CSE-E4810 Machine Learning and Neural Networks (5 cr)

Lecture 9: Independent Component Analysis

Prof. Juha Karhunen

https://mycourses.aalto.fi/course/view.php?id=13086

Introduction

- Independent Component Analysis (ICA) is an important unsupervised (blind) technique for **non-Gaussian** data.
- It has many applications and extensions.
- Our discussion is based on the **tutorial article** *Aapo Hyvärinen and Erkki Oja,* "Independent Component Analysis: Algorithms and Applications," Neural Networks, vol. 13, 2000, pp. 411-430.
- It is still a good introduction to basic ICA.
- This article is available also from the www page `http://research.ics.aalto.fi/ica/`.
- A **comprehensive textbook:** *A. Hyvärinen, J. Karhunen, and E. Oja:* Independent Component Analysis, J. Wiley, 2001.

- See the home page of this book: `http//research.ics.aalto.fi/ica/book/`.
- In Du's and Swamy's book, independent component analysis is discussed in Chapter 14.
- Later on in this chapter some extensions of ICA are presented but mostly only literally with no mathematics.
- An important technique related with ICA is nonnegative matrix factorization (NMF).
- It is discussed briefly Chapter 13 in Du's and Swamy's book.
- This is a long lecture with a lot of stuff and new matters.

Motivation for independent component analysis (ICA)

- Let us start with an example: three people are speaking simultaneously in a room that has three microphones.
- Denote the microphone signals by $x_1(t), x_2(t)$, and $x_3(t)$.
- Each is a weighted sum of the speech signals which we denote by $s_1(t), s_2(t)$, and $s_3(t)$:

$$\begin{aligned} x_1(t) &= a_{11}s_1(t) + a_{12}s_2(t) + a_{13}s_3(t) \\ x_2(t) &= a_{21}s_1(t) + a_{22}s_2(t) + a_{23}s_3(t) \\ x_3(t) &= a_{31}s_1(t) + a_{32}s_2(t) + a_{33}s_3(t) \end{aligned} \tag{1}$$

- Cocktail-party problem: estimate the original speech signals $s_i(t)$ (Figure 1) **using only the recorded signals** in Figure 2.

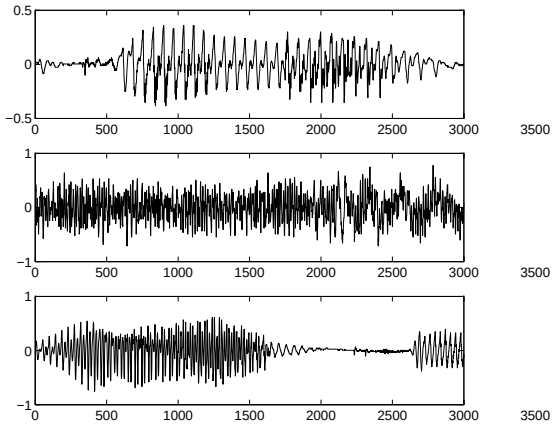

Figure 1: The original speech waveforms.

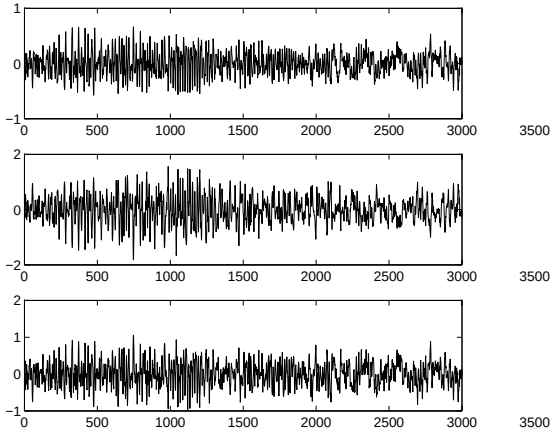

Figure 2: The observed microphone signals.

- As the weights a_{ij} are different, we may assume that the matrix $\mathbf{A} = (a_{ij})$ (although unknown) is invertible.
- Thus there exist another set of weights w_{ij} such that

$$\begin{aligned} s_1(t) &= w_{11}x_1(t) + w_{12}x_2(t) + w_{13}x_3(t) \\ s_2(t) &= w_{21}x_1(t) + w_{22}x_2(t) + w_{23}x_3(t) \\ s_3(t) &= w_{31}x_1(t) + w_{32}x_2(t) + w_{33}x_3(t) \end{aligned} \tag{2}$$

- It turns out that this **blind source separation (BSS)** problem can be solved using **independent component analysis (ICA).**
- In ICA, it suffices to assume that the sources $s_j(t)$ are **non-Gaussian and statistically independent.**
- An ICA solution to the speech separation example is shown in Figure 3.

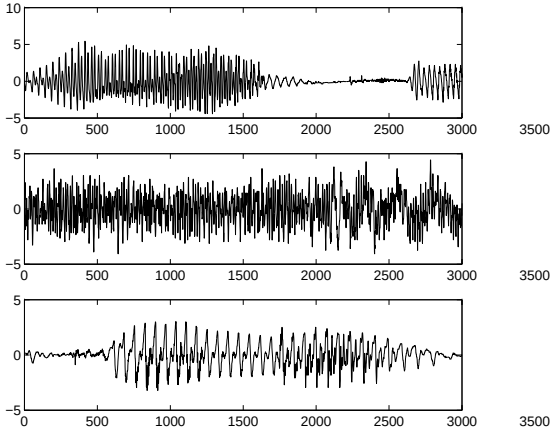

Figure 3: The estimates of the speech waveforms obtained by ICA.

.

Definition of Independent Component Analysis

- Assume that we observe n linear mixtures x_i of n latent variables, the **independent components** s_k.
- ICA model is a linear generative statistical latent variable model

$$x_i = a_{i1}s_1 + a_{i2}s_2 + \ldots + a_{in}s_n, \quad \text{for all } i = 1, \ldots, n \tag{3}$$

where the $a_{ij}, i, j = 1, \ldots, n$ are some real coefficients.

