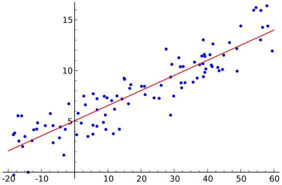
Towards a Regression using Tensors

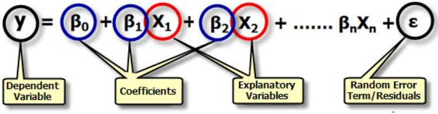
Ming Hou

February 27, 2014

Outline

Classical Linear Regression





- **Predict**

  e.g. speed, road conditions, weather ⇒ traffic accidents rates

- **Identify the key predictors**

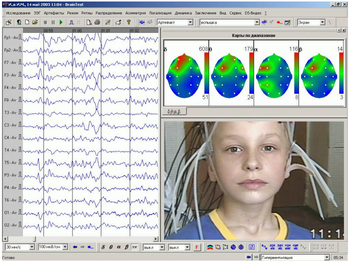
  e.g mental disease status ⇒ the regions of brain

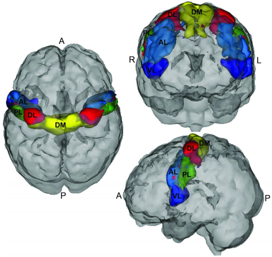
Multi-Dimensional Array Data (Tensors)

- **Neuroscience**
  - **EEG** data: ($time \times frequency \times electrodes$)
  - **fMRI** data: ($time \times x\ axis \times y\ axis \times z\ axis$)
- **Vision**
  - image (video) data:
    ($pixel \times illumination \times expression \times viewpoints$)
- **Chemistry**
  - fluorescence excitation-emission data:
    ($samples \times emission \times excitation$)

Brain Imaging Data Analysis

- Mental health disorders are difficult to diagnose and treat
- Physiology of brain is not well understood
- Neuroimaging can explain the brain physiology
- Several types of neuroimaging **EEG MRI fMRI**



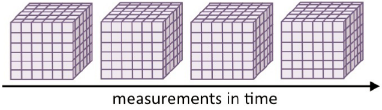


Brain Imaging Data Analysis using Regression

- **Goal** is to find association between **brain images** and **clinical outcomes**.
- Formulate as regression problem
  - clinical outcome as response
  - brain image (multi-dimensional array) as tensor predictor

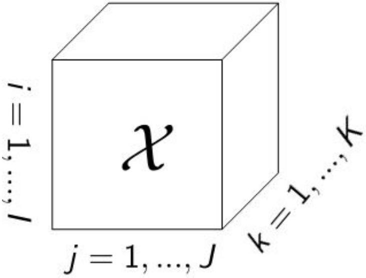
Limitation of Classical Regression

- **Naive approach**: turning an image array as **vector predictor**
  - e.g. a **fMRI** image: 4D array with size $256 \times 256 \times 256 \times 100$
  - yields a huge number of parameters (167 millions!)
  - ignores spatial and temporal correlation
- **New method**: treat each **fMRI** observation as **one tensor predictor** in regression model

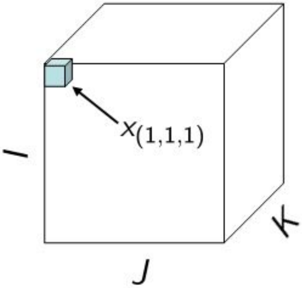


One fMRI Observation from One Subject

What is Tensor?



