Topographic Independent Component Analysis

Presentation outline:

- ICA
- Applications of ICA
- Topography
- Topographic ICA
- Applications of Topographic ICA

ICA

Cocktail-Party Problem

- two people speaking simultaneously
- two microphones in different locations
- we want to separate the two speech signals

General Problem

- n unknown independent source signals s_i, $i = 1..n$ (independent components)
- n observed linear mixtures of these components

$$x_j = a_{j1}s_1 + \cdots + a_{jn}s_n, \text{for } j = 1..n$$

In vector-matrix notation: $\mathbf{x} = \mathbf{A}\mathbf{s}$

Without knowing $\mathbf{A}$, how do we solve for $\mathbf{s}$?

To solve, we will make assumptions about the distributions of s_i, estimate the mixing matrix A, then compute $\mathbf{s} = \mathbf{W}\mathbf{x}$, where $\mathbf{W} = \mathbf{A}^{-1}$.

Assumptions:

- components s_i
  - are statistically independent
  - have zero mean
  - have stationary, non-Gaussian distributions
- observations x_i are noise-free (to simplify the estimation process)

Ambiguities of ICA

1. cannot determine the variances of the independent components

$$x_i = (\frac{1}{c}a_i)(cs_i)$$

   - instead, assume each s_i has unit variance, $E[s_i^2] = 1$
   - still leaves ambiguity of sign

2. cannot determine the order of the components

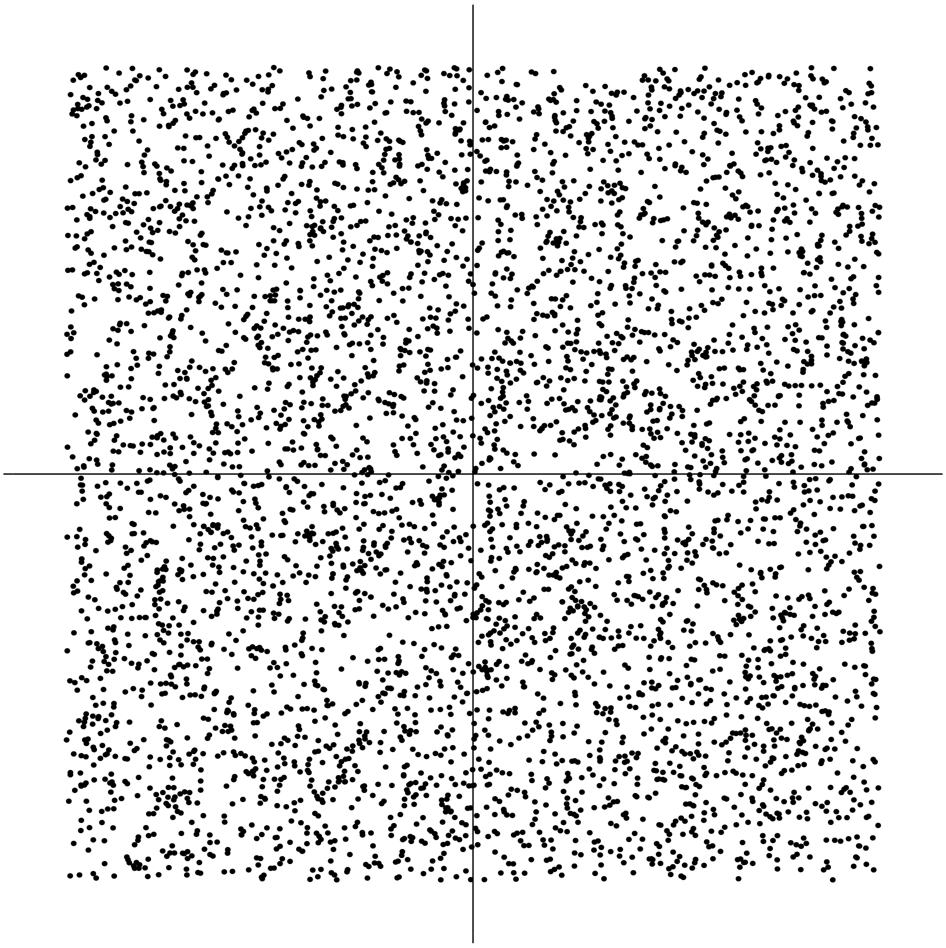
$$\mathbf{x} = \mathbf{A}\mathbf{P}\mathbf{P}^{-1}\mathbf{s}$$