- This is the **basic linear ICA model,** which can be extended in many ways.
- In the basic ICA model, we assume that each mixture x_i as well as each independent component s_j is a **random variable.**
- Using vector-matrix formulation: let

$$\mathbf{x} = (x_1, ..., x_n)^T, \quad \mathbf{s} = (s_1, ..., s_n)^T, \quad \mathbf{A} = (a_{ij}) \tag{4}$$

- Then the basic ICA model is

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{5}$$

- If the columns of the $n \times n$ mixing matrix $\mathbf{A}$ are denoted $\mathbf{a}_j$, the model can also be written as

$$\mathbf{x} = \sum_{i=1}^{n} \mathbf{a}_i s_i \tag{6}$$

- **The ICA problem:** Estimate both $\mathbf{A}$ and $\mathbf{s}$ when only $\mathbf{x}$ is observed and the distribution of $\mathbf{s}$ is unknown.
- Denote the solution formally by

$$\mathbf{s} = \mathbf{W}\mathbf{x} \tag{7}$$

- There $\mathbf{W} = \mathbf{A}^{-1}$ is an $n \times n$ separating matrix which should be determined from the data $\mathbf{x}$ only.

Assumptions made in the basic ICA model

1. The independent components s_i are **statistically independent.**
   - Mathematically, the joint probability density $p(s_i, s_j)$ of s_i and s_j must factorize to the product of the pdf's of s_i and s_j:

$$p(s_i, s_j) = p(s_i)p(s_j) \tag{8}$$

   - Intuitively: knowing the value of random variable s_i does not give information about the value of s_2.
2. The independent components must have **non-Gaussian distributions.**
   - However, one of the independent components can be Gaussian.
   - We **need not know** these non-Gaussian distributions.
3. The unknown **mixing matrix $\mathbf{A}$ is square.**
   - That is, the number of independent components is equal to the

number of observed mixtures.

4. There is **no noise** in the ICA model (5).

- Most of these assumptions can be relaxed in various extensions of basic ICA.
- But the problems and methods needed for solving them become more complicated.

Indeterminacies in the basic ICA model

- Due to the blind nature of the ICA problem, there are several ambiguities in solving it.
  1. **Scaling:** The independent components (source signals) can be found only up to a multiplicative constant:

$$\mathbf{x} = \mathbf{A}\mathbf{s} = (c\mathbf{A})(\frac{1}{c}\mathbf{s}) \tag{9}$$

- Usually the variance of each source is normalized to unity to fix the scale.

2. **Sign:** The sign of found independent components can be chosen freely.
3. **Order:** The order of the independent components cannot be determined
   - Unless some extra criterion is used to that end.

- This is why only the waveforms of the independent components can be recovered without extra prior information.
- But this is sufficient and quite useful in many applications of ICA.
- If there are more than one Gaussian sources (independent components), they can only be estimated up to an orthogonal transformation.
- This is because orthogonal transformation of a multivariate Gaussian

distribution is still Gaussian.

- Another way of realizing this is that uncorrelated Gaussian random variables are also statistically independent.
- This property does not hold in general for any other probability distribution.
- Random variables can be made mutually uncorrelated in infinitely many ways.
- From uncorrelated random variables one can easily generate other sets of uncorrelated random variables by rotating them using an orthogonal transformation matrix.

Comparing PCA with ICA

- Before proceeding, let us compare principal component analysis (PCA) and independent component analysis (ICA).

- Both use a similar simple linear data model

$$\mathbf{x} = \mathbf{A}\mathbf{s} = \sum_{i=1}^{n} \mathbf{a}_i s_i \tag{10}$$

- Furthermore, in both methods only the data vectors $\mathbf{x}$ are known.
- But PCA and ICA differ in assumptions made on the model (10).
- In PCA, the basis vectors $\mathbf{a}_i$ are required to be mutually orthogonal: $\mathbf{a}_i^T \mathbf{a}_j = 0$ for $i \neq j$.
- The variances $\mathrm{E}(s_i^2)$ of the principal components s_i (assuming zero mean $\mathrm{E}(\mathbf{x}) = \mathbf{0}$) have maximal values.
- While in ICA the components s_i are required to be statistically independent (or as independent as possible).
- This is a strong but often natural condition that determines the ICA expansion.

- Without imposing any other constraints onto the basis vectors $\mathbf{a}_i$ of ICA that they are linearly independent.
- The basis vectors of ICA are in general non-orthogonal.
- The PCA expansion has without any extra conditions exactly the same scaling, ordering, and sign ambiguities as the ICA expansion.
- However, usually the scaling indeterminacy is fixed in PCA by requiring that the basis vectors $\mathbf{a}_i$ have unit norm.
- Furthermore, the basis vectors of PCA are ordered according to lowering variances (eigenvalues of the data covariance matrix).
- The sign ambiguity still remains in PCA after these conventions.
- PCA is based on second-order statistics (covariances) of the data.
- Generally speaking, PCA is optimal for Gaussian data.

- Multivariate Gaussian data is determined completely by its:
  - First-order statistics, the mean vector $\mathbf{m} = \mathrm{E}(\mathbf{x})$.
  - Second-order statistics, the covariance matrix $\mathbf{C} = \mathrm{E}\{(\mathbf{x} - \mathbf{m})(\mathbf{x} - \mathbf{m})^T\}$.
- While ICA is based on higher-order statistics.
- Especially after mean centering and whitening which normalize the data with respect to its first-order and second-order statistics.
- ICA is in its own sense optimal for non-Gaussian data.
- Most practical data sets are non-Gaussian, carrying a lot of information in their higher-order statistics.
- Standard linear methods based on at most second-order statistics neglect this extra information.

Example: Uniformly distributed data

- Figure 4 shows a data set (green dots) that is uniformly distributed inside the parallelogram.
- The basis vectors $\mathbf{e}_1$ and $\mathbf{e}_2$ of PCA are shown in red.
- The first basis vector $\mathbf{e}_1$ happens to point out to the corner of the parallelogram.
- Into the direction in which the components of the data would be maximally dependent.
- But the other basis vector $\mathbf{e}_2$ of PCA does not have any natural interpretation; it is just orthogonal to $\mathbf{e}_1$.
- While the basis vectors $\mathbf{a}_1$ and $\mathbf{a}_2$ of ICA (in blue) describe well the underlying uniform distribution.
- Their directions are along the sides of the parallelogram.

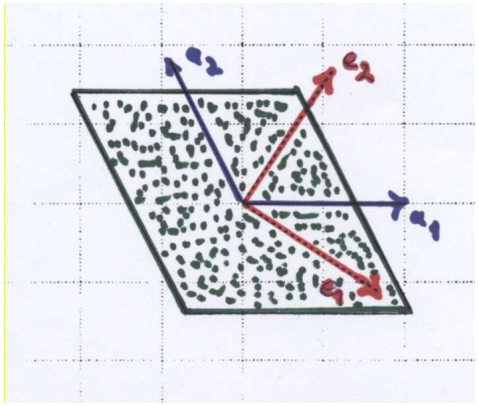

Figure 4: Uniformly distributed data (green) together with the basis vectors of PCA (red) and ICA (blue).

.

