Denoising source separation: a novel approach to ICA and feature extraction using denoising and Hebbian learning

Jaakko Särelä[1] and Harri Valpola[2]

[1] Neural Network Research Centre
[2] Laboratory of Computational Engineering
Helsinki University of Technology, FINLAND

8.5.2005

DSS in source separation

Whitening
- Neuroscience

Denoising
- Procedural formulation of prior information

DSS in real-world problems
- Fast algorithms
- DSS in Climate research

DSS in feature extraction

Selection of information

- In many real-world problems there are plenty of data with a lot of structure.

Selection of information

- In many real-world problems there are plenty of data with a lot of structure.
- Usually only part of the structure is interesting.

Selection of information

- In many real-world problems there are plenty of data with a lot of structure.
- Usually only part of the structure is interesting.
- Which part is interesting depends on the goals.

Selection of information

- In many real-world problems there are plenty of data with a lot of structure.
- Usually only part of the structure is interesting.
- Which part is interesting depends on the goals.
- Source separation and feature extraction are similar selection processes when data dimensionality is high.

Local learning

- Fast learning rules are often local: weight modification needs local information only.

Local learning

- Fast learning rules are often local: weight modification needs local information only.
- Hebbian and anti-Hebbian learning are prime examples of local learning rules: weight change is proportional to pre- and post-synaptic activation.

Part I

DSS in source separation

- Situation: interesting and uninteresting components are observed in mixtures (linear or nonlinear).

Part I

DSS in source separation

- Situation: interesting and uninteresting components are observed in mixtures (linear or nonlinear).
- Task: separate the interesting sources.

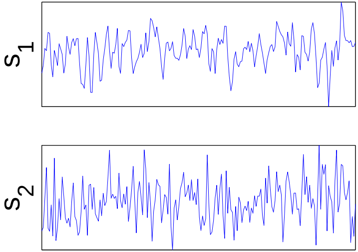
Example

Let's consider a simple source separation task: a source should be recovered from two linear mixtures of two sources.

Example

Let's consider a simple source separation task: a source should be recovered from two linear mixtures of two sources.

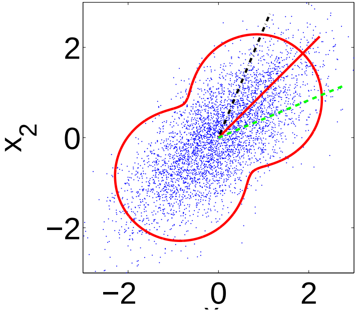
- Source 1 (target) changes slower in time than the other interfering source.



Example

Let's consider a simple source separation task: a source should be recovered from two linear mixtures of two sources.

- Source 1 (target) changes slower in time than the other interfering source.
- Both sources are observed on two channels, but source 1 is relatively stronger on channel 2 and vice versa.



Problem: how to avoid interference

- The contribution of one source to the channels is called the mixing vector $\mathbf{a}$. The mixing vectors of different sources form the mixing matrix $\mathbf{A}$

$$\mathbf{x} = \sum_i \mathbf{a}_i s_i = \mathbf{A}\mathbf{s}$$

Problem: how to avoid interference

- The contribution of one source to the channels is called the mixing vector **a**. The mixing vectors of different sources form the mixing matrix **A**

$$\mathbf{x} = \sum_i \mathbf{a}_i s_i = \mathbf{A}\mathbf{s}$$

- Knowing the mixing vector $\mathbf{a}_i$ is not enough for recovering the source s_i.

Problem: how to avoid interference

- The contribution of one source to the channels is called the mixing vector $\mathbf{a}$. The mixing vectors of different sources form the mixing matrix $\mathbf{A}$

$$\mathbf{x} = \sum_i \mathbf{a}_i s_i = \mathbf{A}\mathbf{s}$$

- Knowing the mixing vector $\mathbf{a}_i$ is not enough for recovering the source s_i.
- Inverse $\mathbf{A}^{-1}$ (so-called unmixing vectors) is required and its computation requires all $\mathbf{a}_i$.

Problem: how to avoid interference

- The contribution of one source to the channels is called the mixing vector $\mathbf{a}$. The mixing vectors of different sources form the mixing matrix $\mathbf{A}$

$$\mathbf{x} = \sum_i \mathbf{a}_i s_i = \mathbf{A}\mathbf{s}$$

- Knowing the mixing vector $\mathbf{a}_i$ is not enough for recovering the source s_i.
- Inverse $\mathbf{A}^{-1}$ (so-called unmixing vectors) is required and its computation requires all $\mathbf{a}_i$.
- Hebbian learning can in general only recover $\mathbf{a}_i$, not $\mathbf{A}^{-1}$.