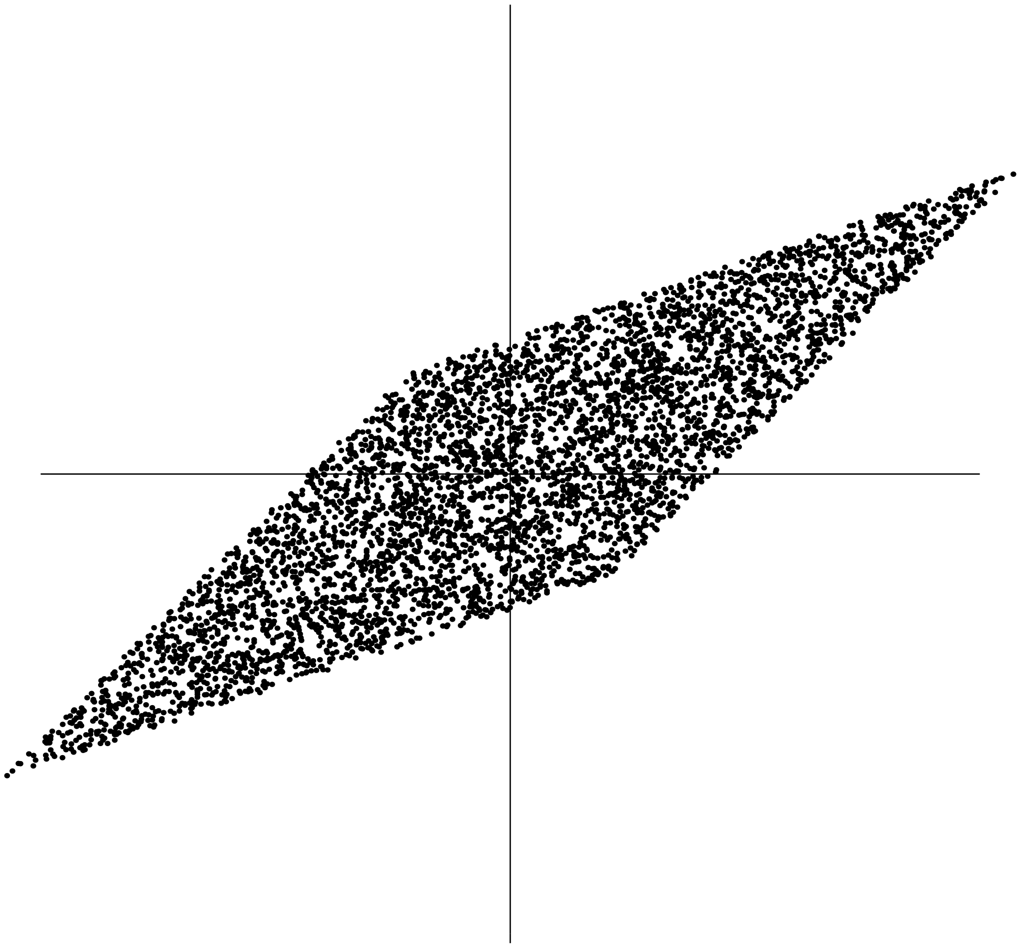
Illustration

- consider two independent components s_1, s_2 each with uniform distribution
- joint density of s_1 and s_2 then uniform on a square
- say the mixing matrix is

$$\mathbf{A} = \begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix}$$

- then the joint distribution of the mixed data x_1, x_2 is uniform on a parallelogram
- problem: how to estimate the mixing matrix A using only the observed mixtures x_1, x_2
- intuitive way: the edges of the parallelogram are in the directions of the columns of A
- however, this method only works with uniformly distributed random variables



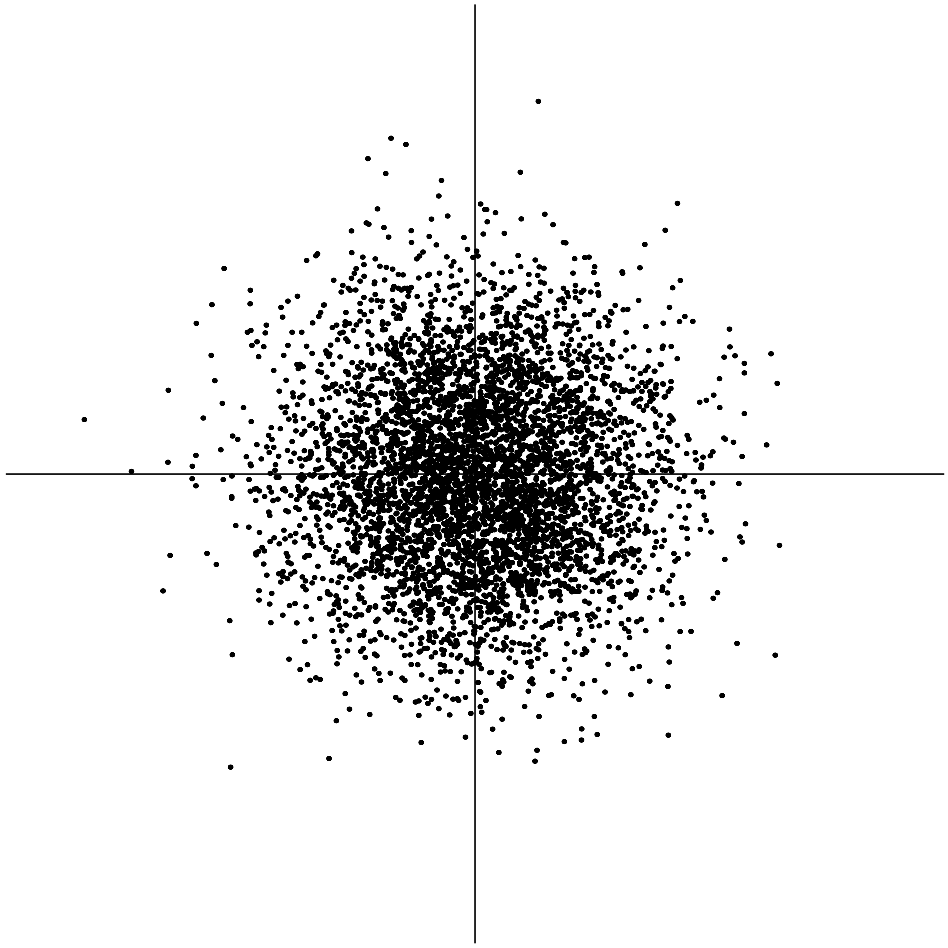


Why Gaussian Variables Are Forbidden

- fundamental restriction: independent components must be non-Gaussian for ICA to be possible
- say s_1, s_2 are Gaussian and the mixing matrix is orthogonal, then

  1. x_1, x_2 are Gaussian, uncorrelated, and of unit variance
  2. their joint density is completely symmetric
  3. joint density does not contain any information about the directions of the columns of A, so A cannot be determined

- it can be proven that the distribution of any orthogonal transformation of the Gaussian (x_1, x_2) is the same as (x_1, x_2)



ICA as Maximizing Non-Gaussianity

Central Limit Theorem: the distribution of the sum of independent random variables tends towards a Gaussian distribution