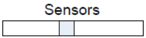
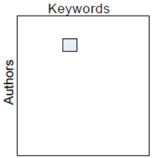
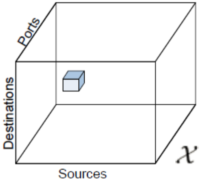
What is Tensor? con't



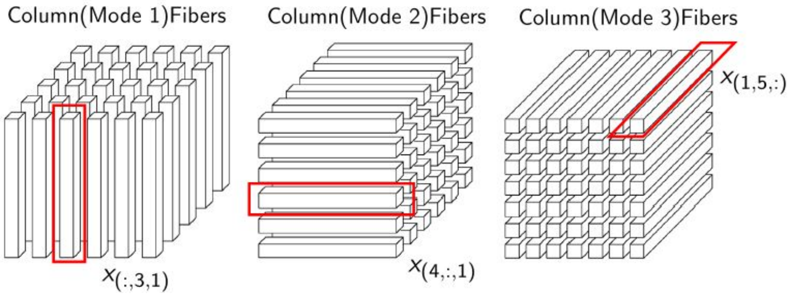
What is Tensor? con't

A tensor is formally denoted as $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$

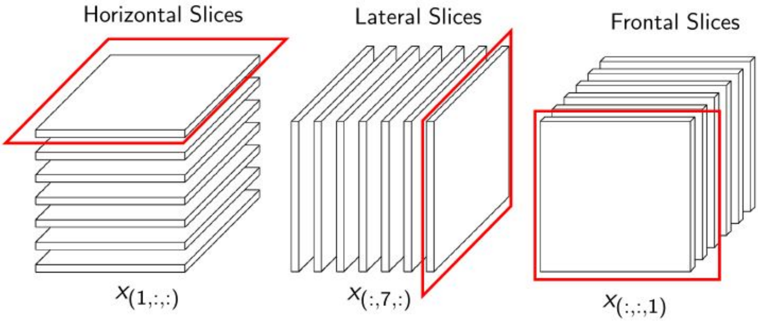
- generalization of vector and matrix
- represented as multi-dimensional array

Order	1st	2^{nd}	3^{rd}
Correspondence	Vector	Matrix	3D array
Example			

Fibers

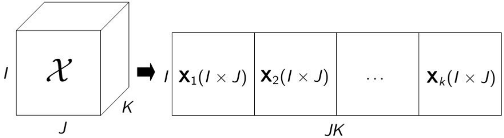

Column(Mode 1)Fibers
Column(Mode 2)Fibers
Column(Mode 3)Fibers
x(:,3,1)
x(4,:,1)
x(1,5,:)

Slices

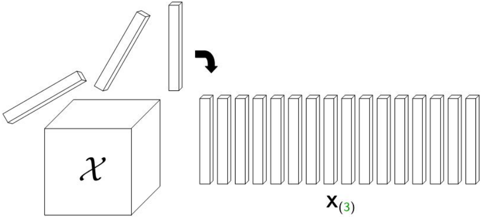


Matricization (Unfolding)

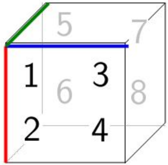
Convert a tensor to a matrix



Tube fibers are rearranged into the columns of a matrix



Matricization (Unfolding) Example



$$\mathbf{X}_{(1)} = \begin{pmatrix} 1 & 3 & 5 & 7 \\ 2 & 4 & 6 & 8 \end{pmatrix}$$

$$\mathbf{X}_{(2)} = \begin{pmatrix} 1 & 2 & 5 & 6 \\ 3 & 4 & 7 & 8 \end{pmatrix}$$

$$\mathbf{X}_{(3)} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{pmatrix}$$

The n-Mode Multiplication

Let $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$, $\mathbf{B} \in \mathbb{R}^{M \times J}$, the 2-mode product of $\mathcal{X}$ with $\mathbf{B}$ is defined by

$$\mathcal{Y} = \mathcal{X} \times_2 \mathbf{B} \in \mathbb{R}^{I \times M \times K}$$

