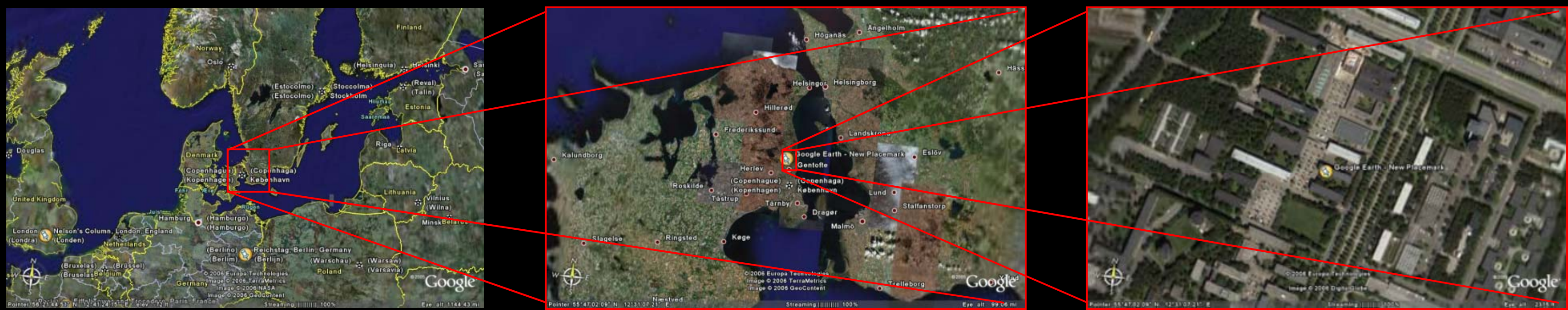
Bayesian Methods for Tensor Decompositions



Morten Mørup
DTU Informatics
Cognitive Systems Group
Technical University of Denmark

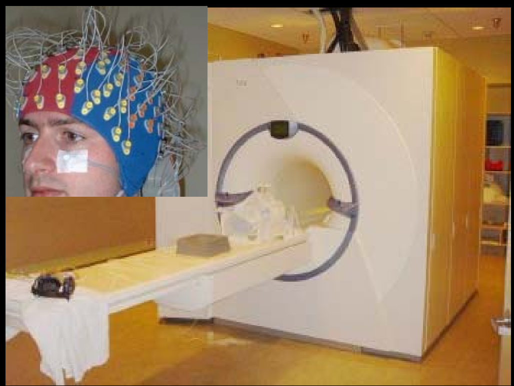
Joint work with Lars Kai Hansen
DTU Informatics
Cognitive Systems Group
Technical University of Denmark

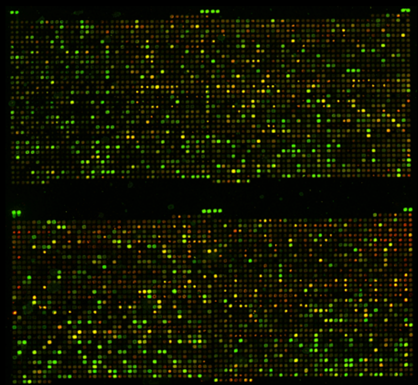
DTU

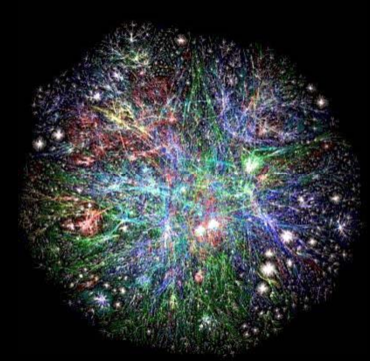
A common problem encountered in Modern Massive Datasets (MMDS)

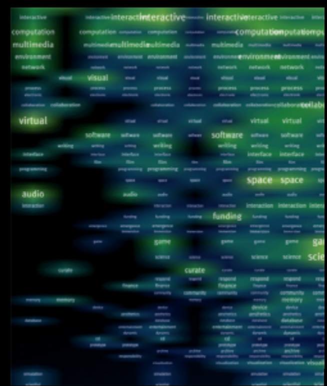
1)Multiple comparisons
2)What is the true number of independent tests (data highly correlated)
3)Data extremely noisy, i.e. low SNR rendering tests insignificant.

$\mathbf{X}^{Space \times Time}$ $\mathbf{X}^{Gene\ seq. \times Samples}$ $\mathbf{X}^{Webpages \times Webpages}$ $\mathbf{X}^{Term \times Document}$









NeuroInformatics **BioInformatics** **ComplexNetworks** **WebDataMining**

Unsupervised Learning attempts to find the hidden causes and underlying structure in the data.

(Multivariate exploratory analysis – driving hypotheses)

Goal of unsupervised Learning

(Ghahramani & Roweis, 1999)


- Perform dimensionality reduction
- Build topographic maps
- Find the hidden causes or sources of the data
- Model the data density
- Cluster data

Purpose of unsupervised learning

(Hinton and Sejnowski, 1999)


- Extract an efficient internal representation of the statistical structure implicit in the inputs

WIRED MAGAZINE: 16.07

2008

SCIENCE : DISCOVERIES

The End of Theory: The Data Deluge Makes the Scientific Method Obsolete

By Chris Anderson 06.23.08

Illustration: Marian Bantjes

THE PETABYTE AGE:
Sensors everywhere. Infinite storage. Clouds of processors. Our ability to capture, warehouse, and understand massive amounts of data is changing science, medicine, business, and technology. As our collection of facts and figures grows, so will the opportunity to find answers to fundamental questions. Because in the

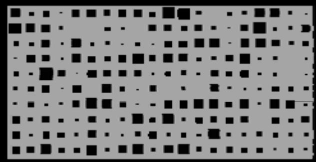
"All models are wrong, but some are useful."

So proclaimed statistician George Box 30 years ago, and he was right. But what choice did we have? Only models, from cosmological equations to theories of human behavior, seemed to be able to consistently, if imperfectly, explain the world around us. Until now. Today companies like Google, which have grown up in an era of massively abundant data, don't

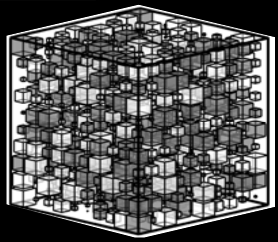
Analysis of massive amounts of data will be the main driving force of all sciences in the future!!

Vector: 1-way array/ 1st order tensor,

Matrix: 2-way array/ 2nd order tensor,



3-way array/3rd order tensor



Multi-way modeling has become an important tool for **Unsupervised Learning** and are in frequent use today in a variety of fields including

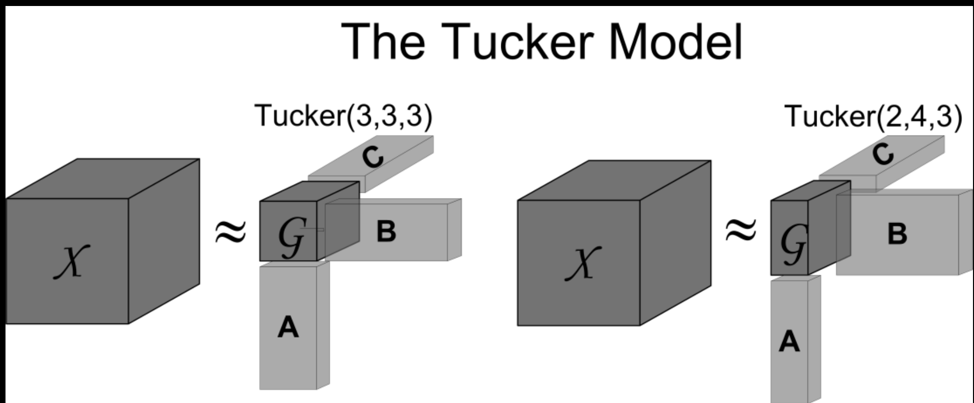
- Psychometrics (Subject x Task x Time)
- Chemometrics (Sample x Emission x Absorption)
- NeuroImaging (Channel x Time x Trial)
- Textmining (User x Query x Webpage or Webpage x Webpage X Anchor text)
- Signal Processing (ICA, i.e. diagonalization of Cummulants)

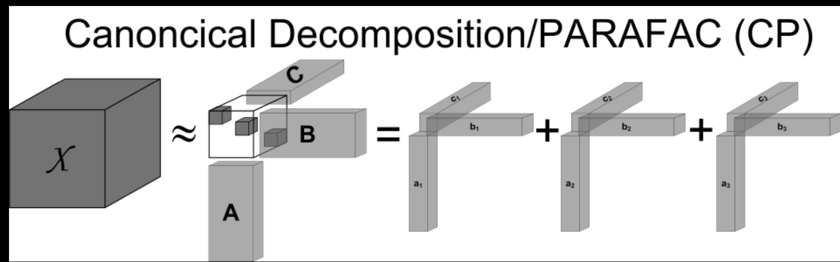
Tensor Decomposition

The two most commonly used tensor decomposition models are the CandeComp/PARAFAC (CP) model and the Tucker model

$$\text{Tucker}(J_1, J_2, \ldots J_N): \quad \mathcal{X}_{i_1,i_2,\ldots,i_N} \approx \sum_{j_1,j_2,\ldots,j_N}^{J_1,J_2,\ldots,J_N} \mathcal{G}_{j_1,j_2,\ldots,j_N} a^{(1)}_{i_1,j_1} a^{(2)}_{i_2,j_2} \cdot \ldots \cdot a^{(N)}_{i_N,j_N}$$

$$\text{CP:} \quad \mathcal{X}_{i_1,i_2,\ldots,i_N} \approx \sum_j^J a^{(1)}_{i_1,j} a^{(2)}_{i_2,j} \cdot \ldots \cdot a^{(N)}_{i_N,j}$$





$$\text{Tucker:} \quad \mathcal{X} = \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_3 \ldots \times_N \mathbf{A}^{(N)}$$

$$\text{CP:} \quad \mathcal{X} = \mathcal{I} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_3 \ldots \times_N \mathbf{A}^{(N)}$$

$$\text{where} \quad (\mathcal{Q} \times_n \mathbf{P})_{i_1,i_2,\ldots,j_n,\ldots i_N} = \sum_{i_n} \mathcal{Q}_{i_1,i_2,\ldots,i_n,\ldots i_N} \mathbf{P}_{j_n,i_n}$$

$$\boxed{\text{SVD: } \boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top = \boldsymbol{S} \times_1 \boldsymbol{U} \times_2 \boldsymbol{V}}$$