- we assume the source signals have structure
- as we mix them, structure is lost
- by choosing an unmixing vector w that maximizes the non-Gaussianity of $\mathbf{w}^T\mathbf{x}$, we should get one independent component

$$\begin{aligned} \mathbf{w}^T\mathbf{x} &= \mathbf{w}^T(\mathbf{A}\mathbf{s}) \\ &= [0 \ldots 010 \ldots 0]\mathbf{s} \\ &= s_i \end{aligned}$$

  ($\mathbf{w}^T$ is a row of $\mathbf{W} \equiv \mathbf{A}^{-1}$)

- subtract contribution of s_i from data and repeat to get more components

Measures of Non-Gaussianity

- **kurtosis**: normalized version of fourth moment of distribution

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

  distribution becomes less Gaussian as kurtosis gets further from zero

- **negentropy**: the more unstructured a random variable, the larger its entropy

  – Gaussian random variables have largest entropy among all random variables with equal variance

  – negentropy J of y, where H is entropy:

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

  in our case, $\mathbf{y} = \mathbf{W}\mathbf{x}$

ICA by Maximum Likelihood Estimation

- popular method for estimating the ICA model
- $h_{ML} = \arg\max_h P(D|h)$

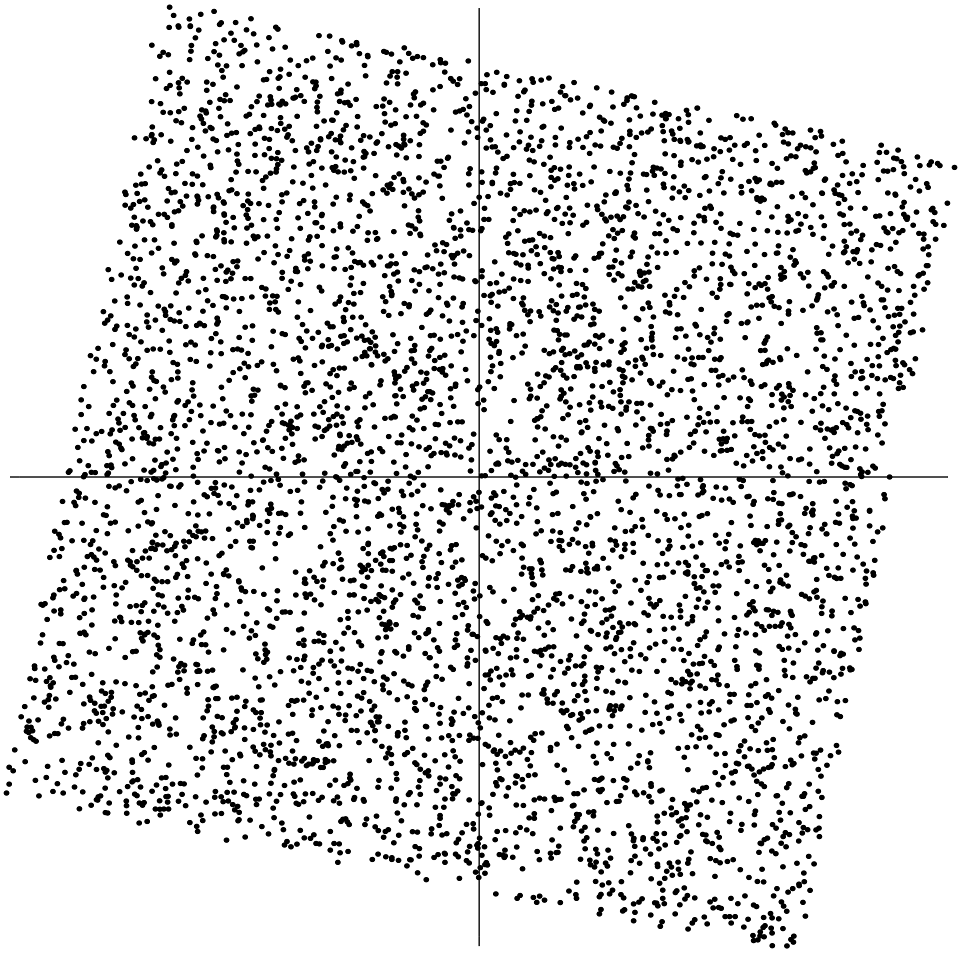
$$L = \prod_{t=1}^{T} f_{\mathbf{s}}(\mathbf{W}\mathbf{x}(t)) |\det \mathbf{W}|$$

$$\log L = \sum_{t=1}^{T} \sum_{i=1}^{n} \log f_{s_i}(\mathbf{w}_i^T \mathbf{x}(t)) + T \log |\det \mathbf{W}|$$

- problem: f_{s_i} must be estimated correctly, otherwise ML estimation will give completely wrong results