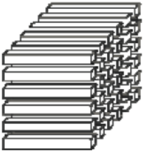
Elementwise

$$y_{imk} = \sum_j x_{ijk} b_{mj}$$

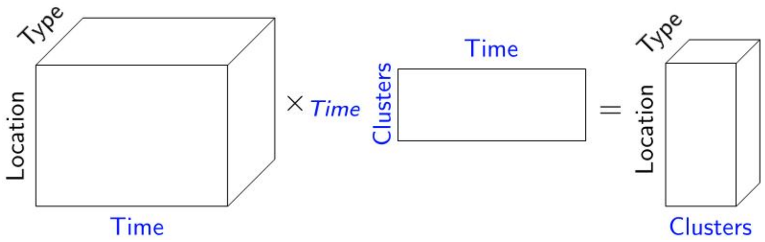
In matrix form

$$\mathbf{Y}_{(2)} = \mathbf{B}\mathbf{X}_{(2)}$$

Multiply each row (mode-2) fiber by **B**



The n-Mode Multiplication Example



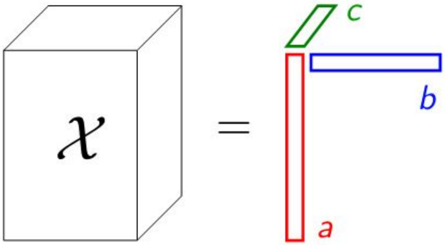
Rank-1 Tensor

3-way outer product

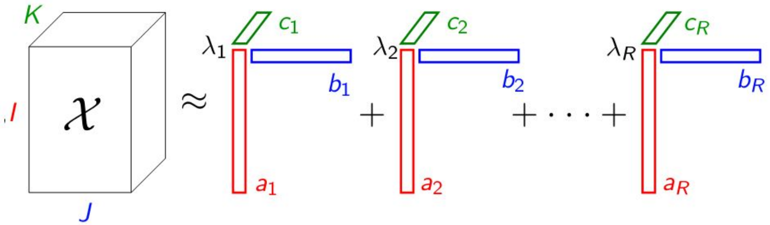
$$\mathcal{X} = a \circ b \circ c$$

Elementwise

$$x_{ijk} = a_i b_j c_k$$

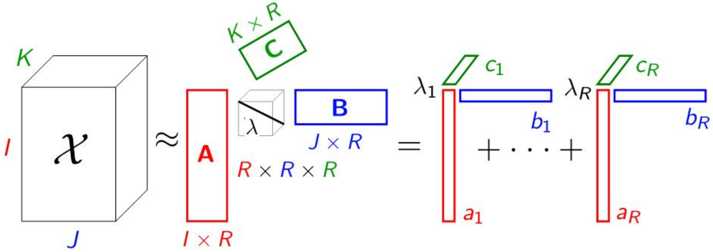


CANDECOMP/PARAFAC Decomposition



$$\mathcal{X} \approx \sum_{r=1}^{R} \lambda_r a_r \circ b_r \circ c_r$$

CANDECOMP/PARAFAC Decomposition con't

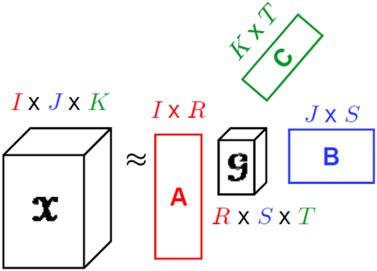


Define **factor matrix** $\mathbf{A} \in \mathbb{R}^{I \times R}$, $\mathbf{B} \in \mathbb{R}^{J \times R}$ and $\mathbf{C} \in \mathbb{R}^{K \times R}$

$$\mathcal{X} \approx \sum_{r=1}^{R} \lambda_r a_r \circ b_r \circ c_r \equiv [\lambda; \mathbf{A}, \mathbf{B}, \mathbf{C}]$$

$$x_{ijk} \approx \sum_{r=1}^{R} \lambda_r a_{ir} b_{jr} c_{kr}$$

Tucker decomposition



Defined by **factor matrix** $\mathbf{A} \in \mathbb{R}^{I \times R}$, $\mathbf{B} \in \mathbb{R}^{J \times S}$ and $\mathbf{C} \in \mathbb{R}^{K \times T}$, and **core tensor** $\mathcal{G} \in \mathbb{R}^{R \times S \times T}$

$$\mathcal{X} \approx \mathcal{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C} \equiv [\mathcal{G}; \mathbf{A}, \mathbf{B}, \mathbf{C}]$$

$$x_{ijk} = \sum_{r=1}^{R} \sum_{r=1}^{S} \sum_{r=1}^{T} g_{rst} a_{ir} b_{js} c_{kt}$$

