Bayesian inference

image generation

prior knowledge

$$P(H|D) = \frac{P(D|H)\,P(H)}{P(D)}$$

?

Simple example

$$y = x + n$$

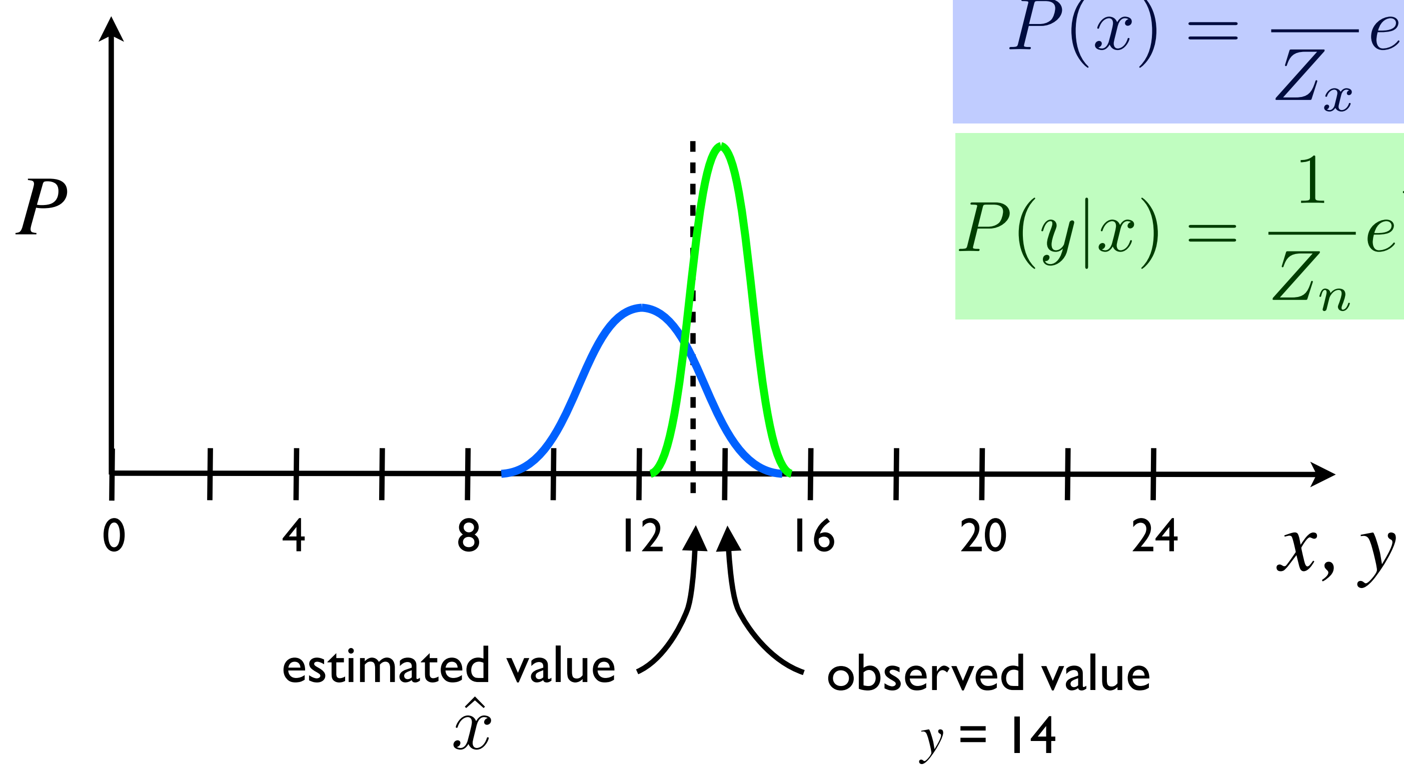
You observe y, what is x?

