Reconhecimento de padrões baseado em álgebra multilinear

Israel Gonçalves de Oliveira

prof.israel@gmail.com

9º Encontro: Machine Learning
Grupo Meetup Machine Learning Porto Alegre
Parque Científico e Tecnológico da PUCRS (Tecnopuc)

Porto Alegre, 25 de abril de 2018.

Resumo

Uma classe de métodos na área de reconhecimento de padrões que tem chamado atenção é a dos métodos multilineares. Diferente dos métodos lineares e dos não lineares, a representação e análise dos dados é realizada na sua forma natural: alta dimensionalidade e, por vezes, esparsa. Objetiva-se com tais métodos a preservação de correlações entre dimensões as quais seriam perdidas completa ou parcialmente quando convertidas em representações com menos dimensões.

Resumo

É apresentada a estrutura básica do tensor e a decomposição tensorial, bem como um método do estado da arte e uma formulação ainda não publicada. São apresentados, também, algumas problemáticas no uso de tensores aplicados a métodos de aprendizado de máquina e uma proposta de trabalho futuro.

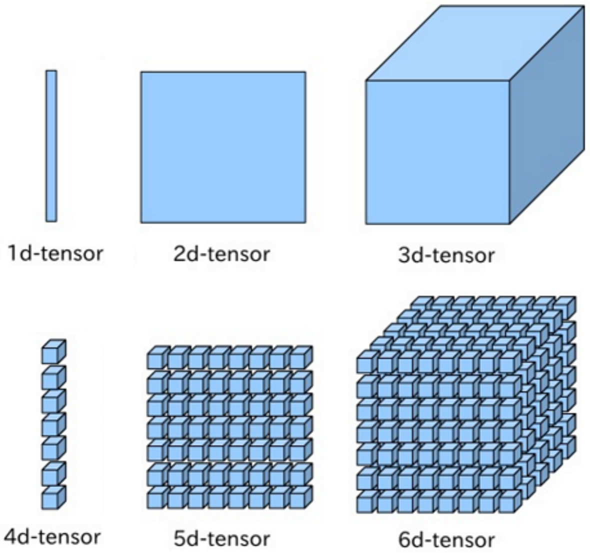
Contents

Tópicos

- Tensores: o que são? Por que usar?
- Tensores em Machine Learning.
- *Multilinear Discriminant Analisys* (MDA).
- Projeção tensorial: espaço dos fatores.
- Invariância no espaço de fatores.
- Críticas ao uso de tensores em Machine Learning.

Tensores

Representação tensorial

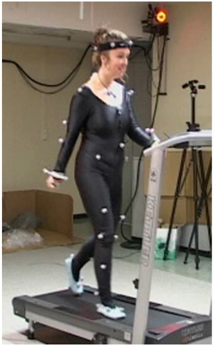


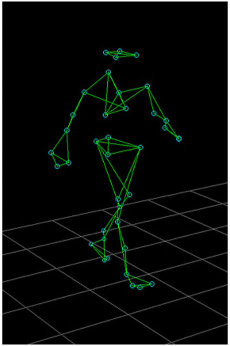
Representação tensorial

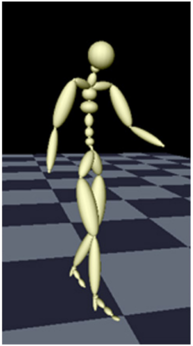
$$1 \qquad \begin{bmatrix} 1 \\ 2 \end{bmatrix} \qquad \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \qquad \begin{bmatrix} \begin{bmatrix} 1 & 2 \end{bmatrix} & \begin{bmatrix} 3 & 2 \end{bmatrix} \\ \begin{bmatrix} 1 & 7 \end{bmatrix} & \begin{bmatrix} 5 & 4 \end{bmatrix} \end{bmatrix}$$