DTU

Important Question:

What constitutes an adequate number of components?

i.e. determining J for CP and $J_1, J_2, \ldots, J_N$ for Tucker, is an open problem, particularly difficult for the Tucker model as the number of components are specified for each modality separately

Notice:

- CP-model unique

$$\mathcal{X} \approx (\mathcal{D} \times_1 \boldsymbol{Q} \times_2 \boldsymbol{R} \times_3 \boldsymbol{S}) \times_1 (\boldsymbol{A}\boldsymbol{Q}^{-1}) \times_2 (\boldsymbol{B}\boldsymbol{R}^{-1}) \times_3 (\boldsymbol{C}\boldsymbol{S}^{-1}) = \widetilde{\mathcal{D}} \times_1 \widetilde{\boldsymbol{A}} \times_2 \widetilde{\boldsymbol{B}} \times_3 \widetilde{\boldsymbol{C}}.$$

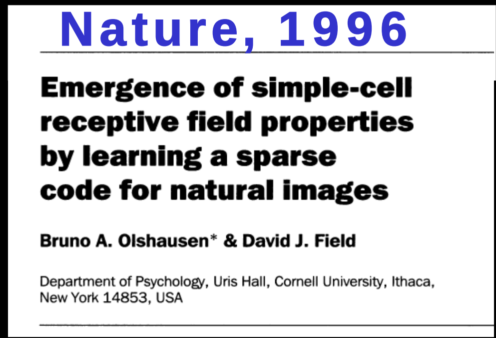
- Tucker model not unique

$$\mathcal{X} \approx (\mathcal{G} \times_1 \boldsymbol{Q} \times_2 \boldsymbol{R} \times_3 \boldsymbol{S}) \times_1 (\boldsymbol{A}\boldsymbol{Q}^{-1}) \times_2 (\boldsymbol{B}\boldsymbol{R}^{-1}) \times_3 (\boldsymbol{C}\boldsymbol{S}^{-1}) = \widetilde{\mathcal{G}} \times_1 \widetilde{\boldsymbol{A}} \times_2 \widetilde{\boldsymbol{B}} \times_3 \widetilde{\boldsymbol{C}}.$$

However, contrary to SVD neither of the two models are nested, i.e., factors for the smaller models not in general contained in larger model!

Sparse Coding

Bruno A. Olshausen



Nature, 1996

Emergence of simple-cell receptive field properties by learning a sparse code for natural images

Bruno A. Olshausen* & David J. Field

Department of Psychology, Uris Hall, Cornell University, Ithaca, New York 14853, USA

David J. Field

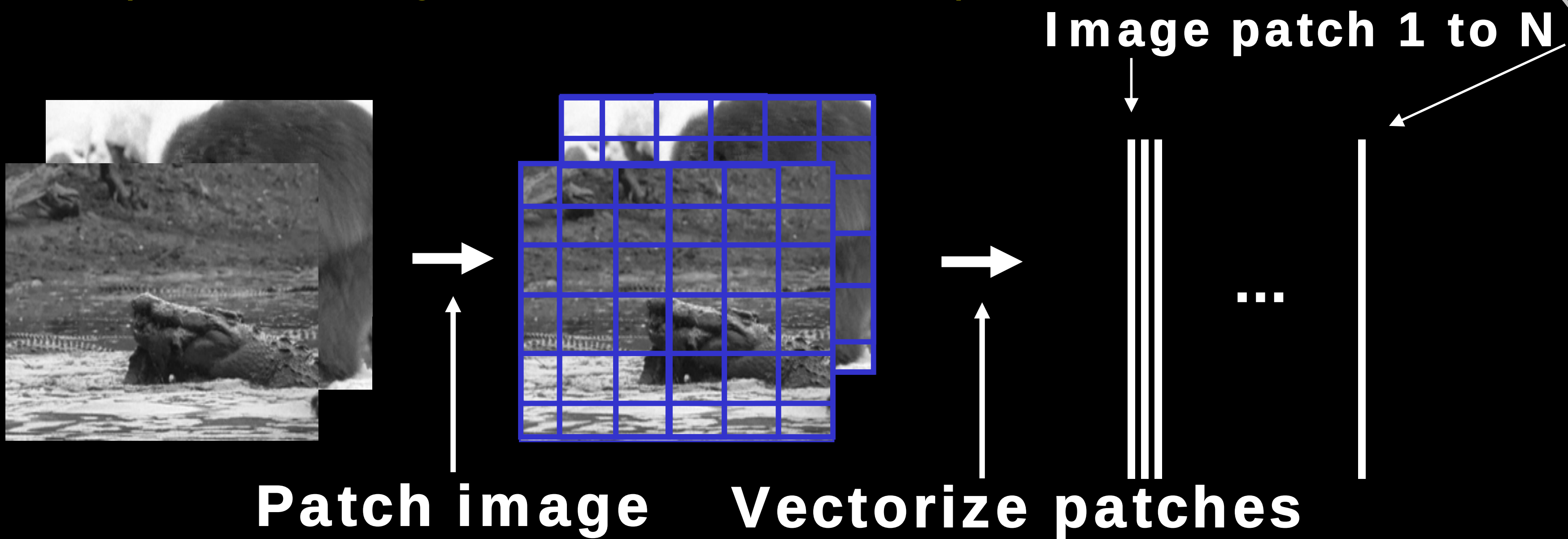
$$\operatorname{argmin}_{\boldsymbol{A},\boldsymbol{S}} \underbrace{D(\boldsymbol{X},\boldsymbol{AS})}_{\text{Preserve Information}} + \lambda \underbrace{sp(\boldsymbol{S})}_{\text{Preserve Sparsity (Simplicity)}}$$

Tradeoff parameter

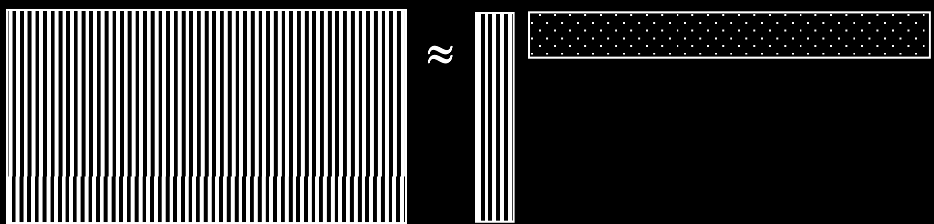
Open problem to choose tradeoff/regularisation parameter λ

$$C(\mathbf{A},\mathbf{S}) = \frac{1}{2}\|\mathbf{X}^{pixels \times patches} - \mathbf{A}^{pixels \times feat.}\mathbf{S}^{feat. \times patches}\|_F^2 + \lambda|\mathbf{S}|_1$$

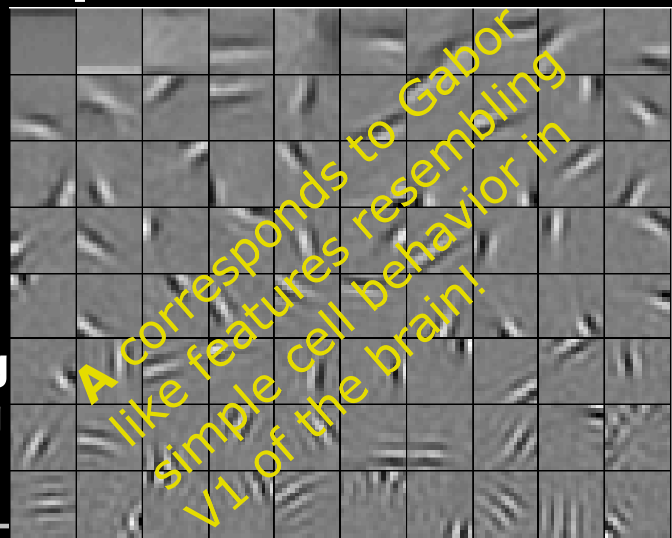
Sparse Coding relates to how brain process information!



$$C(\mathbf{A},\mathbf{S}) = \frac{1}{2}\|\mathbf{X}^{pixels\times patches} - \mathbf{A}^{pixels\times feat.}\mathbf{S}^{feat.\times patches}\|_F^2 + \lambda|\mathbf{S}|_1$$