Generalized Linear Regression Model

The standard linear regression model $\mathbf{x} \in \mathbb{R}^p$, $y = \beta^T \mathbf{x} + \alpha + \varepsilon$, $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ can be written

$$\mu = \beta^T \mathbf{x} + \alpha \quad y \sim \mathcal{N}(\mu, \sigma^2)$$

where $\mu = \mathbb{E}(Y|\mathbf{x})$

A **generalized linear regression model** (**GLM**) extends this to

$$g(\mu) = \beta^T \mathbf{x} + \alpha \quad y \sim \mathcal{EF}(\mu, \phi)$$

- $\mathcal{EF}(\mu, \phi)$ is any exponential family distribution (e.g. Normal, Poisson, Binomial)
- $g(\cdot)$ is any smooth monotonic link function
- $\beta^T \mathbf{x} + \alpha (= \eta)$ is the linear predictor

Generalized Linear Regression Model con't

In classical **GLM** Y belongs to an exponential family with **PMF**

$$p(y|\theta,\phi) = \exp\{\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)\}$$

The **GLM** relates $\mathbf{x} \in \mathbb{R}^p$ to the mean $\mu = \mathbb{E}(Y|\mathbf{x})$ by

$$g(\mu) = \eta = \alpha + \beta^T \mathbf{x}$$

The **GLM** for the matrix predictor $\mathbf{X}$ given by

$$g(\mu) = \eta = \alpha + \gamma^T \mathbf{z} + \beta_1^T \mathbf{X} \beta_2$$

Generalized Linear Regression Model for Tensor Predictor

The **GLM** with the systematic part for tensor predictor given by

$$g(\mu) = \eta = \alpha + \gamma^T \mathbf{z} + < \mathcal{B}, \mathcal{X} >$$

- D-dimensional tensor predictor $\mathcal{X} \in \mathbb{R}^{p_1 \times \cdots \times p_D}$
- D-dimensional coefficient tensor $\mathcal{B} \in \mathbb{R}^{p_1 \times \cdots \times p_D}$
- $\mathcal{B}$ has $\prod_{d=1}^{D} p_d$ parameters, which is ultrahigh dimensional and far exceeds sample size

Generalized Linear CP Tensor Regression

- Univariate outcome Y belongs to exponential family
- Tensor covariate $\mathcal{X} \in \mathbb{R}^{p_1 \times \cdots \times p_D}$
- Assume coefficient tensor $\mathcal{B}$ has a rank-R decomposition $[\mathbf{B}_1, ..., \mathbf{B}_D]$ where $\mathbf{B}_d \in \mathbb{R}^{p_d \times R}$

Generalized linear CP tensor regression model (Zhou et al. 2013) with the systematic part given by

$$g(\mu) = \eta = \alpha + \gamma^T \mathbf{z} + < \sum_{r=1}^{R} \beta_1^{(r)} \circ \cdots \circ \beta_D^{(r)}, \mathcal{X} >$$

$$= \alpha + \gamma^T \mathbf{z} + < (\mathbf{B}_D \odot \cdots \odot \mathbf{B}_1)\mathbf{1}_R, vec(\mathcal{X}) >$$

Generalized Linear CP Tensor Regression con't

Generalized linear CP tensor regression model given by

$$g(\mu) = \eta = \alpha + \gamma^T \mathbf{z} + < (\mathbf{B}_D \odot \cdots \odot \mathbf{B}_1)\mathbf{1}_R, vec(\mathcal{X}) >$$

- substantial reduction in dimensionality to the scale of $R \times \sum_{d=1}^{D} p_d$

  e.g For a 128-by-128-by-128 **MRI** image, the dimensionality reduce from **2,097,157** to **1157** using rank-3 decomposition