Maximization of non-Gaussianity

- An important principle leading to ICA is maximization of non-Gaussianity.
- That is, one tries to find components s_i that are as non-Gaussian as possible.
- It turns out that this principle yields the independent components.
- We justify this criterion heuristically in the following, but it can be shown to hold rigorously.
- Consider estimation of one independent component by

$$\mathbf{w}^T\mathbf{x} = \sum_{j=1}^{n} w_j x_j \tag{11}$$

- There $\mathbf{w}^T$ is some row vector of the separating matrix $\mathbf{W}$.

- Recall that $\mathbf{s} = \mathbf{W}\mathbf{x}$, where $\mathbf{W} = \mathbf{A}^{-1}$, and $\mathbf{x} = \mathbf{A}\mathbf{s}$.
- Let us now change the variables: $\mathbf{z} = \mathbf{A}^T\mathbf{w}$. Then

$$\mathbf{w}^T\mathbf{x} = \mathbf{w}^T\mathbf{A}\mathbf{s} = \mathbf{z}^T\mathbf{s} \,, \tag{12}$$

  which is a linear combination of the sources.
- Assume now for simplicity that all the components s_i have identical distribution, and apply the
- **Central limit theorem:** Sum of independent and identically distributed (i.i.d.) random variables tends toward Gaussian distribution.
- Since $\mathbf{z}^T\mathbf{s}$ is such a sum, it is more Gaussian than any of the s_i.
- If its non-Gaussianity is maximized, it must tend toward one of the independent components s_i.
- Therefore, if we choose the vector $\mathbf{w}$ so that the non-Gaussianity of

$\mathbf{w}^T\mathbf{x}$ is maximized, we get an estimate of some independent component s_i.

- In practice, weighted sum of random variables tends toward Gaussianity rapidly when the number of terms in the sum grows.
- This holds even if the distributions of the random variables are quite different, far from being i.i.d.
- Thus one approach to ICA is:
  1. Construct a measure of the non-Gaussianity of the estimates of the components s_i;
  2. Maximize that measure.
- We shall consider in more detail **two measures of non-Gaussianity** which have rather different properties.
- Namely **kurtosis** and **negentropy.**

Kurtosis

- Kurtosis is a classical measure of non-Gaussianity.
- For a scalar random variable y it is defined by

$$\text{kurt}(y) = \text{E}\{y^4\} - 3(\text{E}\{y^2\})^2 \tag{13}$$

- Assuming that y has unit variance (and mean zero), $\text{E}\{y^2\} = 1$.
- The kurtosis is then simply the fourth moment

$$\text{kurt}(y) = \text{E}\{y^4\} - 3 \tag{14}$$

- It can be shown that the kurtosis is zero for Gaussian y.
- Two types of non-Gaussianity are distinguished based on the value of kurtosis.
- **Super-Gaussian** signals or random variables y have positive kurtosis.

- Their probability density functions (pdf's) have typically a sharper peak and longer tail than the Gaussian pdf.
- **Sub-Gaussian** random signals have negative kurtosis.
- Their pdf's are flatter than the Gaussian pdf or multimodal.
- Figure 5 shows an example of the super-Gaussian Laplacian pdf and Figure 6 the sub-Gaussian uniform density.
- Non-Gaussianity can be measured by the absolute value of kurtosis.
- **Properties of kurtosis:**
  - \+ Computationally and theoretically simple
  - − Sensitive to outliers (not robust), depending on the fourth moment of the data.
  - − Non-symmetric: the degree of non-Gaussianity of super-Gaussian and sub-Gaussian signals cannot be compared directly.

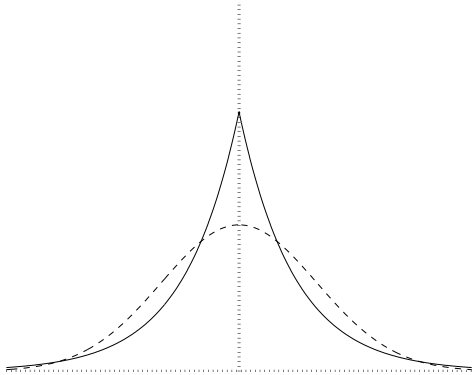

Figure 5: Super-Gaussian Laplacian probability density (solid line) compared with the Gaussian probability density (dashed line)

.

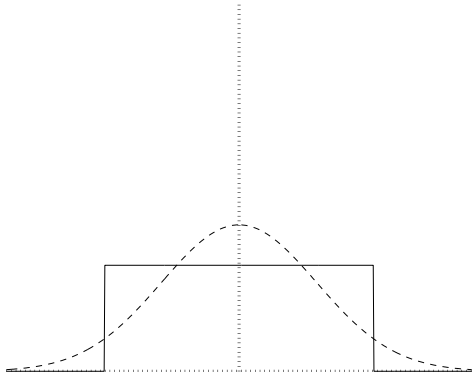

Figure 6: Sub-Gaussian uniform probability density (solid line) compared with the Gaussian probability density (dashed line)

.

Information-theoretic criteria

Entropy and differential entropy

- The entropy H is a well-known information-theoretic measure of the information contents and randomness of a random variable.
- Consider first a **discrete random variable** Y with possible values $y_1, y_2, \ldots, y_n$.
- The **entropy** of Y is then defined as

$$H(Y) = -\sum_{i=1}^{n} p(y_i) \log p(y_i) \tag{15}$$

- There $p(y_i)$ is the probability of the value y_i.
- The base of the logarithm is arbitrary.
- The values of entropy satisfy $0 \leq H(Y) \leq \log(n)$.

- The discrete entropy H attains its maximum value for the **uniform distribution** $p(y_i) = 1/n$, $i = 1, 2, \ldots, n$.
- **Interpretations of discrete entropy:**
  - Average amount of information ("surprise") or
  - Average reduction of uncertainty obtained by observing the value of Y, or
  - Optimal average code-length required for transmitting the values of Y.
- Assume now that $\mathbf{Y}$ is a **continuous valued random vector,** with values $\mathbf{y}$.
- Then the quantity corresponding to the entropy is the **differential entropy,** defined by

$$h(\mathbf{Y}) = -\int p(\mathbf{y}) \log p(\mathbf{y}) d\mathbf{y} \tag{16}$$

- The maximum of the differential entropy is achieved when $\mathbf{Y}$ is **Gaussian.**
- Under the constraint that the random vectors $\mathbf{Y}$ compared have equal covariance matrices.
- Uniform distribution is the "least interesting" one (most random) among the discrete distributions.
- And the Gaussian distribution among the continuous distributions.
- Small entropy means that the random variable is not so random.
- It then contains a lot of deterministic information having some structure.
- We are often interested to find such structural information from the data studied.
- This gives another justification for maximizing the non-Gaussianity.

