Gaussian Processes & Kernelization for Tensor-Based Models

Liqing Zhang

MOE-MS Joint Lab for
Intelligence Computing and Intelligent Systems
Shanghai Jiao Tong University

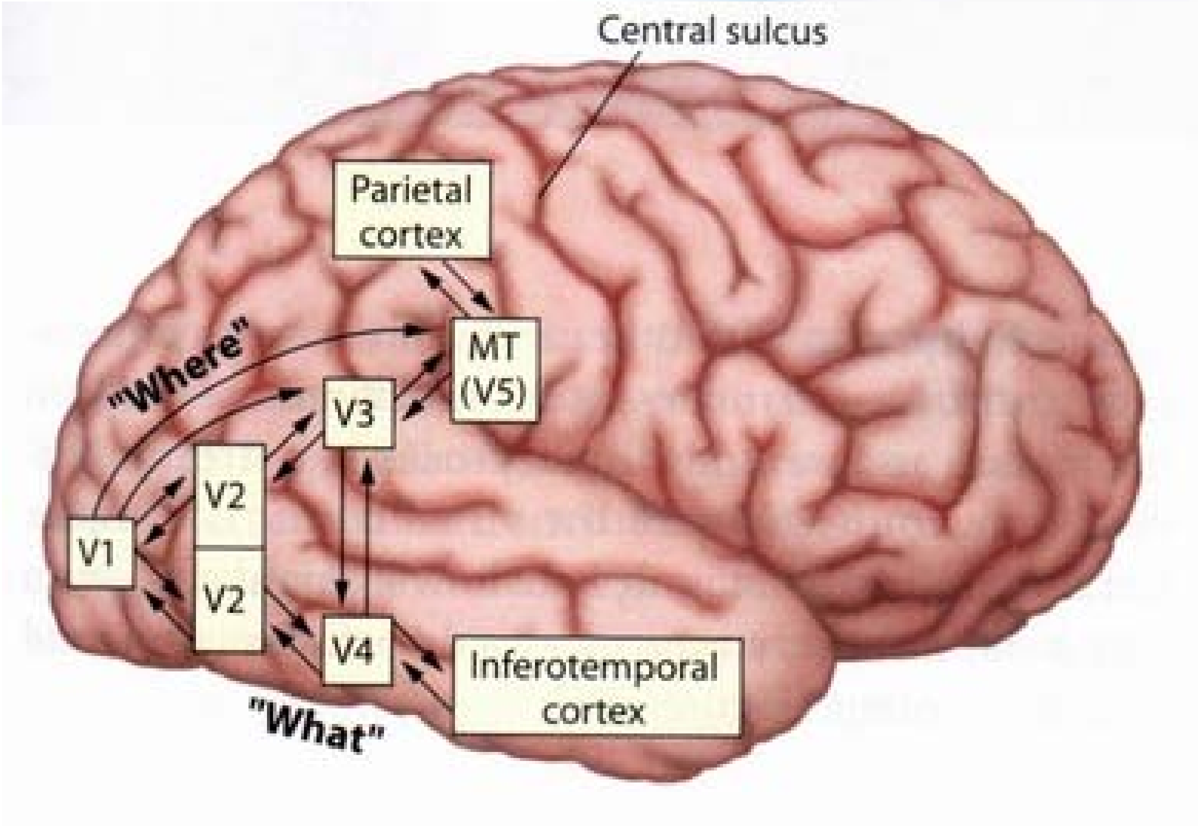
MLA2013 Fudan
2013.11.2

BCMI Brain-like Computing & Machine Intelligence

Outline

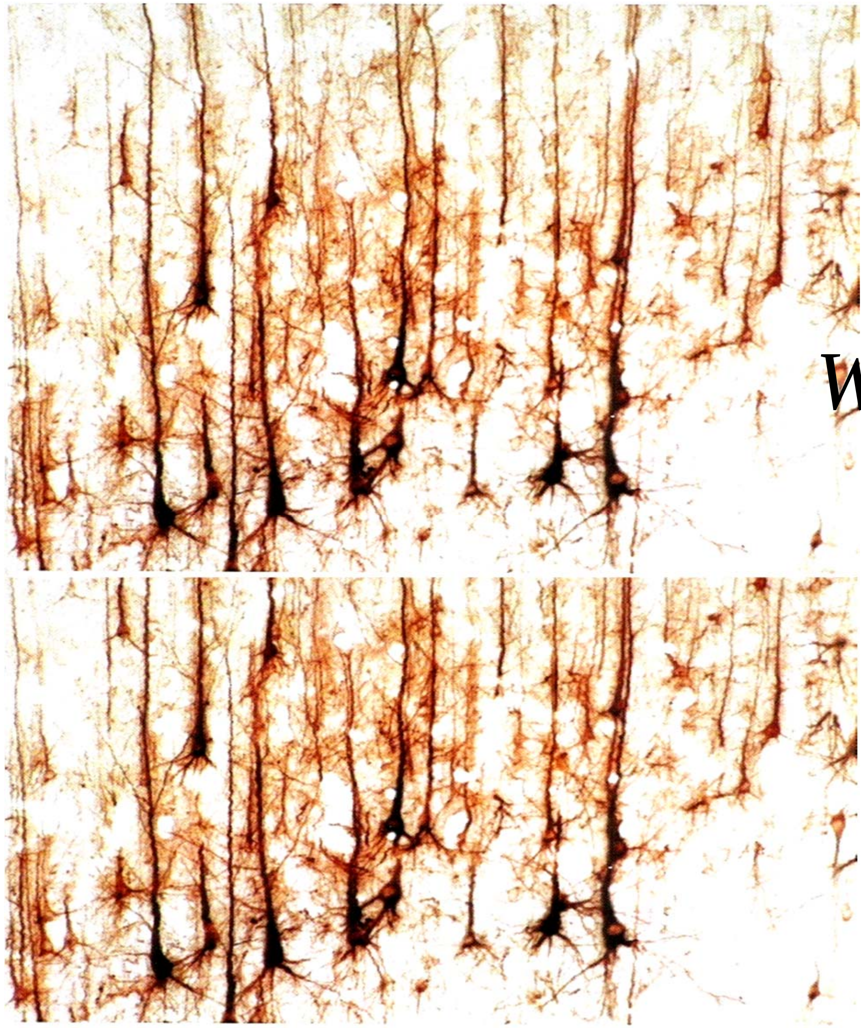
- Background and Motivation
- Tensor Factorization/Decomposition
- Gaussian Processes
- Probabilistic Kernels for Tensors
- Constrained Tensor Factorization
  - ✓ Nonnegative Tenor Factorization
  - ✓ Discriminative Tensor Factorization
- Multilinear regression
- Conclusions and Perspectives

