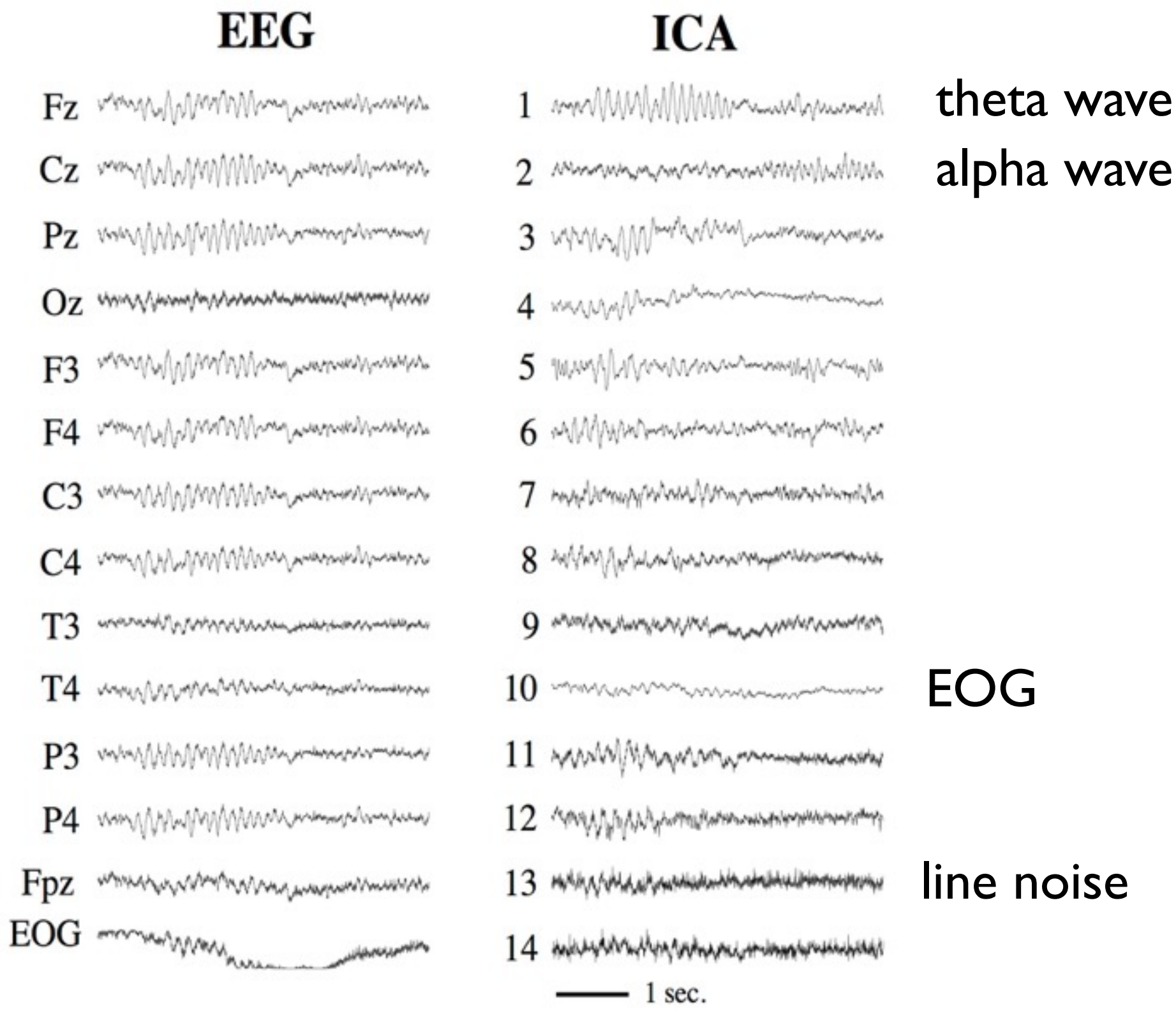
The "Independent Components" of Natural Scenes are Edge Filters

ANTHONY J. BELL,*† TERRENCE J. SEJNOWSKI*

Received 16 July 1996; in revised form 9 April 1997






EEG
ICA
Fz
Cz
Pz
Oz
F3
F4
C3
C4
T3
T4
P3
P4
Fpz
EOG
1
2
3
4
5
6
7
8
9
10
11
12
13
14
1 sec.
theta wave
alpha wave
EOG
line noise

ICA is not a general solution for finding independent components of data

Assumptions of the model:

1. linear superposition: $\mathbf{x} = \mathbf{A}\mathbf{s}$

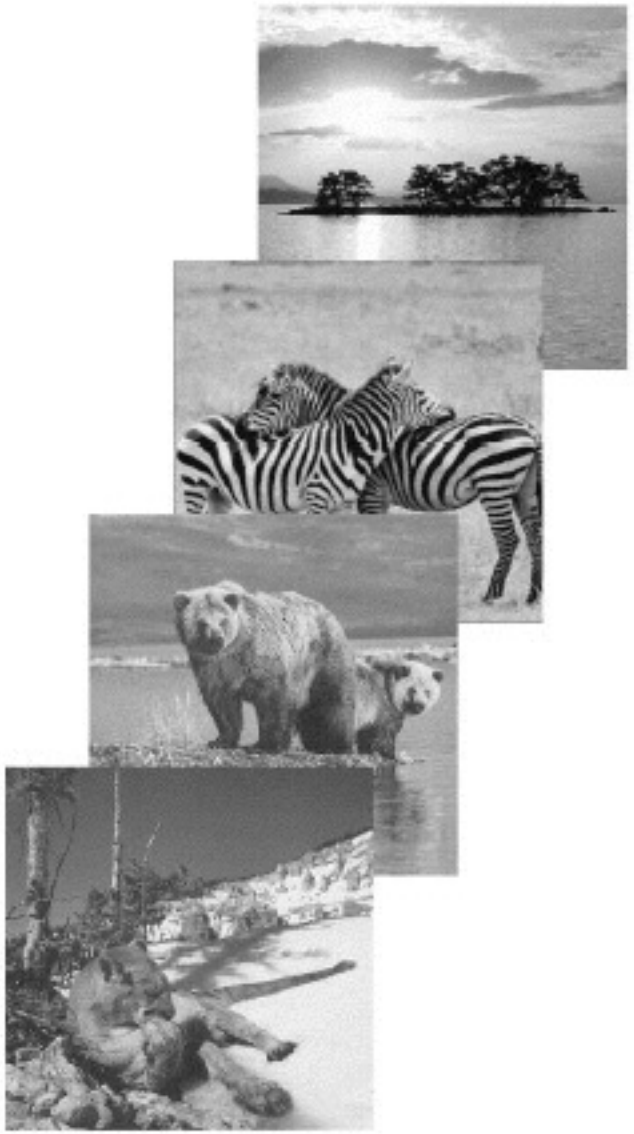
2. shape of the prior over each of the components:

$$p(s_i) \propto e^{-C(s_i)}$$

3. factorial prior over the entire set of components:

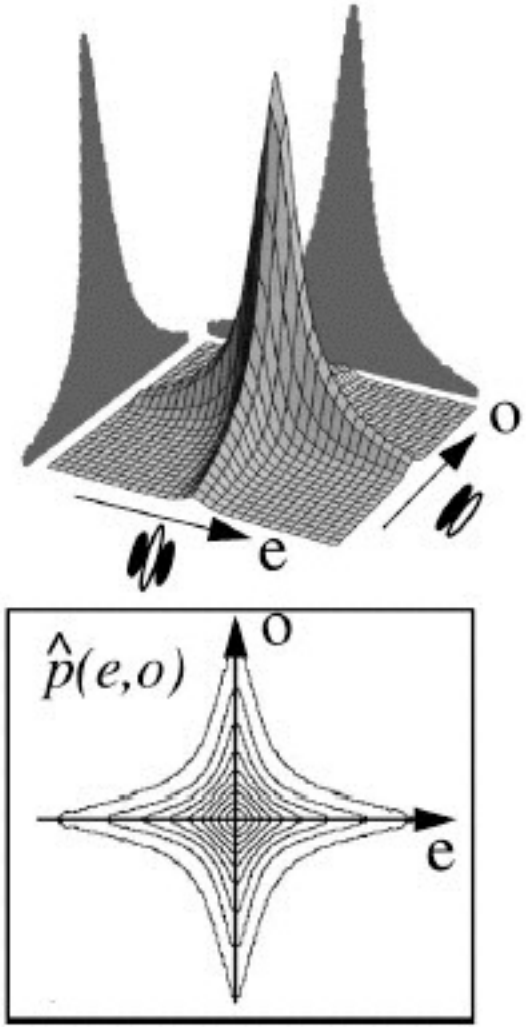
$$p_s(\mathbf{s}) \propto \prod_i p_{s_i}$$