Negentropy

- **Negentropy** $J(\mathbf{Y})$ is defined for a continuous random vector $\mathbf{Y}$ as

$$J(\mathbf{Y}) = h(\mathbf{Y}_{\text{Gauss}}) - h(\mathbf{Y}) \tag{17}$$

- There $\mathbf{Y}_{\text{Gauss}}$ is Gaussian random vector having the same covariance matrix as $\mathbf{Y}$.
- Negentropy measures the deviation from the maximum of the differential entropy $h(\mathbf{Y})$ attained when $\mathbf{Y}$ is Gaussian.
- **Properties of negentropy:**
  - $+$ Well justified, takes into account all higher-order statistics;
  - $+$ Invariant to invertible linear transformations;
  - $-$ Computationally difficult: requires the estimation of the probability density of $\mathbf{Y}$.

- In general, reliable estimation of the probability density and entropy is very difficult for multivariate densities having long tails.
- In practical ICA algorithms, negentropy is usually approximated with either high-order moments or other contrast functions.

Mutual information

- Consider first two discrete scalar random variables X and Y.
- Their relation can be measured by the **mutual information**

$$I(X;Y) = H(X) - H(X|Y) \tag{18}$$

- There the **conditional entropy**

$$H(X|Y) = H(X,Y) - H(Y) \tag{19}$$

is the amount of uncertainty remaining about X after Y has been observed.

- In (19), the **joint entropy** $H(X,Y)$ of X and Y is defined as

$$H(X,Y) = -\sum_x \sum_y p(x,y) \log p(x,y) \tag{20}$$

- There $p(x,y)$ is the joint probability density of the discrete random variables X and Y.
- The summations in (20) are taken over all possible values x_i and y_j of X and Y.
- The conditional entropy $H(X|Y)$ is nonnegative and at most equal to the entropy $H(X)$ of X (if X and Y are statistically independent):

$$0 \leq H(X|Y) \leq H(X) \tag{21}$$

- The mutual information I is
  - Symmetric: $I(X;Y) = I(Y;X)$

  - Nonnegative: $I(X;Y) \geq 0$
  - Equal to zero only if X and Y are statistically independent
- The equations are the same for differential entropy h.
- All expressions can be generalized to more than two variables, e.g.

$$I(Y_1;\ldots;Y_n) = \sum_{i=1}^{n} H(Y_i) - H(Y_1,\ldots,Y_n) \tag{22}$$

- For two continuous random vectors $\mathbf{X}$ and $\mathbf{Y}$

$$I(\mathbf{X};\mathbf{Y}) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p_{\mathbf{X},\mathbf{Y}}(\mathbf{x},\mathbf{y}) \log(\frac{p_{\mathbf{x}}(\mathbf{x} \mid \mathbf{y})}{p_{\mathbf{x}}(\mathbf{x})}) d\mathbf{x} d\mathbf{y} \tag{23}$$

- The subscripts show the random variable of the pdf.
- Applying the Bayes formula to (23), it is easy to see that

$$I(\mathbf{X};\mathbf{Y}) = h(\mathbf{X}) + h(\mathbf{Y}) - h(\mathbf{X},\mathbf{Y}) \tag{24}$$

- The relationships of the quantities defined are illustrated in Figure 7.
- They hold for discrete and continuous random variables and vectors as well with appropriate changes in notations.

Kullback-Leibler divergence

- Consider first two probability densities $p(y)$ and $q(y)$ defined at n values $y_1, y_2, \ldots, y_n$ of the discrete scalar random variable y.
- The **Kullback-Leibler (KL) divergence** is defined for these pdf's as

$$D(p,q) = \sum_{i=1}^{n} p(y_i) \log \frac{p(y_i)}{q(y_i)} \tag{25}$$

- KL divergence is a measure of the dissimilarity ("distance") of the two probability distributions with pdf's p and q.
- For two pdf's $p(\mathbf{x})$ and $q(\mathbf{x})$ of the continuous random vector $\mathbf{X}$,

the Kullback-Leibler divergence is defined as

$$D(p,q) = \int_{-\infty}^{\infty} p(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x})} d\mathbf{x} \tag{26}$$

- In both cases, **the KL divergence has the following properties:**
  - $D(p,q) \geq 0$.
  - $D(p,q) = 0$ only if the two pdf's are the same: $p = q$.
  - But $D(p,q)$ is not symmetric, and hence it is not a proper distance measure.
- KL divergence measures the average amount of additional information contained in p, given that the distribution of q is known.
- KL divergence is a sensible measure of the similarity of two probability distributions, although it is not symmetric.

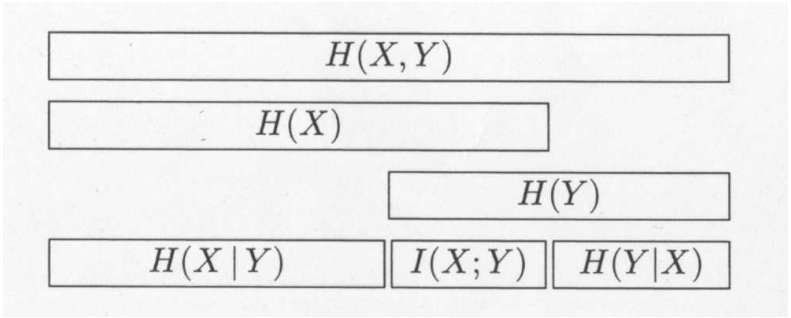

Figure 7: A schematic diagram describing the relations among the mutual information $I(X;Y)$ and the entropies $H(X)$ and $H(Y)$.

Relationships between various criteria in ICA

- It is easy to show that

$$I(\mathbf{X};\mathbf{Y}) = D(p_{\mathbf{X},\mathbf{Y}}, p_{\mathbf{X}}p_{\mathbf{Y}}) \tag{27}$$

  $\Rightarrow$ Mutual information measures the difference of the joint distribution $p_{\mathbf{X},\mathbf{Y}}(\mathbf{x},\mathbf{y})$ from the factored distribution $p_{\mathbf{X}}(\mathbf{x})p_{\mathbf{Y}}(\mathbf{y})$.
- For independent random variables the joint distribution is equal to the factored distribution.
- Therefore **mutual information is a natural measure of the deviation from independence.**
- It takes into account all kinds of dependencies between the random variables.
- Assume now that the values of the random vector $\mathbf{Y}$ are obtained from linear transformation $\mathbf{y} = \mathbf{W}\mathbf{x}$ like in ICA.