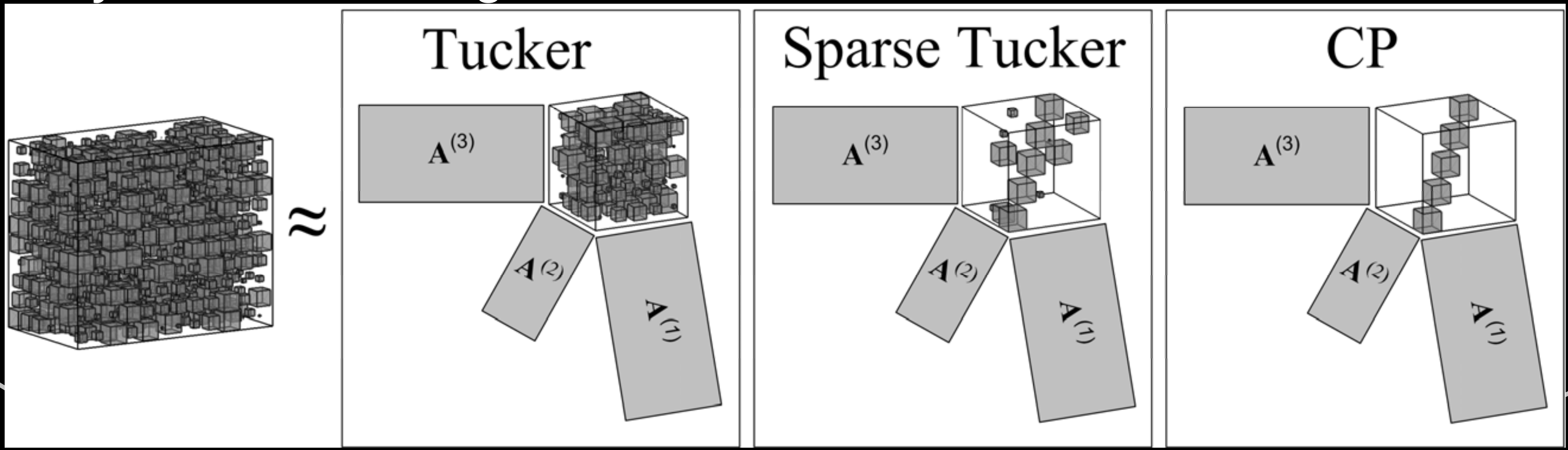


Contrary to compressive sensing Sparse coding attempts to both learn dictionary and encoding at the same time (This problem is not convex!)



Agenda

- To use sparse coding to Simplify the Tucker core forming a unique representation as well as enable interpolation between the Tucker (full core) and CP (diagonal core) model.
- To use sparse coding to turn off excess components in the CP and Tucker model and thereby select the model order.
- To tune the pruning/regularization strength from data by Automatic Relevance Determination (ARD) based on Bayesian learning.



Bayesian Learning and the Principle of Parsimony

William of Ockham

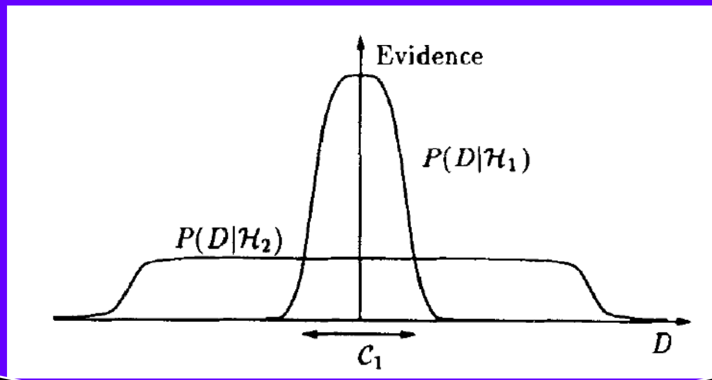
The explanation of any phenomenon should make as few assumptions as possible, eliminating those that make no difference in the observable predictions of the explanatory hypothesis or theory.

To get the posterior probability distribution, multiply the prior probability distribution by the likelihood function and then normalize

Thomas Bayes

David J.C. MacKay

Bayesian learning embodies Occam's razor, i.e. Complex models are penalized.



Many inference paradigms in Bayesian Learning

- **Maximum a posteriori estimation (MAP)**
  seeks optimal solution (admit standard optimization) however, the approach does not take parameter uncertainty into account
- **Sampling methods**
  Marcov Chain Monte Carlo (MCMC)
- **Variational methods (VB) and Belief Propagation (BP)**
  Approximate likelihood $P(\theta)$ by factorized form $Q(\theta)$ that is tractable
  VB: minimize the Kulback Leibler divergence $KL(P(\theta)|Q(\theta))$
  BP: minimize the Kulback Leibler divergence $KL(Q(\theta)|P(\theta))$

(Notice: Bayesian Learning based on MAP admits direct use of all your favorite standard optimization tools)

Automatic Relevance Determination (ARD)

- Automatic Relevance Determination (ARD) is a hierarchical Bayesian approach widely used for model selection
- In ARD hyper-parameters explicitly represents the relevance of different features by defining their range of variation.
  (i.e., Range of variation$\rightarrow$0 $\Rightarrow$ Feature removed)

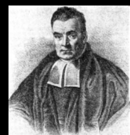
A motivating example: A Bayesian formulation of the Lasso /Basis Pursuit Denoising (BPD) problem

$$\text{LASSO/BPD:} \quad \arg\min_{\mathbf{s}} \frac{1}{2\sigma^2}\|\mathbf{x}^I - \mathbf{A}^{I\times J}\mathbf{s}^J\|_F^2 + \lambda|\mathbf{s}|_1$$

$$P(\mathbf{x}|\mathbf{A},\mathbf{s},,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}^I} e^{-\frac{\|\mathbf{x}-\mathbf{A}\mathbf{s}\|_F^2}{2\sigma^2}}$$ **Likelihood**

$$P(\mathbf{s}|\lambda) = (\frac{\lambda}{2})^J e^{-\lambda|\mathbf{s}|_1}$$ **Prior**

$$P(\mathbf{s}|\mathbf{A},\mathbf{x},\sigma^2,\lambda) = \frac{P(\mathbf{x}|\mathbf{A},\mathbf{s},\sigma^2)P(\mathbf{s}|\lambda)}{P(\mathbf{x})}$$ **Bayes**



$$\begin{aligned} -\log P(\mathbf{s}|\mathbf{A},\mathbf{x},\sigma^2,\lambda) &= -\log P(\mathbf{x}|\mathbf{A},\mathbf{s},\sigma^2) - \log P(\mathbf{s}|\lambda) + \log P(\mathbf{x}) \\ &= \underbrace{\frac{\|\mathbf{x}-\mathbf{A}\mathbf{s}\|_F^2}{2\sigma^2} + \lambda|\mathbf{s}|_1}_{\text{BPD / LASSO term}} + \underbrace{\frac{I}{2}\log 2\pi\sigma^2 - J\log\frac{\lambda}{2}}_{\text{Normalization terms}} + Const. \end{aligned}$$

$$\boxed{\frac{\partial \log P(\mathbf{s}|\mathbf{A},\mathbf{x},\sigma^2,\lambda)}{\partial\lambda} = 0 \Rightarrow \lambda = \frac{J}{|\mathbf{s}|_1}}$$

ARD in reality a ℓ_0-norm optimization scheme. As such ARD based on Laplace prior corresponds to ℓ_0-norm optimization by re-weighted ℓ_1-norm

In particular if we define λ for each entry in $\mathbf{s}$, i.e.

$$\frac{1}{2\sigma^2}\|\mathbf{x}^I - \mathbf{A}^{I\times J}\mathbf{s}^J\|_F^2 + \sum_j \lambda_j|\mathbf{s}_j|$$

Corresponding to the Laplace prior $P(\mathbf{s}|\boldsymbol{\lambda}) = \prod_j \frac{\lambda_j}{2} e^{-\lambda_j|s_j|}$ optimizing for λ_j gives $\lambda_j = \frac{1}{|s_j|}$ such that

$$\frac{1}{2\sigma^2}\|\mathbf{x}^I - \mathbf{A}^{I\times J}\mathbf{s}^J\|_F^2 + \sum_j \frac{|\mathbf{s}_j|}{|\widetilde{\mathbf{s}}_j|}$$

ℓ_0 norm by re-weighted ℓ_2 follows by imposing Gaussian prior instead of Laplace

Notice that we are all the time monotonically decreasing

$$-\log P(\mathbf{s}|\mathbf{A}, \mathbf{x}, \sigma^2, \lambda)$$

Sparse Tucker decomposition by ARD

$$P(\mathcal{X}|\mathcal{R},\sigma^2) = (2\pi\sigma^2)^{-\frac{I_1 I_2 \cdots I_N}{2}} e^{-\frac{\|\mathcal{X}-\mathcal{R}\|_F^2}{2\sigma^2}}$$

$$P(\mathcal{G}|\alpha^{\mathcal{G}}) = \left(\frac{\alpha^{\mathcal{G}}}{2}\right)^{J_1 J_2 \cdots J_N} e^{-\alpha^{\mathcal{G}}|\mathcal{G}|_1}$$

$$P(\mathbf{A}^{(n)}|\boldsymbol{\alpha}^{(n)}) = \prod_{j_n}\left(\frac{\alpha^{(n)}}{2}\right)^{I_n} e^{-\alpha_j^{(n)}|\mathbf{a}_j|_1}$$

$$L = P(\mathcal{G}, \boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(N)}|\mathcal{X}, \sigma^2, \alpha^{\mathcal{G}}, \boldsymbol{\alpha}^{(1)}, \ldots, \boldsymbol{\alpha}^{(n)})$$
$$\propto P(\mathcal{X}|\mathcal{R},\sigma^2)P(\mathcal{G}|\alpha^{\mathcal{G}})P(\boldsymbol{A}^{(1)}|\boldsymbol{\alpha}^{(1)})\cdots P(\boldsymbol{A}^{N}|\boldsymbol{\alpha}^{(N)}).$$