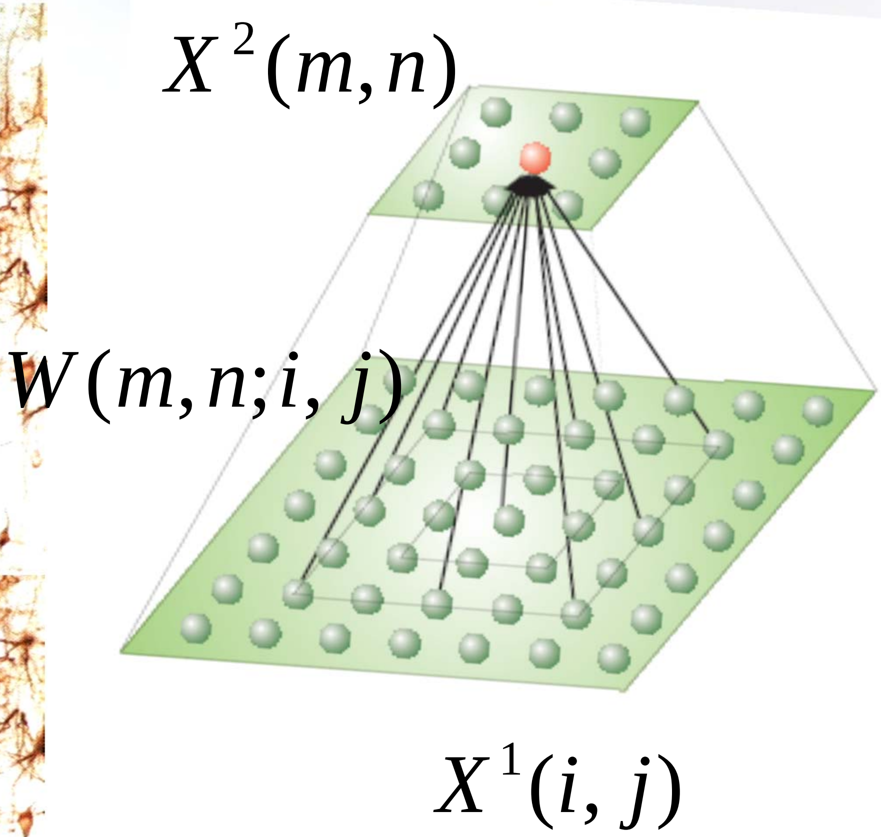
Cortical Network



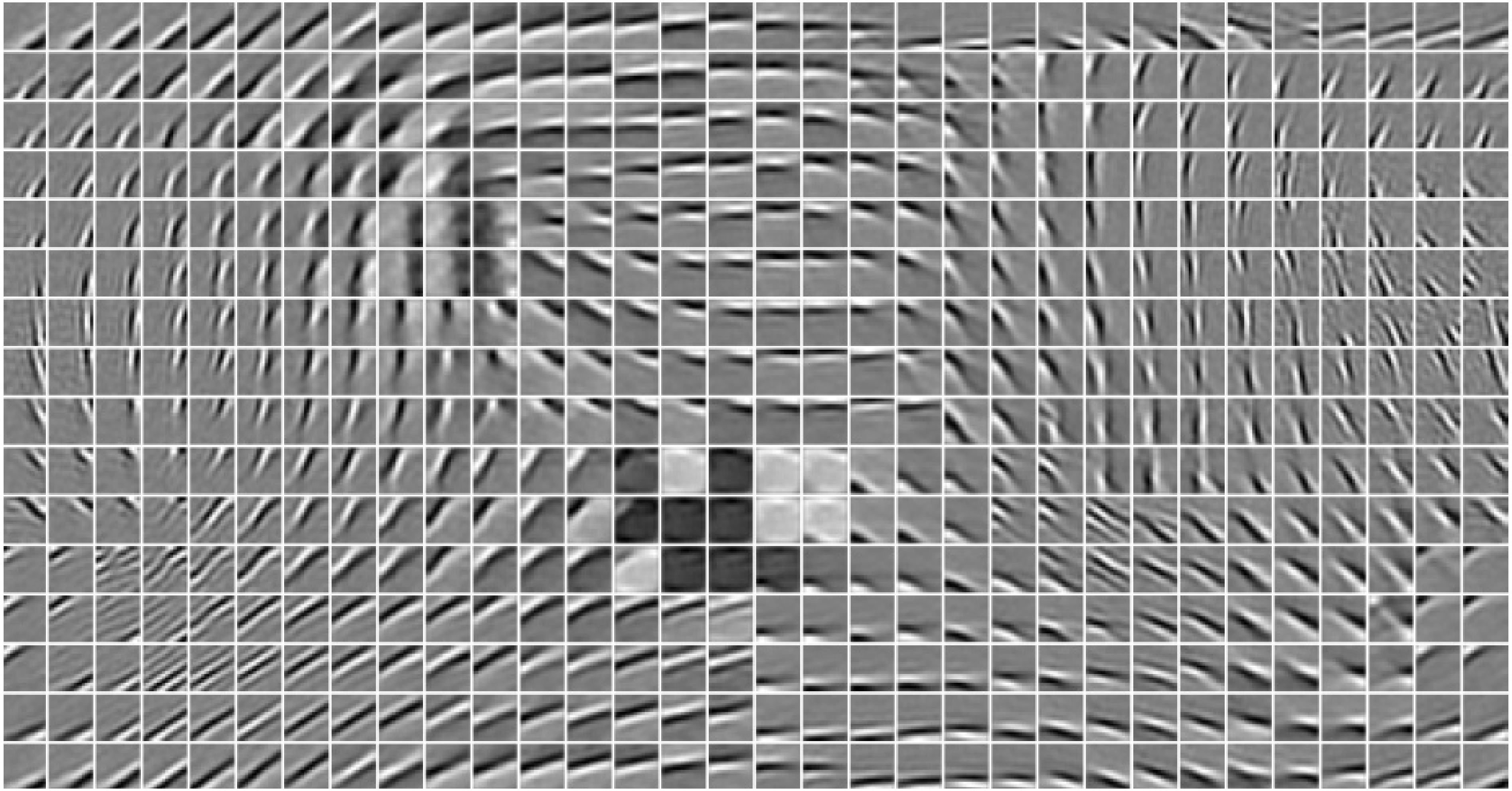
Tensor Structures

- Cortical Neural Networks

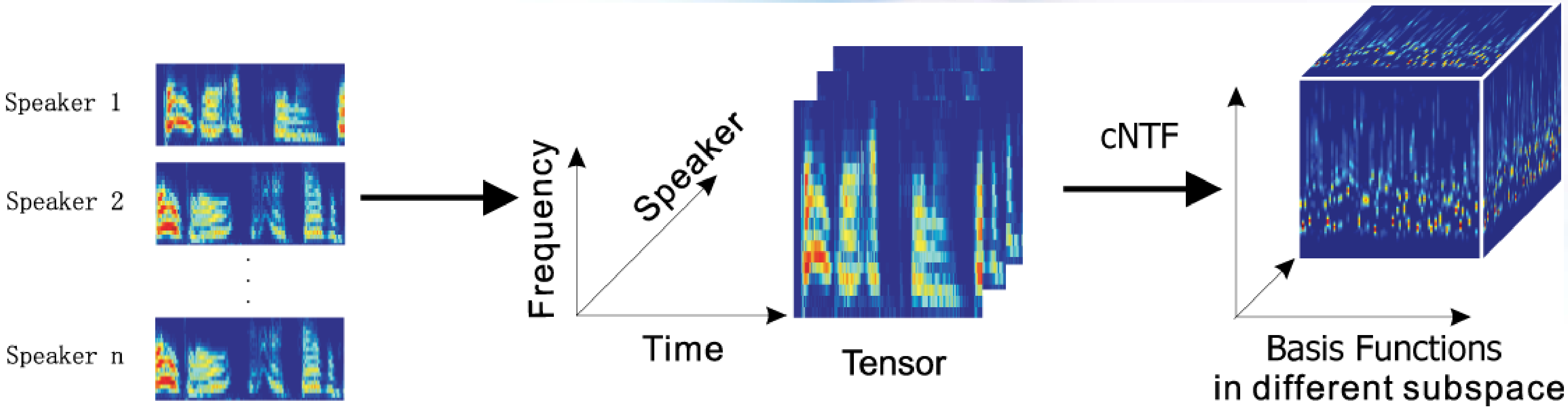




Receptive Field



Speech Tensor Representation



In order to extract robust features based on tensor structure, we model the cochlear power feature of different speakers as 3-order tensor $\mathcal{X} \in \mathbb{R}^{N_f \times N_t \times N_s}$.

Each feature tensor is an array with three modals

frequency $\times$ time $\times$ speaker identity

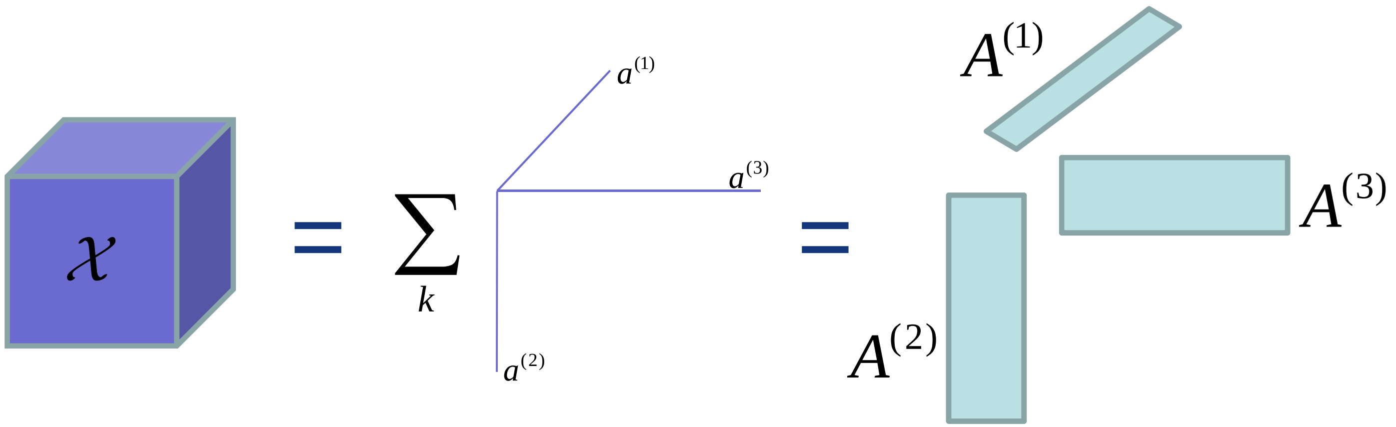
Matrix Factorization-Extension

- Matrix Factorization

$$\mathcal{A} = \sum_{r=1}^{R} \lambda_r v_r v_r^T$$

- PARAFAC Model

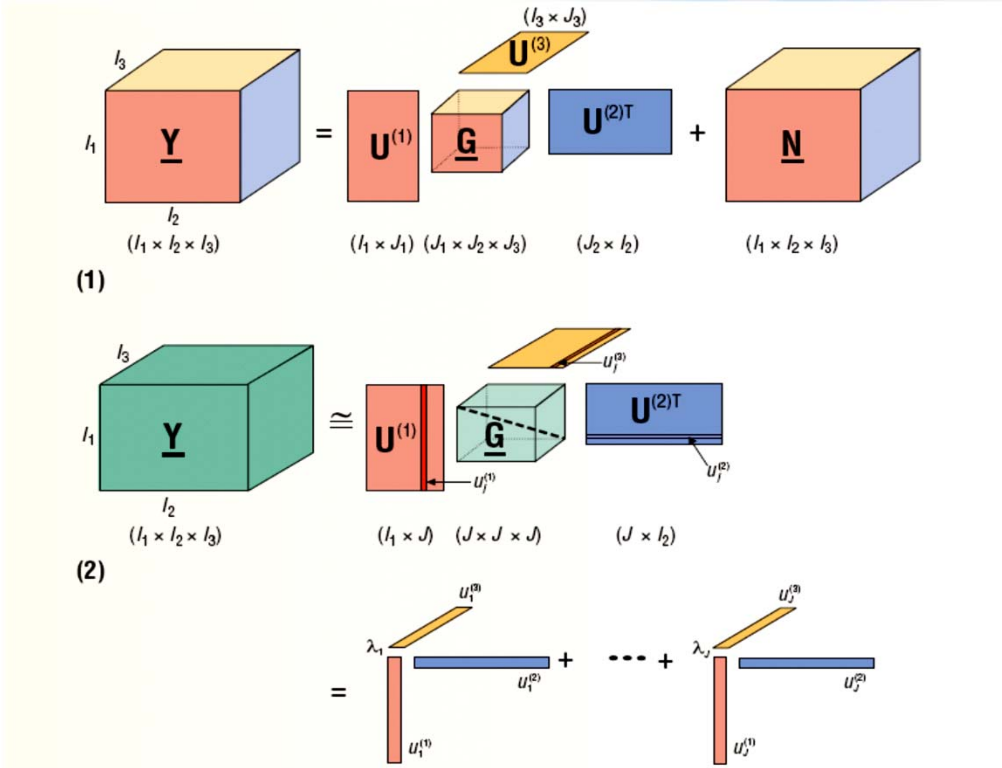
$$\mathcal{X} = \sum_{r=1}^{R} \lambda_r a_r^{(1)} \otimes a_r^{(2)} \otimes \cdots \otimes a_r^{(M)}$$



Tensor Decomposition

Tucker

$$\mathcal{Y} = \mathcal{G} \times_1 U^{(1)} \times_2 U^{(2)} \cdots \times_n U^{(n)} + \mathcal{N}$$



PARAFAC:

$$\mathcal{Y} = \sum_{r=1}^{R} \lambda_r u_r^{(1)} \otimes u_r^{(2)} \otimes \cdots \otimes u_r^{(n)} = \Lambda \times_1 U^{(1)} \times_2 U^{(2)} \cdots \times_n U^{(n)}$$

Supervised Learning

- Supervised Learning

Observations: $\mathbf{x}_i \Rightarrow y_i, \quad i = 1, 2, \cdots, N$

Objective: To find some unknown function f, such that

$$y_i = f(\mathbf{x}_i), \quad i = 1, 2, \cdots, N$$

- Regression Problem:

To define some parametric models

$$y = f_\theta(\mathbf{x}) + \varepsilon$$

where ε is a model error term, and usually assume Gaussian distributed. The parameters θ is identified via maximum likelihood or maximum posterior. Typical eaxmple is the linear regression model

$$f_\theta(\mathbf{x}) = \sum_{k=1}^{K} \alpha_k \phi_k(\mathbf{x}), \quad \boldsymbol{\theta} = (\alpha_1, \alpha_2, \cdots \alpha_K)^T.$$

Supervised Learning

- Supervised Learning

  Observations: $\mathbf{x}_i \Rightarrow y_i, \quad i = 1, 2, \cdots, N$

  Objective: To find some unknown function f, such that

  $$y_i = f(\mathbf{x}_i), \quad i = 1, 2, \cdots, N$$

- General Approach

  To infer a prabability function $p(f \mid \mathbf{X}, \mathbf{y})$, given the data $(\mathbf{X}, \mathbf{y})$.