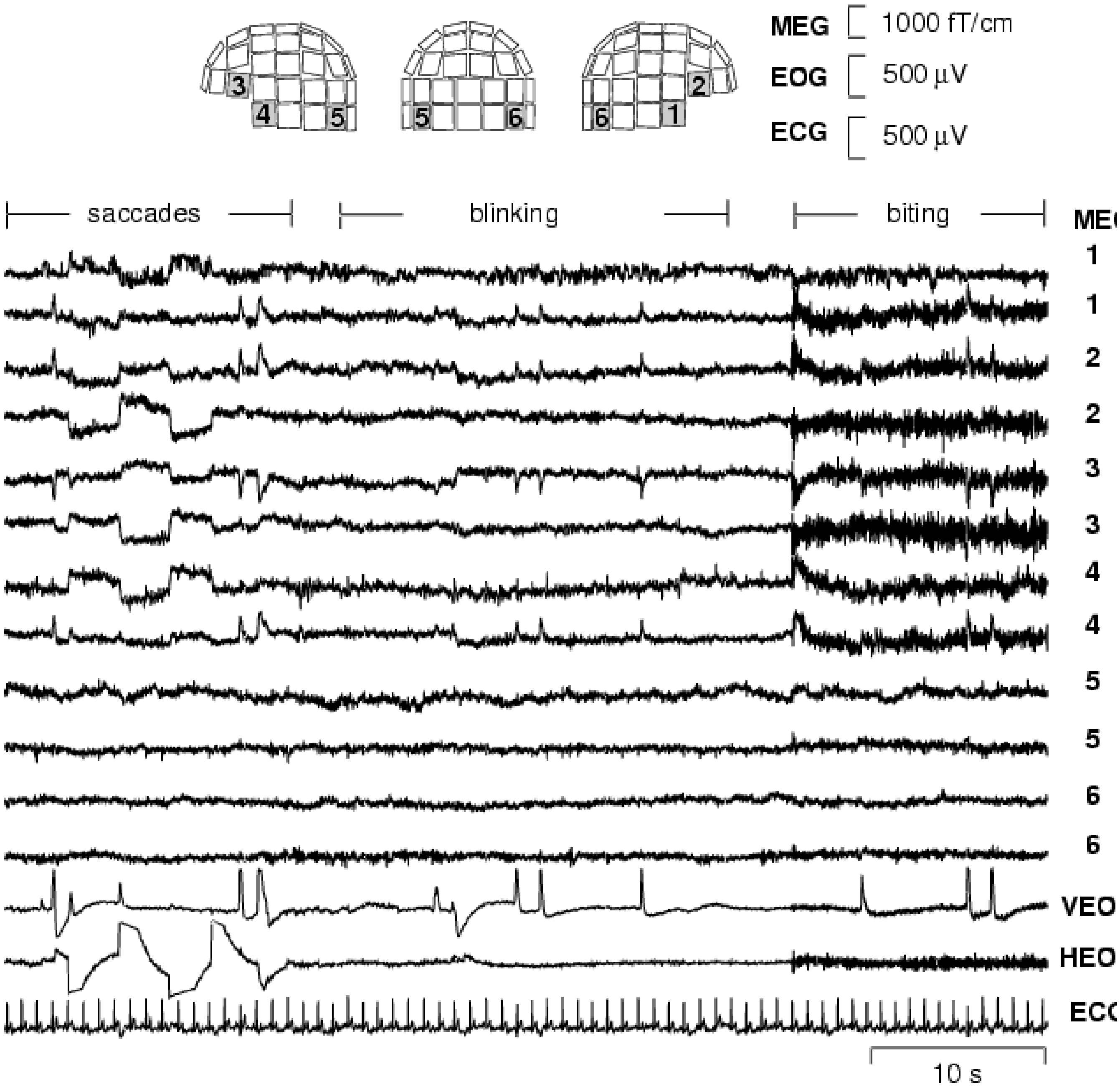
Preprocessing

1. centering x
   - subtract its mean vector $\mathbf{m} = E[\mathbf{x}]$, making x zero-mean
   - add $\mathbf{A}^{-1}\mathbf{m}$ back to centered estimate of s
2. whitening x
   - linear transformation applied to make observed signal data uncorrelated and of unit variance
   - constrains the new mixing matrix, $\tilde{A}$, to be orthogonal
   - now we only need to estimate $n(n-1)/2$ parameters, rather than the n^2 elements of A