- Zhou et al.(2013) showed that this low rank tensor model could provide a sound recovery of many low rank signals

Estimation

Given n iid data $\{(y_i, \mathcal{X}_i, \mathbf{z}_i), i = 1, ..., n\}$ the log-likelihood

$$\ell(\alpha, \gamma, \mathbf{B}_1, ..., \mathbf{B}_D) = \sum_{i=1}^{n} \frac{y_i\theta - b(\theta)}{a(\phi)} + \sum_{i=1}^{n} c(y_i, \phi)$$

find the parameters $(\alpha, \gamma, \mathbf{B}_1, ..., \mathbf{B}_D)$ that maximizes this function

Estimation con't

Generalized linear CP tensor regression model given by

$$g(\mu) = \eta = \alpha + \gamma^T \mathbf{z} + < (\mathbf{B}_D \odot \cdots \odot \mathbf{B}_1)\mathbf{1}_R, vec(\mathcal{X}) >$$

A **key observation** is although $g(\mu)$ is not linear in $(\mathbf{B}_1, ..., \mathbf{B}_D)$ jointly, it is linear in each $\mathbf{B}_d$ separately

When updating $\mathbf{B}_d \in \mathbb{R}^{p_d \times R}$, the inner product part can be written as

$$< \mathbf{B}_d, \mathbf{X}_{(d)}(\mathbf{B}_D \odot \cdots \odot \mathbf{B}_{d+1} \odot \mathbf{B}_{d-1} \odot \cdots \odot \mathbf{B}_1) >$$

this yields the **block relaxation algorithm**, which converges to a stationary point

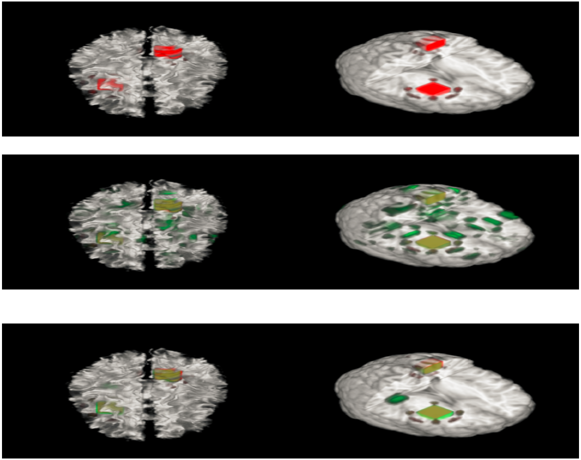
Sparsity Regularization

Maximize a regularized log-likelihood function

$$\ell(\alpha, \gamma, \mathbf{B}_1, ..., \mathbf{B}_D) - \sum_{d=1}^{D} \sum_{r=1}^{R} \sum_{i=1}^{p_d} P_\lambda(|\beta_{di}^{(r)}|, \rho)$$

- scalar penalty function $P_\lambda(|\beta|, \rho)$
- **power family** $P_\lambda(|x|, \rho) = \rho|\beta|^\lambda$, $\lambda \in (0, 2]$
- in particular lasso $(\lambda = 1)$

ADHD-200 Data Results



[taken from (Zhou et al. 2013)]

Future Work Plan

- Extending the linear CP/Tucker tensor regression model to the linear $\mathcal{H}$-Tucker tensor regression model
    - like CP model, the number of parameters is free from exponential dependence on D
    - preserve the flexibility of Tucker model
- Comparing the performance of different tensor regression models (Tucker, $\mathcal{H}$-Tucker) when applying different regularization approaches (sparsity regularization, trace norm regularization)

Future Work Plan con't

- Finding the appropriate model and algorithm to address the multi-block tensor regression problems
- Combing the kernel concept and partial least squares (PLS) techniques to deal with tensor (multi-block tensor) regression problem
- Applying tensor regression approaches listed above to the applications such as neuroimaing data analysis, brain signal data analysis to test if the improved performance can be achieved