Thus the negative log likelihood based on Laplace priors is proportional to

$$-\log L \propto \mathrm{c} + \frac{1}{2\sigma^2}\|\mathcal{X}-\mathcal{R}\|_F^2 + \sum_n \sum_{j_n} \alpha_{j_n}^{(n)}|\boldsymbol{a}_{j_n}^{(n)}|_1 + \alpha^{\mathcal{G}}|\mathcal{G}|_1$$
$$+\frac{1}{2}I_1 I_2 \cdots I_N \log\sigma^2 - \sum_n \sum_{j_n} I_n \log\alpha_{j_n}^{(n)} - J_1 J_2 \cdots J_n \log\alpha^{\mathcal{G}}.$$

Maximum a posteriori (MAP) estimation

Brakes into standard Lasso/BPD sub-problems of the form

$$\arg\min_{\boldsymbol{A}^{(n)}} \frac{1}{2\sigma^2}\|\boldsymbol{X}_{(n)} - \boldsymbol{A}^{(n)}\boldsymbol{Z}_{(n)}\|_F^2 + \sum_j \lambda_j |\boldsymbol{a}_j^{(n)}|_1$$

Update of regularization parameters by ARD

$$\alpha_{\mathcal{G}} = \frac{J_1 J_2 \cdots J_N}{|\mathcal{G}|_1}, \quad \boldsymbol{\alpha}_d^{(n)} = \frac{J_n}{|\boldsymbol{A}_d^{(n)}|_1}$$

Solving efficiently the Lasso/BPD sub-problems

Algorithm 1 Gradient Based Sparse Coding (GBSC): $A = \text{GBSC}(X, Z, \lambda)$, solves $\arg\min_A \frac{1}{2}\|X - AZ\|_F^2 + \sum_j \lambda_j |a_j|_1$

1: **repeat**
2: Take gradient step according to LS-objective
3: $A^{new} \leftarrow A^{old} - \mu(AZ - X)Z^\top$
4: Take gradient step according to l_1-regularization
5: **if** $|a_{i,j}^{new}| < \mu\lambda_j$ **then**
6: $a_{i,j}^{new} = 0$
7: **else**
8: $a_{i,j}^{new} = a_{i,j}^{new} - \mu\lambda_j \operatorname{sign}(a_{i,j}^{new})$
9: **end if**
10: Estimate μ by line-search
11: **until** Convergence

	100×256	256×256	1000×256	2500×256
SIGNSEARCH	0.0750 ± 0.0359	0.1984 ± 0.1342	$\mathbf{0.3734 \pm 0.1759}$	$\mathbf{1.6969 \pm 0.6441}$
CONJUGATE GRADIENT	0.4172 ± 0.0651	1.1219 ± 0.2560	9.0297 ± 1.8055	45.6297 ± 12.0142
LARS	0.0453 ± 0.0226	$\mathbf{0.1313 \pm 0.0787}$	0.4313 ± 0.1477	1.9813 ± 0.6342
NNQP	0.5703 ± 0.0696	0.9313 ± 0.0748	2.8719 ± 0.1389	15.5047 ± 0.7882
GBSC	$\mathbf{0.0125 \pm 0.0066}$	0.3172 ± 0.2121	2.0688 ± 1.0760	22.8828 ± 12.2846

Notice, in general the alternating sub-problems for Tucker estimation have J<I!

DTU

The Tucker ARD Algorithm

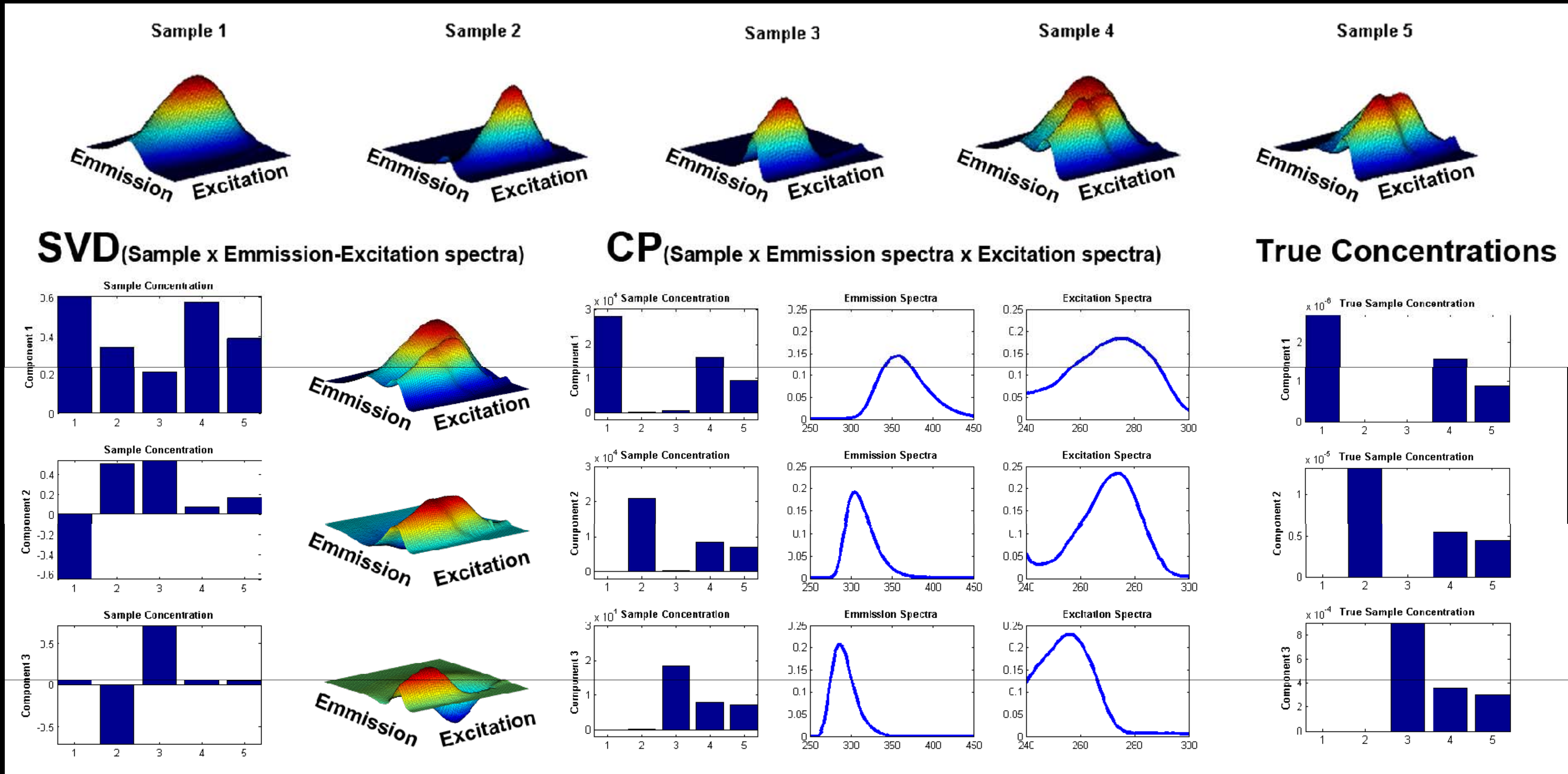
Algorithm 2 Sparse Tucker estimation based on Automatic Relevance Determination (ARD)

```
set $J_1, J_2, \ldots, J_n$ large enough to encompass all potential models, $\sigma^2 = \|\mathcal{X}\|_F^2/(I_1 I_2 \cdots I_n(1+10^{\text{SNR}/10}))$, set $\alpha_{\mathcal{G}} = 0$, $\boldsymbol{\alpha}^{(n)} = \mathbf{0}$ and initialize by random $\boldsymbol{A}^{(n)}$ for $n = 1, 2, \ldots, N$
repeat
    $\boldsymbol{Q} = \boldsymbol{A}^{(1)} \otimes \boldsymbol{A}^{(2)} \otimes \ldots \otimes \boldsymbol{A}^{(N)}$, vec($\mathcal{G}$) $\leftarrow$ gbsc(vec($\mathcal{X}$), $\boldsymbol{Q}, \sigma^2 \alpha_{\mathcal{G}}$),
    $\alpha_{\mathcal{G}} = \min\{\frac{J_1 J_2 \cdots J_N}{|\mathcal{G}|_1}, \frac{1}{\epsilon}\}$
    $\mathcal{R} = \mathcal{G} \times_1 \boldsymbol{A}^{(1)} \times_2 \boldsymbol{A}^{(2)} \times_3 \ldots \times_N \boldsymbol{A}^{(N)}$
    for n=1:N do
        $\boldsymbol{Z}_{(n)} = (\mathcal{R} \times_n \boldsymbol{A}^{(n)\dagger})_{(n)}$, $\boldsymbol{A}^{(n)} \leftarrow$ gbsc($\boldsymbol{X}_{(n)}, \boldsymbol{Z}_{(n)}, \sigma^2 \alpha^{(n)}$), $\alpha_d^{(n)} = \min\{\frac{J_n}{|\boldsymbol{A}_d^{(n)}|_1}, \frac{1}{\epsilon}\}$
        If $\alpha_{j_n}^{(n)} = \frac{1}{\epsilon}$ then $J_n = J_n - 1$, $\boldsymbol{A}^{(n)} = \boldsymbol{A}_{\setminus j_n}^{(n)}$, $\mathcal{G} = \mathcal{G}_{\setminus j_n}$,
        $\alpha^{(n)} = \alpha_{\setminus j_n}^{(n)}$ end
        $\boldsymbol{R}_{(n)} = \boldsymbol{A}^{(n)} \boldsymbol{Z}_{(n)}$
    end for
until convergence
```