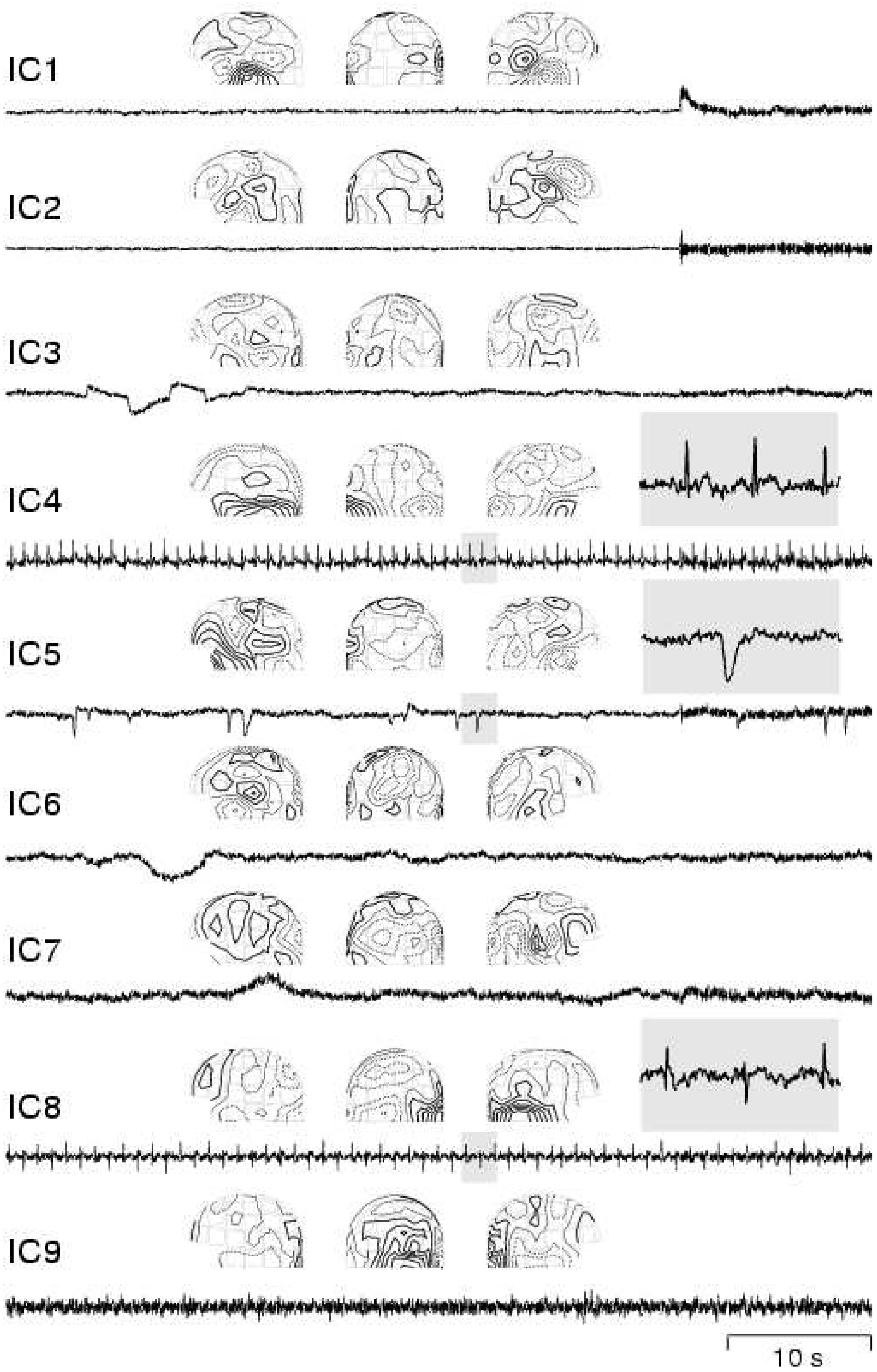


Separation of Artifacts in MEG Data

- Magnetoencephalography measures activity of the cortical neurons
- records 2 signals at each of 61 locations over the scalp $\rightarrow 122$ observed signals
- problem: extract essential features of neuromagnetic signals in the presence of artifacts (e.g. eye blinking, movement, sensor malfunctions)
- assumption: brain activity and artifacts are separate processes, so the signals they generate are statistically independent
- data: amplitudes $x_i(t)$ of the 122 signals at a certain time
- method: use PCA to whiten and reduce dimensionality of data, use ICA to determine some rows of W
- results: artifacts were isolated



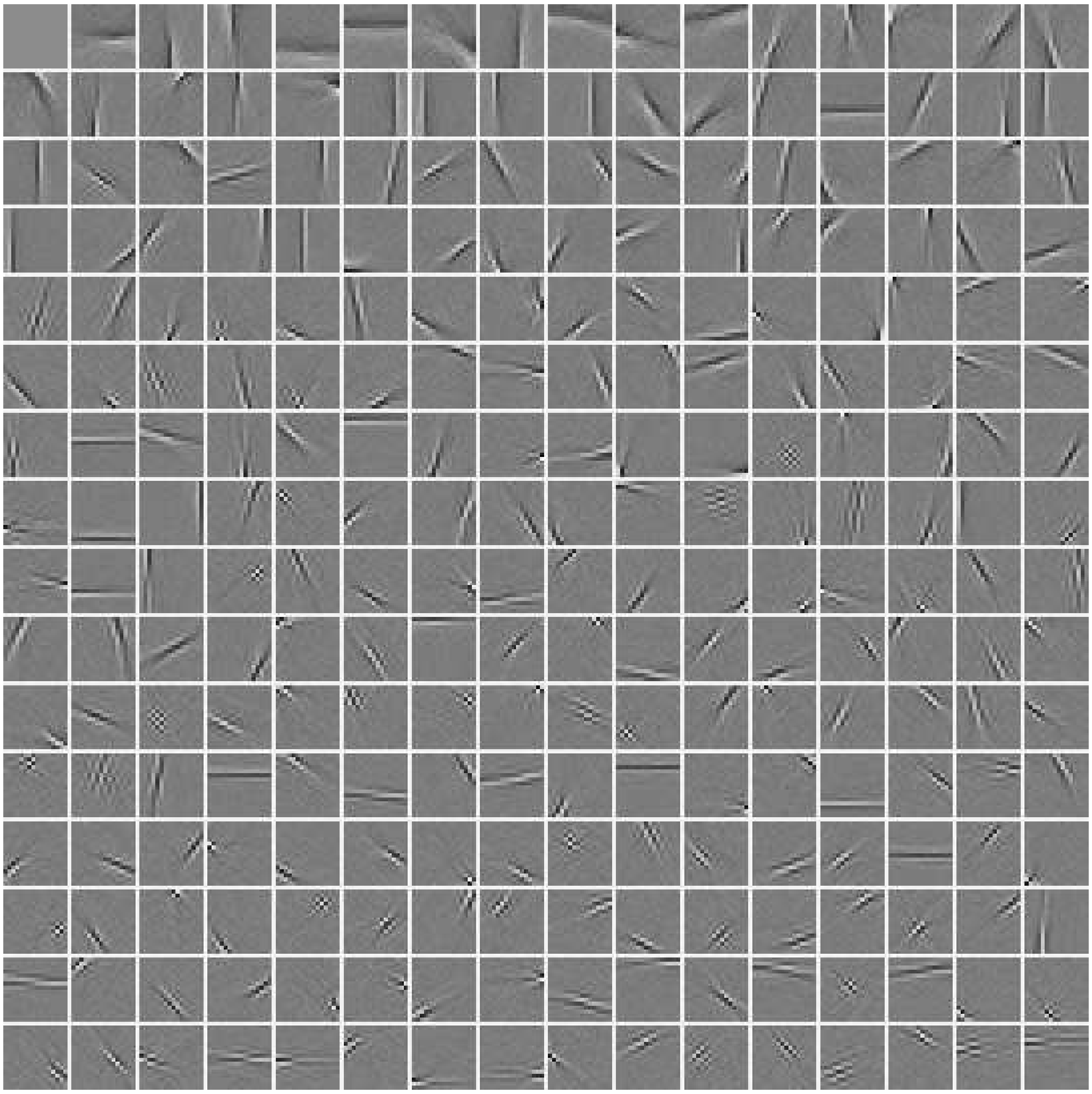
3
4
5
5
6
6
1
2
MEG 1000 fT/cm
EOG 500 µV
ECG 500 µV
saccades
blinking
biting
ME
1
1
2
2
3
3
4
4
5
5
6
6
VEO
HEO
ECG
10 s