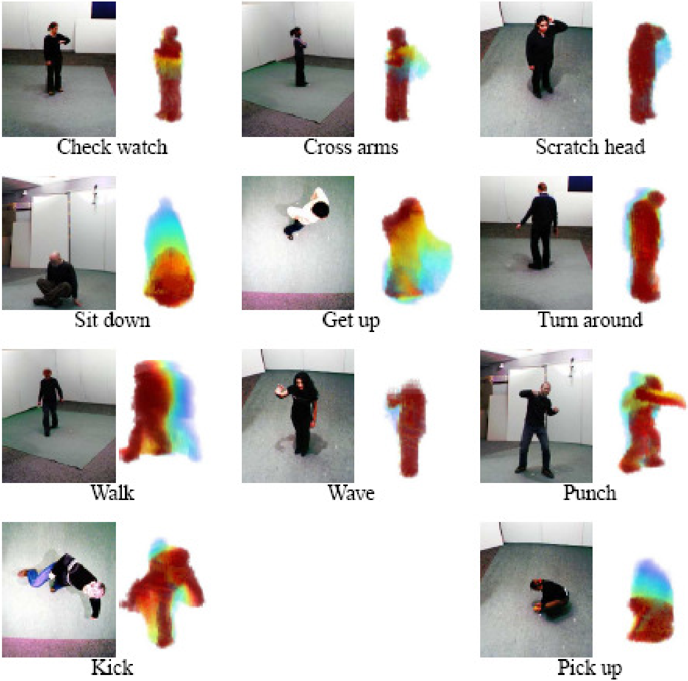
Representação tensorial

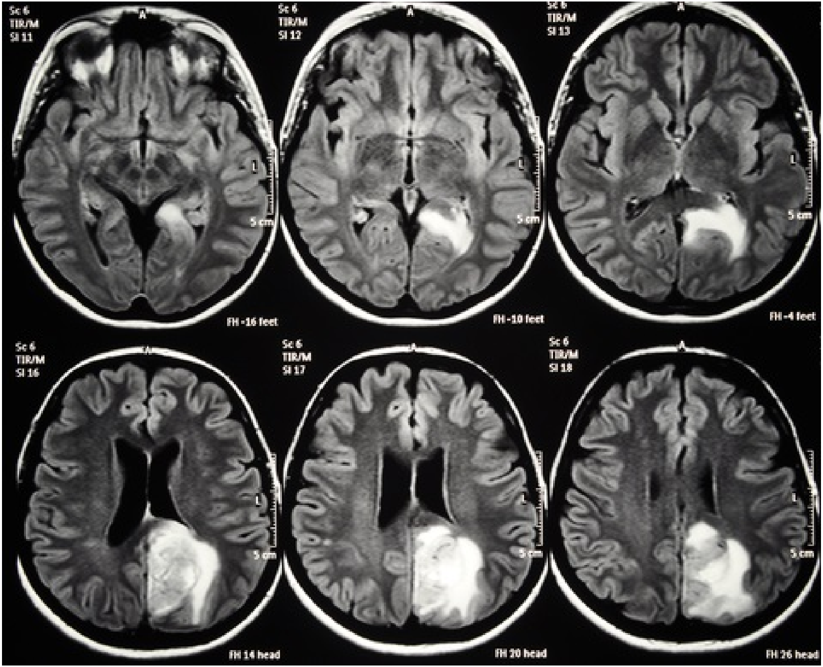
- 2d-tensor: valores de coordenadas de uma partícula no tempo, imagem em escala de cinza, binária etc.
- 3d-tensor: coordenas de várias partículas no tempo (como na captura de movimento), imagem RGB, vídeo preto e branco etc.
- 4d-tensor: vídeo colorido, captura temporal 3D etc.
- 5d-tensor: síntese artificial de dados visando alguma conveniência.
- 6d-tensor:...