  To predict

  $$p(y_* \mid x_*, \mathbf{X}, \mathbf{y}) = \int p(y_* \mid f, x_*) p(f \mid \mathbf{X}, \mathbf{y}) df$$

  Gaussian Process infer $p(f \mid \mathcal{D})$

  Parametric Model infer $p(\theta \mid \mathcal{D})$

BCMI Brain-like Computing & Machine Intelligence

Gaussian Processes

- **Problem:** It is difficult to represent a distribution over a function
- **Solution:** To define a distribution over the function's values at a finite, but arbitrary
- **Gaussian Processes:**
  - ✓ Definition: A GP is a collection of random variables, any finite number of which have joint Gaussian Distribution

Data set: $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$

Random Variables: $\{f(\mathbf{x}_1), f(\mathbf{x}_2), \cdots, f(\mathbf{x}_N)\}$

Assume $p(f(\mathbf{x}_1), f(\mathbf{x}_2), \cdots, f(\mathbf{x}_N))$ is jointly Gaussian

with mean $\mu(\mathbf{x})$ and covariance $\Sigma(\mathbf{x})$, given by $\Sigma_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$.

GP for Regression

Training data set: $\{(\mathbf{x}_i, y_i), i = 1, 2, \cdots, N\}, y_i = f(\mathbf{x}_i)$

Given a test set $\mathbf{X}_*$ of size $N_* \times d$,

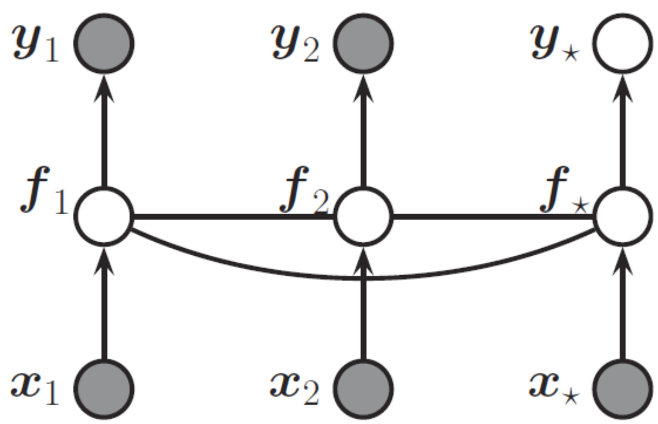
To predict the function outputs $f_*(\mathbf{X}_*)$

Consider the joint distribution in the following form

$$\begin{pmatrix} f \\ f_* \end{pmatrix} \prec \mathcal{N}\left(\begin{pmatrix} \mu \\ \mu_* \end{pmatrix}, \begin{pmatrix} K & K_* \\ K_*^T & K_{**} \end{pmatrix}\right)$$

where $K = \kappa(X, X)$ is $N \times N, K_* = \kappa(X, X_*)$ is $N \times N_*$,

$K_{**} = \kappa(X_*, X_*)$ is $N_* \times N_*$



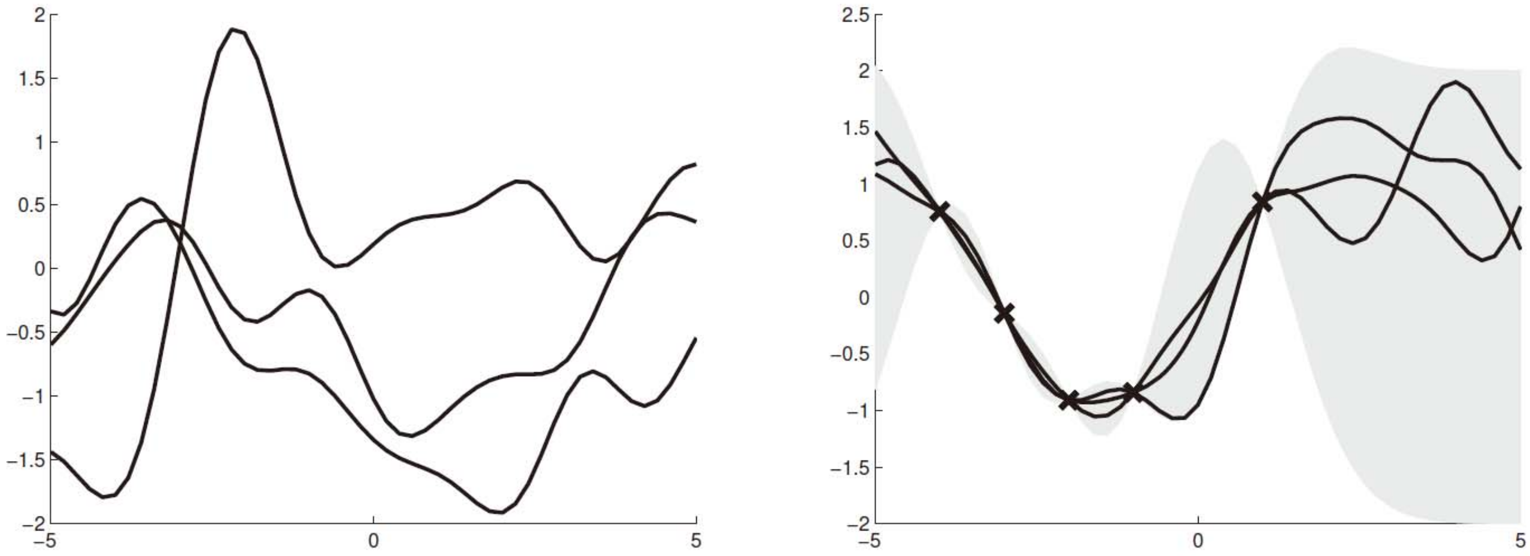
GP for Regression(II)

- By the standard rules of the conditioning Gaussians, the posterior has the following form

$$p(f_* \mid X_*, X, f) = \mathcal{N}\left(f_* \mid \mu_*, \Sigma_*\right),$$

$$\mu_* = \mu(X_*) + K_*^T K^{-1}(f - \mu(X)),$$

$$\Sigma_* = K_{**} - K_*^T K^{-1} K_* .$$



(a) (b)

GP Prediction (noisy)

The noisy observation model is

$$y = f(x) + \varepsilon, \qquad \varepsilon \sim \mathcal{N}\left(0, \sigma_y^2\right),$$

The covariance of the noisy observations

$$\mathrm{cov}(y_p, y_q) = \kappa(x_p, x_q) + \sigma_y^2 \delta_{pq},$$

$$\mathrm{cov}[\mathbf{y} \mid \mathbf{X}] = \mathbf{K} + \sigma_y^2 \mathbf{I}_N \triangleq \mathbf{K}_y$$

The joint probability function is given by

$$\begin{pmatrix} y \\ f_* \end{pmatrix} \prec \mathcal{N}\left(0, \begin{pmatrix} K_y & K_* \\ K_*^T & K_{**} \end{pmatrix}\right)$$

The posterior density function is

$$p(f_* \mid X_*, X, f) = \mathcal{N}\left(f_* \mid \mu_*, \Sigma_*\right),$$

$$\mu_* = K_*^T K_y^{-1} f, \qquad \Sigma_* = K_{**} - K_*^T K_y^{-1} K_* .$$

GP Prediction (noisy) (2)

The posterior density function is

$$p(f_* \mid X_*, X, f) = \mathcal{N}\left(f_* \mid \mu_*, \Sigma_*\right),$$

$$\mu_* = K_*^T K_y^{-1} y, \qquad \Sigma_* = K_{**} - K_*^T K_y^{-1} K_*.$$

In particular, $d = 1$, we have

$$p(f_* \mid X_*, X, f) = \mathcal{N}\left(f_* \mid K_*^T K_y^{-1} y, k_{**} - k_*^T K_y^{-1} k_*\right),$$

where $k_* = [\kappa(x_*, x_1), \cdots, \kappa(x_*, x_N)], k_{**} = \kappa(x_*, x_*)$. The posterior mean

$$\overline{f}_* = K_*^T K_y^{-1} y = \sum_{i=1}^{N} \alpha_i \kappa(x_i, x_*)$$

where $\alpha = K_y^{-1} y$.

GP for Tensor Variate

Classification Problem:

M-th Order Tensor $\mathcal{X}_n \in \mathbb{R}^{I_1 \times \cdots \times I_M}$, n=1,2,$\cdots$,N

Classes Label $y_n \in \{1, 2, \cdots, C\}$,

Latent function $f_n = (f_n^1, f_n^2, \cdots, f_n^C)^T = f(\mathcal{X}_n)$

Denote

$$\mathcal{X} = [\mathcal{X}_1 \ \mathcal{X}_2 \cdots \mathcal{X}_N],$$

$$\mathbf{f} = [f_1^1, f_2^1, \cdots, f_N^1, f_1^2, \cdots, f_N^2, \cdots\cdots, f_1^C, \cdots, f_N^C]^T$$

Prediction:

$$p(f_* \mid \mathcal{X}_*, \mathcal{X}, y)$$

Gaussian Prior with zero mean:

$$p(f \mid \mathcal{X}) = \mathcal{N}(0, K)$$

where K is a $CN \times CN$ blocked diagonal covariance matrix

$$K = diag(K^1, K^2, \cdots, K^C),$$

$K_{ij}^c = k(\mathcal{X}_i, \mathcal{X}_j) = \mathrm{cov}(f_i^c, f_j^c)$ with the class c.

Typical Covariance function

$$k(\mathcal{X}_i, \mathcal{X}_j \mid \Theta) = \sigma^2 \exp\left(-\frac{1}{2l^2} \left\langle \mathcal{X}_i, \mathcal{X}_j \right\rangle_2 \right),$$

where $\Theta = \{\sigma^2, l\}$.

Observation model: The multinomial probit

$$p(y_i \mid f_i) = E_{u_i}\left[\prod_{j=1, j \neq y_i}^{C} \Phi(u_i + f_i^{y_i} - f_i^{j}) \right]$$

where Φ is the cumulative density function of the standard normal distribution, and the auxiliary variable u_i is distributed as $\mathcal{N}(0,1)$.