IC1
IC2
IC3
IC4
IC5
IC6
IC7
IC8
IC9
10 s

Feature Extraction From Images

- image patches can be analyzed as a linear superposition of basis vectors (representing basic image features, e.g. edges, gradients, etc.)
- idea: use ICA to extract these basic features
- data: 16×16 pixel image patches taken at random locations from monochrome photographs depicting wild-life scenes $\rightarrow$ vector length 256
- method:
  - data centered by subtracting mean gray-scale value
  - solve for the 256 independent components using ICA



Topography

competitive learning

- neural net - on any input only one output unit, the "winner", is active at a time
- winner has largest net input
- only weights of winning unit are updated
- usually no significance in the geometric arrangements of our output units
- consider networks in which location of the winning unit conveys some info
  $\rightarrow$ nearby outputs correspond to nearby input patterns
- **topographic map**: a mapping that preserves neighborhood relations

Neighborhood Functions

define the distance between output units in a topography

- topography: 1-D line or ring, 2-D grid or torus
- $h(i, j) = 1$ when $i = j$
- $h(i, j)$ falls off as distance between i and j increases

examples

- $h(i, j) = \exp(-|unit_i - unit_j|^2/2\sigma^2)$
- $h(i, j) = \delta_{ij}$, the Kronecker delta function

Kohonen's Algorithm

- use ordinary competitive learning, but update weights going to the neighbors of the winning unit as well as those of the winning unit itself
- usually output units arranged in a 1-d or 2-d array

example

- map from 2d input space to a linear array of 50 output units
- weights of an output will correspond to a point in the input space
  $\rightarrow$ input close to this point will cause high activation in the output unit

Utility of Topography

- data visualization - topography provides additional information
- dimensionality reduction for future learning
- modeling - topographic mappings occur in the brain

e.g retinotopic map (2D) : retina $\rightarrow$ visual cortex

somatosensory map (1D) : skin $\rightarrow$ somatosensory cortex $\rightarrow$ image of body surface

tonotopic map (1D) : ear $\rightarrow$ auditory cortex $\rightarrow$ sounds of different frequencies laid out in order

Generative Topographic Mapping

- a “principled alternative” to Kohonen maps
- instead of mapping data from input space to, say, a 2d grid, assume data was generated from point (latent variables) in a 2d grid (latent space) then mapped non-linearly to the data space (+ Gaussian noise)
- a very similar model is used in Topographic ICA

Topographic ICA

- in classic ICA the estimated components, in practice, retain a residual dependence
- utilize the residual dependence to produce an ordering of the components
- idea: formulate a generative model for the components such that nearby components are not independent (relax the assumption of independence)

Independence

Estimated components may not be independent because

1. there may be no independent components

2. estimated distributions of s_i may not be correct

Usually require components to be uncorrelated

uncorrelated $\not\Rightarrow$ independent

Instead, try to *'maximize independence'*

The Generative Model

Dependency modeled

- cannot use linear correlation, since many ICA estimation methods require that the s_i's be uncorrelated
- instead use correlation of energies
  if s_i, s_j close in topography then
  $$\text{cov}(s_i^2, s_j^2) = E[s_i^2 s_j^2] - E[s_i^2]E[s_j^2] \neq 0$$
- idea: random variables have correlated energies if they tend to both be close to 0 or both far from 0 in absolute value at the same time

The Generative Model continued

- define sources $s_i = \sigma_i z_i$ where z_i are zero-mean, unit variance independent components and σ_i are standard deviation variables

- σ_i are generated to be related for neighboring s_i

- to do this, first define the topography via a neighborhood function, $h(i, j)$ (for components i and j)

$$\text{e.g.}\quad h(i,j) = \begin{cases} 1 & \text{if } |i-j| \leq m \\ 0 & \text{otherwise} \end{cases}$$

- next define

$$\sigma_i = \phi\Big(\sum_{k=1}^{n} h(i,k) u_k\Big)$$

where u_k are latent random variables, and ϕ is a non-linear function

Maximizing Likelihood

- joint probability density of $\mathbf{s}$ and $\mathbf{u}$

$$p(\mathbf{s}, \mathbf{u}) = p(\mathbf{s}|\mathbf{u})p(\mathbf{u})$$

u_i's are independent
s_i's are independent given their variances

$$p(\mathbf{s}, \mathbf{u}) = \prod_i p_i^s(\frac{s_i}{\phi(\sum_k h(i,k)u_k)})\frac{p_i^u(u_i)}{\phi(\sum_k h(i,k)u_k)}$$

- obtain the marginal density of $\mathbf{s}$, $f_{\mathbf{s}}$, by integration

$$f_{\mathbf{s}}(\mathbf{s}) = \int \prod_i p_i^s(\frac{s_i}{\phi(\sum_k h(i,k)u_k)})\frac{p_i^u(u_i)}{\phi(\sum_k h(i,k)u_k)}d\mathbf{u}$$

ML continued

- the maximum likelihood is (as before)

$$L(\mathbf{W}) = \prod_{i=1}^{T} f_{\mathrm{s}}(\mathbf{W}\mathbf{x}(t))|\det \mathbf{W}|$$

using our expression for f_{s} we get $L(\mathbf{W}) =$

$$\prod_{t=1}^{T} \int \prod_{i} p_i^s\left(\frac{s_i}{\phi(\sum_k h(i,k)u_k)}\right)\frac{p_i^u(u_i)}{\phi(\sum_k h(i,k)u_k)}|\det \mathbf{W}|d\mathbf{u}$$