Check watch
Cross arms
Scratch head
Sit down
Get up
Turn around
Walk
Wave
Punch
Kick
Pick up


Sc 6
TIR/M
SI 11
FH -16 feet
Sc 6
TIR/M
SI 12
FH -10 feet
Sc 6
TIR/M
SI 13
FH -4 feet
Sc 6
TIR/M
SI 16
FH 14 head
Sc 6
TIR/M
SI 17
FH 20 head
Sc 6
TIR/M
SI 18
FH 26 head
5 cm

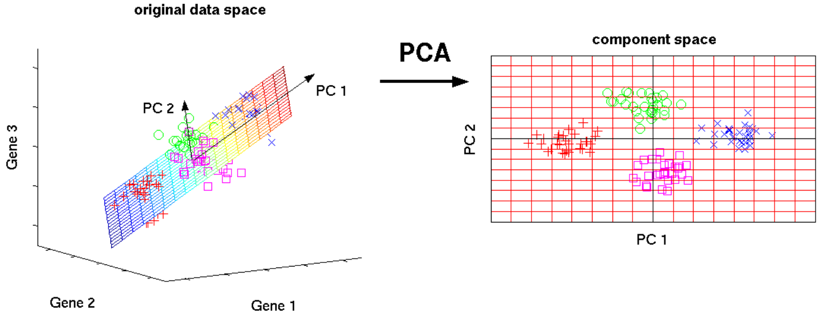
Representação tensorial

- Amostras na sua forma natural podem ser melhor representadas com tensores de ordem superior.
- Dados volumosos.
- Informação relevante nas dimensões superiores (correlações).
- No geral, dados esparsos.

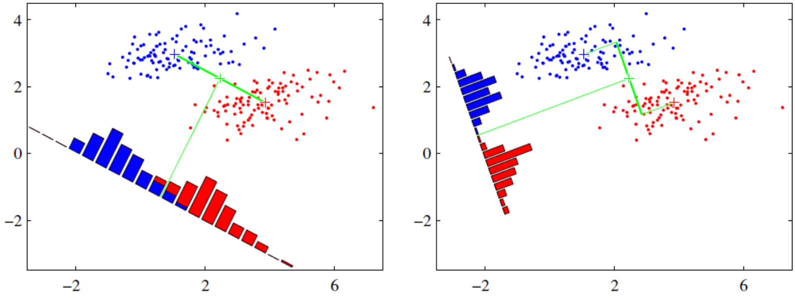
$$
\begin{pmatrix}
1.0 & 0 & 5.0 & 0 & 0 & 0 & 0 & 0 \\
0 & 3.0 & 0 & 0 & 0 & 0 & 11.0 & 0 \\
0 & 0 & 0 & 0 & 9.0 & 0 & 0 & 0 \\
0 & 0 & 6.0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 7.0 & 0 & 0 & 0 & 0 \\
2.0 & 0 & 0 & 0 & 0 & 10.0 & 0 & 0 \\
0 & 0 & 0 & 8.0 & 0 & 0 & 0 & 0 \\
0 & 4.0 & 0 & 0 & 0 & 0 & 0 & 12.0
\end{pmatrix}
$$

Tensores em ML

- *Linear subspace learning*: *principal component analysis* (PCA), *independent component analysis* (ICA), *linear discriminant analysis* (LDA) and *canonical correlation analysis* (CCA).

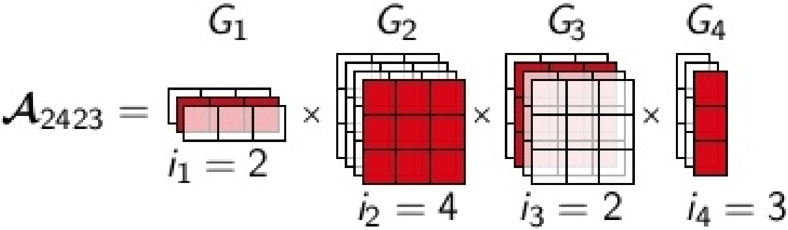


- Multilinear subspace learning: *multilinear principal component analysis* (MPCA), *multilinear independent component analysis* (MICA), *multilinear discriminant analysis* (MDA) and *multilinear ou tensor canonical correlation analysis* (MCCA e TCCA).