Comparing apples and oranges

- Often it is difficult to see some target pattern in the data because it is masked by interference from some other, stronger patterns.

Comparing apples and oranges

- Often it is difficult to see some target pattern in the data because it is masked by interference from some other, stronger patterns.
- In such cases it is particularly difficult to use simple Hebbian-type algorithms for finding the target patterns.

Comparing apples and oranges

- Often it is difficult to see some target pattern in the data because it is masked by interference from some other, stronger patterns.
- In such cases it is particularly difficult to use simple Hebbian-type algorithms for finding the target patterns.
- Related problem: how to decide if something is large from visual image if the distance to different objects can vary?

Comparing apples and oranges

- Often it is difficult to see some target pattern in the data because it is masked by interference from some other, stronger patterns.
- In such cases it is particularly difficult to use simple Hebbian-type algorithms for finding the target patterns.
- Related problem: how to decide if something is large from visual image if the distance to different objects can vary?
- Solution: normalise (distance in the size example, variance in Hebbian learning).

Whitening: removing correlation structure

- Whitening (a.k.a. sphering) normalises the variance structure (data will be decorrelated and variance is isotropic = the same in every direction).

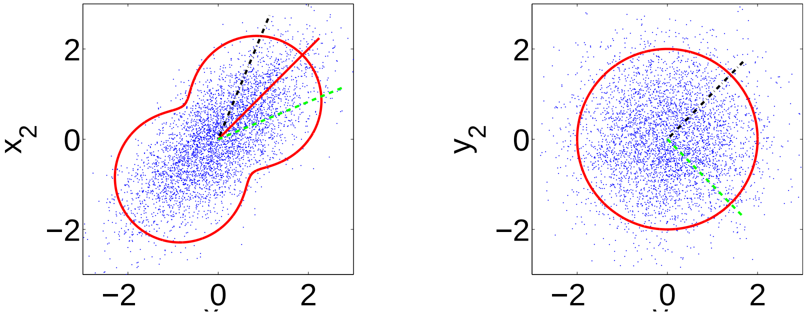
Whitening: removing correlation structure

- Whitening (a.k.a. sphering) normalises the variance structure (data will be decorrelated and variance is isotropic = the same in every direction).
- Can be implemented by PCA + normalisation of variances.

Whitening: removing correlation structure

- Whitening (a.k.a. sphering) normalises the variance structure (data will be decorrelated and variance is isotropic = the same in every direction).
- Can be implemented by PCA + normalisation of variances.
- The data can also be rotated "back to the original" after normalisation.

Whitening



Result: PCA doesn't see any structure in the data but the mixing vectors become (more) orthogonal.

Whitening in the brain

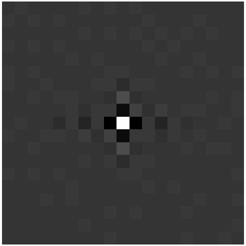
- Interestingly, decorrelation and normalisation are ubiquitous in the brain: many systems have lateral inhibition and "gain control".

Whitening in the brain

- Interestingly, decorrelation and normalisation are ubiquitous in the brain: many systems have lateral inhibition and "gain control".
- For instance, retinal on-center-off-surround cells and thalamic "relay cells".

Whitening in the brain

- Interestingly, decorrelation and normalisation are ubiquitous in the brain: many systems have lateral inhibition and "gain control".
- For instance, retinal on-center-off-surround cells and thalamic "relay cells".
- Symmetric whitening computed from natural images:



Making things different again

After whitening PCA doesn't see any structure, but what if we "disturb" the data a bit.

Making things different again

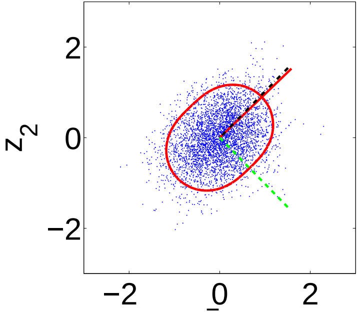
After whitening PCA doesn't see any structure, but what if we "disturb" the data a bit.

- Remember that our target source changed slowly. What if we low-pass filter the data.

Making things different again

After whitening PCA doesn't see any structure, but what if we "disturb" the data a bit.