- Assume further that the components y_i, $i = 1, 2, \ldots, n$, of the vector $\mathbf{y}$ are constrained to be uncorrelated and have unit variance.
- Then it can be shown that

$$I(Y_1, \ldots, Y_n) = C - \sum_{i=1}^{n} J(y_i) \tag{28}$$

  where J is the negentropy, and C is a constant independent of the matrix $\mathbf{W}$.
- Therefore, we can conclude that
  Maximization of negentropy
  $\Leftrightarrow$
  Minimization of mutual information
  $\Leftrightarrow$
  Finding as independent components as possible.

ICA by minimization of mutual information

- From the previous considerations, we get an **alternative definition of independent component analysis (ICA):**
- ICA is a linear transformation $\mathbf{s} = \mathbf{W}\mathbf{x}$, such that the mutual information of the components s_i of the vector $\mathbf{s}$ is minimized.
- This is a natural definition since mutual information is a well justified measure of deviation from independence.
- Note that now we need not assume that the source signals $\mathbf{s}$ come from the data model $\mathbf{x} = \mathbf{A}\mathbf{s}$.
- The goal may simply be to find **as independent components as possible.**
- Note that the found components may still have some dependencies remaining.

- These considerations give a justification for the more heuristic maximization of nongaussianity (negentropy).
- This ICA finds directions in which the negentropy is maximized.
- More specifically, projections of the data vectors $\mathbf{x}$ onto these directions have maximum negentropy.
- Usually the projections are constrained to be uncorrelated which simplifies computation.
- ICA can be derived also using the maximum likelihood and Infomax principles but we shall not consider them here.

Relation to projection pursuit

- There are many methods for vizualizing and exploring multidimensional data.
- One can see at most in three dimensions, so some mapping methods

to two (or three) dimensions are needed.

- **Projection pursuit** is a method for finding "interesting" projections of multidimensional data:
  1. First a suitable measure of interestingness (called an "index") must be chosen and defined.
  2. The interesting projections are then found by maximizing that measure.
- Projection pursuit is especially useful for interactive exploratory data analysis.
- The commonly used indices measure the deviation from Gaussian distribution.
- One tries to find projections which are as non-Gaussian as possible.
- They contain interesting structural information on the data studied.

- Most projections of high-dimensional data sets are usually almost Gaussian.
- They don't contain any interesting information about the structure of the data.
- **Relation to ICA:**
  - Measures of non-Gaussianity $\Leftrightarrow$ Measures of interestingness
  - Independent components $\Leftrightarrow$ Interesting directions (projections)
- Note that no assumptions about independence or data models are made in projection pursuit.
- Therefore, the solution obtained by the ICA algorithms reveals:
  - The "real" independent components (source signals) if the ICA model holds;
  - The projection pursuit directions if the ICA model does not hold.

An example: Oil pipeline data

- Taken from the book M. Girolami, "Self-Organising Neural Networks: Independent Component Analysis and Blind Source Separation", Springer 1999, pp. 249-252.
- Oil pipeline data consists of 12-component data vectors.
- They measure the quantity of oil in a multi-phase pipeline carrying oil, water, and gas.
- Three flow regimes can occur within the pipeline: laminar, annular, and homogenous.
- These three distinct physical sources describe the data.
- The oil flow data is described in more detail on pp. 678-681 in the book C. Bishop, "Pattern Recognition and Machine Learning", Springer 2006.

- The data was projected to two dimensions for analyzing it and for finding clusters using different methods:
  - Generative Topographic Mapping (GTM), Figure 8
  - Independent Component Analysis (ICA), Figure 9
  - Principal Component Analysis (PCA), Figure 10
- GTM is a **nonlinear** mapping method based on a generative model, learned using the maximum likelihood method.
- It tries to realize self-organizing map (SOM) in a theoretically justified way.
- We shall discuss self-organizing map on the next lecture.
- The GTM method spreads the data fairly evenly in Figure 8
- However, in this projection pursuit application GTM performs clearly worse than standard **linear ICA** in Figure 9

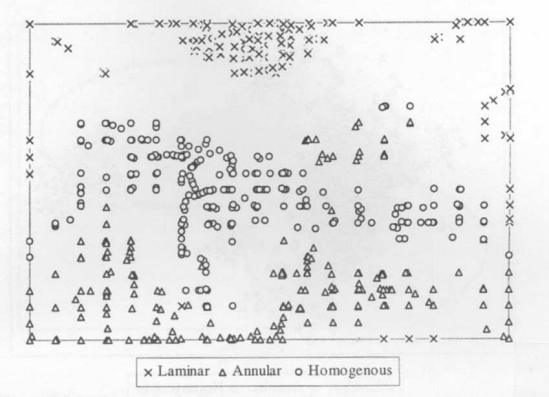

Figure 8: Plot of the oil flow data in the two-dimensional latent space of the GTM method.

.

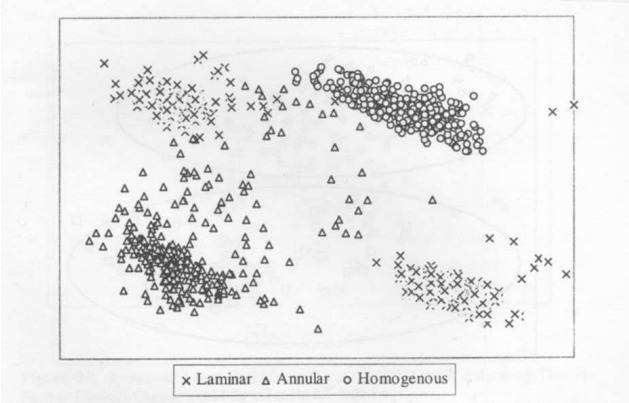

Figure 9: Plot of the oil flow data projected onto a two-dimensional ICA subspace.

.

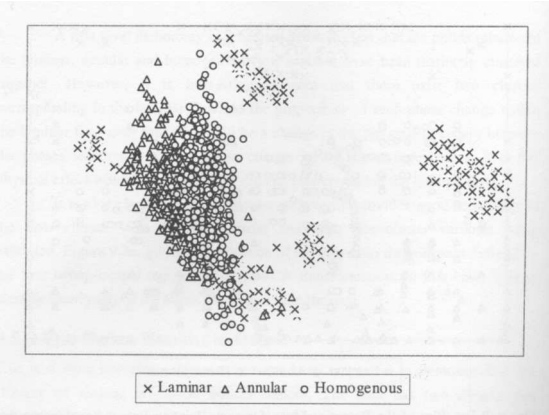

Figure 10: Plot of the oil flow data projected onto a two-dimensional PCA subspace.