GP for Tensor Variate

The conditional posterior distribution

$$p(f \mid \mathcal{D}, \Theta) = \frac{1}{Z} p(f \mid \mathcal{X}, \Theta) \prod_{n=1}^{N} p(y_n \mid f_n)$$

where $Z = \int p(f \mid \mathcal{X}, \Theta) \prod_{n=1}^{N} p(y_n \mid f_n) df$ is known as the marginal likelihood

The observation model results in an analytically intractable posterior distribution and approximate methods are needed for integration over latent variables

Probabilistic Product Kernels for Tensors

Commonly used kernels in vectors

Linear kernel: $k(\mathcal{X},\mathcal{X}^{'})=\left\langle vec(\mathcal{X}),vec(\mathcal{X}^{'})\right\rangle$,

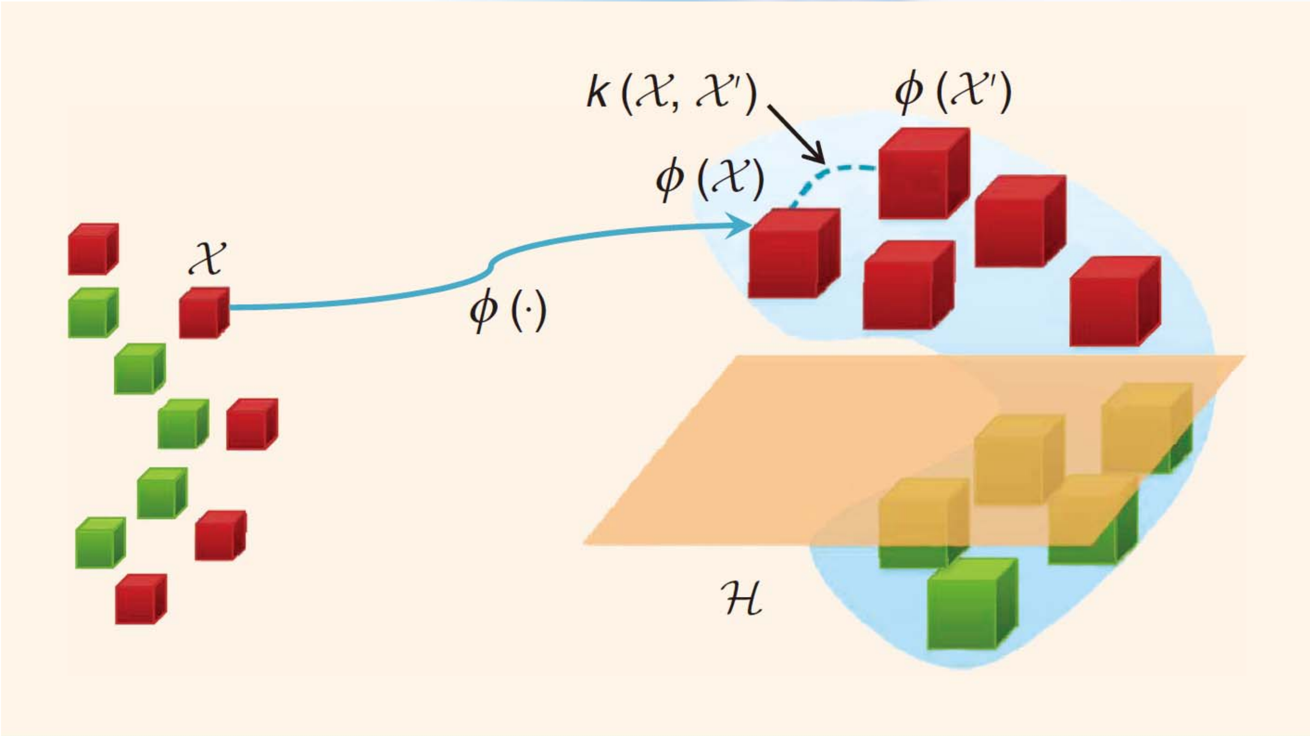
Gaussian-RBF $k(\mathcal{X},\mathcal{X}^{'})=\exp\left(-\frac{1}{2\beta^2}\left\|\mathcal{X}-\mathcal{X}^{'}\right\|_F^2\right).$

- Tensor based Kernels

$\mathcal{X}_n \in R^{I_1\times I_2\cdots\times I_M}$, $\mathrm{X}_n^{(m)}$ is its mode-m matriczation, which is considered as an ensemble of I_m-dim multivarate with $I_1\times\cdots\times \hat{I}_2\cdots\times I_M$ samples, generated from $p(x\,|\,\lambda_n^m)$

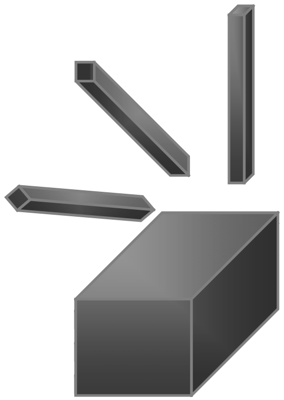
AAAI2013, Zhao et al, IEEE SPM 2013

Similarity Measure

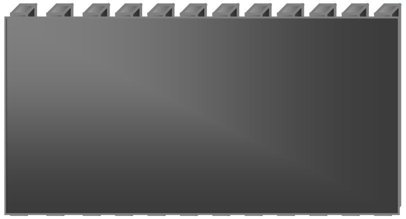


Tensor observations are mapped into RKHS space H by a nonlinear mapping function. The kernel function is particularly defined as a similarity measure between two tensors.

Matricization



$\mathcal{V}$



$$\mathbf{V}_{(3)} \approx \mathbf{A}^{(3)}\mathbf{Z}^{(3)}$$

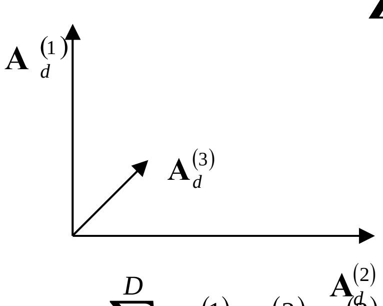
$$\mathbf{Z}^{(3)} = \left(\mathbf{A}^{(2)} | \otimes | \mathbf{A}^{(1)}\right)^T$$

$$\mathbf{V}_{(1)} \approx \mathbf{A}^{(1)}\mathbf{Z}^{(1)}$$

$$\mathbf{Z}^{(1)} = \left(\mathbf{A}^{(3)} | \otimes | \mathbf{A}^{(2)}\right)^T$$

$$\mathbf{V}_{(2)} \approx \mathbf{A}^{(2)}\mathbf{Z}^{(2)}$$

$$\mathbf{Z}^{(2)} = \left(\mathbf{A}^{(3)} | \otimes | \mathbf{A}^{(1)}\right)^T$$



$$\mathcal{V}_{i_1 i_2 i_3} \approx \sum_{d=1}^{D} \mathbf{A}_{i_1 d}^{(1)} \mathbf{A}_{i_2 d}^{(2)} \mathbf{A}_{i_3 d}^{(3)}$$

Probabilistic Product Kernels for Tensors

Mode-m similarity measure

$$S_m(\mathcal{X}\|\mathcal{X}^{'}) = sKL\left(p(x\,|\,\lambda_{\mathcal{X}}^{m})\middle\|q(x\,|\,\lambda_{\mathcal{X}^{'}}^{m})\right)$$

where p,q represent mode-m probabilty density function for $\mathcal{X}$ and $\mathcal{X}^{'}$. Therefore, the probabilty kernel for tensors is given by

$$k(\mathcal{X}\|\mathcal{X}^{'}) = \alpha^2 \prod_{m=1}^{M} \exp\left(-\frac{1}{2\beta_m^2} S_m(\mathcal{X}\|\mathcal{X}^{'})\right)$$

where α denotes a magnitude parameter and $[\beta_1,\cdots,\beta_M]$ play the role of width-scales, which are identified by automatic relevance determination(ARD).

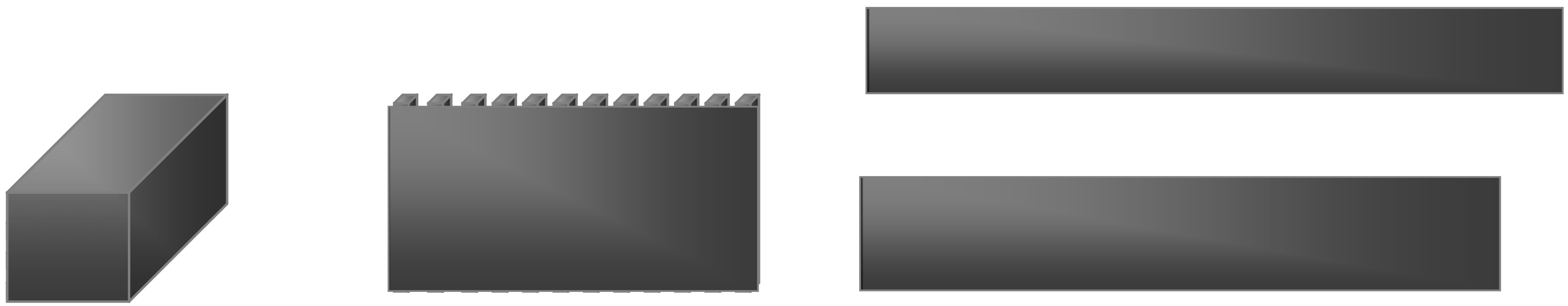
Denote parameter set $\Theta = \left\{\alpha, \beta_m \,|\, m = 1,\cdots,M\right\}$

AAAI2013, Zhao et al, IEEE SPM 2013

Generalization

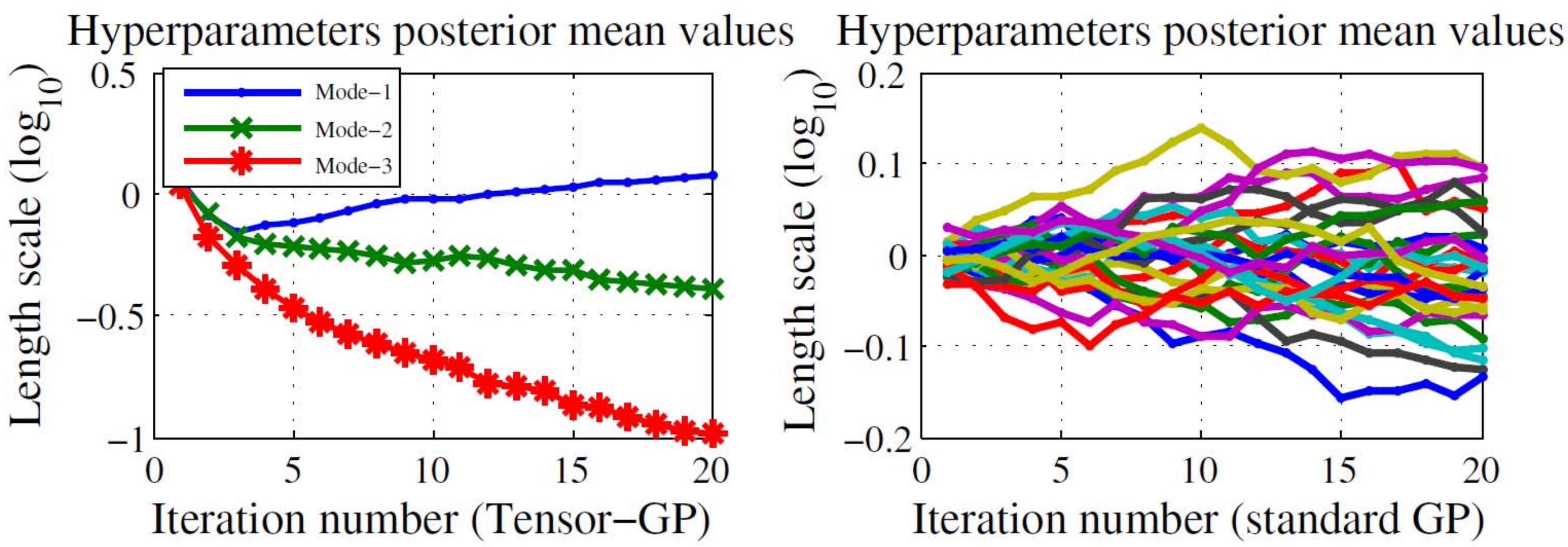
Assume $p(x \mid \lambda^m)$ is Gaussian, the model parameter $\lambda^m = \{\mu_m, \Sigma_m\}$ can be easily estimated from $\mathrm{X}_n^{(m)}$, the mode-m matricization of $\mathcal{X}$.