$$P(x|y) \propto P(y|x)\, P(x)$$

likelihood prior

$$P(x) = \frac{1}{Z_x} e^{-\frac{(x-\mu_x)^2}{2\sigma_x^2}}$$

$$P(y|x) = \frac{1}{Z_n} e^{-\frac{(y-x)^2}{2\sigma_n^2}}$$



How to compute $\hat{x}$?

$$P(x|y) \propto P(y|x)\, P(x)$$

$$= \frac{1}{Z_n} e^{-\frac{(y-x)^2}{2\sigma_n^2}} \frac{1}{Z_x} e^{-\frac{(x-\mu_x)^2}{2\sigma_x^2}}$$

$$-\log P(x|y) = \frac{(y-x)^2}{2\sigma_n^2} + \frac{(x-\mu_x)^2}{2\sigma_x^2} + \text{const.}$$

$$-\frac{\partial}{\partial x} \log P(x|y) = -\frac{(y-x)}{\sigma_n^2} + \frac{(x-\mu_x)}{\sigma_x^2} = 0$$

$$\Rightarrow \boxed{\hat{x} = \frac{\sigma_x^2\, y + \sigma_n^2\, \mu_x}{\sigma_x^2 + \sigma_n^2}}$$

Wiener filter

NOISE REMOVAL VIA BAYESIAN WAVELET CORING

Eero P. Simoncelli

Computer and Information Science Dept.
University of Pennsylvania
Philadelphia, PA 19104

Edward H. Adelson

Brain and Cognitive Science Dept.
Massachusetts Institute of Technology
Cambridge, MA 02139

original image

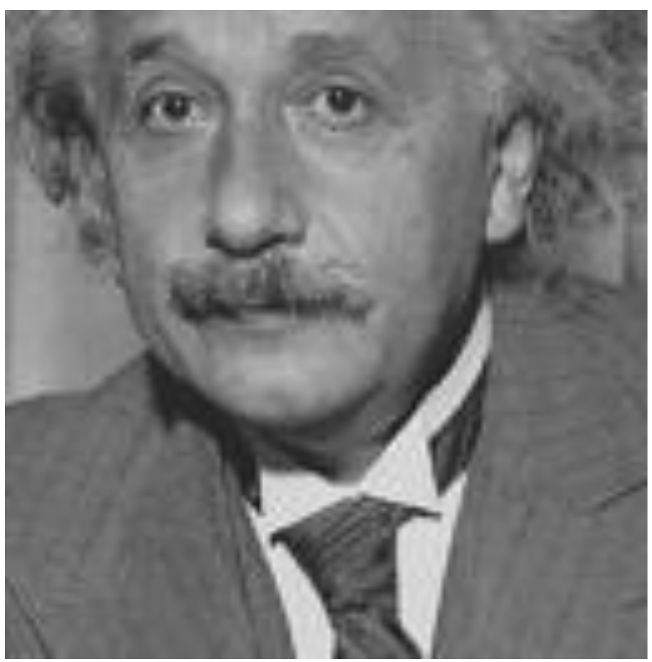
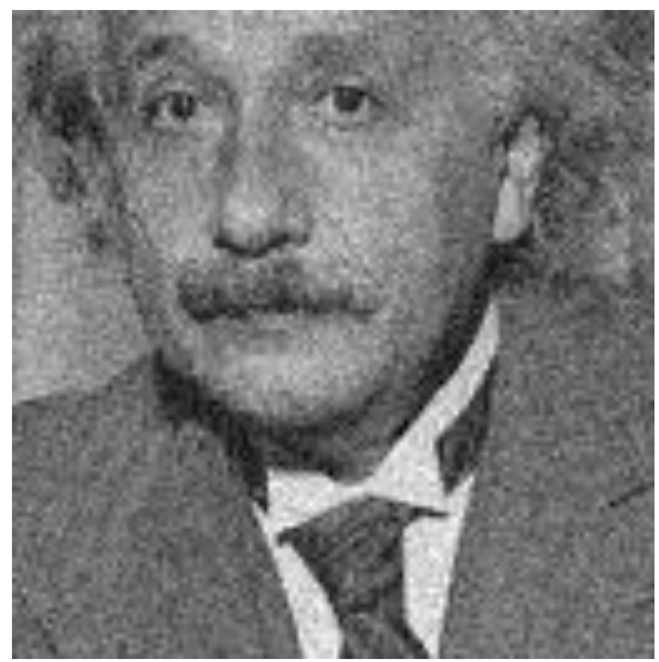


