FEATURE EXTRACTION AND SELECTION METHODS

1

Feature extraction and selection methods

- The task of the feature extraction and selection methods is to obtain the most ***relevant information*** from the original data and represent that information in a ***lower dimensionality*** space.

2

Selection methods

- When the cost of the acquisition and manipulation of all the measurements is high we must make a selection of features.
- The goal is to select, among all the available features, those that will perform better.
- Example: which features should be used for classifying a student as a good or bad one:
  - Available features: marks, height, sex, weight, IQ.
  - Feature selection would choose marks and IQ and would discard height, weight and sex.
- We have to choose P variables in a set of M variables so that the separability is maximal.

3

Extraction methods

- The goal is to build, using the available features, those that will perform better.
- Example: which features should be used for classifying a student as a good or bad one:
  - Available features: marks, height, sex, weight, IQ.
  - Feature extraction may choose marks + IQ^2 as the best feature (in fact, it is a combination of two features).
- The goal is to transform the origin space X in a new space Y to obtain new features that work better. This way, we can compress the information.

4

Principal Component Analysis

- PCA = Karhunen-Loeve transform = Hotelling transform
- PCA is the most popular feature extraction method
- PCA is a linear transformation
- PCA is used in face recognition systems based on appearance

5

Principal Component Analysis

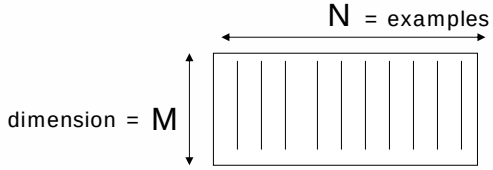
- PCA has been successfully applied to human face recognition.
- PCA consists on a transformation from a space of high dimension to another with more reduced dimension.
- If the data are highly correlated, there is redundant information.
  - PCA decreases the amount of redundant information by decorrelating the input vectors.
  - The input vectors, with high dimension and correlated, can be represented in a lower dimension space and decorrelated.
  - PCA is a powerful tool to compress data.

6

PCA by Maximizing Variance (I)

We will derive PCA by maximizing the variance in the direction of principal vectors.

Let us suppose that we have N M-dimensional vectors $\mathbf{x}_j$ aligned in the data matrix **X**.



Let **u** be a direction (a vector of lenght 1). The projection of the j-th vector $\mathbf{x}_j$ onto the vector **u** can be calculated in the following way:

$$p_j = \vec{u}^T \cdot \vec{x}_j = \sum_{i=1}^{M} u_i x_{ij}$$

7

PCA by Maximizing Variance (II)

We want to find a direction **u** that maximizes the variance of the projections of all input vectors $\mathbf{x}_j$, j=1,..N.

The function to maximize is:

$$J^{PCA}(\vec{u}) = \sigma^2(p_j) = \frac{1}{N}\sum_{j=1}^{N}(p_j - \bar{p})^2 = \ldots = \vec{u}^T C \vec{u}$$

where **C** is the covariance matrix of the data matrix **X.**

$$C = \frac{1}{N}\hat{X}\cdot\hat{X}^T \qquad \hat{X} = X - \mu \cdot 1_{1xN}$$

$$\mu = [\mu_1, \ldots, \mu_m]^T$$

Using the technique of Lagrange multipliers, the solution to this maximization problem is to compute the **eigenvectors** and the **eigenvalues** of the covariance matrix **C.**

MORE INFO in PCA.pdf

8

PCA by Maximizing Variance (III)

The largest eigenvalue equals the maximal variance, while the corresponding eigenvector determines the direction with the maximal variance.

By performing singular value decomposition (SVD) of the covariance matrix **C** we can diagonalize **C:**

$$C = U \Lambda U^T$$

in such a way that the orthonormal matrix **U** contains the eigenvectors $\mathbf{u_1}, \mathbf{u_2}, .. \mathbf{u_N}$ in its columns and the diagonal matrix Λ contains the eigenvalues $\lambda_1, \lambda_2, .. \lambda_N$ on its diagonal.

The eigenvalues and the eigenvectors are arranged with respect to the descending order of the eigenvalues, thus $\lambda_1 \geq \lambda_2 \geq .. \geq \lambda_N$. Therefore, the most of the variability of the input random vectors is contained in the first eigenvectors. Hence, the eigenvectors are called principal vectors.