Statistical dependencies



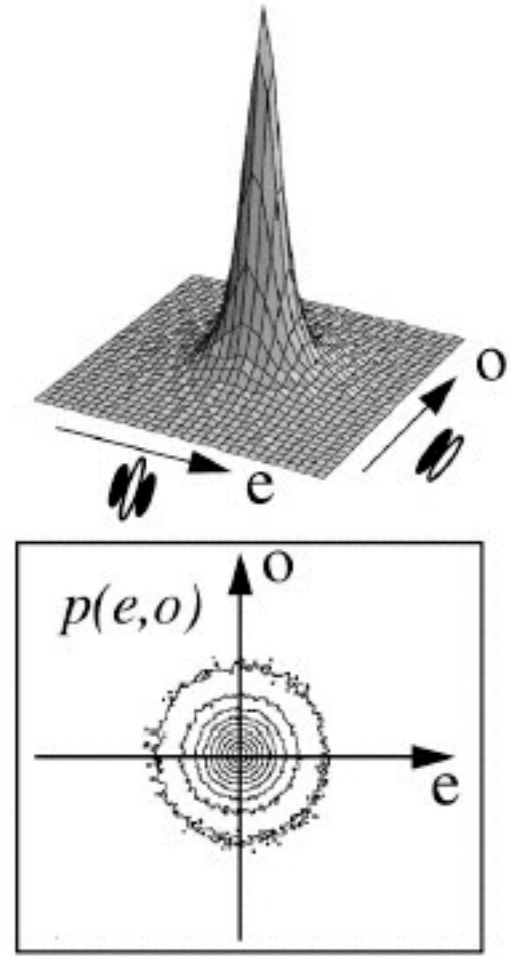
Predicted bivariate activity distribution

$\hat{p}(e,o) = p(e) \cdot p(o)$

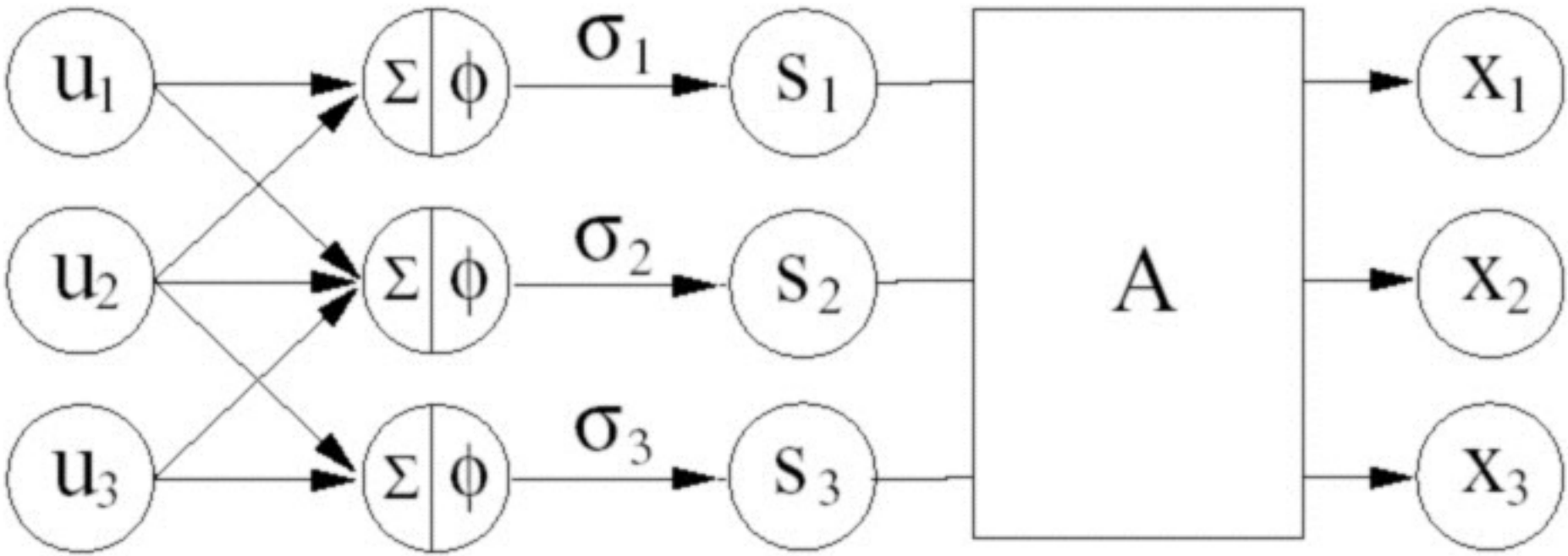


Measured bivariate activity distribution

$p(e,o)$



‘Topographic ICA’
(Hyvarinen & Hoyer)



‘Topographic ICA’
(Hyvarinen & Hoyer)