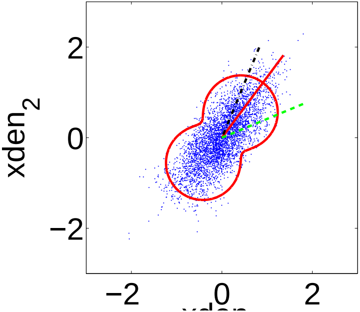
- Remember that our target source changed slowly. What if we low-pass filter the data.



Making things different again

After whitening PCA doesn't see any structure, but what if we "disturb" the data a bit.

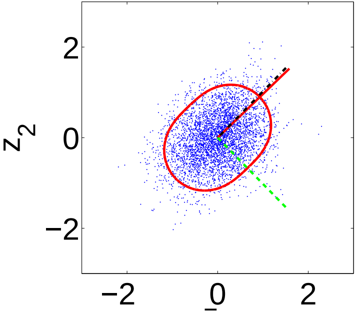
- Remember that our target source changed slowly. What if we low-pass filter the data.
- Whitening was important...



Making things different again

After whitening PCA doesn't see any structure, but what if we "disturb" the data a bit.

- Remember that our target source changed slowly. What if we low-pass filter the data.
- Whitening was important...



Theoretical justification

- Denoising can be viewed as prior information in procedural form.

Theoretical justification

- Denoising can be viewed as prior information in procedural form.
- DSS can be justified as an EM-algorithm for source separation: E-step = denoising using prior information, M-step = estimation of a new mixing vector.

Theoretical justification

- Denoising can be viewed as prior information in procedural form.
- DSS can be justified as an EM-algorithm for source separation: E-step = denoising using prior information, M-step = estimation of a new mixing vector.
- Denoising can thus (but does need to) be derived from prior information $p(\mathbf{s})$.

Theoretical justification

- Denoising can be viewed as prior information in procedural form.
- DSS can be justified as an EM-algorithm for source separation: E-step = denoising using prior information, M-step = estimation of a new mixing vector.
- Denoising can thus (but does need to) be derived from prior information $p(\mathbf{s})$.
- Whitening means that mixing vector = unmixing vector; sources can be extracted one by one.

Nonlinear denoising

- In our example the denoising was applied to the data. This is possible only with linear denoising.

Nonlinear denoising

- In our example the denoising was applied to the data. This is possible only with linear denoising.
- EM-connection suggests that the source estimates should be denoised. Like power method with denoising embedded in the iterations or neural PCA with denoising as the “activation function” of the neurons.

Nonlinear denoising

- In our example the denoising was applied to the data. This is possible only with linear denoising.
- EM-connection suggests that the source estimates should be denoised. Like power method with denoising embedded in the iterations or neural PCA with denoising as the “activation function” of the neurons.
- With simple nonlinearities DSS realises independent component analysis (ICA).

Standard methods work

- Regular PCA works for linear denoising.

Standard methods work

- Regular PCA works for linear denoising.
- Power method required with nonlinear denoising.

Standard methods work

- Regular PCA works for linear denoising.
- Power method required with nonlinear denoising.
- Just as PCA, DSS can be applied for very large datasets.

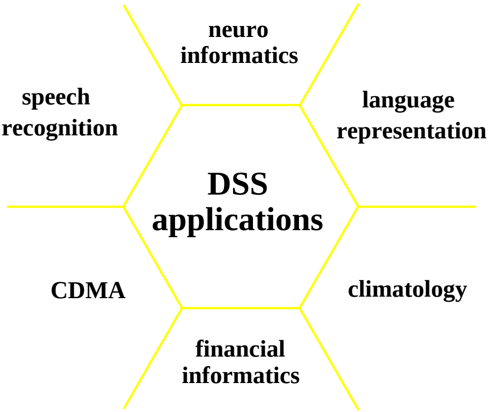
Standard methods work

- Regular PCA works for linear denoising.
- Power method required with nonlinear denoising.
- Just as PCA, DSS can be applied for very large datasets.
- Either deflation (one-by-one extraction) or symmetric separation can be used.

Standard methods work

- Regular PCA works for linear denoising.
- Power method required with nonlinear denoising.
- Just as PCA, DSS can be applied for very large datasets.
- Either deflation (one-by-one extraction) or symmetric separation can be used.
- Note: linear denoising + symmetric separation can only identify the signal subspace.