9

Computing PCA

Steps to compute the PCA transformation of a data matrix X:

- **Center the data** $\overline{X}$
- **Compute the covariance matrix** C
- **Obtain the eigenvectors and eigenvalues of the covariance matrix** U , Λ
- **Project the original data in the eigenspace**

$$P = U^T \cdot \overline{X}$$

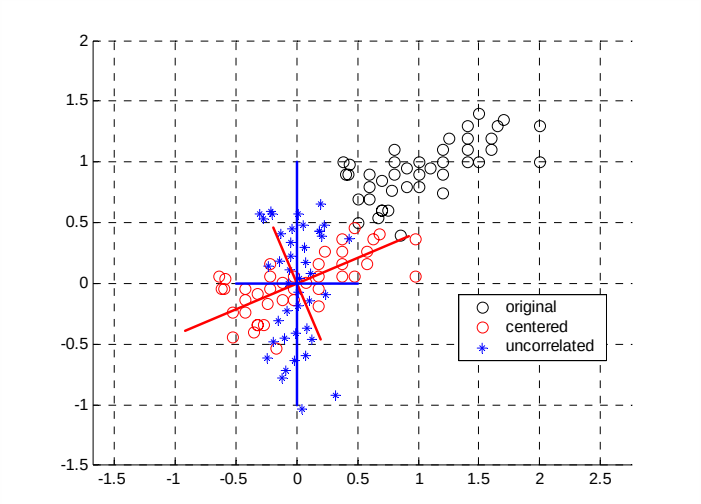
U can be used as a linear transformation to project the original data of high dimension into a space of lower dimension.

Matlab code:

```
%number of examples
N=size(X,2);
%dimension of each example
M=size(X,1);
%mean
meanX=mean(X,2);
%centering the data
Xm=X-meanX*ones(1,N);
%covariance matrix
C=(Xm*Xm')/N;
%computing the eigenspace:
[U D]=eig(C);
%projecting the centered data
   over the eigenspace
P=U'*Xm;
```

10

PCA of a bidimensional dataset



11

Computing PCA of a set of images

This approach to the calculation of principal vectors is very clear and widely used. However, if the size of the data vector M is very large, which is often the case in the field of computer vision, the covariance matrix **C** becomes very large and eigenvalue decomposition of **C** becomes unfeasible.

But, if the number of input vectors is smaller than the size of these vectors (N<M), PCA can be sped up using a method proposed by Murakami and Kumar which is also known as Turk and Pentland´s trick.

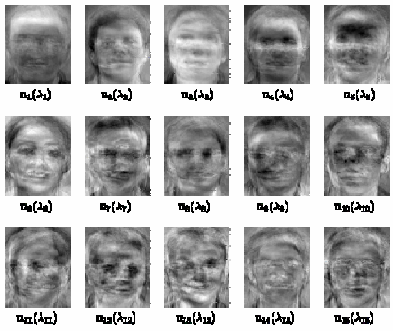
When we work on Matlab, we use **pc_evectors.m** to apply the Turk and Pentland´s trick and compute the PCA transformation of a set of images

MORE INFO in PCA.pdf

12

Face recognition using PCA (I)

Eigenfaces for Recognition, *Turk, M. & Pentland, A. , Journal of Cognitive Neuroscience, 3, 71-86, 1991.*



13

Linear Discriminant Analysis (I)

- LDA = Fisher analysis
- LDA is a linear transformation
- LDA is also used in face recognition
- LDA seeks directions that are efficient for discrimination between classes

- In PCA, the subspace defined by the vectors is the one that better describes the conjunct of data.
- LDA tries to discriminate between the different classes of data.