.

- The ICA basis vectors for the mapping are chosen so that they correspond to maximally subgaussian (providing useful clusters) or nongaussian independent components.
- The linear mapping provided by the two first principal components corresponding to the largest eigenvalues in Figure 10 is also worse than the ICA mapping.

Preprocessing

- In most ICA methods, the data $\mathbf{x}$ is normalized with respect to its first and second-order statistics.
- This makes practical computation of ICA simpler.
- After that, ICA methods can concentrate on the higher-order statistics of the data studied.
- Utilizing higher-order statistics is a fundamental characteristics of

ICA.

- The first-order statistics of the data is its mean vector $\mathbf{m}_x = \mathrm{E}(\mathbf{x})$.
- In practice, $\mathbf{m}_x$ is estimated from the available samples and subtracted from $\mathbf{x}$.
- The mean can be added back to the estimated source or independent component vector $\mathbf{s}$ after computing the separating matrix $\mathbf{W}$.
- Simply add $\mathbf{W}\mathbf{m}$ to the estimated $\mathbf{s}$.
- As discussed earlier, **whitening** normalizes the data with respect to its the second-order statistics.
- Which corresponds to the "Gaussian" structure in the data.
- In whitening, the data $\mathbf{x}$ is transformed linearly as

$$\tilde{\mathbf{x}} = \mathbf{V}\mathbf{x} \tag{29}$$

so that the covariance matrix of the zero-mean whitened data $\tilde{\mathbf{x}}$

$$\mathrm{E}[\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T] = \mathbf{I} \tag{30}$$

- The components of the whitened data vectors $\tilde{\mathbf{x}}$ are uncorrelated and have unit variance.
- The whitening transformation can be computed in many ways.
- A standard method is to use PCA, which is based on the eigendecomposition of the covariance matrix $\mathbf{C}_{xx} = \mathrm{E}[\mathbf{x}\mathbf{x}^T]$ of the original data vectors $\mathbf{x}$.
- Let the eigenvalues of $\mathbf{C}_{xx}$ be $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \geq 0$.
- They are usually arranged in descending order into the diagonal matrix

$$\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n) \tag{31}$$

- The corresponding eigenvectors of $\mathbf{e}_i$ of $\mathbf{C}_{xx}$ are the column vectors

of the matrix

$$\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n] \tag{32}$$

- The whitening transformation is then

$$\tilde{\mathbf{x}} = \mathbf{\Lambda}^{-1/2}\mathbf{E}^T\mathbf{x} \tag{33}$$

- It transforms the mixing matrix to $\tilde{\mathbf{A}}$:

$$\tilde{\mathbf{x}} = \mathbf{\Lambda}^{-1/2}\mathbf{E}^T\mathbf{A}\mathbf{s} = \tilde{\mathbf{A}}\mathbf{s} \tag{34}$$

- The new mixing matrix $\tilde{\mathbf{A}}$ becomes orthogonal:

$$\mathrm{E}[\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T] = \tilde{\mathbf{A}}E[\mathbf{s}\mathbf{s}^T]\tilde{\mathbf{A}}^T = \tilde{\mathbf{A}}\tilde{\mathbf{A}}^T = \mathbf{I} \tag{35}$$

- The separating matrix $\tilde{\mathbf{W}}$ for the whitened data $\tilde{\mathbf{x}}$ becomes orthogonal, too:

$$\mathbf{s} = \tilde{\mathbf{W}}\tilde{\mathbf{x}} = \tilde{\mathbf{W}}\tilde{\mathbf{A}}\mathbf{s} \tag{36}$$

$$\Rightarrow \tilde{\mathbf{W}} = \tilde{\mathbf{A}}^{-1} = \tilde{\mathbf{A}}^T \tag{37}$$

- Orthogonality after whitening simplifies the computation of ICA.
- Instead of the original n^2 entries in $\mathbf{W}$, only the $n(n-1)/2$ different elements in the orthogonal separating matrix $\tilde{\mathbf{W}}$ needs to be estimated.
- A problem with PCA whitening is that it may amplify noise.
- For small eigenvalues λ_i, the corresponding element $\lambda_i^{-1/2}$ in the matrix $\mathbf{\Lambda}^{-1/2}$ becomes large.
- This problem can be handled by compressing the dimensionality of the data in context with PCA whitening.
- The principal components $\mathbf{e}_i^T\mathbf{x}$ corresponding to small eigenvalues λ_i are discarded.

Algorithms for computing ICA

- For computing the independent components $\mathbf{s} = \mathbf{W}\mathbf{x}$, many algorithms and approaches have been proposed.
- They usually estimate the separating matrix $\mathbf{W} = \mathbf{A}^{-1}$ for the mixing model $\mathbf{x} = \mathbf{A}\mathbf{s}$.
- In the following, we discuss two most popular algorithms:
  - The natural gradient algorithm.
  - The fixed point algorithm(s).

The natural gradient algorithm

- The natural gradient algorithm is a simple neural (adaptive) ICA algorithm.
- It can be derived from several starting points:

  - Maximum likelihood principle;
  - The Infomax principle;
  - Minimization of the mutual information.
- We have skipped the maximum likelihood and Infomax criteria in this lecture.
- But they are discussed a little in the tutorial paper by Hyvärinen and Oja.
- We skip the somewhat involved derivations here, too.
- Denote by $\mathbf{w}_i^T$ the i:th row vector of the $n \times n$ separating matrix $\mathbf{W}$.
- Then $y_i = \mathbf{w}_i^T \mathbf{x}$ is an estimate of an independent component s_j.
- Denote by $p_i(s_i)$, $i = 1, \ldots, n$, the probability densities of the n independent components (source signals) $\mathbf{s} = [s_1, s_2, \ldots, s_n]^T$.

- The **natural gradient algorithm** is

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \eta(k)[\mathbf{I} - \mathbf{g}(\mathbf{y}(k))\mathbf{y}^T(k)]\mathbf{W}(k) \tag{38}$$

- There $\mathbf{y}(k) = \mathbf{W}(k)\mathbf{x}(k)$, and $\eta(k)$ is a small learning parameter on iteration k.
- The i:th component $g_i(y_i)$ of the vector $\mathbf{g}(\mathbf{y}) = \mathbf{g}(\mathbf{W}\mathbf{x})$ is

$$g_i(y_i) = -\frac{d \log p_i(y_i)}{dy_i} = -\frac{dp_i(y_i)/dy_i}{p_i(y_i)} \tag{39}$$

- Hence the nonlinearities g_i depend on the densities p_i of the true independent components (sources) s_i.
- These densities are usually unknown.
- Fortunately it suffices in practice to know whether the independent component is sub-Gaussian or super-Gaussian.
- This can also be estimated directly from the data.