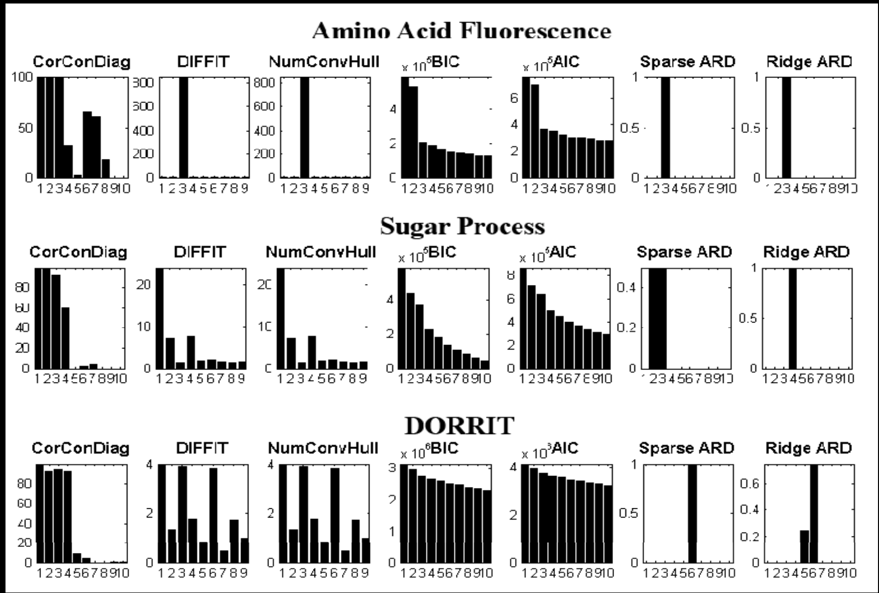
CP follows setting $\mathcal{G}=I$

Estimating the number of components comes at the same cost as fitting one conventional model!

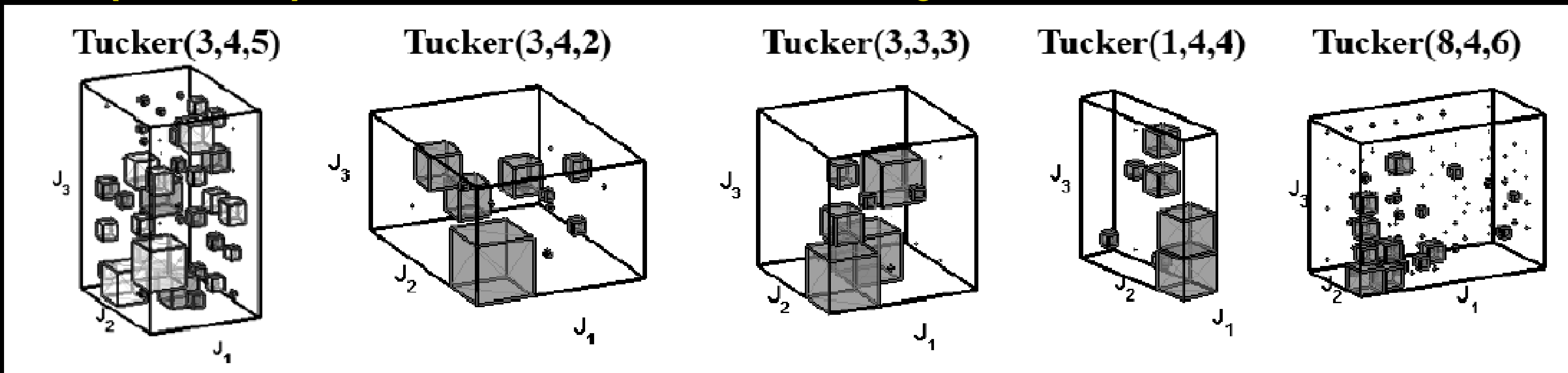
Results on Fluorscence spectroscopy data



CP models

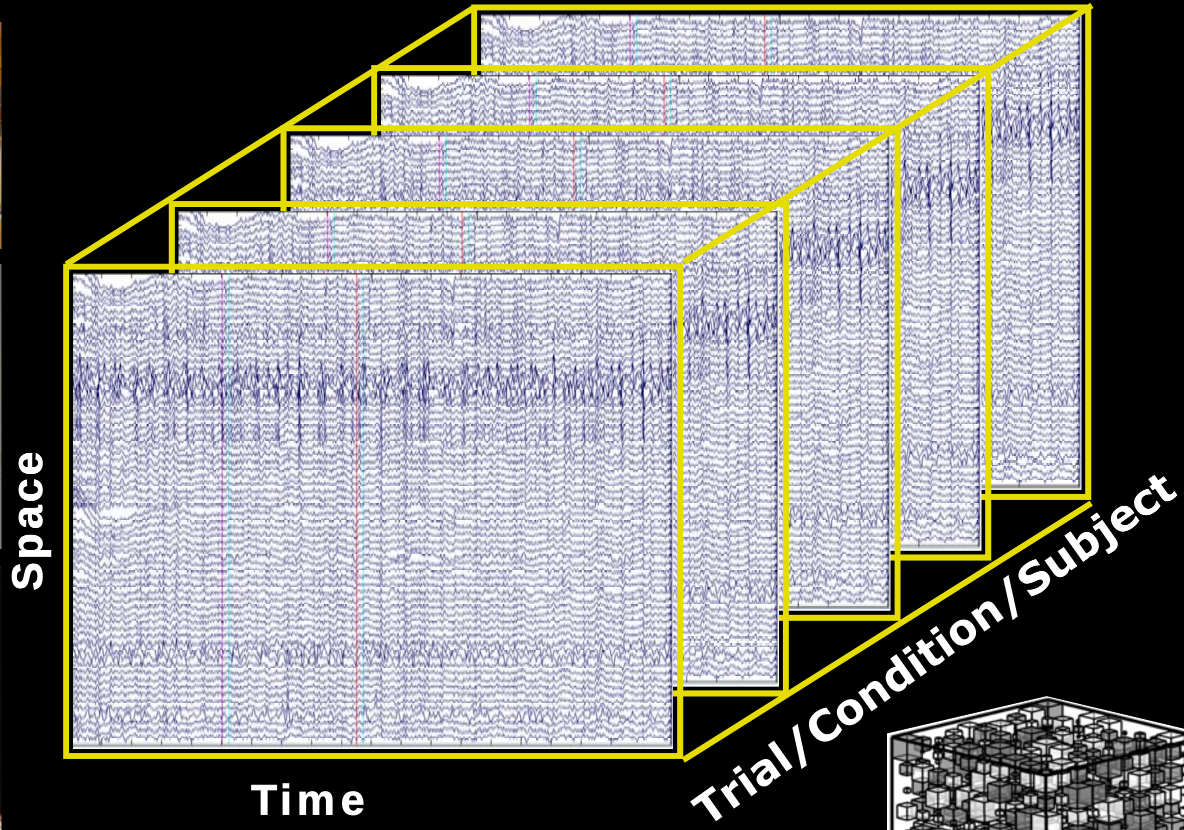


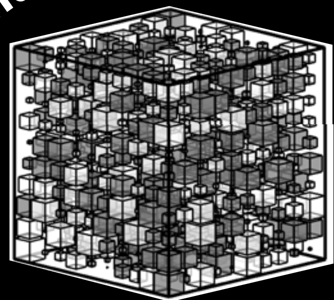
Tucker(10,10,10) models were fitted to the data, given are below the extracted cores



Synthetic Tucker(3,4,5) Tucker(3,6,4) Tucker(3,3,3) Tucker(?,4,4) Tucker(4,4,4)

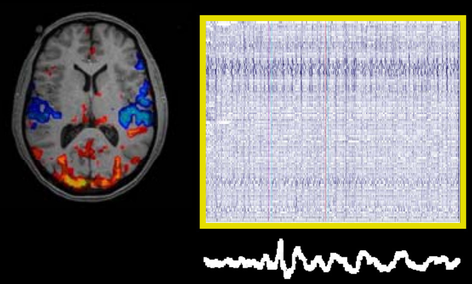
From 2-way to multi-way analysis of NeuroImaging data





Multilinear modelling

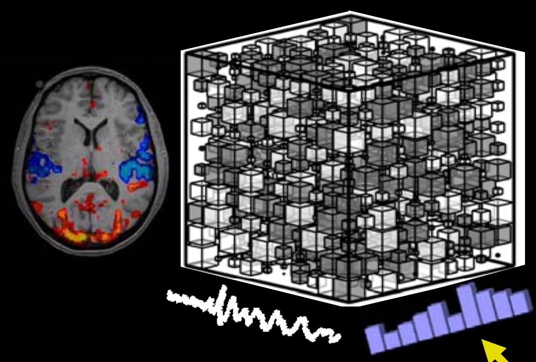
Bilinear Model: $\mathbf{X}^{\text{Voxel}\times\text{Time}} \approx \sum_d \mathbf{a}_d^{Voxel}\mathbf{b}_d^{\text{Time}}$