- Much less parameters, better generalization



$\mathcal{X} \in \mathbf{R}^{L \times M \times N}$ $\quad$ $X_1 \in \mathbf{R}^{L \times MN}$

Simulations

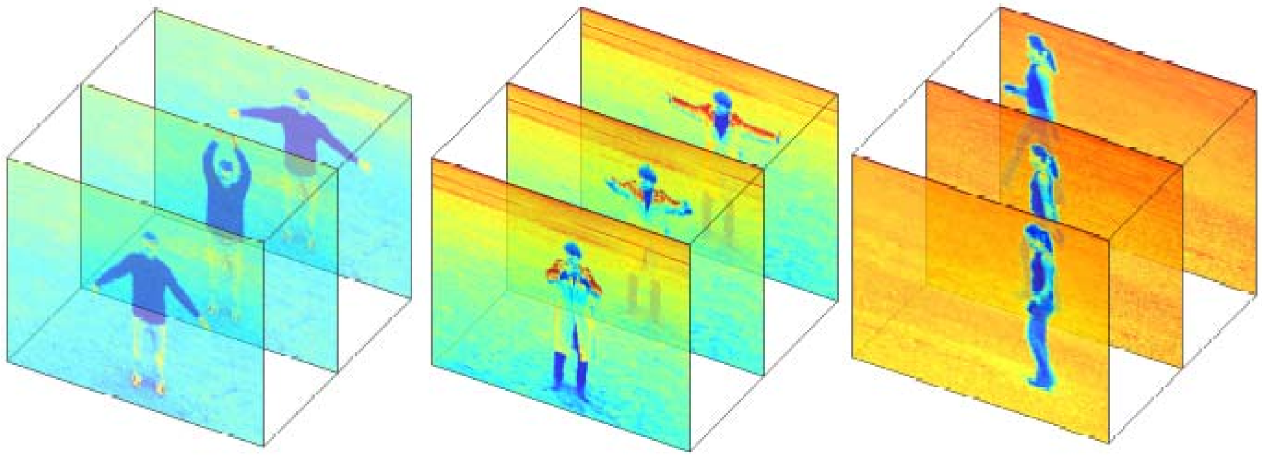


(Artificial Data) Evolution of estimated posterior means for the inverse squared length scale hyper-parameters on a dataset generated by CP model.

AAAI2013, Zhao et al, IEEE SPM 2013

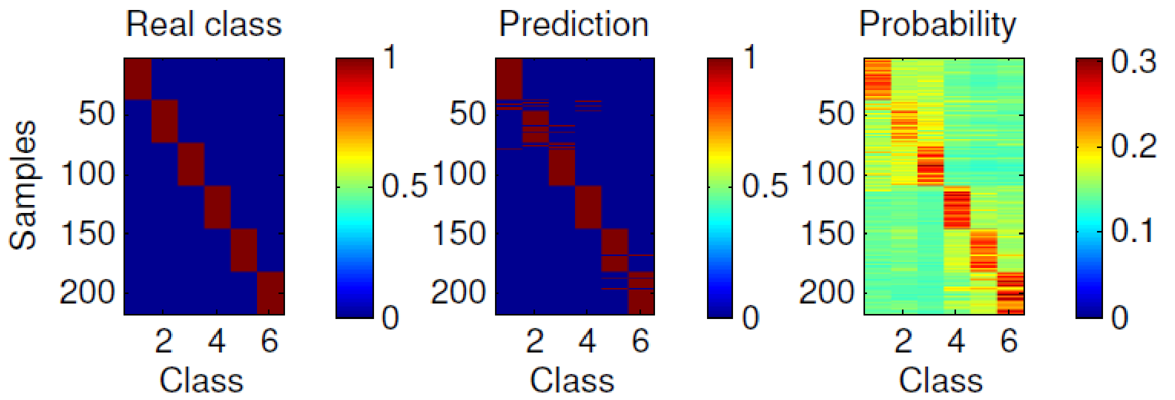
Apps: Visual Action Classification

- Data Preprocessing
  - ✓ Each video is space-time aligned and uniformly resized to 20 X 20 X 32, which are then be represented by a third-order tensor X
  - ✓ 16 person videos for training, 9 person videos for test



Three examples of video sequences for hand waving, hand clapping and walking actions

Simulation Results



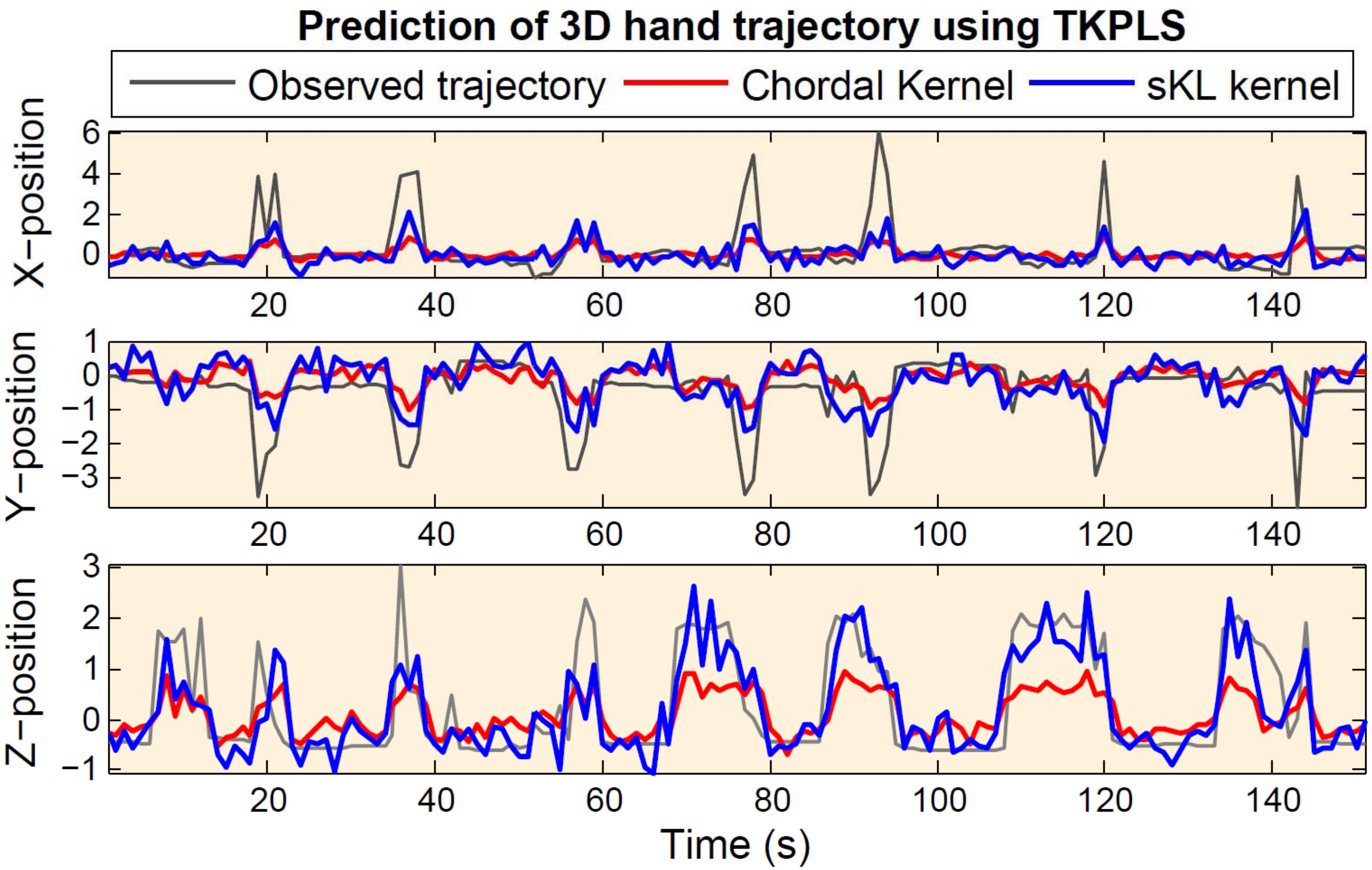
Classification results and probability of predictions on the test set.