Tensores em ML

- Generalização das variantes de MDA e MPCA: MDA com e sem MSD. Método atualmente com melhores resultados (em *multilinear subspace learning*)
- *Tensor Train Decomposition*: *TT-format*, representação enxuta com uma combinação de fatores (*TT-cores*).



Tensores em ML

- **Tensorizing Neural Network*: Uso da representação *TT-format*, alternativa ao uso das *Fully Connected Layers* com as TT-Layers (TensorNet) [1]. Menor custo computacional, menor generalização dos dados.
- **Tensor Processing Unit* (TPU): Hardware ASIC para aplicações de aprendizado de máquina, especificamente para redes neurais artificiais e compatível com a API TensorFlow [2].

Tensores em ML: prós e contras

Prós:

- Aproveitamento de informações potencialmente relevantes oriundas das correlações entre as dimensões.
- Operações lineares e, então, paralelizáveis: possível uso de GPU, processamento multicores em geral.
- APIs com representações otimizadas para dados esparsos.
- Alternativa mais eficiente à vetorização (ou matricialização), com o uso do MDA para "eliminar"dimensões superiores de forma mais eficiente do que um simples empilhamento ou soma.
- Facilmente associado com diferentes métodos de classificação (SVM, ANN etc.).

Tensores em ML: prós e contras

Contras:

- Sem unicidade prática nas decomposições (problema do tipo NP, solução aproximada por um problema de otimização: máxima semelhança, menor erro etc.). Interfere significativamente na modelagem com dados esparsos.
- Sensível a variância espacial (nas dimensões). Métodos tensoriais oferecem baixa robustez à desalinhamentos na obtenção das amostras.
- Não existe convolução 3D.(?)
- Altamente sensível a variação de parâmetros, baixa linearidade entre os parâmetros e a performance final de classificação.
- Poucas APIs (nenhuma para C++, com os produtos tensoriais). As duas APIs mais eficientes são para MATLAB [3].
- Busca exaustiva dos melhores parâmetros.

MDA

- Para cada modo é obtido um subespaço otimizado:

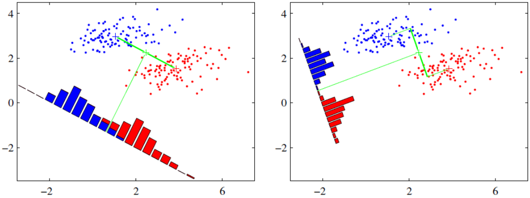
$$\{U_k^*\} = \arg\left\{\max_{U_k} \frac{\mathrm{Tr}(U_k^T S_B U_k)}{\mathrm{Tr}(U_k^T S_W U_k)}\right\}$$

  e com MSD

$$\{U_k^*\} = \arg\left\{\max_{U_k} \mathrm{Tr}(U_k^T S_B - \gamma S_W U_k)\right\}$$

- Aumenta o espalhamento entre classes: afasta as amostras de classes diferentes (S_B: *between-class scatter covariance matrix*). Semelhante ao PCA.
- Diminui o espalhamento entre amostras de mesma classe: aproxima as amostras de mesma classes (S_W: *within-class scatter covariance matrix*).

MDA II



- Transforma as amostras para o novo espaço tensorial:

$$
\begin{aligned}
\mathbf{Z} &= \mathbf{X} \times_1 U_1 \times_2 U_2 \times_3 \cdots \times_p U_p \\
\mathbf{Z}_{c_j} &= \mathbf{X}_{c_j} \times_1 U_1 \times_2 U_2 \times_3 \cdots \times_p U_p
\end{aligned}
$$

com dimensionalidade reduzida e, consequentemente, menor custo computacional e discriminando melhor a informação contida nas amostras (promovendo maior precisão de classificação).
Procedimento para o MDA [4]:

MDA III

Multilinear Discriminant Analysis:

Given the sample set $\tilde{\mathbf{X}} \in \mathbb{R}^{m_1 \times m_2 \times \ldots \times m_n \times N}$, their class labels $c_i \in \{1, 2, \ldots, N_c\}$, and the final lower dimensions $m'_1 \times m'_2 \times \ldots \times m'_n$.