DSS applications

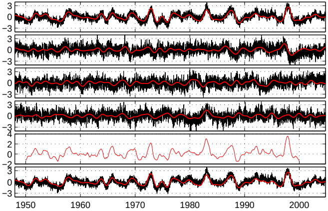


DSS in Climate research

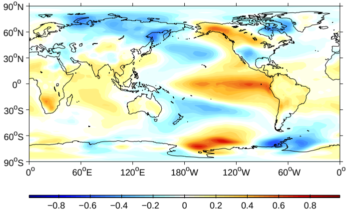
Several global daily measurements during several tens of years: surface temperature, sea level pressure, precipitation, etc.

DSS in Climate research

Several global daily measurements during several tens of years: surface temperature, sea level pressure, precipitation, etc.



Climatological slow components. The first is caused by the El Niño effect.



The spatial map corresponding to the El Niño component.

Part II

DSS in feature extraction

- Situation: A task, such as recognition or motor action should be performed.

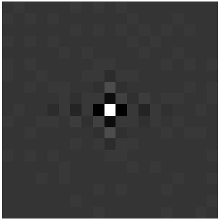
Part II

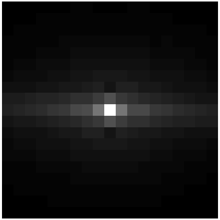
DSS in feature extraction

- Situation: A task, such as recognition or motor action should be performed.
- Task: Find a feature representation for the situation that facilitates the task.

PCA and DSS features from natural images

Symmetric PCA gives on-center/off-surround features.

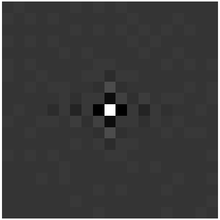




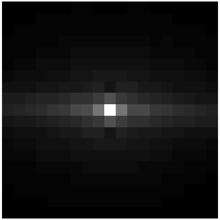
PCA and DSS features from natural images

Symmetric PCA gives on-center/off-surround features.

PCA feature

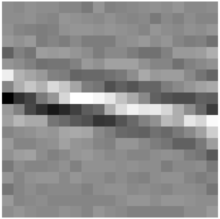


activating pattern

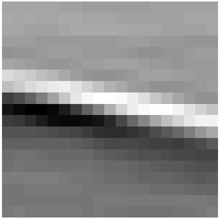


ICA-DSS gives edge-detectors resembling simle cell outputs in V1.

DSS feature



activating pattern



Learning invariant representations

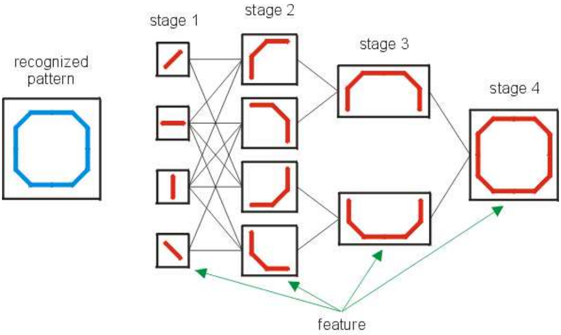
- Invariance = being unsensitive to something:
  - translation
  - rotation
  - scaling
  - ...

Learning invariant representations

- Invariance = being unsensitive to something:
  - translation
  - rotation
  - scaling
  - ...
- It is as important to lose most information as to remain sensitive to the "essential features".

How: hierarchical grouping

- grouping of individual features.
- hierarchy of feature extraction stages.
- the higher the layer, the more complex and invariant the features.



Hierarchies in DSS

- Stacking DSS layers does not bring anything new.

Hierarchies in DSS

- Stacking DSS layers does not bring anything new.
- Solution: make the layer nonlinear somehow.

Nonlinear feature expansion

Linear regression can be made nonlinear by including as inputs nonlinear functions of the inputs.

Nonlinear feature expansion

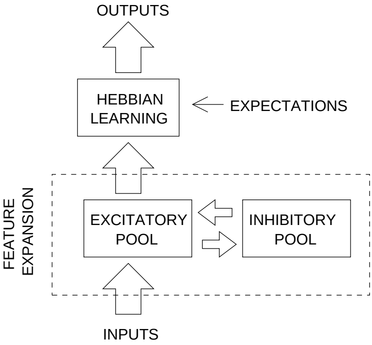
Linear regression can be made nonlinear by including as inputs nonlinear functions of the inputs.

Many alternatives:

- fixed nonlinearities

Nonlinear feature expansion

Linear regression can be made nonlinear by including as inputs nonlinear functions of the inputs.