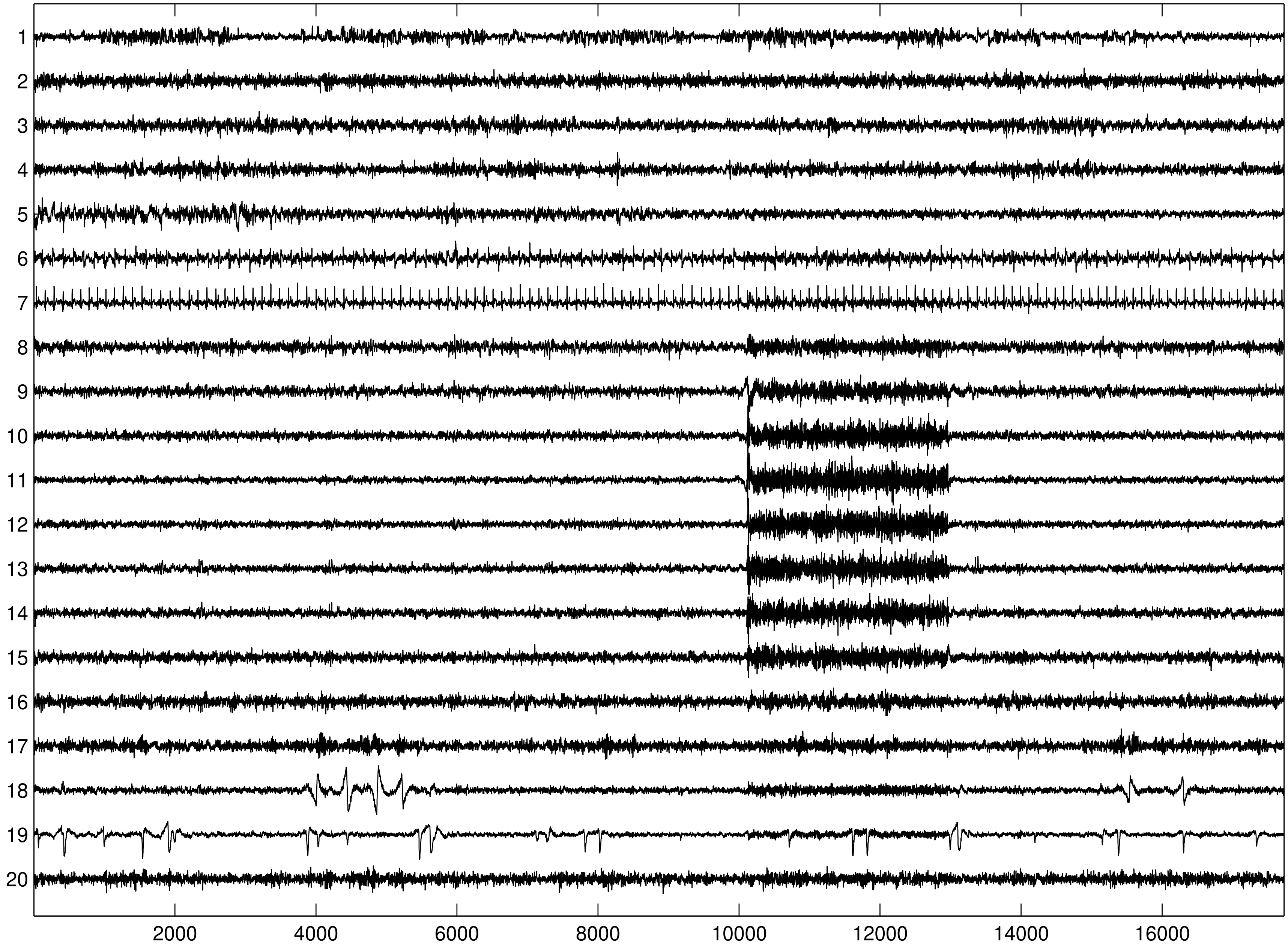
- problem: the above equation contains an intractable integral, so in practice it must be approximated

$$\log \tilde{L}(\mathbf{W}) = \sum_{t=1}^{T} \sum_{j=1}^{n} G\left(\sum_{i=1}^{n} h(i,j)(\mathbf{w}_i^T \mathbf{x}(t)^2)\right)$$

$$+T \log|\det \mathbf{W}|$$

Separation of Artifacts in MEG Data

- same data as in ICA experiment
- method:
  - 122 dimensional data first reduced to 20 dimensions by PCA
  - applied band-pass filter to eliminate low-frequency noise
  - topography: a 1-dimensional ring
- results:
  - signals corresponding to bites (9-15) are adjacent; can also determine that these are ordered according to the side of the head they originated from
  - eye artifacts (18 = saccades, 19 = blinks) are adjacent
  - heart beat (7) and digital watch(6) appear as adjacent (coincidence)