Table 1: Confusion matrix (average accuracy 94%)

	Walk	Run	Jog	Box	H-C	H-W
Walk	1.0	0	0	0	0	0
Run	.08	.78	.06	.08	0	0
Jog	.03	.03	.94	0	0	0
Box	0	0	0	1.0	0	0
H-C	0	0	0	0	.98	.02
H-W	0	0	0	0	.08	.92

AAAI2013, Zhao et al, IEEE SPM 2013

Applications- ECoG Decoding



Decoding of 3D movement trajectories from ECoG

ICASSP2013, Zhao et al

Outline

- Background and Motivation
- Tensor Factorization/Decomposition
- Gaussian Processes
- Probabilistic Kernels for Tensors
- Constrained Tensor Factorization
  - ✓ Nonnegative Tenor Factorization
  - ✓ Discrimitive Tensor Factorization
- Multilinear regression
- Conclusions and Perspectives

Nonnegative Tensor Factorization (cNTF)

- Cost Function
  - ✓ Least Square

$$J_{LS}(A^{(d)}) = \sum_{d=1}^{M}\left(\frac{1}{2}\sum_{p=1}^{N_d}\sum_{q=1}^{N_{\bar{d}}}\left([X_{(d)}]_{pq} - [A^{(d)}SZ^{(d)}]_{pq}\right)^2 + \alpha\sum_{p\neq q}[A^{(d)T}A^{(d)}]_{pq}\right)$$

  - ✓ K-L Divergence

$$J_{KL}(A^{(d)}) = \sum_{d=1}^{M}\left(\sum_{p=1}^{N_d}\sum_{q=1}^{N_{\bar{d}}}\left([X_{(d)}]_{pq}\log\frac{[X_{(d)}]_{pq}}{[A^{(d)}SZ^{(d)}]_{pq}} - [X_{(d)}]_{pq} + [A^{(d)}SZ^{(d)}]_{pq}\right)\right.$$
$$\left. + \alpha\sum_{p\neq q}[A^{(d)T}A^{(d)}]_{pq}\right)$$

Constrained Nonnegative Tensor Factorization (cNTF)

- **Update Rules**
  - ✓ Least Squared Error

$$A_{ij}^{(d)} \leftarrow A_{ij}^{(d)} \frac{[X_{(d)} Z^{(d)T} S^{T}]_{ij}}{[A^{(d)} S Z^{(d)} Z^{(d)T} S^{T}]_{ij} + \alpha \sum_{p \neq j} [A^{(d)T}]_{pi}}$$

  - ✓ K-L Divergence

$$A_{ij}^{(d)} \leftarrow A_{ij}^{(d)} \frac{\sum_{k} [S Z^{(d)}]_{jk} \frac{[X_{(d)}]_{ik}}{[A^{(d)} S Z^{(d)}]_{ik}}}{\sum_{k} [S Z^{(d)}]_{jk} + \alpha \sum_{p \neq j} [A^{(d)T}]_{pi}}$$

Discriminative model

- Logistic Regression Model

$$\log \frac{p(y=+1 \mid X)}{p(y=-1 \mid X)} = f(X,\theta) = X \prod_{d=1}^{m} \times_d w_d + b$$

- Objective Function:

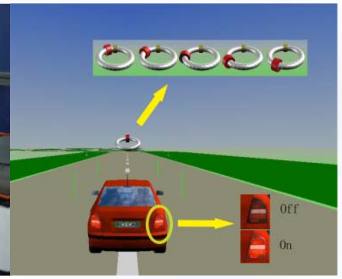
$$\min_{w_d|_{d=1}^{m},b} \sum_{n=1}^{N} \log(1+\exp(-y_n f(X_n,\theta))) + \sum_{d=1}^{m} \left\{ \begin{array}{l} \lambda_d^1 \|w_d\|_2^2 + \lambda_d^2 \|w_d\|_1 \\ -\lambda_d^3 w_d^T K_d w_d \end{array} \right\}$$

K evaluates the degree of correlation between two samples according to their distance

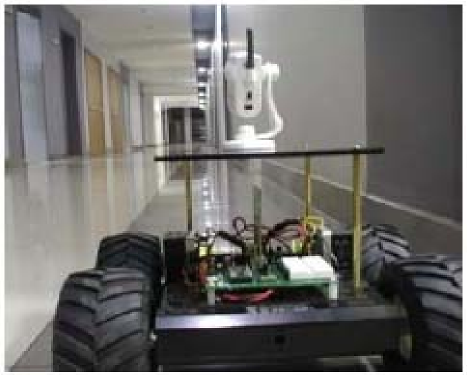
BCMI
Brain-like Computing
&
Machine Intelligence

Brain Computer Interface

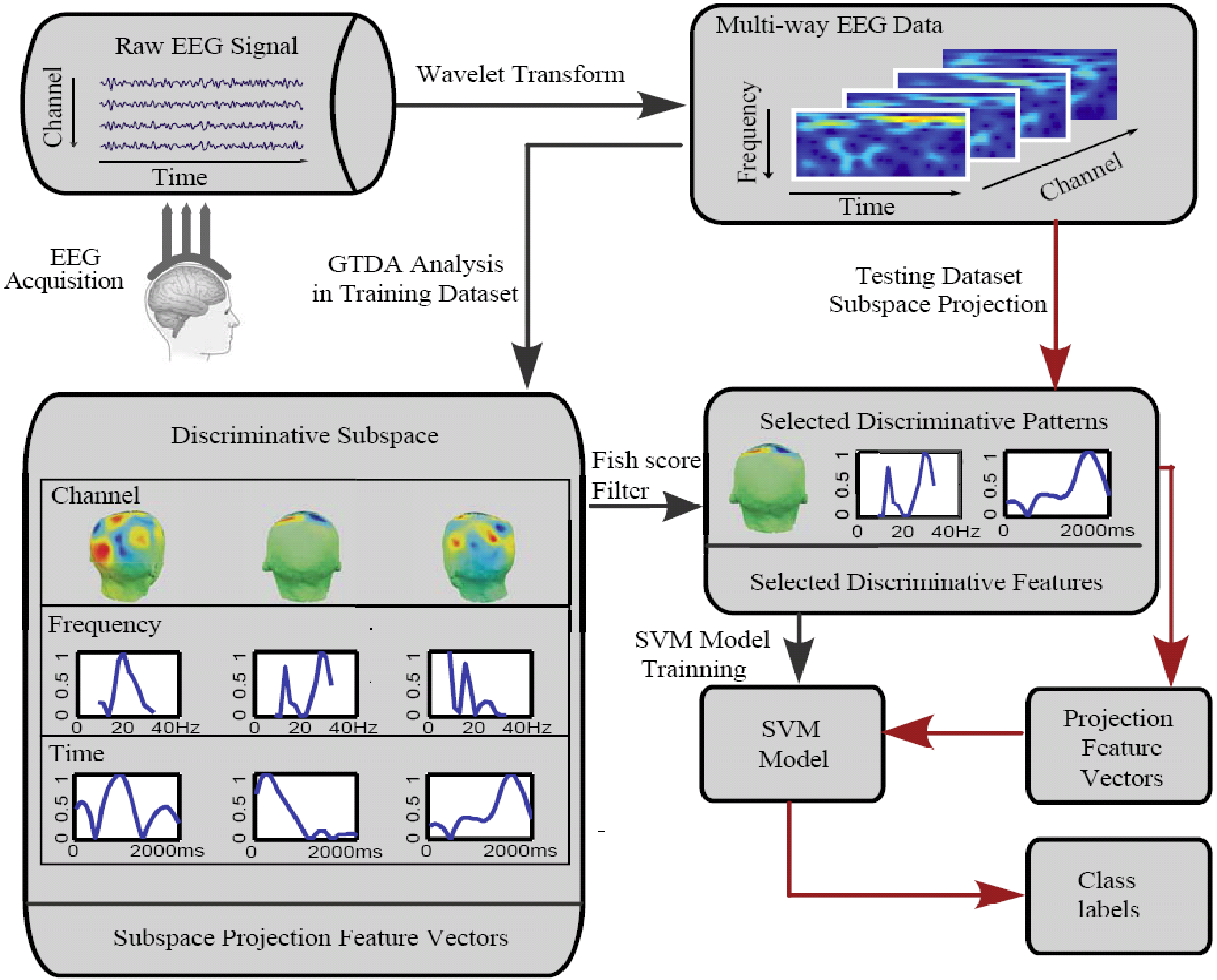
- BCI Car-Driving Systems
- BCI Wheelchair System
- BCI Remote Control System
- BCI based Rehabilitation
- BCI based Vigilance Detection