Many alternatives:

- fixed nonlinearities
- competition and positivity constraint

How to create expectations that drive the learning?

- The big question is: how to recognize that features belong to the same object?

How to create expectations that drive the learning?

- The big question is: how to recognize that features belong to the same object?
- Most common criterion is temporal proximity: features that often appear roughly at the same times probably represent the same object.

How to create expectations that drive the learning?

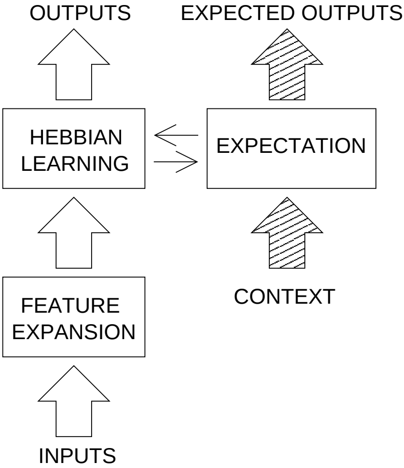
- The big question is: how to recognize that features belong to the same object?
- Most common criterion is temporal proximity: features that often appear roughly at the same times probably represent the same object.
- Contextual proximity is a better criterion: features that appear in the same context probably represent the same object.

How to create expectations that drive the learning?

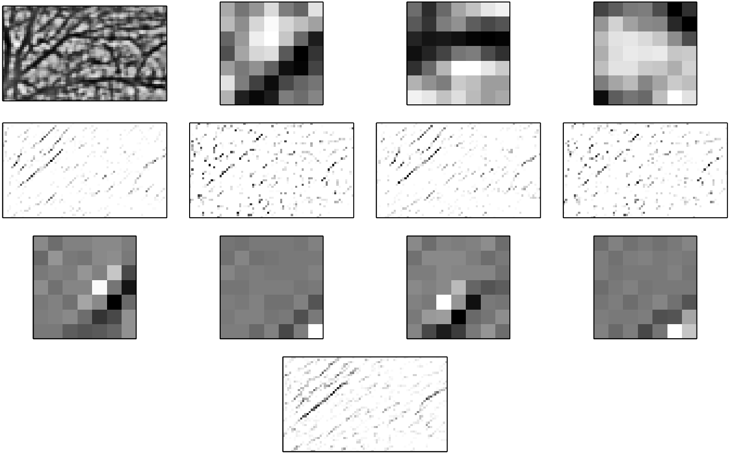
- The big question is: how to recognize that features belong to the same object?
- Most common criterion is temporal proximity: features that often appear roughly at the same times probably represent the same object.
- Contextual proximity is a better criterion: features that appear in the same context probably represent the same object.
- Temporal proximity is often a special case because contexts tend to evolve slowly.

Expectation-driven learning

- inputs drive outputs.
- expectations drive learning (modulate only).
- without the nonlinear feature expansion, equivalent to canonical correlation analysis.



Results



Hierarchical architecture

- It makes sense to stack nonlinear feature extractors into a hierarchy.

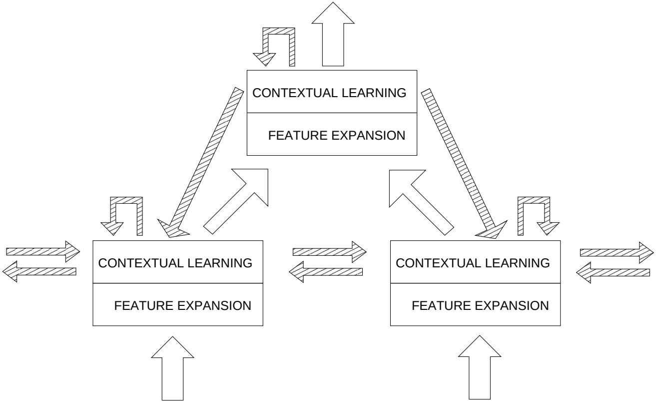
Hierarchical architecture

- It makes sense to stack nonlinear feature extractors into a hierarchy.
- Context derived from "all over the place" can guide learning.

Hierarchical architecture

- It makes sense to stack nonlinear feature extractors into a hierarchy.
- Context derived from “all over the place” can guide learning.
- Learning aims to find a “coherent representation of the world”.