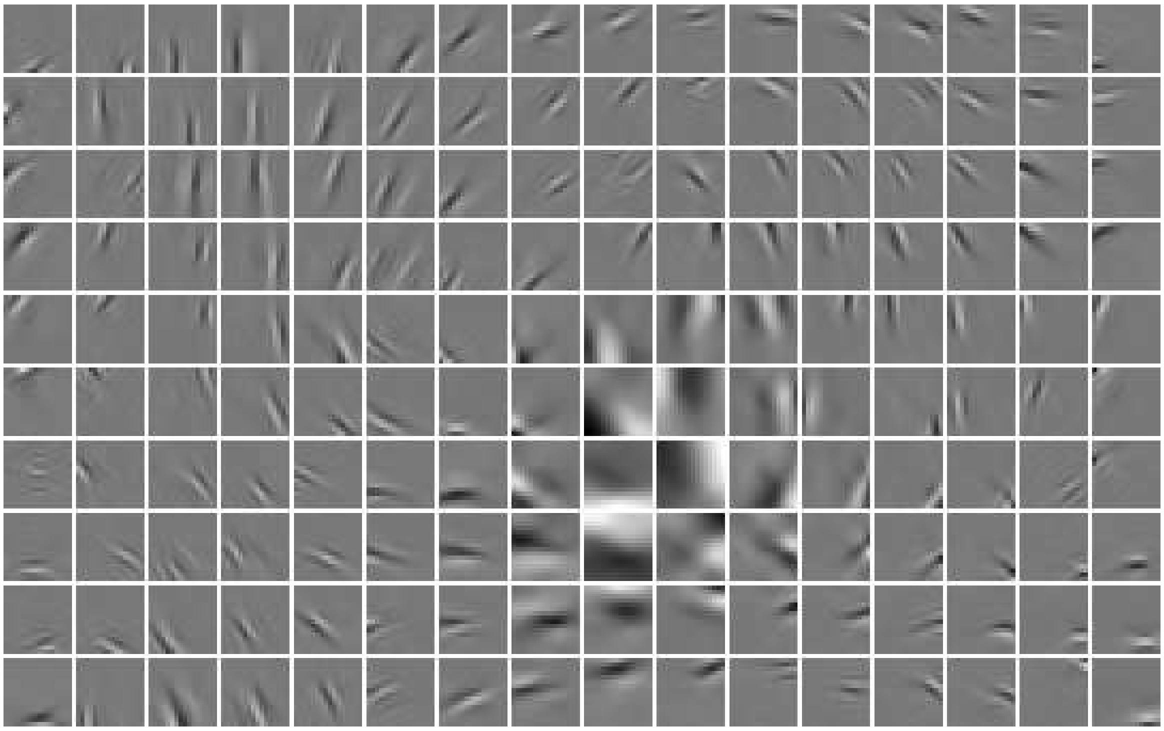

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
2000
4000
6000
8000
10000
12000
14000
16000

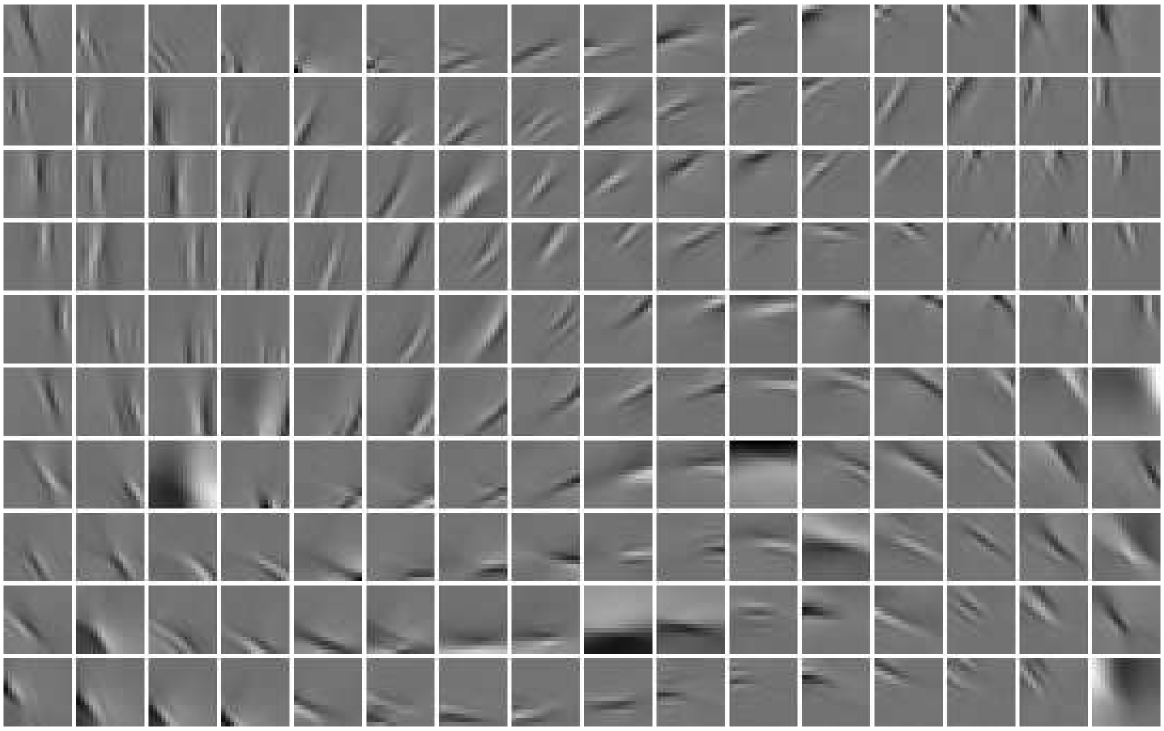
Feature Extraction from Images

- same wild-life scenes dataset as in ICA experiment
- method
  - data was low-pass filtered using PCA, reducing the dimension of the data vector
  - the 160 principal components with the largest variances were retained; data was then whitened by normalizing these variances
  - topography: 2d torus lattice
  - 3 different neighborhood sizes used:
    1. 3×3 square, $h(i, j) = 1$ if i, j adjacent, even obliquely, $= 0$ otherwise
    2. 5×5 square, similar h
    3. 5 non-obliquely adjacent elements

Feature Extraction Continued

- results
  - smaller 3×3 neighborhood: spatial frequency is important factor
  - larger 5×5 neighborhood: orientation and location most important; more elongated features estimated (different basis vectors)
  - last neighborhood was too small, giving very weak topographic ordering, similar to ordinary ICA
  - basis vectors produced are image filters





Discussion

- How useful is adding topography to ICA?
- more principled approaches to selecting
  - neighborhood function
  - distributions for u_i's
  - type of topography
    ("intrinsic dimensionality" of data)
- time/space complexity?
- desire independent components, why build dependence into model?