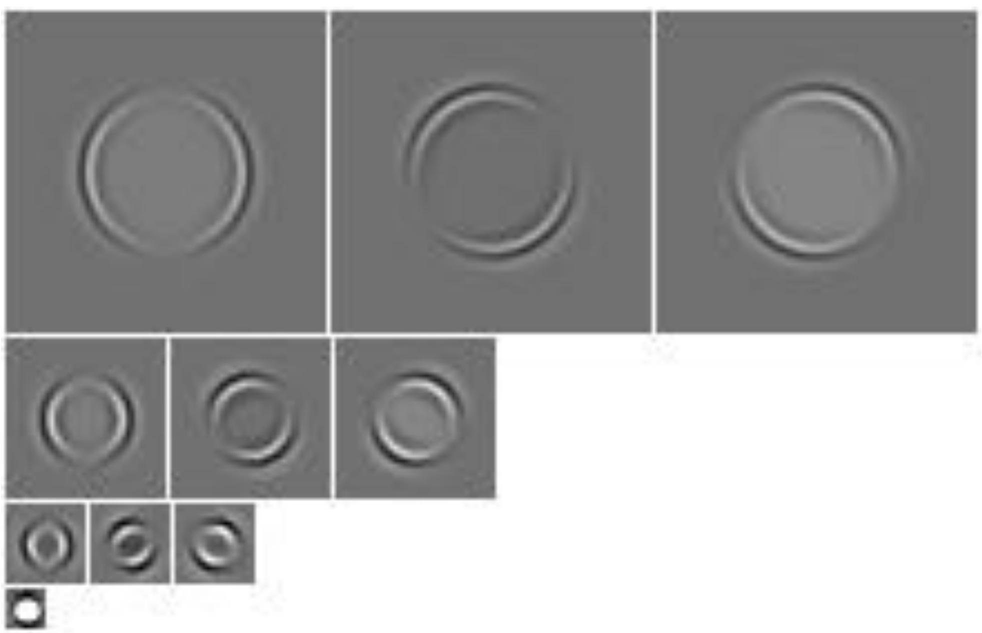
image + noise



NOISE REMOVAL VIA BAYESIAN WAVELET CORING

Eero P. Simoncelli *Edward H. Adelson*

Oriented wavelet decomposition of a circular disc



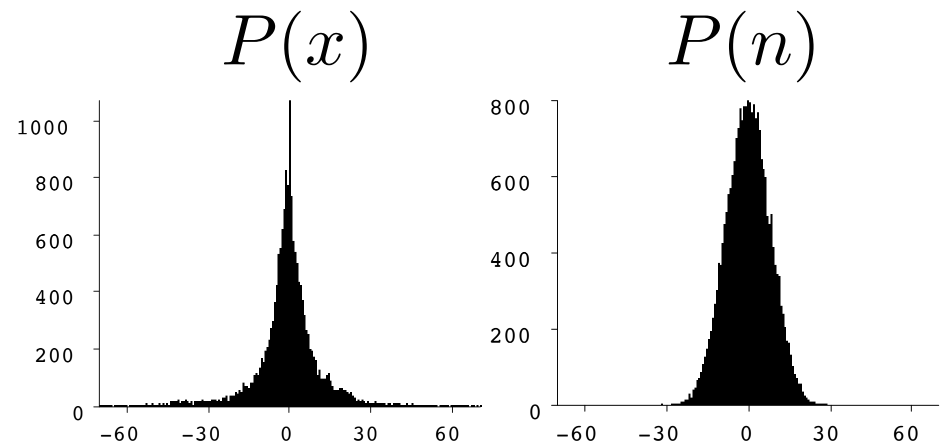


Figure [illegible]. Histograms of a mid-frequency subband in an octave-bandwidth wavelet decomposition for two different images. Left: The“Einstein” image. Right: A white noise image with uniform pdf.