14

Linear Discriminant Analysis (I)

- We have a conjunct of N vectors of dimension M in the data matrix MxN.
- We have C classes and k vectors per class.
- We want to find the transformation matrix W that better describes the subspace that discriminates between classes, after projecting the data in the new space.
- The objective is to make maximum the distance between classes S_b and minimizing S_w.

$$P = W \cdot X \qquad W \text{ are the eigenvectors of } \mathrm{S_w}^{-1}\mathrm{S_b}$$

"between class" scatter matrix

$$\mathrm{S_b} = \sum_{j=1}^{C} (\mu_j - \mu)(\mu_j - \mu)^T$$

"within-class" scatter matrix

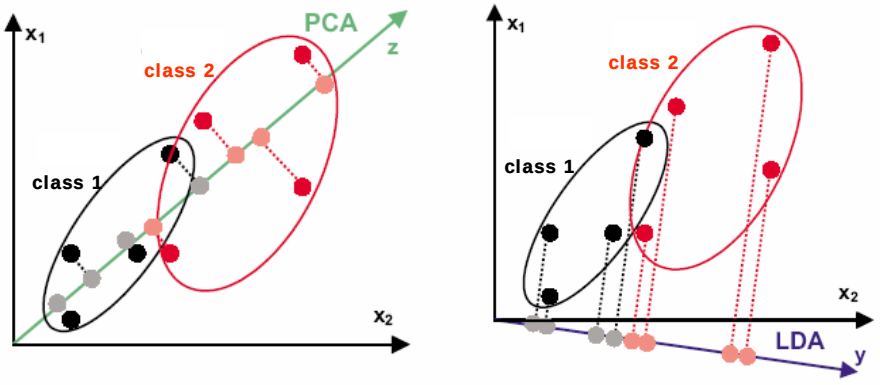
$$\mathrm{S_w} = \sum_{j=1}^{C} \sum_{i=1}^{K} (\mathrm{x}_i^j - \mu_j)(\mathrm{x}_i^j - \mu_j)^T = \sum_{j=1}^{C} \hat{\mathrm{X}}^j \hat{\mathrm{X}}^{jT}$$

$$\mathrm{X} = [\mathrm{X}^1 | \mathrm{X}^2 | \ldots | \mathrm{X}^j | \ldots | \mathrm{X}^C]$$

$$\mathrm{X}^1 = [\mathrm{x}_1^1, \mathrm{x}_2^1, \ldots, \mathrm{x}_i^1, \ldots, \mathrm{x}_K^1]$$
$$\mathrm{X}^j = [\mathrm{x}_1^j, \mathrm{x}_2^j, \ldots, \mathrm{x}_i^j, \ldots, \mathrm{x}_K^j]$$
$$\mathrm{X}^C = [\mathrm{x}_1^C, \mathrm{x}_2^C, \ldots, \mathrm{x}_i^C, \ldots, \mathrm{x}_K^C]$$

Linear Discriminant Analysis (II)

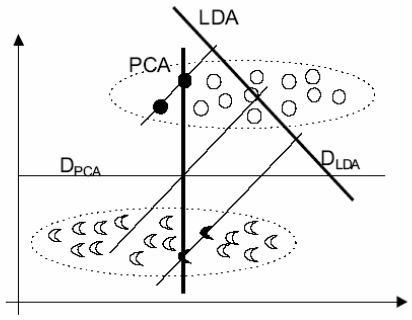
- The figure shows the effect of LDA transform in a conjunct of data composed of 2 classes.



16

Linear Discriminant Analysis (III)

Limitations of LDA



- LDA works better than PCA when the training data are well representative of the data in the system.
- If the data are not representative enough, PCA performs better.

17

Independent Component Analysis (I)

•**I**ndependent **C**omponent **A**nalysis

•ICA is a statistical technique that represents a multidimensional random vector as a linear combination of nongaussian random variables ('independent components') that are as independent as possible.

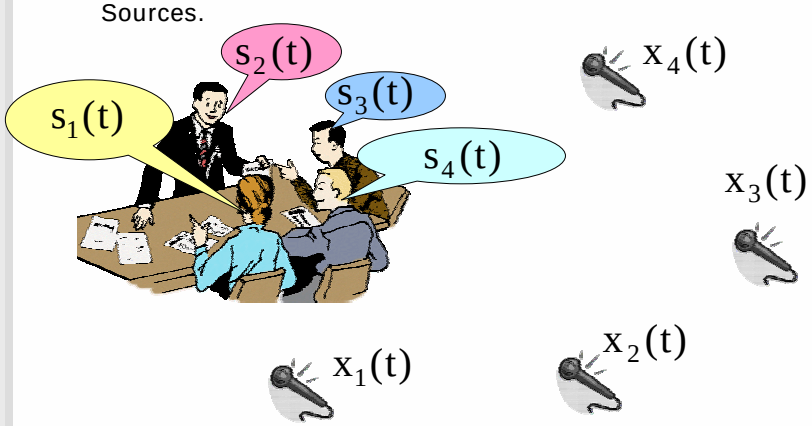
•ICA is somewhat similar to PCA.