$\sum_d$ 

Assumption: Data **instantaneous** mixture of temporal signatures.
(PCA/ICA/NMF)

Trilinear Model: $\mathbf{X}^{\text{Voxel}\times\text{Time}\times Trial} \approx \sum_d \mathbf{a}_d^{Voxel}\mathbf{b}_d^{\text{Time}}\mathbf{c}_d^{Trial}$



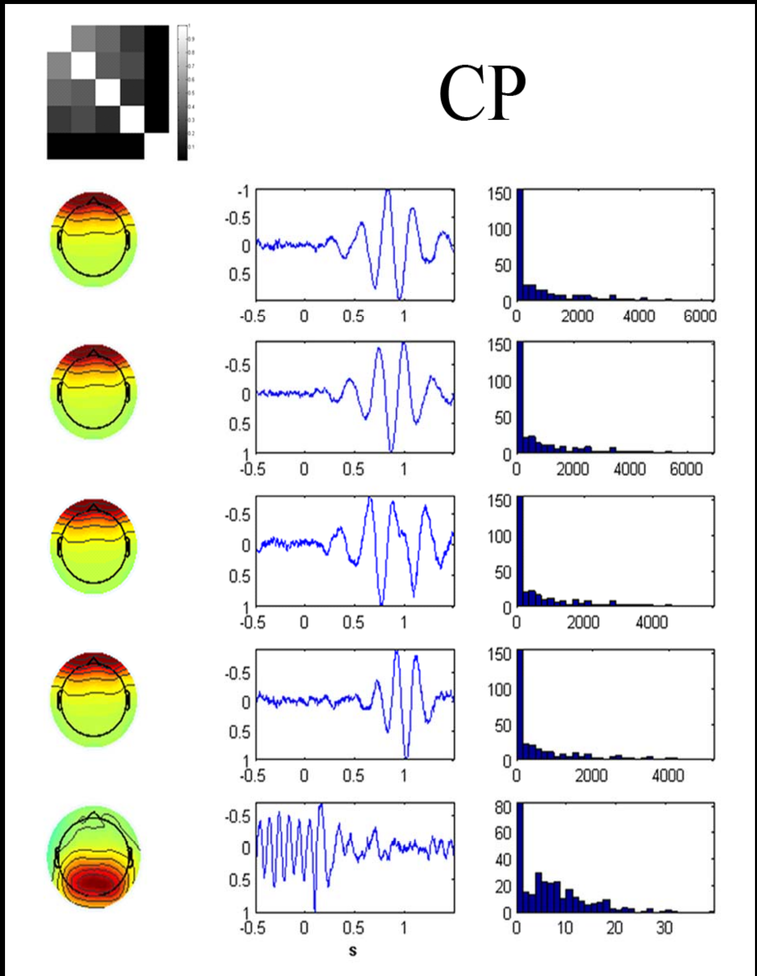
$\sum_d$ 

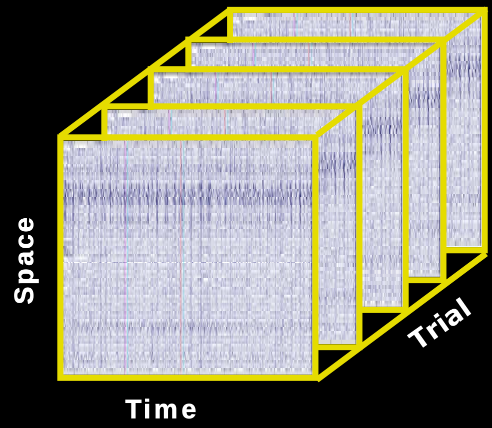
Assumption: Data **instantaneous** mixture of temporal signatures that are expressed to **various degree** over the Subjects/trials
(Canoncial Decomposition, Parallel Factor (CP))

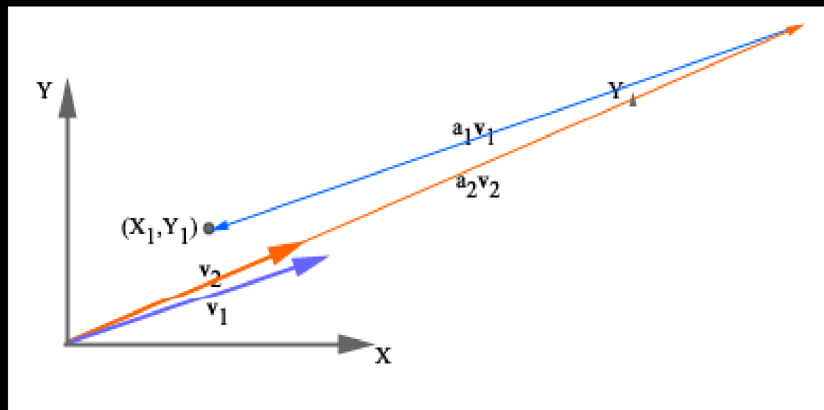
(weighted averages over the trials)

DTU

Unfortunately, Violation of multi-linearity causes degeneracy

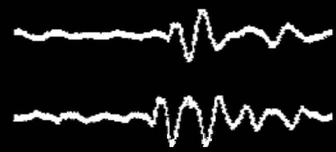




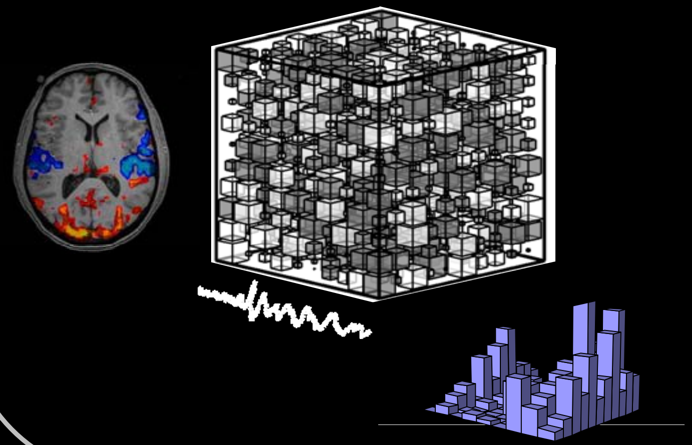


Common Fixes: Impose orthogonality, reguarlization or non-negativey constraints by analyzing Data transformed to a time-frequency domain representation

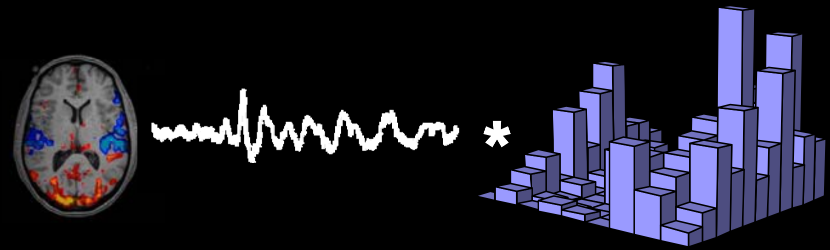
Modeling Shape (and delay) Variability



convolutive CP: $$x_{i,k}(t) \approx \sum_{d,\tau} a_{i,d} b_d(t-\tau) c_{k,d}(\tau)$$

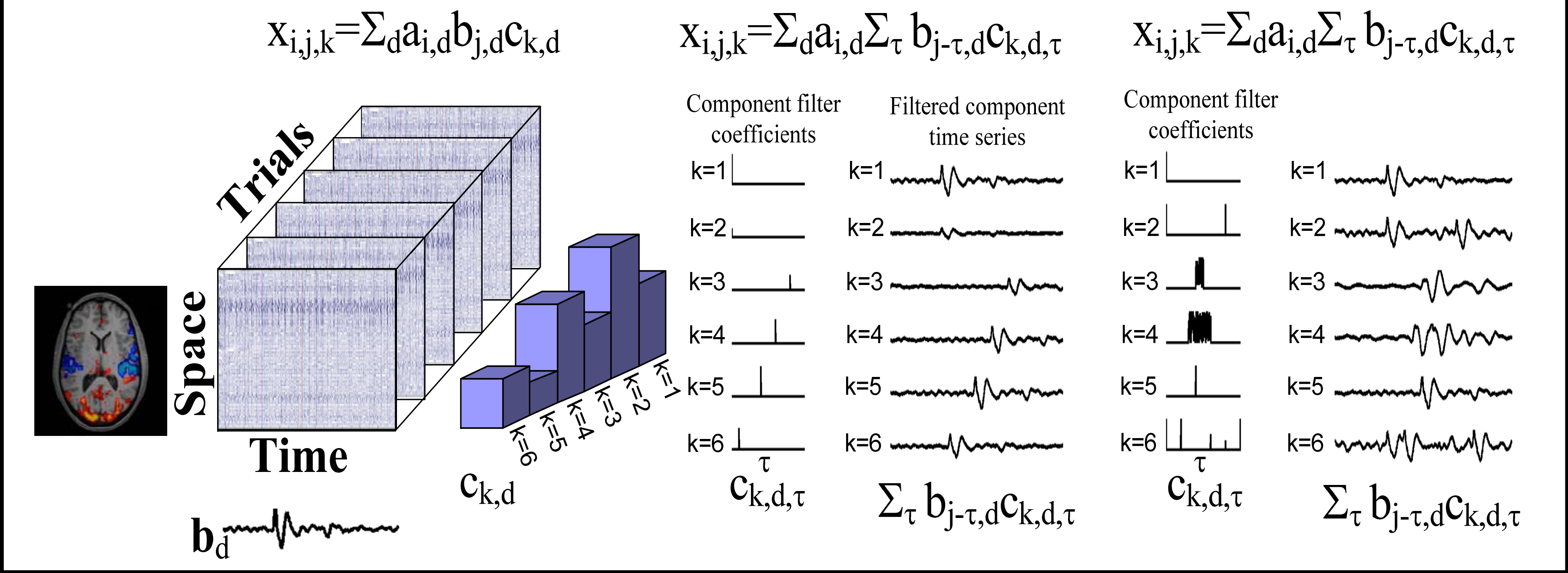


$$\sum_d$$



(Mørup et al., Nips workshop on New Directions in Statistical Learning for Meaningful and Reproducible fMRI Analysis 2008)