1. Initialize $U_1^0 = I_{m_1}, U_2^0 = I_{m_2}, \ldots, U_n^0 = I_{m_n}$;
2. For $t = 1, 2, \ldots, T_{max}$ do

a) For $k = 1, 2, \ldots, n$ do

$$\mathbf{Y}_i = \mathbf{X}_i \times_1 U_1^t \ldots \times_{k-1} U_{k-1}^t \times_{k+1} U_{k+1}^{t-1} \ldots \times_n U_n^{t-1}$$

$$Y_i^k \Leftarrow_k \mathbf{Y}_i$$

$$S_B = \sum_{\substack{j=1}}^{\prod_{o \neq k} m_o} S_B^j, S_B^j = \sum_{c=1}^{N_c} n_c (\bar{Y}_c^{k,j} - \bar{Y}^{k,j})(\bar{Y}_c^{k,j} - \bar{Y}^{k,j})^{\mathrm{T}}$$

$$S_W = \sum_{\substack{j=1}}^{\prod_{o \neq k} m_o} S_W^j, S_W^j = \sum_{i=1}^{N} (Y_i^{k,j} - \bar{Y}_{c_i}^{k,j})(Y_i^{k,j} - \bar{Y}_{c_i}^{k,j})^{\mathrm{T}}$$

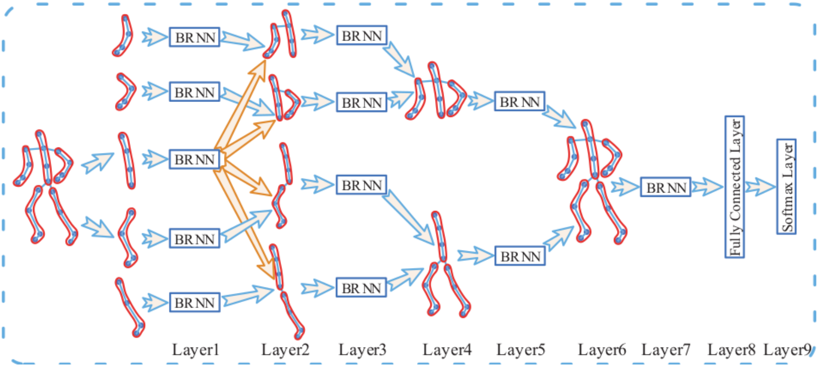
$$S_B U_k^t = S_W U_k^t \Lambda_k, \quad U_k^t \in \mathbb{R}^{m_k \times m'_k}$$

b) If $t > 2$ and $\| U_k^t - U_k^{t-1} \| < m'_k m_k \varepsilon, k = 1, .., n$, break;