- For super-Gaussian independent components, the componentwise nonlinearity is often chosen to be

$$g(y) = 2\tanh(y) \tag{40}$$

- For sub-Gaussian independent components, one can use

$$g(y) = y - \tanh(y) \tag{41}$$

- Prewhitening of the data vectors $\mathbf{x}$ is theoretically not necessary in the natural gradient algorithm.
- However, it is highly recommendable in practice.
- Note that at convergence, the algorithm satisfies a nonlinear decorrelation condition $\mathrm{E}\{\mathbf{g}(\mathbf{y})\mathbf{y}^T\} = \mathbf{I}$.
- If $\mathbf{g}(\mathbf{y}) = \mathbf{y}$, the natural gradient algorithm reduces to a neural prewhitening algorithm.

- **Pros and cons of the natural gradient algorithm:**
  - \+ Theoretically well justified.
  - \+ Simple neural adaptive algorithm.
  - – Converges still slowly because of stochastic gradient used.
  - – Requires different nonlinearities for sub-Gaussian and super-Gaussian independent components.

Fixed-point algorithms

- Aapo Hyvärinen (and Erkki Oja) developed fast ICA algorithms called FastICA in 1996.
- In the former Lab. of Computer and Information Science at HUT belonging now to our Aalto University Dept. of Computer Science.
- They are fixed-point iterations maximizing non-Gaussianity.
- Non-Gaussianity is measured by an approximation of negentropy.

- The resulting basic fixed-point iteration for estimating one independent component $\mathbf{w}^T\mathbf{z}$ is

$$\mathbf{w} \leftarrow \mathrm{E}\{\mathbf{z}g(\mathbf{w}^T\mathbf{z})\} - \mathrm{E}\{g'(\mathbf{w}^T\mathbf{z})\}\mathbf{w} \tag{42}$$

- Here g' is the derivate of the nonlinear function g.
- $\mathbf{z}$ is whitened data vector $\mathbf{x}$.
- The resulting **FastICA algorithm** is summarized in Table 1.
- The function g can be chosen from

$$g_1(y) = \tanh(a_1 y) \tag{43}$$

$$g_2(y) = y\exp(-y^2/2) \tag{44}$$

$$g_3(y) = y^3 \tag{45}$$

1. Center the data $\mathbf{x}$ to make its mean zero.
2. Whiten the centered data $\mathbf{x}$ to give $\mathbf{z}$.
3. Choose an initial (e.g., random) vector $\mathbf{w}$ of unit norm.
4. Let $\mathbf{w}^* \leftarrow \mathrm{E}\{\mathbf{z}g(\mathbf{w}^T\mathbf{z})\} - \mathrm{E}\{g'(\mathbf{w}^T\mathbf{z})\}\mathbf{w}$.
5. Normalize $\mathbf{w}^*$: $\mathbf{w} \leftarrow \mathbf{w}^*/\|\mathbf{w}^*\|$.
6. If not converged, go back to step 4.

Table 1: **The FastICA algorithm for estimating one independent component.** The expectations are estimated in practice as an average over the available data sample.

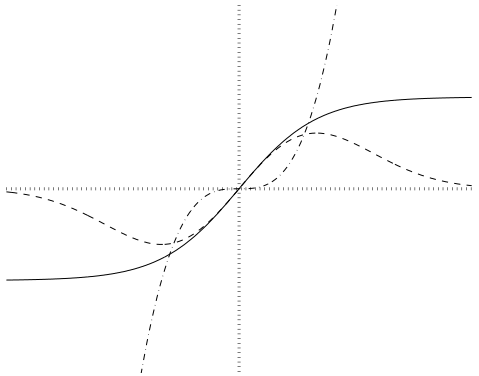

Figure 11: The robust nonlinearities g_1 and g_2, given by the solid and the dashed line, respectively. The third power used in kurtosis-based methods is given by the dash-dotted line.

- These functions are depicted in Figure 11.
- Using the nonlinearity $g_3(y) = y^3$ corresponds to maximizing the absolute value of kurtosis.
- This leads to the basic fixed-point update rule

$$\mathbf{w}^* \leftarrow \mathrm{E}\{\mathbf{z}(\mathbf{w}^T\mathbf{z})^3\} - 3\mathbf{w}$$

- A particularly simple, but not so robust version.
- Kurtosis may depend on only a few observations in the tails of the distribution or on outliers.
- The above fixed-point algorithm converges to some row vector $\mathbf{w}^T$ of the separating matrix $\mathbf{W}$.
- Thus it provides an estimate to only one independent component $\mathbf{w}^T\mathbf{z}$.

- For estimating more or all independent components, different weight vectors $\mathbf{w}_1, \ldots, \mathbf{w}_n$ must orthonormalized.
- The weight vectors and independent components can be estimated either:
  - Sequentially one-by-one.
  - Or symmetrically in parallel.

Sequential estimation

- Assume that $p-1$ weight vectors $\mathbf{w}_1, \ldots, \mathbf{w}_{p-1}$ have already been estimated.
- The p:th weight vector $\mathbf{w}_p$ is orthogonalized against them using the well-known Gram-Schmidt formula:

$$\mathbf{w}_p \leftarrow \mathbf{w}_p - \sum_{j=1}^{p-1} (\mathbf{w}_p^T\mathbf{w}_j)\mathbf{w}_j$$

- This vector is then used in the one-unit fixed-point rule.

Symmetric estimation

- In symmetric estimation, a fixed-point iteration is first carried out for all the weight vectors $\mathbf{w}_1, \ldots, \mathbf{w}_n$.
- Then they are orthonormalized symmetrically:

$$\mathbf{W} \leftarrow (\mathbf{W}\mathbf{W}^T)^{-1/2}\mathbf{W}$$

- Let the eigendecomposition of the symmetric matrix $\mathbf{W}\mathbf{W}^T$ be $\mathbf{E}\mathbf{D}\mathbf{E}^T$.
- The columns of the matrix $\mathbf{E}$ contain the eigenvectors.
- The elements of the diagonal matrix $\mathbf{D}$ contain the respective eigenvalues.
- Then $(\mathbf{W}\mathbf{W}^T)^{-1/2} = \mathbf{E}\mathbf{D}^{-1/2}\mathbf{E}^T$.