•ICA has many applications in data analysis, source separation, and feature extraction.

18

ICA – cocktail party problem

- Cocktail party problem
  - ICA is a statistical technique for decomposing a complex dataset into independent sub-parts. Here we show how it can be applied tothe problem of separation of Blind Sources.



19

ICA – cocktail party problem

- Cocktail party problem

Estimate the sources $s_i(t)$
from the mixed signals $x_i(t)$

20

ICA – cocktail party problem

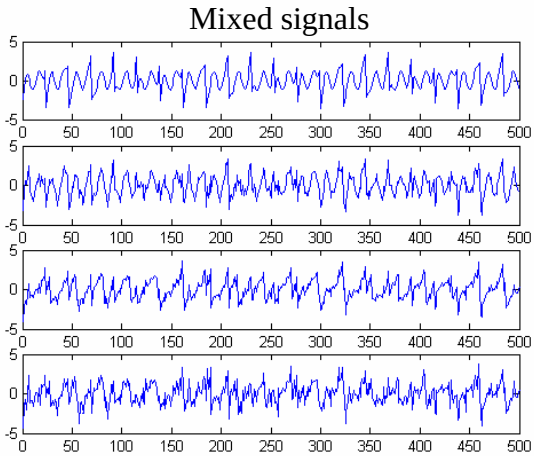
- Linear model:

$$x_1(t) = a_{11}s_1(t) + a_{12}s_2(t) + a_{13}s_3(t) + a_{14}s_4(t)$$
$$x_2(t) = a_{21}s_1(t) + a_{22}s_2(t) + a_{23}s_3(t) + a_{24}s_4(t)$$
$$x_3(t) = a_{31}s_1(t) + a_{32}s_2(t) + a_{33}s_3(t) + a_{34}s_4(t)$$
$$x_4(t) = a_{41}s_1(t) + a_{42}s_2(t) + a_{43}s_3(t) + a_{44}s_4(t)$$

- We can model the problem as X=AS
  - S = 4D vector containing the independent source signals.
  - A = mixing matrix.
  - X = Observed signals.

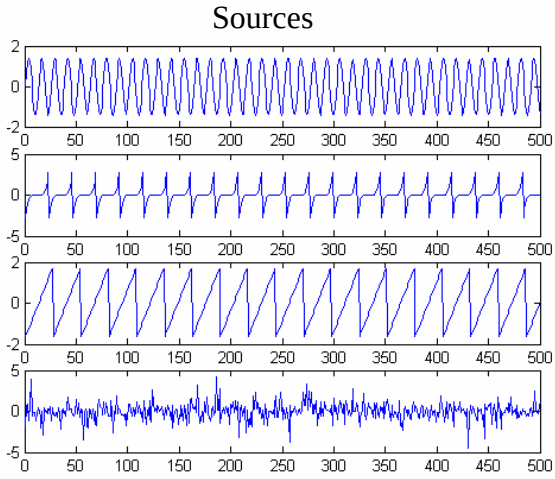
21

ICA – cocktail party problem



22

ICA – cocktail party problem



23

ICA – cocktail party problem

Estimate the sources $s_i(t)$
from the mixed signals $x_i(t)$

ICA: One possible solution is to assume that the sources are **independent.**

$$p(s_1, s_2, \ldots, s_n) = p(s_1)p(s_2)\ldots p(s_n)$$

24

ICA – cocktail party problem

MODEL ICA

$$X = A \cdot S$$

Mixed signals — Mixing matrix — Sources

25

ICA – cocktail party problem

ESTIMATING THE SOURCES

$$S = W \cdot X$$

Sources — Separation matrix — Mixed signals

IC's

$$W = A^{+} \rightarrow X \approx AS$$

26

Computing IC's

- Typically, in ICA algorithms, **W** is sought such that the rows of it have maximally non-gaussian distributions and are mutually uncorrelated.
- A simple way to do this is to first whiten the data and then seek orthogonal non-normal projections.
- We want to find arrows ω_i / $s_i = w_i^T \cdot x$ have maximally non-gaussian distributions and mutually uncorrelated.