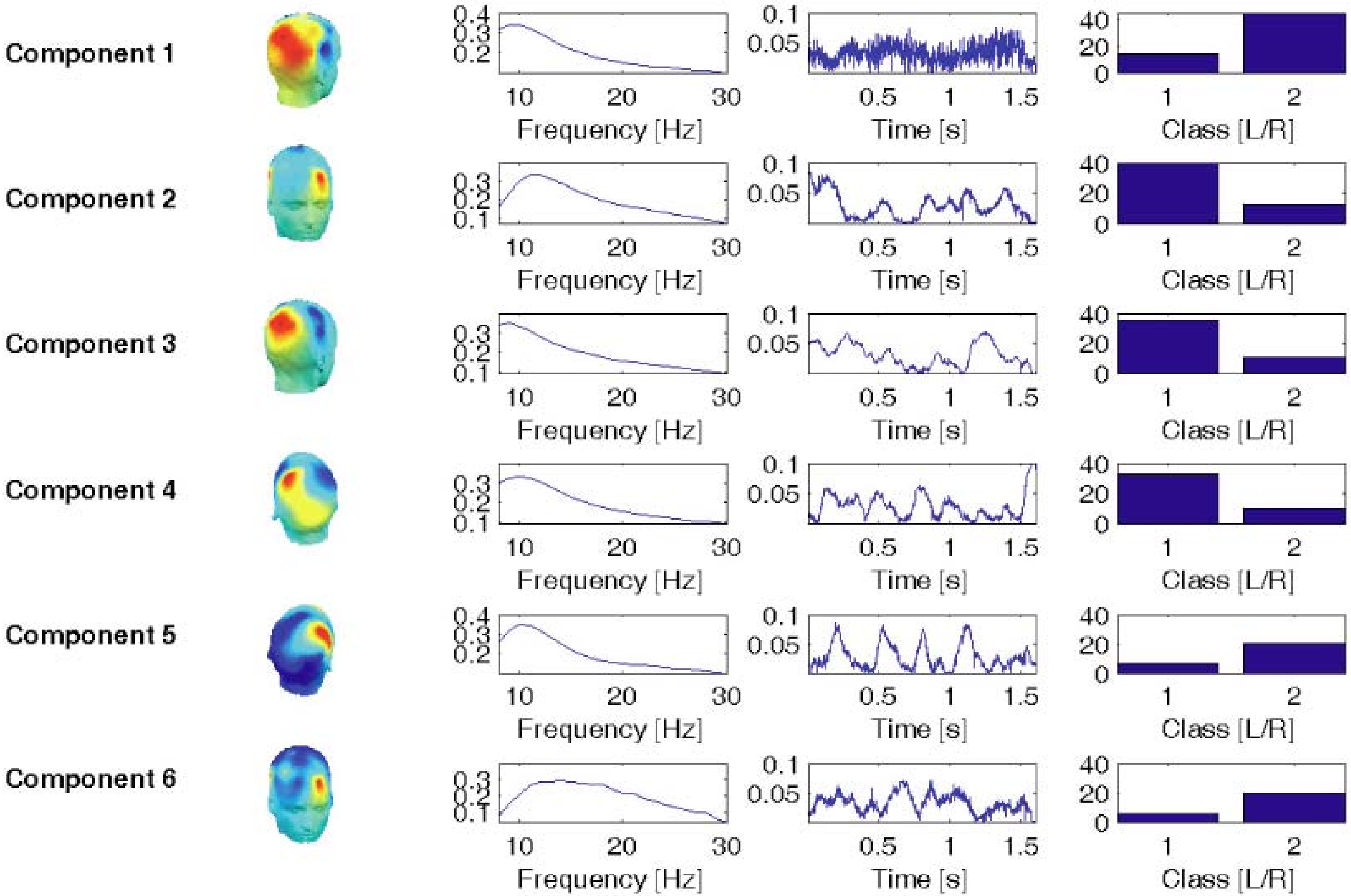
BCI Tasks



Tensor Feature Extraction



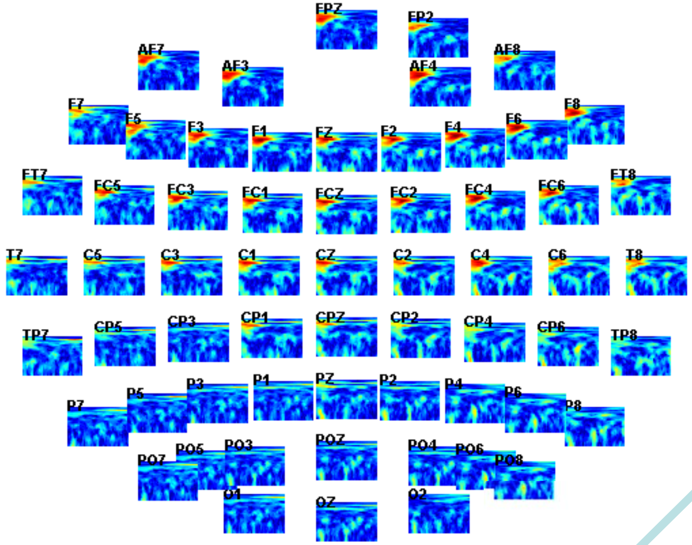
Tensor Feature Extraction



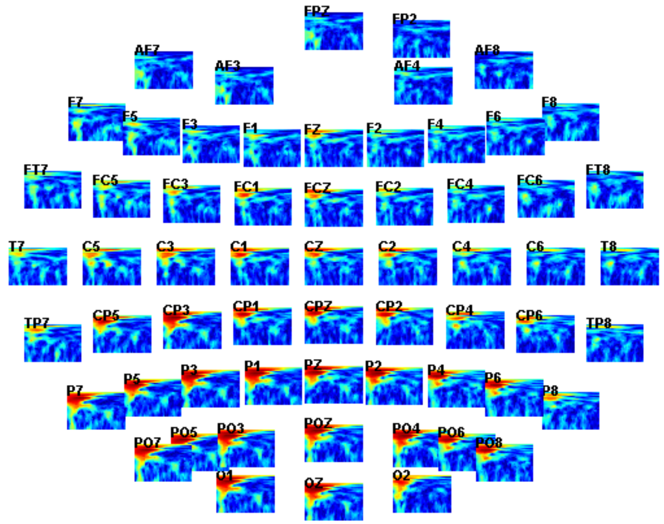
Li & Zhang, IEEE NSRE 2009

BCMI
Brain-like Computing & Machine Intelligence

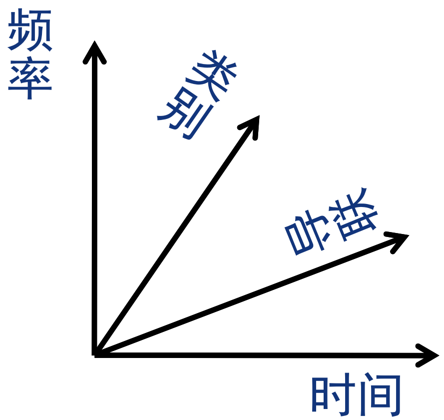
Apps: 视听觉刺激诱发电位

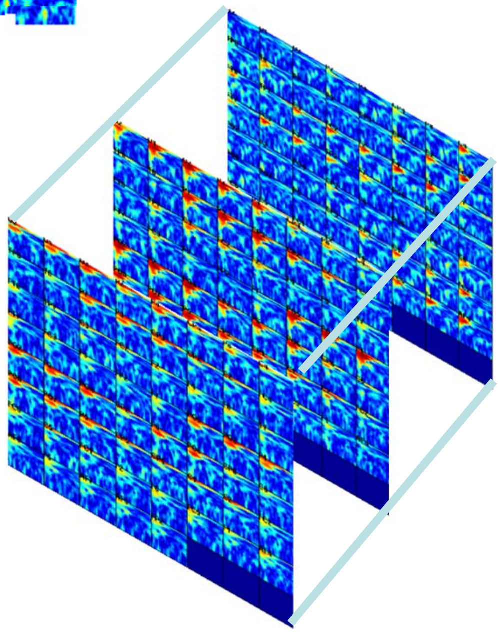


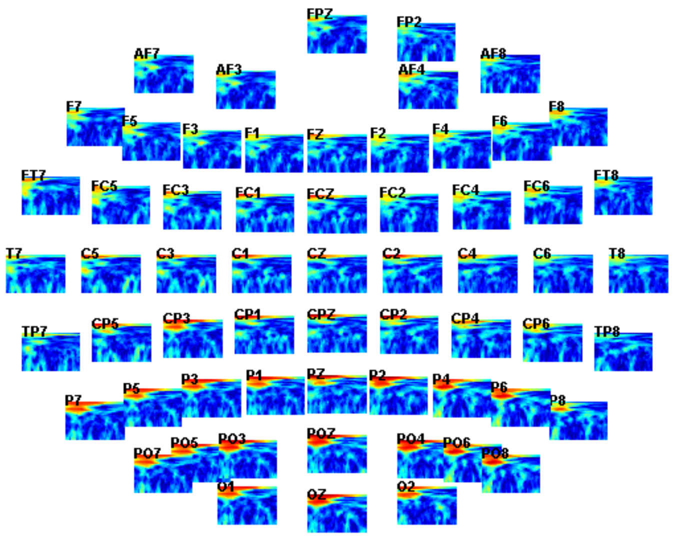
ITPC - 声音刺激



ITPC – 视觉刺激







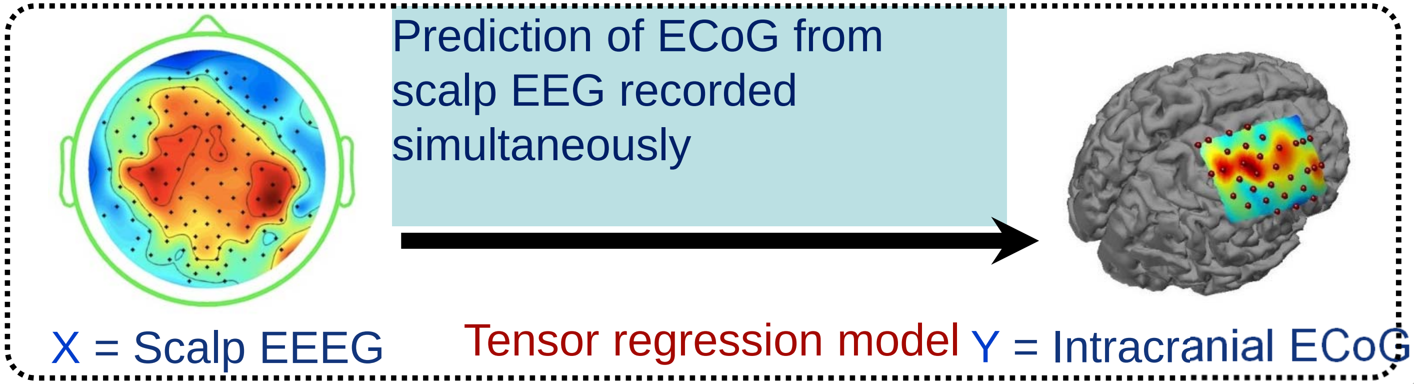
ITPC – 声音+视觉 stimulus

Outline

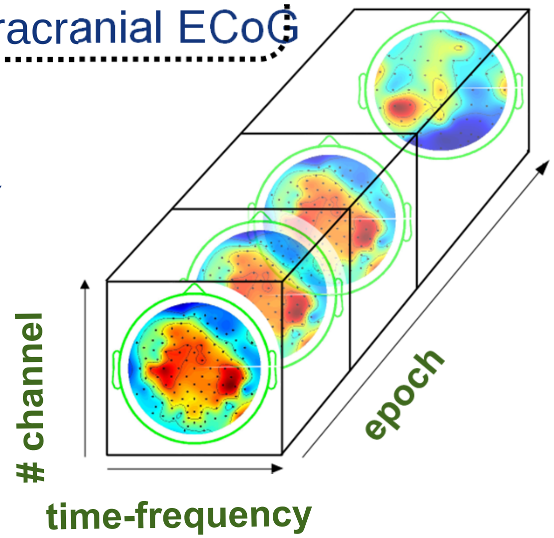
- Background and Motivation
- Tensor Factorization/Decomposition
- Gaussian Processes
- Probabilistic Kernels for Tensors
- Constrained Tensor Factorization
  - ✓ Nonnegative Tenor Factorization
  - ✓ Discriminative Tensor Factorization
- Multilinear regression
- Conclusions and Perspectives

Multilinear regression and applications

- **Tensor representation of multidimensional data**
  - EEG, ECoG (spatial, temporal, frequency, epoch,...)
  - Physical meaning - ease of interpretation
- **From multivariate to multi-way array processes - partial least squares (PLS)**

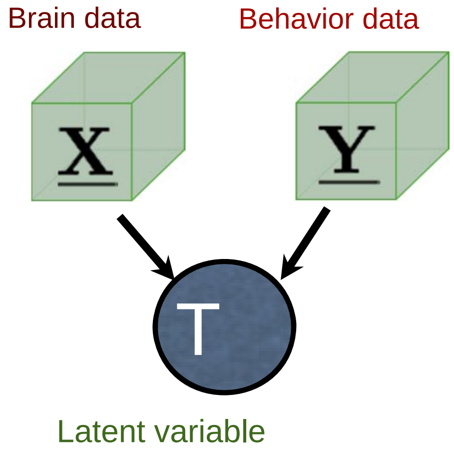


- **Standard PLS applied on matricization of both X and Y**
- Small sample size problem
- Overfitting problem (high dimension of subspace basis)
- Lack of physical interpretation for loadings