- A free Matlab package for FastICA algorithms is available at `http://research.ics.aalto.fi/ica/fastica/`.
- There are also implementations in Python, R, and C++ software.
- **Pros and cons of fixed-point algorithms:**
  - \+ Fast (even cubic) convergence.
  - \+ Computationally much more efficient than the natural gradient algorithm.
  - \+ Can be used for quite large problems.
  - \+ The same nonlinearity can be applied both to sub-Gaussian and super-Gaussian independent components.
  - − Non-neural and non-adaptive batch algorithms.
  - − Require prewhitening of the data.

Practical applications of ICA

- The two most popular applications areas of ICA are currently:
- **Speech and audio applications,** especially:
  - The "cocktail party problem": separation of voices or music or sounds.
  - This problem is in practice much more difficult than the standard ICA problem.
  - It is complicated by reverberations, time delays etc.
- **Biomedical applications** are studied a lot, too.
  - ICA and BSS can be applied to various biomedical signals obtained using EEG, ECG, MEG, and fMRI techniques.
  - The task is to separate interesting signals or remove disturbing artefacts.

- There are many other applications of ICA.
- In the tutorial article A. Hyvärinen and E. Oja, "Independent Component Analysis: Algorithms and Applications," three application of ICA are discussed:
  1. Separation of artifacts in magnetoencephalography (MEG) data.
  2. Finding hidden factors in financial data.
  3. Reducing noise in natural images.
- In the following, we discuss a fourth application of ICA.

Feature extraction from natural images

- A fundamental problem in signal and image processing is to find suitable representations for image, audio etc. data.
- They are useful in tasks like data compression and noise suppression.
- Data representations are often based on discrete linear

transformations.

- Standard linear transformations widely used in image processing are the Fourier, Haar, Walsh-Hadamard, and discrete cosine transforms.
- They can be computed using fast FFT (Fast Fourier Transform) type algorithms.
- But the basis images or vectors of these transforms are **fixed** and the same for all types of images.
- It would be very useful to estimate the linear transformation from the data itself.
- Allowing the transform to adapt ideally to the kind of data that is being processed.
- ICA was applied in our laboratory to this task for a set of images showing natural scenes.

- Each image was first normalized so that the pixels (picture elements) had zero mean and unit variance.
- A set of 10,000 image pacthes (windows) of the size 16×16 pixels were taken at random locations from these images.
- Furthermore, local mean was subtracted from each image window.
- Then the image windows were scanned to 256-dimensional data vectors row-by-row.
- For removing noise, the dimension of the data vectors was reduced to 160.
- The data set preprocessed in this way was used as an input to the FastICA algorithm, using the tanh nonlinearity.
- Figure 12 shows the 160 ICA basis images obtained.
- These basis images can be considered as the independent features of

images.

- Every image window is a linear sum of these windows.
- The basis images in Figure 12 are clearly localized in space, as well as in frequency and orientation.
- They are sensitive to edges and lines in various orientations.
- There are also a few basis images that correspond to global features.
- These ICA basis images resemble closely Gabor features and wavelets that are used extensively in digital image processing.
- They are much more meaningful than the global features provided by PCA.
- The PCA features corresponding to smaller eigenvalues resemble a checkerboard, and are not regarded useful.

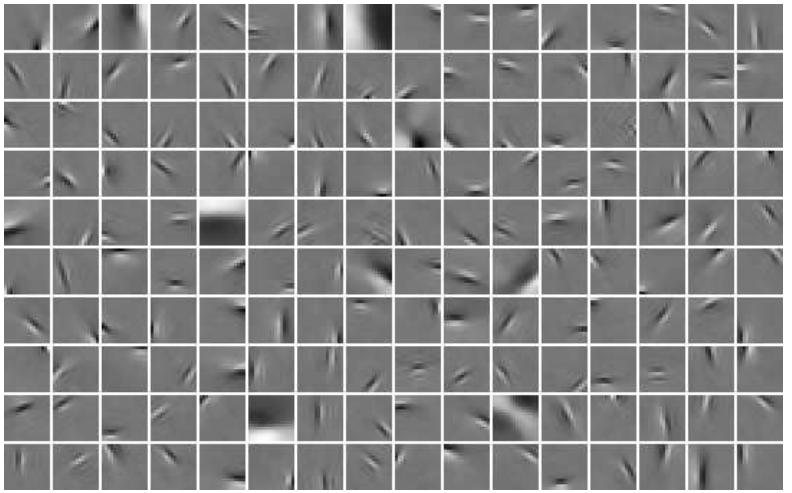

Figure 12: Basis functions in ICA of natural images.

Linear blind source separation (BSS)

- In linear **blind source separation (BSS),** one tries to separate the original source signals from their linear mixtures.
- Assuming that the sources are independent and the mixing model is linear, $\mathbf{x} = \mathbf{A}\mathbf{s}$, one can apply linear ICA methods directly to BSS.
- Another major group of linear BSS methods utilizes **time structure** of the sources.
- **Second-order temporal statistics** are then sufficient for achieving blind separation.
- The sources can be even Gaussian provided that they have different autocorrelation sequences.
- ICA neglects possible temporal structure of the sources or independent components, treating them as random variables.

- On the other hand, it works for temporally uncorrelated sources.
- Ideally, both spatial independence and temporal structure should be taken into account in estimation.
- Still other blind source separation methods are based on **time-frequency representations** or **nonstationarity of signals**.
- Each major group of BSS methods has its own strengths and weaknesses.
- They are best applicable in somewhat different situations.

Extensions and modifications of basic linear ICA

- Noisy ICA; requires more sophisticated methods.
- Overcomplete bases: the number of independent components is larger than the number of mixtures.

- Taking into account the temporal structure in the data.
- ICA and BSS for nonlinear mixture models.
- Separation of convolutive mixtures containing time delays.
- Separation of correlated or non-independent sources.
- Nonstationary sources, time dependent mixing matrices.
- Semi-blind problems: some prior information on the source signals and/or mixtures is available.
- Sparse signal representations and sparse component analysis.
- Nonnegative matrix factorizations (NMF), restricting the source signals having nonnegative values only (for example pixels in digital images).

References

- See the homepage `http://research.ics.aalto.fi/ica/` of our former ICA group for further information and useful links.
- A **comprehensive textbook** : *A. Hyvärinen, J. Karhunen, and E. Oja,* Independent Component Analysis, Wiley 2001.
- A newer review paper: *S. Choi et al.*, "Blind Source Separation and Independent Component Analysis: A Review", Neural Information Processing - Letters and Reviews, Vol. 6, No. 1, January 2005.
- An extensive book on more recent develoments, extensions, and applications: *P. Comon and C. Jutten,* Handbook of Blind Source Separation - Independent Component Analysis and Applications, Academic Press 2010.