27

PCA, WHITENING, ICA (I)

- **PCA:**

  uncorrelated data

  (the covariance matrix of the PCA transformed data has the eigenvalues in its diagonal)
- **WHITENING**:

  PCA + scaling

  (the covariance matrix of the whiten data is the identity)
- **ICA:**

  WHITENING + rotation

28

WHITENING

- **WHITENING**:

PCA + scaling

$$\mathbf{X} \approx \sum_{j=1}^{K} \sigma_j \mathbf{u}_j \mathbf{v}_j^T = \widetilde{\mathbf{U}} \widetilde{\mathbf{\Lambda}} \widetilde{\mathbf{V}}^T$$

$$\widetilde{\mathbf{Z}} = \widetilde{\mathbf{\Lambda}}^{-1} \widetilde{\mathbf{U}}^T \mathbf{X} = \widetilde{\mathbf{V}}^T$$

29

ICA (I)

- **ICA:**

WHITENING + rotation

$$\mathbf{S} = \mathbf{R}^T \widetilde{\mathbf{\Lambda}}^{-1} \widetilde{\mathbf{U}}^T \mathbf{X} = \mathbf{R}^T \widetilde{\mathbf{Z}}$$

R is a rotation that maximizes the non-gaussianity of the projections

30

ICA (II)

- **ICA model:**

$$\mathbf{X} \approx \widetilde{\mathbf{U}}\widetilde{\mathbf{\Lambda}}\widetilde{\mathbf{V}}^T = \underbrace{\widetilde{\mathbf{U}}\widetilde{\mathbf{\Lambda}}\mathbf{R}}_{\mathbf{A}}\underbrace{\mathbf{R}^T\widetilde{\mathbf{V}}^T}_{\mathbf{S}} = \mathbf{AS}$$

31

ICA (III)

FastICA : is a free MATLAB program that implements the fast-fixed point algorithm.

32

PCA, WHITENING, ICA (II)

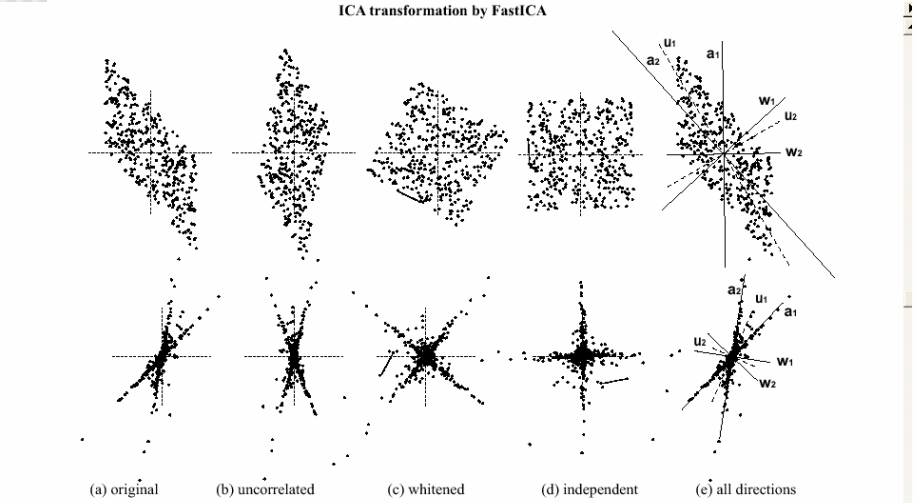
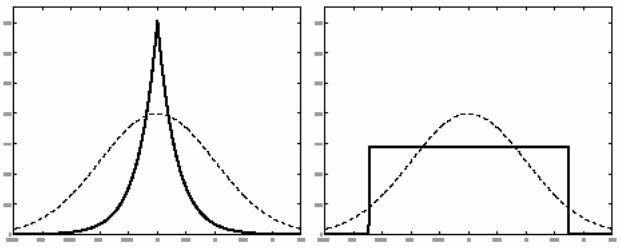


Fig. 2. Two artificial examples: a subgaussian dataset (1st. row) and a supergaussian dataset (2nd. row) (a), both transformed by PCA (b), whitening (c) and using the ICA model by means of FastICA (d). The last figure in each row (e) shows the original dataset with the ICA and PCA directions.