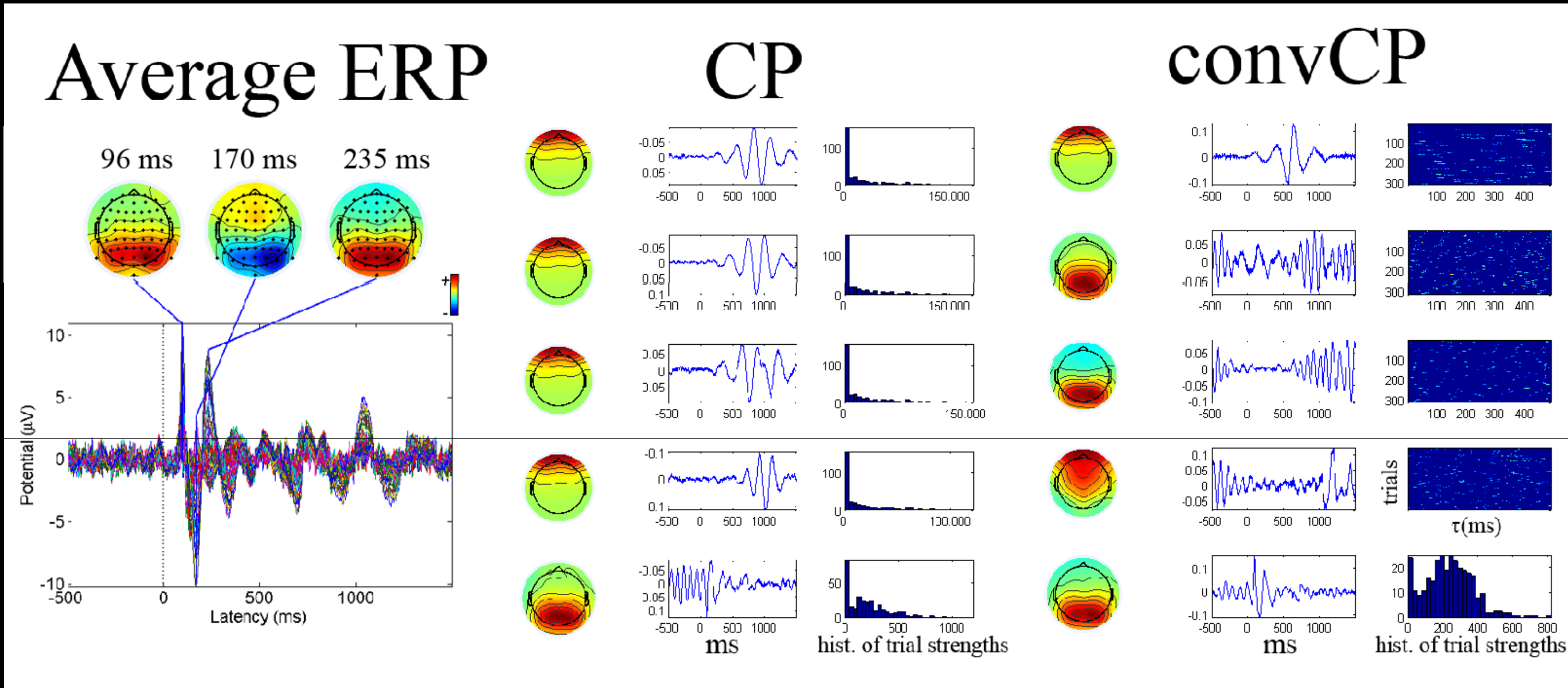
CP, ShiftCP and ConvCP



ConvCP: Can model arbitrary number of component delays within the trials and account for shape variation within the convolutional model representation. Redundancy between what is coded in C and B resolved by imposing sparsity on C. Number of components and sparsity regularisation identifed through ARD.

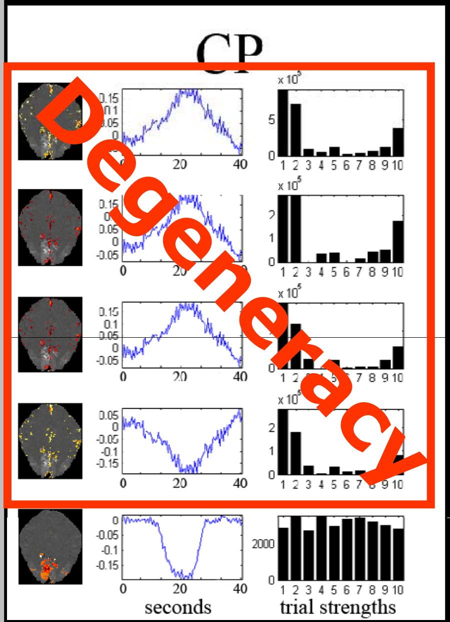
(Mørup et al., Nips workshop on New Directions in Statistical Learning for Meaningful and Reproducible fMRI Analysis 2008)

Convolutive Multi-linear decomposition



DTU

Analysis of fMRI data



Each trial consists of a visual stimulus delivered as an annular full-field checkerboard reversing at 8 Hz.

λ' is ℓ_1 sparsity regularization imposed on third mode

(Mørup et al., Nips workshop on New Directions in Statistical Learning for Meaningful and Reproducible fMRI Analysis 2008)

Convolutive bi-linear model form a Latent Causal Modeling framework

Channel Specific Input Functions

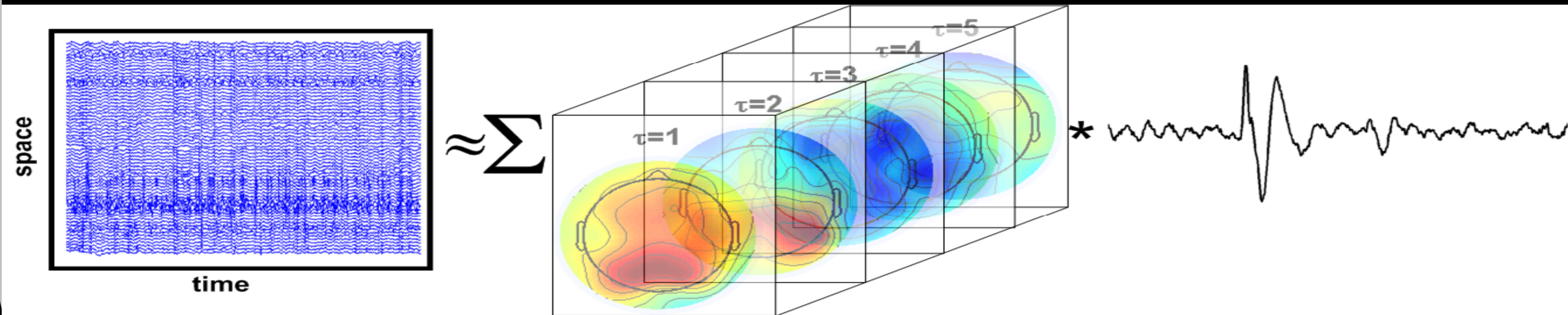
$$\text{DTF:} \quad x_i(t) = \sum_{\tau=0}^{\Upsilon-1}\sum_{j=1}^{J} h_{i,j}(\tau)e_j(t-\tau)$$

$$\text{SLCM:} \quad x_i(t) = \sum_{\tau=0}^{\Upsilon-1}\sum_{d=1}^{D} a_{i,d}(\tau)s_d(t-\tau)+\varepsilon_i(t).$$

Noise

Transfer Functions

Latent Sources



$\mathbf{x}(t)=\sum_\tau \mathbf{A}(\tau)\mathbf{s}(t-\tau)=\sum_\tau \mathbf{A}(\tau)\mathbf{Q}\mathbf{Q}^{-1}\mathbf{s}(t-\tau)=\tilde{\mathbf{A}}(\tau)\tilde{\mathbf{s}}(t-\tau)$ → **We impose sparsity on $\mathbf{A}(\tau)$**

(Mørup et al., Nips workshop on Connectivity Inference in Neuroimaging 2009)

Bayesian inference admit estimation of model order and degree of sparsity through Automatic Relevance Determination