$$y = x + n$$

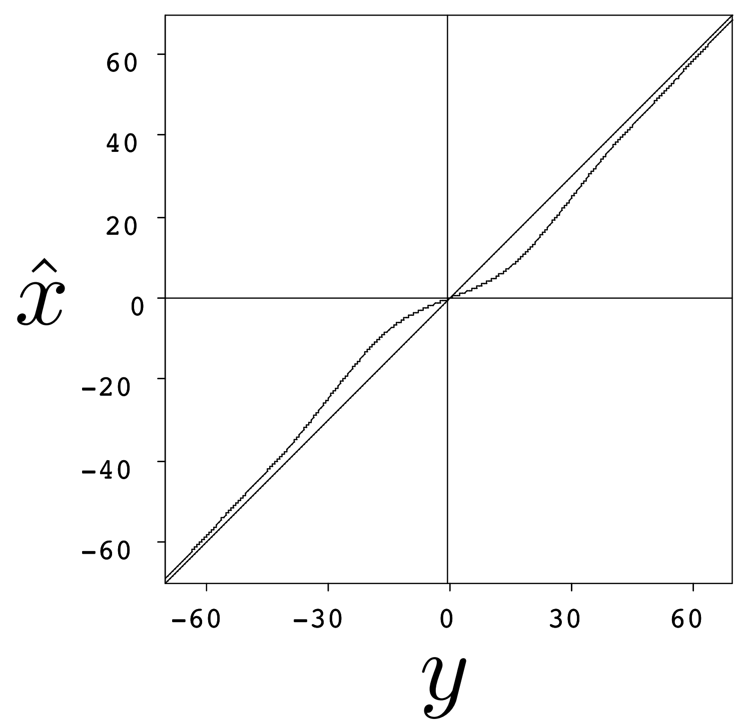
$$y = x + n$$

$$P(x) = \frac{1}{Z_s} e^{-|\frac{x}{s}|^p}$$

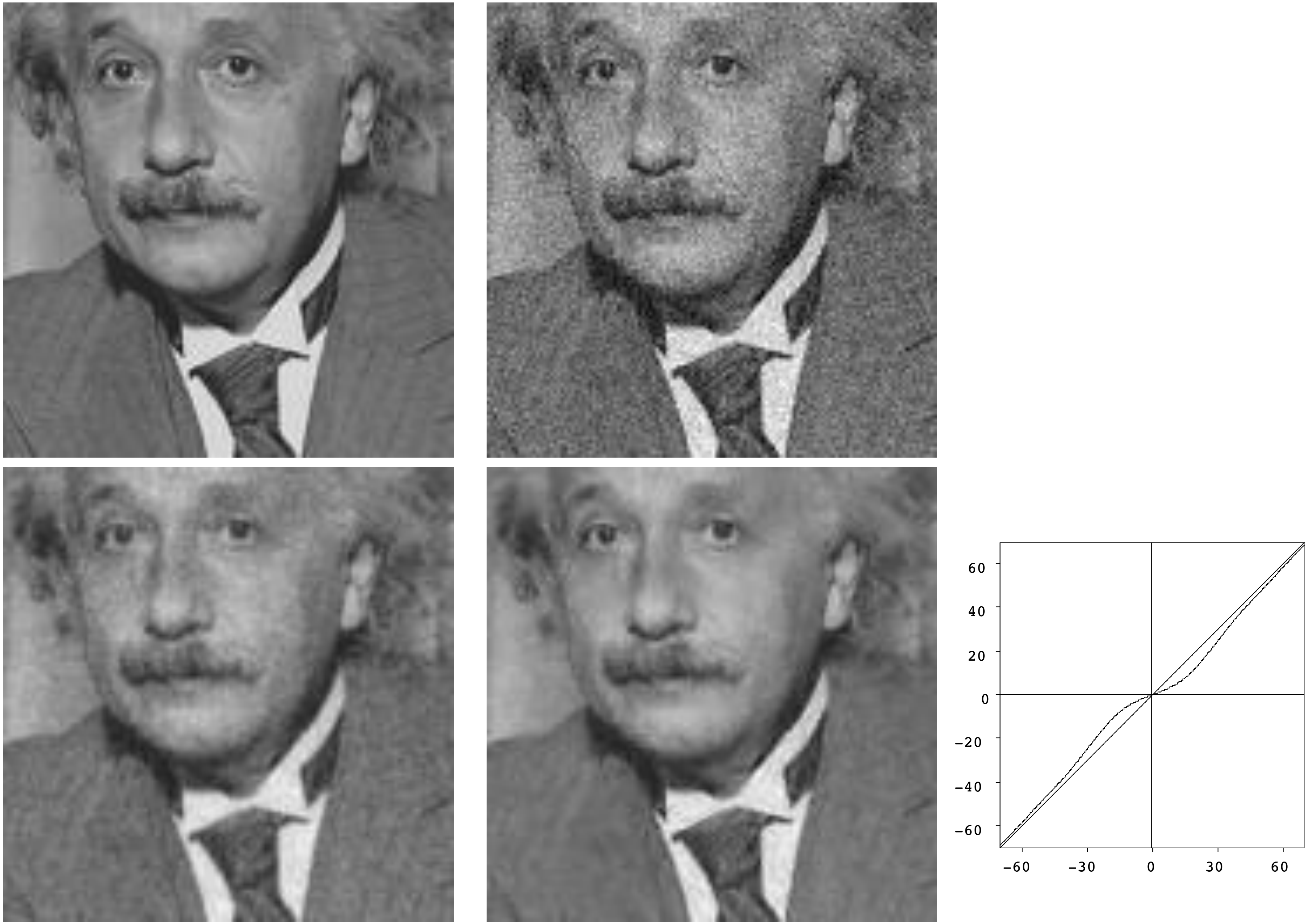
$$P(x|y) \propto P(y|x)\, P(x)$$

MAP estimate:

$$\hat{x} = \arg\min_x \left[\frac{|y - x|^2}{2\sigma_n^2} + |\frac{x}{s}|^p \right]$$