Non gaussianity (I)

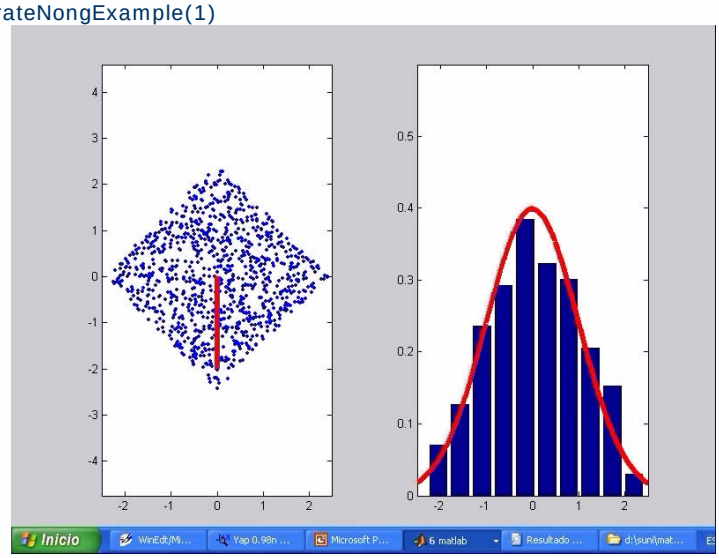


SUPERGAUSSIAN ***SUBGAUSSIAN***

34

No gaussianity (II)

generateNongExample(1)



35

ICA in CNS (Computational Neuroscience) (I)

- BSS aplications with EEG and MEG signals.
  - The brains activity is measured through Electroencephalograms. Those signals are a mixture of different activities in the brain and other external noises.
  - ICA solves correctly the problem of extracting the original activity signals
- Modeling the performance of the neurons in area V1 of mammalian cortex.
  - Spikes
  - Receptive fields
  - Natural images
- Some studies propose that the behaviour of one kind of neurons can be computationally described throught the ICA analysis of this natural inputs.

36

Spikes

- SPIKES: electrical signal in neurons

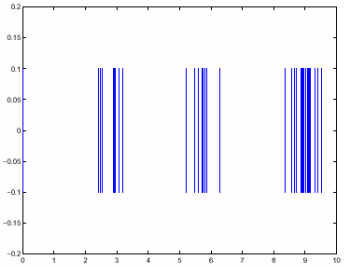


Figure 1: An inhomogeneus Poisson spike train

37

Receptive fields

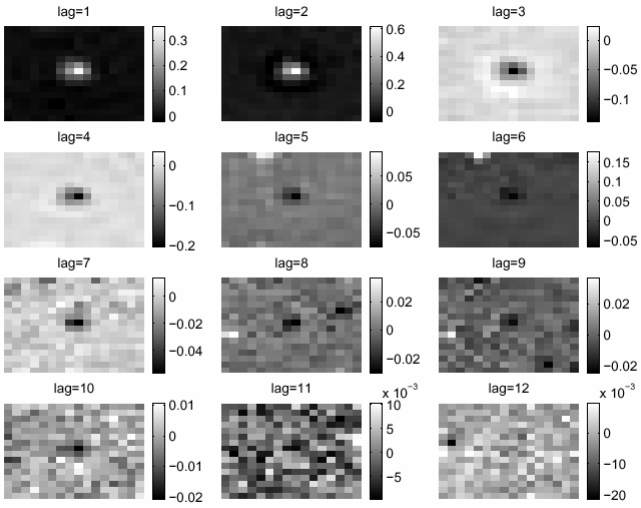
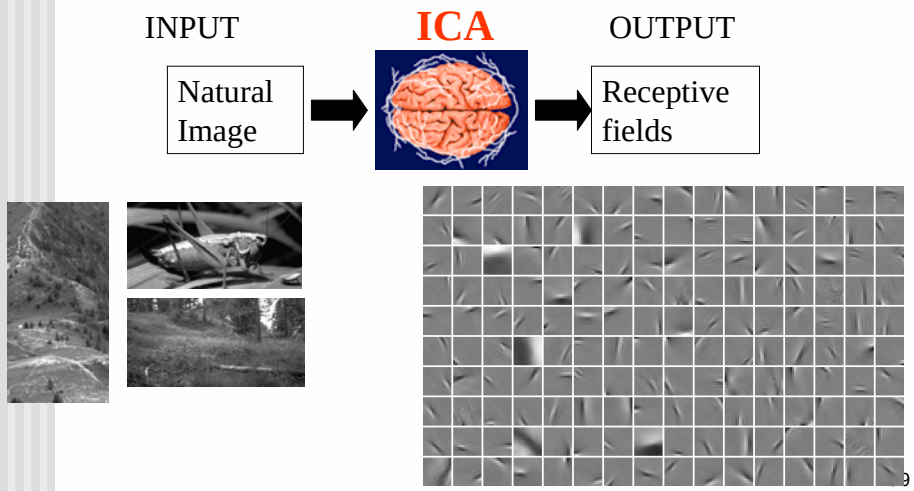