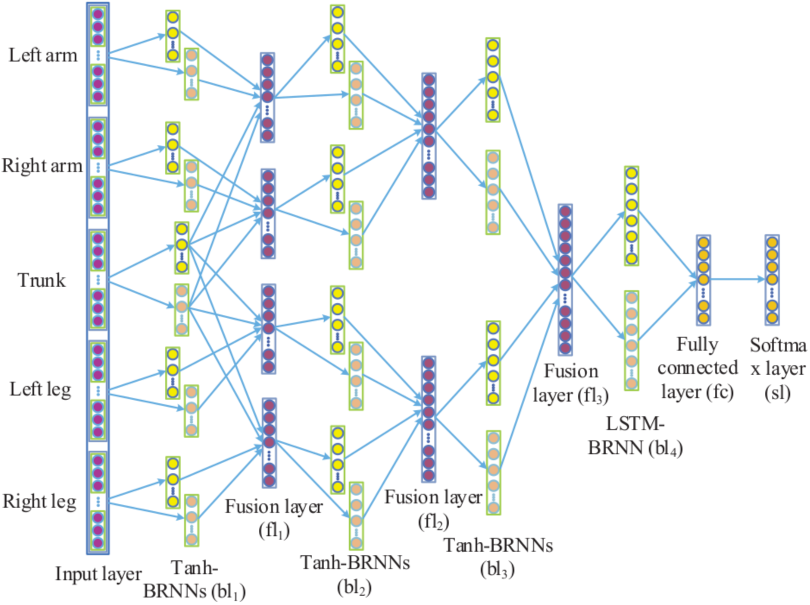
3. Output the projections $U_k = U_k^t \in \mathbb{R}^{m_k \times m'_k}, k = 1, \ldots, n$.

- Projeção otimizada com uma generalização das variações do MDA disponíveis em [5, 6, 7, 4].
- Classificação por k-NN, mas poucos resultados bons com $k > 1$ (Weizmann e MHAD).
- Métricas: Frobenius (TensorLAB [8]), Norma-2 (C++/Eigen [9]). Ambos são mais rápidos e a função 'norm' do Eigen oferece resultados muito superiores à função equivalente do MATLAB. Demais operações com a TensorToolbox [10], da Sandia Labs.
- Representações Tensor Train (TT-cores) aplicados ao MDA não resultaram em maior precisão de acerto de classificação, pelo contrário.

- - Hierarchical Recurrent Neural Network for Skeleton Based Action Recognition (2015) [11].

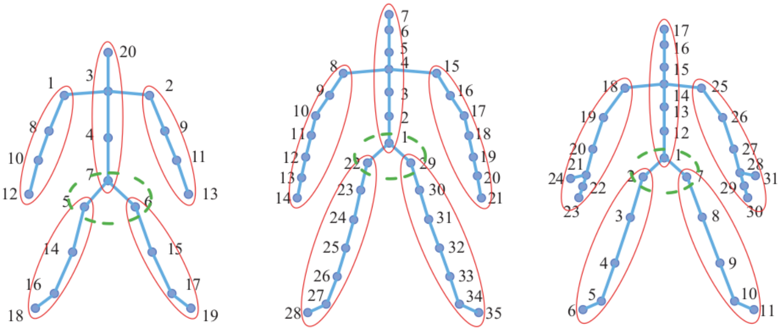


Resultados



Left arm
Right arm
Trunk
Left leg
Right leg
Input layer
Tanh-BRNNs (bl_1)
Fusion layer (fl_1)
Tanh-BRNNs (bl_2)
Fusion layer (fl_2)
Tanh-BRNNs (bl_3)
Fusion layer (fl_3)
LSTM-BRNN (bl_4)
Fully connected layer (fc)
Softmax layer (sl)

Resultados



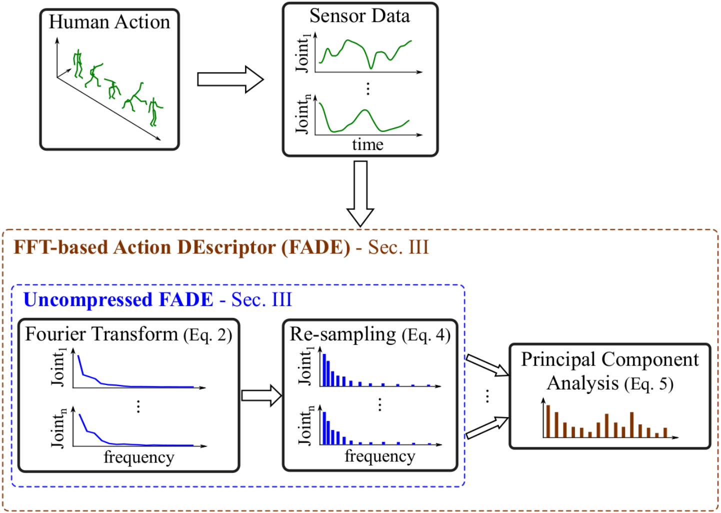
MSR Action3D Dataset   Berkeley MHAD   HDM05