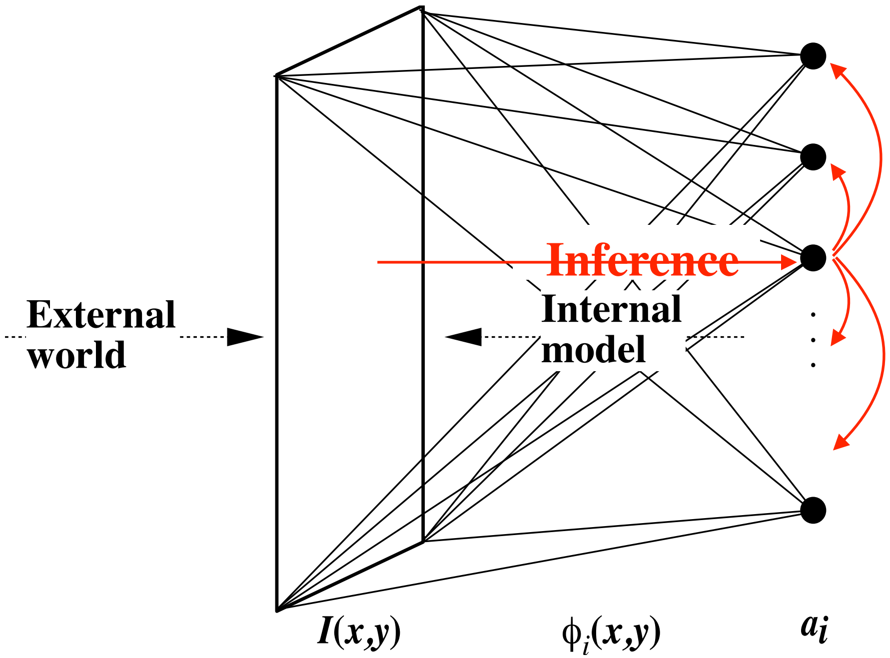


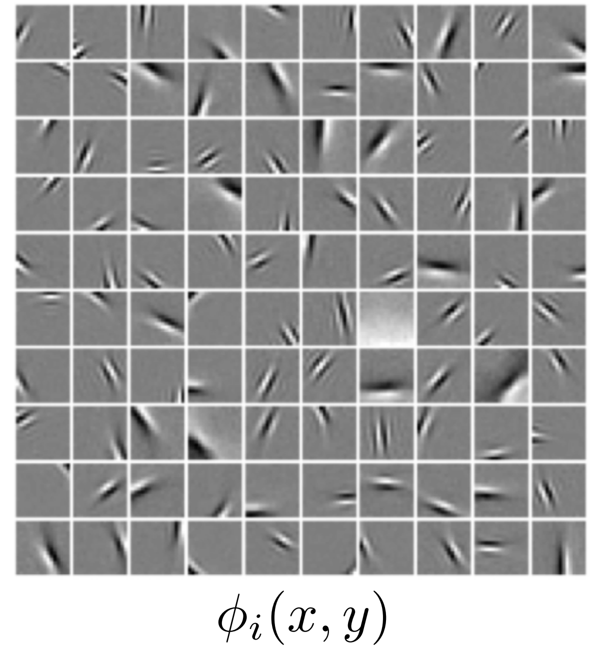
original image | image + noise



Wiener filter | wavelet coring

Sparse coding model





Inference: $$P(\mathbf{a}|\mathbf{I};\boldsymbol{\Phi}) \propto P(\mathbf{I}|\mathbf{a};\boldsymbol{\Phi})\, P(\mathbf{a})$$

$$\hat{\mathbf{a}} = \arg\min_{\mathbf{a}} |\mathbf{I} - \Phi\,\mathbf{a}|^2 + \lambda \sum_i C(a_i)$$

Learning: $$\hat{\boldsymbol{\Phi}} = \arg\max_{\boldsymbol{\Phi}} \langle \log P(\mathbf{I}|\boldsymbol{\Phi})\rangle$$

$$P(\mathbf{I}|\boldsymbol{\Phi}) = \int P(\mathbf{I}|\mathbf{a},\boldsymbol{\Phi})\, P(\mathbf{a})\, d\mathbf{a}$$

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

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

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

Its the ICA learning rule!

$$\Delta \mathbf{A} \propto \frac{\partial}{\partial \mathbf{A}} \langle \log p(\mathbf{x}) \rangle$$

$$= \frac{\partial}{\partial \mathbf{A}} \left[-\sum_i C(s_i) - \log \det \mathbf{A} \right]$$

$$= \langle [\mathbf{A}^T]^{-1} \mathbf{z}(\mathbf{s})\, \mathbf{s}^T - [\mathbf{A}^T]^{-1} \rangle$$

Pre-multiplying by ***A A***^{T} (natural gradient) yields:

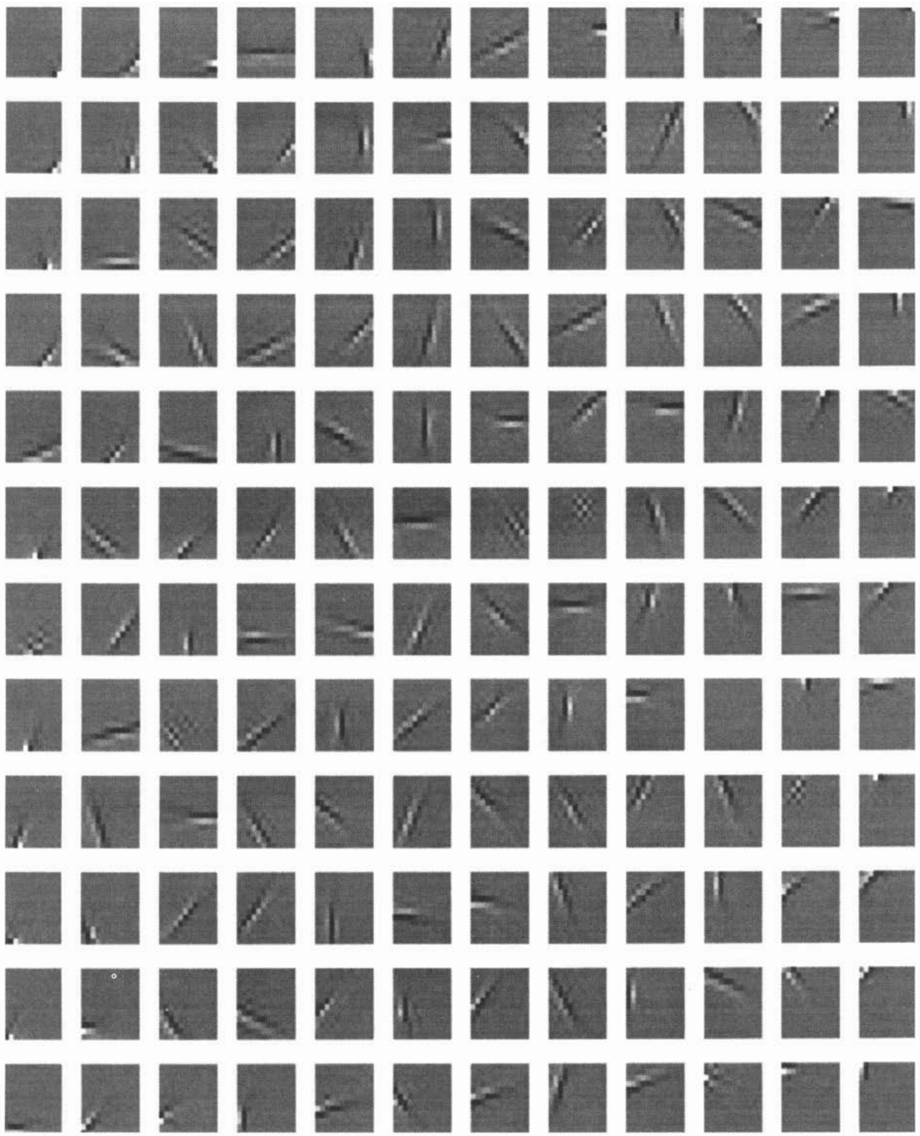
$$\Delta \mathbf{A} \propto \langle \mathbf{A}\, \mathbf{z}\, \mathbf{s}^T - \mathbf{A} \rangle$$