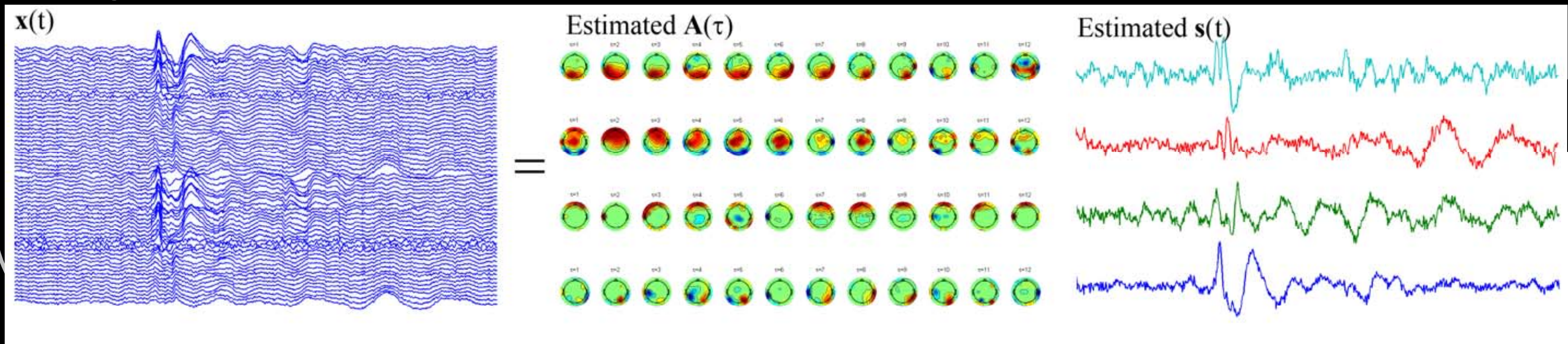
$$\textbf{SLCM:} \quad x_i(t) \;=\; \sum_{\tau=1}^{\Upsilon-1} \sum_{d=1}^{D} a_{i,d}(\tau) s_d(t-\tau) + \varepsilon_i(t).$$

$$\begin{aligned}
\varepsilon_i(t) &\sim Normal(0, \sigma^2) \\
\sigma^{-2} &\sim Gamma(1, \kappa \|\boldsymbol{X}\|_F^2) \\
\boldsymbol{a}_d(\tau) &\sim Laplace(0, \beta_d) \\
\beta_d &\sim Gamma(1, \alpha) \\
s_d(t) &\sim \delta(1 - \textstyle\sum_t s_d(t)^2)
\end{aligned}$$

$$\log P(\boldsymbol{X}, \mathcal{A}, \boldsymbol{S}, \sigma^{-2}, \boldsymbol{\beta} | \kappa, \alpha) = \begin{cases} -\frac{\sigma^{-2}}{2} \sum_t^T \|\boldsymbol{x}(t) - \sum_\tau \boldsymbol{A}(\tau)\boldsymbol{s}(t-\tau)\|_F^2 \\ -\frac{1}{2} IT \log(\sigma^{-2}) - \kappa \|\boldsymbol{X}\|_F^2 \sigma^{-2} \\ +\sum_d I\Upsilon \log \beta_d - \beta_d(\alpha + \sum_i^I \sum_\tau^\Upsilon |a_{i,d}(\tau)|) \\ +const. \\ \quad s.t. \quad \sum_t s_d(t)^2 = 1 \end{cases}$$

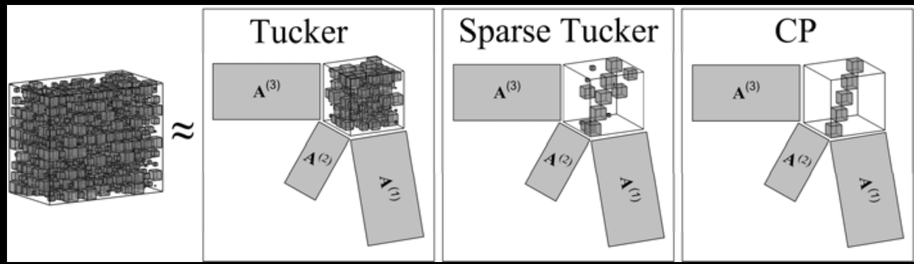
Regularization strength learned from data, i.e. $\beta_d^{\mathrm{MAP}} = \dfrac{I\Upsilon}{\alpha + \sum_i^I \sum_\tau^\Upsilon |a_{i,d}(\tau)|}$

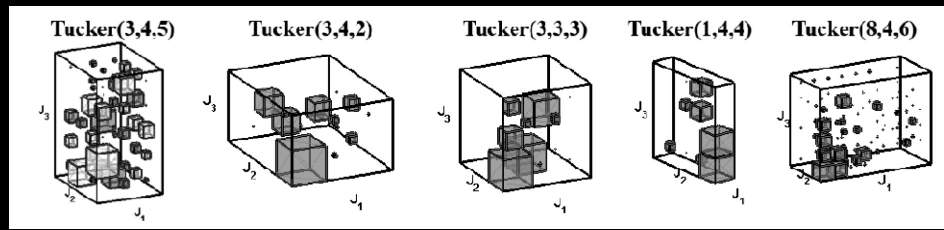
Analysis of EEG data



Conclusion

- Imposing sparseness on the core enable to interpolate between Tucker and CP model
- ARD from Bayesian learning based on MAP estimation form a simple framework to tune the pruning in sparse coding models giving the model order.
- ARD framework especially useful for the Tucker model where the order is specified for each mode separately which makes exhaustive evaluation of all potential models expensive.
- ARD-estimation based on MAP is closely related to ℓ_0 norm estimation based on reweighted ℓ_1 and ℓ_2 norm. Thus, ARD form a principled framework for learning the sparsity pattern.





$$\frac{1}{2\sigma^2}\|\mathbf{x}^I - \mathbf{A}^{I\times J}\mathbf{s}^J\|_F^2 + \sum_j \lambda_j |\mathbf{s}_j|$$

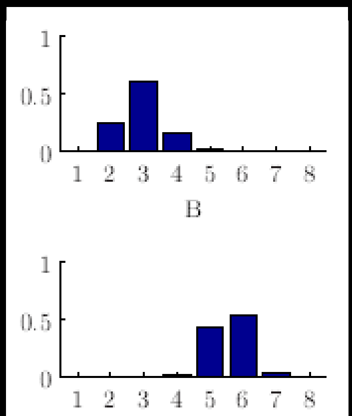
Corresponding to the Laplace prior $P(\mathbf{s}|\boldsymbol{\lambda}) = \prod_j \frac{\lambda_j}{2} e^{-\lambda_j |s_j|}$ optimizing for λ gives $\lambda = \frac{1}{|s_j|}$ such that

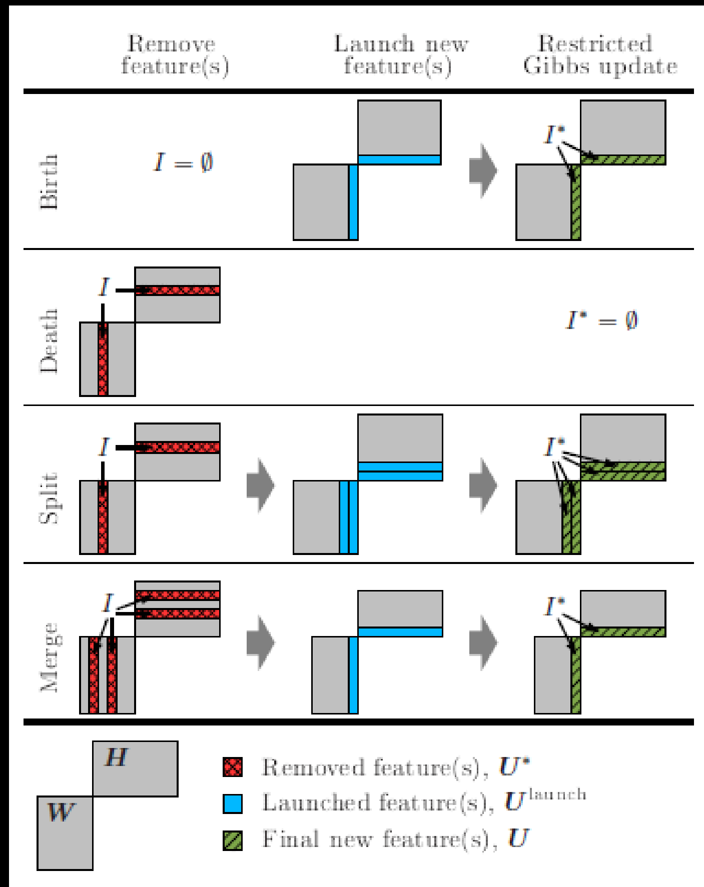
$$\frac{1}{2\sigma^2}\|\mathbf{x}^I - \mathbf{A}^{I\times J}\mathbf{s}^J\|_F^2 + \sum_j \frac{|\mathbf{s}_j|}{|\tilde{\mathbf{s}}_j|}$$

DTU

Current research

- Non-parametric efficient sampling approaches for model order estimation based on reversible jump MCMC for the described models.
  (See also Schmidt and Mørup, Infinite Non-negative Matrix Factorization, to appear EUSIPCO 2010)





References

M. Mørup, Applications of tensor (multi-way array) factorizations and decompositions in data mining, to appear Wiley DMKD 2010.

M. N. Schmidt, M Mørup, Infinite Non-negative Matrix Factorization, to appear Eusipco 2010

M. Mørup, L.K. Hansen, Automatic Relevance, Determination for multi-way models, Journal of Chemometrics, 2009

M. Mørup, Kristoffer H. Madsen, L.K. Hansen, Latent Causal Modeling of Neuroimaging Data, NIPS workshop on Connectivity inference in Neuroimaging data, 2009

M. Mørup, L.K. Hansen, S.M. Arnfred, L.-K. Lim, K.M. Madsen, Shift Invariant Multilinear Decomposition of Neuroimaging Data, NeuroImage vol. 42(4), pp.1439-50, 2008

M. Mørup, Kristoffer H. Madsen, L.K. Hansen Modeling trial based neuroimaging data, Nips workshop on New Directions in Statistical Learning for Meaningful and Reproducible fMRI Analysis, 2008