Resultados

- 1 experimento com HDM05: 96.2% (batido com 99,9-100.0%)
  1 experimento com MSR: 96.0% (não batido*, menos de 60%).
  1 experimento com Berkeley MHAD*: 100% (foi conseguido 98.91%, é possível chegar a 100% com outra configuração de atores para teste e para treino).
  Berkeley Multimodal Human Action Database (MHAD).
  Hochschule der Medien (Faculdade de Mídia), 2005 (HDM05).
  (Microsoft Research) MSR Action 3D.

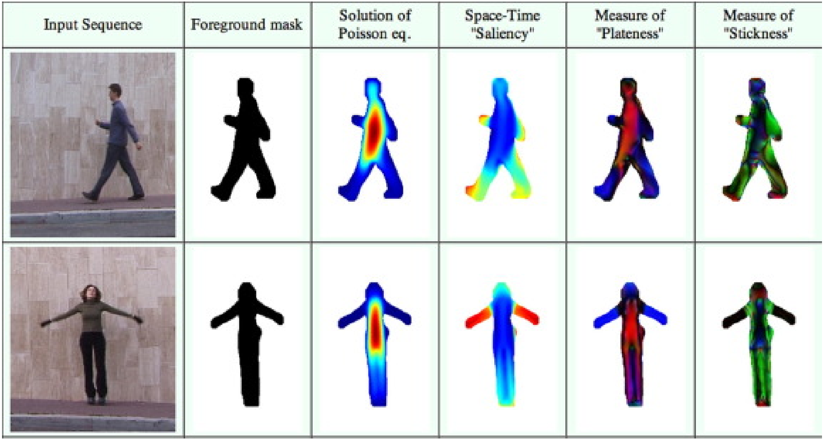
Resultados

- - Encoding human actions with a frequency domain approach (2016) [12]. 2 experimentos com HDM05: 96.2% e 88.0% (batidos com 99,9-100%)

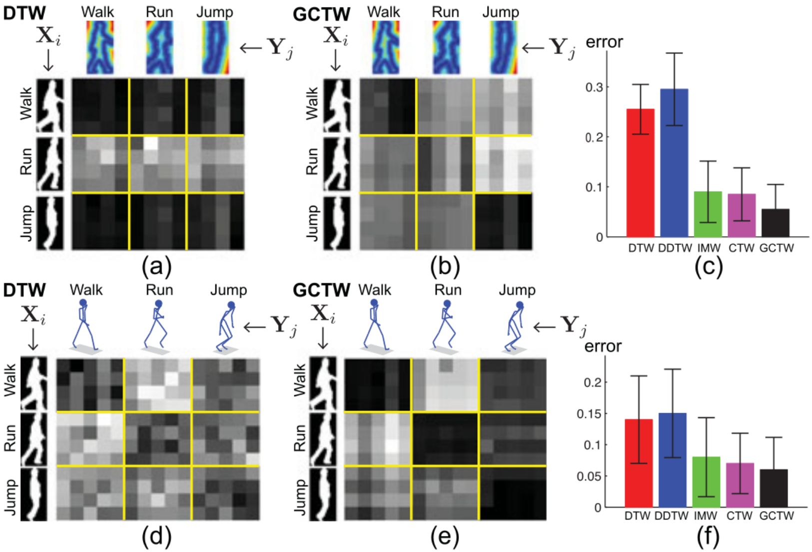


Resultados

- Generalized Canonical Time Warping (2016) [13]: 1 experimento com Weizmann: 94.5%, batido com 98,33%.