NIPS2011 PAMI2013

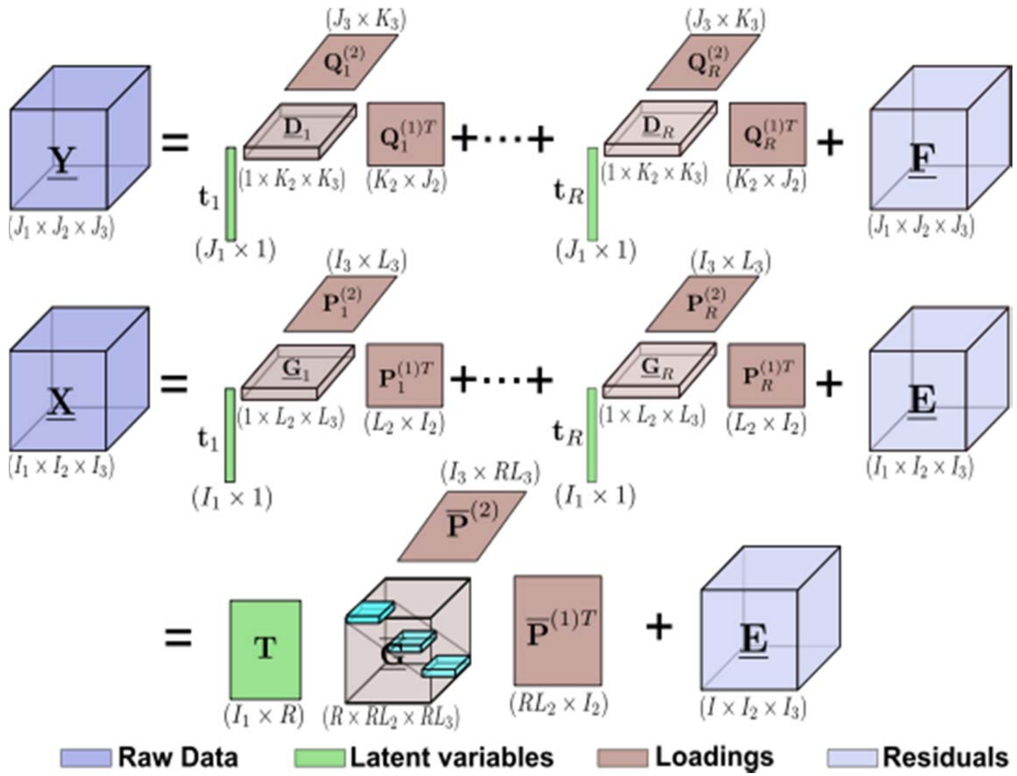
Proposed approach



Objective function

$$\min_{\{\mathbf{P}^{(n)},\mathbf{Q}^{(m)}\}} \left\| \underline{\mathbf{X}} - [\![\underline{\mathbf{G}}; \mathbf{t}, \mathbf{P}^{(1)}, \ldots, \mathbf{P}^{(N-1)}]\!] \right\|^2 + \left\| \underline{\mathbf{Y}} - [\![\underline{\mathbf{D}}; \mathbf{t}, \mathbf{Q}^{(1)}, \ldots, \mathbf{Q}^{(M-1)}]\!] \right\|^2$$

$$\text{s. t.} \quad \{\mathbf{P}^{(n)T}\mathbf{P}^{(n)}\} = \mathbf{I}_{L_{n+1}}, \quad \{\mathbf{Q}^{(m)T}\mathbf{Q}^{(m)}\} = \mathbf{I}_{K_{m+1}},$$



Extension of PLS to higher-order tensor data - HOPLS

- Goal: to predict a tensor Y from a tensor X
- Approach: to extract the common latent variables

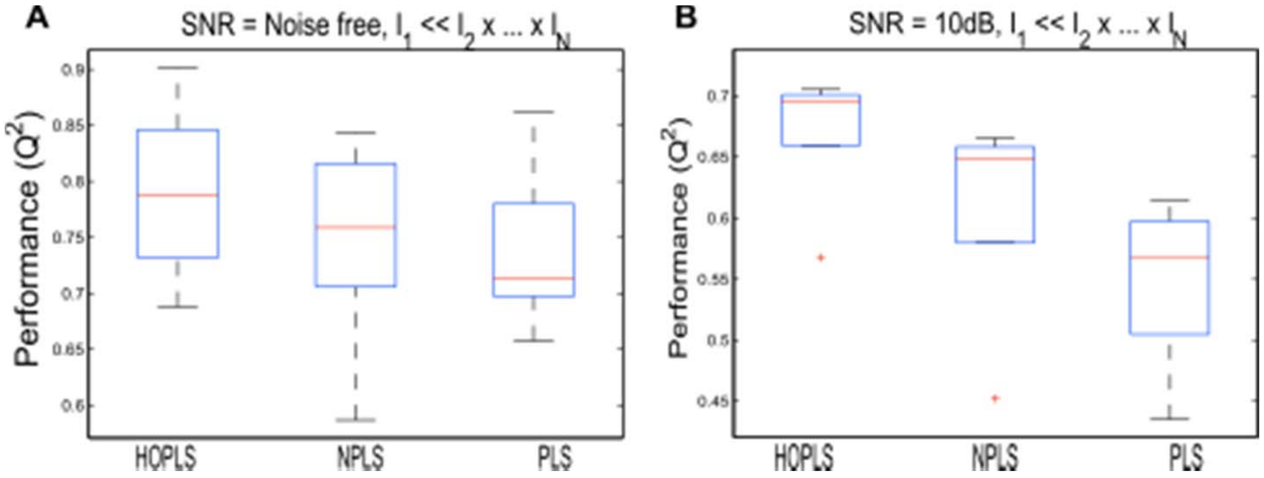
Properties:

- Flexible multilinear regression framework
- Projection on tensor subspace basis
- Efficient optimization algorithm using HOOI on the *n*-mode cross-covariance tensor

NIPS2011 PAMI2013

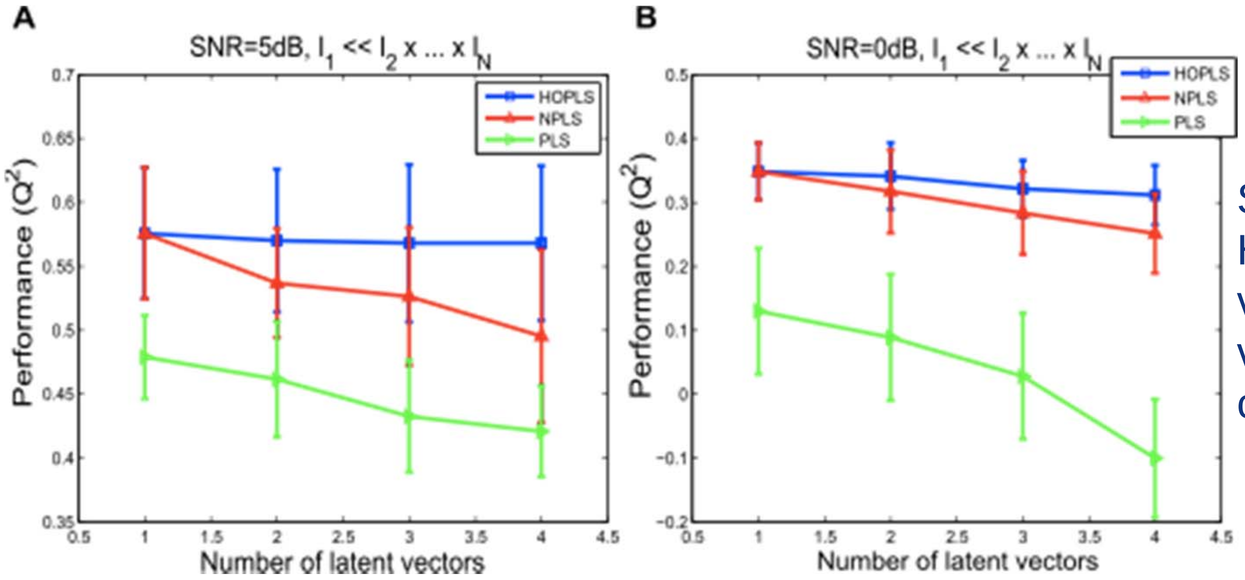
Key advantages

Small sample size



HOPLS: better prediction performance and enhanced robustness to noise

Robustness against overfitting and noise



Stability of the performance of HOPLS, NPLS and PLS for a varying number of latent vectors under different noise conditions

NIPS2011, Zhao et al

Conclusions and Perspectives

- New Kernelization for Tensor Data
- Discriminative Tensor Feature Extraction
- Multilinear PLS for Tensor Data
- Perspectives:
  - ✓ Theory on Tensor Decomposition
  - ✓ Algorithms for Tensor Features
  - ✓ Fast Algorithms for Tensor Operations
  - ✓ Dynamical Tensor Features

BCMI

Brain-like Computing & Machine Intelligence

Acknowledgements

- Prof . Baoliang Lv
- Mr. Ye Liu
- Mr. Wang Hang
- Prof. Yi Wu
- Dr. Jie Jia
- Dr. Qiang Wu
- Dr. Andrzej Cichocki
- Dr. Jianting Cao
- Dr. Qibin Zhao
- Dr. Jie Li
- Dr. Junhua Li

References

- Qibin Zhao, Cesar F. Caiafa, Danilo P. Mandic, Zenas C. Chao, Yasuo Nagasaka, Naotaka Fujii, Liqing Zhang and Andrzej Cichocki, Higher-Order Partial Least Squares (HOPLS): A Generalized Multi-Linear Regression Method, ***IEEE PAMI,*** 35(7): 1660-1673, July, 2013

- Qibin Zhao, Guoxu Zhou, Tülay Adali, Liqing Zhang, and Andrzej Cichocki, "Kernelization of Tensor-Based Models for Multiway Data Analysis: Processing of Multidimensional Structured Data," Signal Processing Magazine, IEEE , vol.30, no.4, pp.137,148, July 2013

- Junhua Li, Jianyi Liang, Qibin Zhao, Jie Li, Kan Hong, Liqing Zhang and Andrzej Cichocki, Design of Assistive Wheelchair System Directly Steered by Human Thoughts, International Journal of Neural Systems, Vol. 23, No. 3 1350013 (12 pages), 2013

- Q Zhao, G Zhou, T Adali, L Zhang, A Cichocki Kernel-based tensor Partial Least Squares for reconstruction of limb movements. In: Machine Learning for Signal Processing, ICASSP2013, May 26-31, Vancouver, Canada, 2013

- Qibin Zhao, Liqing Zhang and Andrzej Cichocki, A Tensor-Variate Gaussian Process for Classification of Multidimensional Structured Data, AAAI 2013, July 14-18, 2013, Bellevue, USA

- Junhua Li, Liqing Zhang, Active training paradigm for motor imagery BCI, ***Experimental Brain Research***, Volume 219, Number 2 (2012), 245-254

- Q. Zhao, C. Caiafa, D. Mandic, L. Zhang, et al, A Multilinear Subspace Regression Method Using Orthogonal Tensors Decompositions, NIPS 2011, Granada, Spain, 2011

- Jie Li, Liqing Zhang, Regularized tensor discriminant analysis for single trial EEG classification in BCI, Pattern Recognition Letters 31: 619–628, (2010)

- Qibin Zhao, Liqing Zhang and Andrzej Cichocki, EEG-based asynchronous BCI control of a car in 3D virtual reality environments, Chinese Science Bulletin, 54(1):78-87, 2009

- Jie Li, Liqing Zhang, et al, A Prior Neurophysiologic Knowledge Free Tensor-based Scheme for Single Trial EEG Classification, IEEE Transactions on Neural Systems and Rehabilitation Engineering, 17(2):107-115,2009

BCMI
Brain-like Computing
&
Machine Intelligence

Thanks for Your Attention!