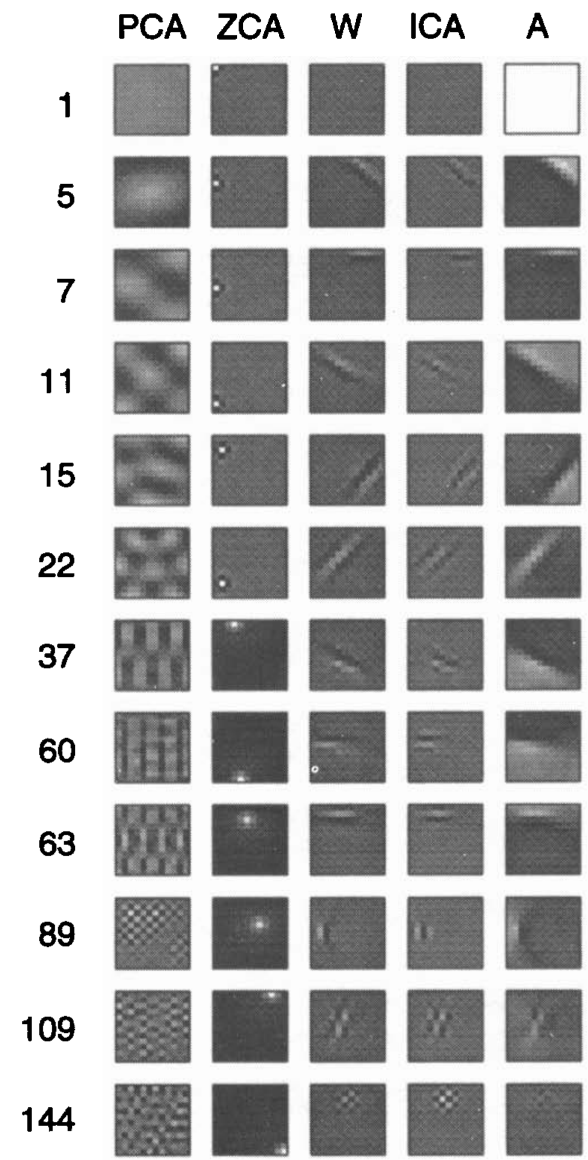
$$= \langle [\mathbf{x} - \mathbf{A}(\mathbf{s} - \mathbf{z})]\, \mathbf{s}^T - \mathbf{A} \rangle$$

The "Independent Components" of Natural Scenes are Edge Filters

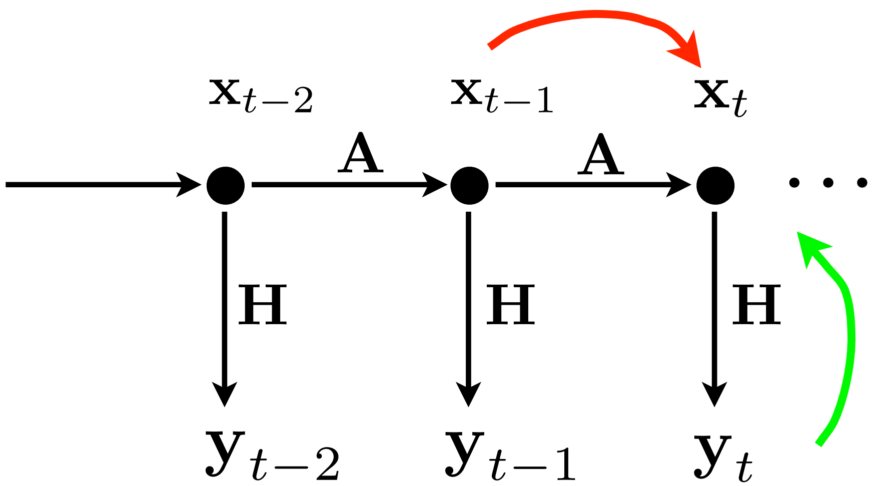
ANTHONY J. BELL,*† TERRENCE J. SEJNOWSKI*

Received 16 July 1996; in revised form 9 April 1997





First-order Markov process ('Kalman filter')



Linear generative model:

$$\mathbf{x}_t = \mathbf{A}\,\mathbf{x}_{t-1} + \mathbf{w}_{t-1}$$

$$\mathbf{y}_t = \mathbf{H}\,\mathbf{x}_t + \mathbf{n}_t$$

Prediction:

$$P(\mathbf{x}_t|\mathbf{y}_0 ... \mathbf{y}_{t-1}) = \int_{-\infty}^{\infty} P(\mathbf{x}_t|\mathbf{x}_{t-1})\, P(\mathbf{x}_{t-1}|\mathbf{y}_0 ... \mathbf{y}_{t-1})\, d\mathbf{x}_{t-1}$$

Update:

$t \leftarrow t+1$

$$P(\mathbf{x}_t|\mathbf{y}_0 ... \mathbf{y}_t) \propto P(\mathbf{y}_t|\mathbf{x}_t)\, P(\mathbf{x}_t|\mathbf{y}_0 ... \mathbf{y}_{t-1})$$