Hierarchical architecture



Neocortical structure

The structure of the model discussed here is very much inspired by the neocortex:

- hierarchy of areas creating an increasingly abstract, invariant representation

Neocortical structure

The structure of the model discussed here is very much inspired by the neocortex:

- hierarchy of areas creating an increasingly abstract, invariant representation
- excitatory-inhibitory interaction creating nonlinear "raw material"

Neocortical structure

The structure of the model discussed here is very much inspired by the neocortex:

- hierarchy of areas creating an increasingly abstract, invariant representation
- excitatory-inhibitory interaction creating nonlinear "raw material"
- grouping simple features to get complex ones

Neocortical structure

The structure of the model discussed here is very much inspired by the neocortex:

- hierarchy of areas creating an increasingly abstract, invariant representation
- excitatory-inhibitory interaction creating nonlinear "raw material"
- grouping simple features to get complex ones
- strong, decorrelated bottom-up stimuli

Neocortical structure

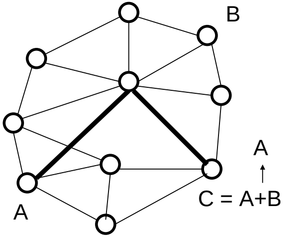
The structure of the model discussed here is very much inspired by the neocortex:

- hierarchy of areas creating an increasingly abstract, invariant representation
- excitatory-inhibitory interaction creating nonlinear "raw material"
- grouping simple features to get complex ones
- strong, decorrelated bottom-up stimuli
- competition modulated by context

Attention

- Attentional filtering decides which information reaches global context.
- Attention has a strong goal-directed component.
- Hypothesis: attention mediates goal information in perceptual learning.

Attention



Conclusion

- Basic idea in DSS: whitening or other normalisation makes learning sensitive to denoising or other such operations (e.g., combination of several datasets).

Conclusion

- Basic idea in DSS: whitening or other normalisation makes learning sensitive to denoising or other such operations (e.g., combination of several datasets).
- DSS is flexible, robust, fast and is suitable for analysing large datasets.

Conclusion

- Basic idea in DSS: whitening or other normalisation makes learning sensitive to denoising or other such operations (e.g., combination of several datasets).
- DSS is flexible, robust, fast and is suitable for analysing large datasets.
- With nonlinear feature expansion, DSS can be stacked in layers to get a powerful nonlinear feature extractor.

Conclusion

- Basic idea in DSS: whitening or other normalisation makes learning sensitive to denoising or other such operations (e.g., combination of several datasets).
- DSS is flexible, robust, fast and is suitable for analysing large datasets.
- With nonlinear feature expansion, DSS can be stacked in layers to get a powerful nonlinear feature extractor.
- DSS combines attention and learning under the same framework.

More information in the Web:

- Denoising source separation. J. Särelä and H. Valpola. *Journal of Machine Learning Research*, 6:233-272, 2005. Available at `http://www.jmlr.org/papers/v6/sarela05a.html`

More information in the Web:

- Denoising source separation. J. Särelä and H. Valpola. *Journal of Machine Learning Research*, 6:233-272, 2005. Available at `http://www.jmlr.org/papers/v6/sarela05a.html`
- Behaviourally meaningful representations from normalisation and context-guided denoising. H. Valpola. *Internal Technical report*, 2004. Available at `http://cogprints.ecs.soton.ac.uk/archive/00003633/`

More information in the Web:

- Denoising source separation. J. Särelä and H. Valpola. *Journal of Machine Learning Research*, 6:233-272, 2005. Available at http://www.jmlr.org/papers/v6/sarela05a.html
- Behaviourally meaningful representations from normalisation and context-guided denoising. H. Valpola. *Internal Technical report*, 2004. Available at http://cogprints.ecs.soton.ac.uk/archive/00003633/
- Development of representations, categories and concepts—a hypothesis. Accepted in *CIRA 2005 special session on ontogenetic robotics.*

More information in the Web:

- Denoising source separation. J. Särelä and H. Valpola. *Journal of Machine Learning Research*, 6:233-272, 2005. Available at http://www.jmlr.org/papers/v6/sarela05a.html
- Behaviourally meaningful representations from normalisation and context-guided denoising. H. Valpola. *Internal Technical report*, 2004. Available at http://cogprints.ecs.soton.ac.uk/archive/00003633/
- Development of representations, categories and concepts—a hypothesis. Accepted in *CIRA 2005 special session on ontogenetic robotics.*
- DSS project pages http://www.cis.hut.fi/projects/dss.

Thank you for your attention

Powered by beamer-style for LaTeX