Figure 2: The spike-triggered average images for each of the 12 time steps

38

Simple experiment



NMF (I)

Non-negative matrix factorization (**NMF**) is a recently developed technique for finding parts, and it is based on linear representations of non-negative data.

Given a non-negative data matrix **X**, NMF finds an approximate factorization

$$X \approx W \cdot H$$

into non-negative factors **W** and **H**. The non-negativity constraints make the representation purely additive (allowing no subtractions), in contrast to many other linear representations such as PCA or ICA.

Motivation: In most real systems, the variables are non negative. PCA and ICA offer results complicated to interpret.

W and H are chosen as the matrix that minimize reconstruction error.

40

NMF (II)

NMF as a feature extraction method in faces

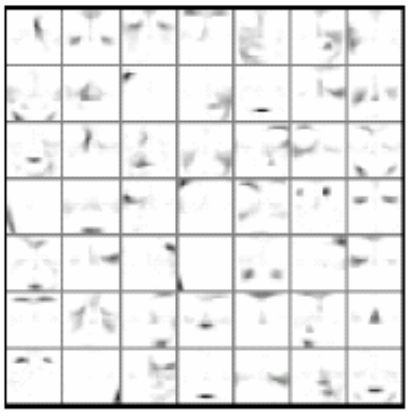
The importance of NMF is that it has capacity of obtaining significant features in collections of real biological data.

When applied to X = Faces, NFM generates base vectors that are intuitive features of the faces (eyes, mouth, nose…)

41

NMF (III)

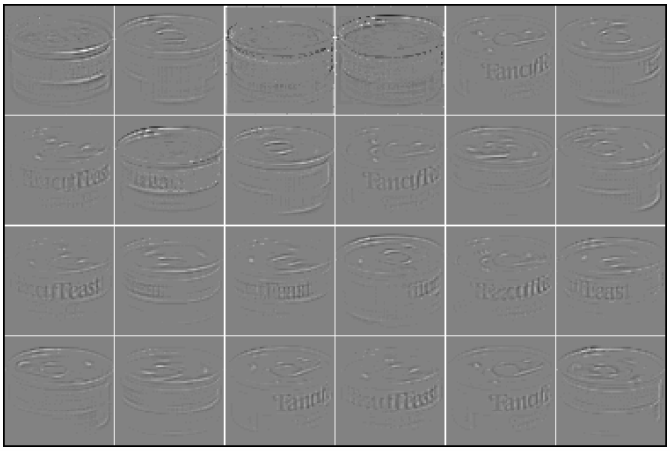
- **NMF → local features**



42

NMF (IV)

- **NMF → local features**



43

NMF (V)

- NMF presents features that make it adequate for applications in object recognition.
- It allows extracting local features as shown on the previous figure. Some images extract the text, others the top side, others the general shape of the object…
- It can be useful in presence of occusions.
  - In this case, it is not possible to extract global features, but we can extract local ones.
- It can be useful also to identify objects in non-structured environments.
- At last, we can use it to extract categories of objects.

44