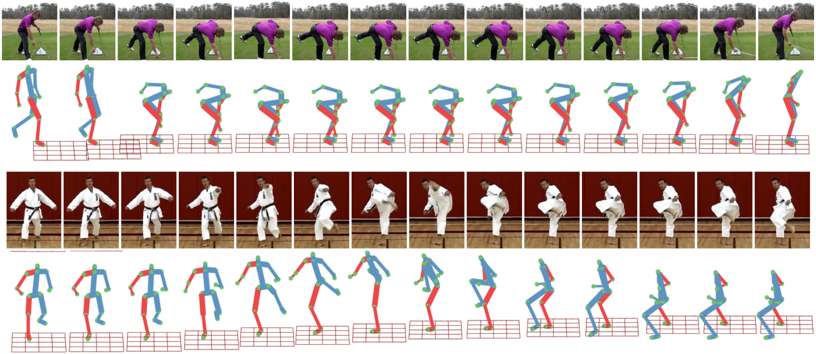
Resultados



DTW
Walk
Run
Jump
$\mathbf{X}_i$
$\leftarrow \mathbf{Y}_j$
Walk
Run
Jump
(a)
GCTW
Walk
Run
Jump
$\mathbf{X}_i$
$\leftarrow \mathbf{Y}_j$
Walk
Run
Jump
(b)
error
0.3
0.2
0.1
0
DTW DDTW IMW CTW GCTW
(c)
DTW
Walk
Run
Jump
$\mathbf{X}_i$
$\leftarrow \mathbf{Y}_j$
Walk
Run
Jump
(d)
GCTW
Walk
Run
Jump
$\mathbf{X}_i$
$\leftarrow \mathbf{Y}_j$
Walk
Run
Jump
(e)
error
0.2
0.15
0.1
0.05
0
DTW DDTW IMW CTW GCTW
(f)

Resultados

- Learning robust features for gait recognition by Maximum Margin Criterion [14] (2016). Combinação de variações do PCA e LDA. 1 experimento com CMU Mocap: 91.02%, batido com 95.74%



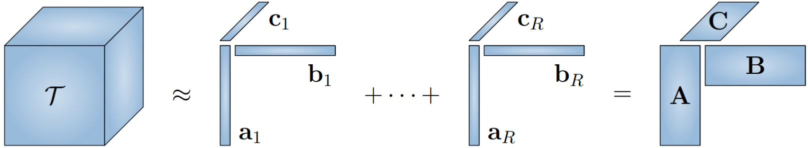
Projeção tensorial

Decomposição Tensorial

- Considere, agora, N amostras (faces, por exemplo) e cada amostra uma matriz $X_n \in \mathbb{R}^{l \times c}$, ao empilhar as amostras obtém-se um tensor de terceira ordem $\mathcal{T} \in \mathbb{R}^{l \times c \times N}$. A decomposição canônica poliádica (*canonical polyadic decomposition*) em fatores do tensor $\mathcal{T}$, com posto (*rank*) R pode ser escrita como [8]:

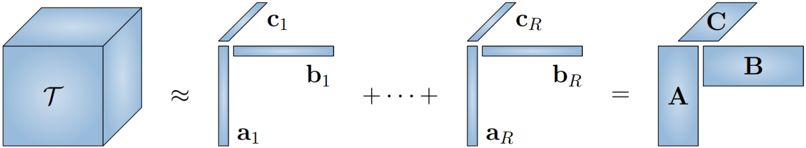
$$\mathcal{T} = \sum_{r=1}^{R} a_r \otimes b_r \otimes c_r + \epsilon(R) = A \otimes B \otimes C + \epsilon(R) \tag{1}$$

com $a_r \in \mathbb{R}^l$, $b_r \in \mathbb{R}^c$, $c_r \in \mathbb{R}^N$, $A \in \mathbb{R}^{l \times R}$, $B \in \mathbb{R}^{c \times R}$ e $C \in \mathbb{R}^{N \times R}$.



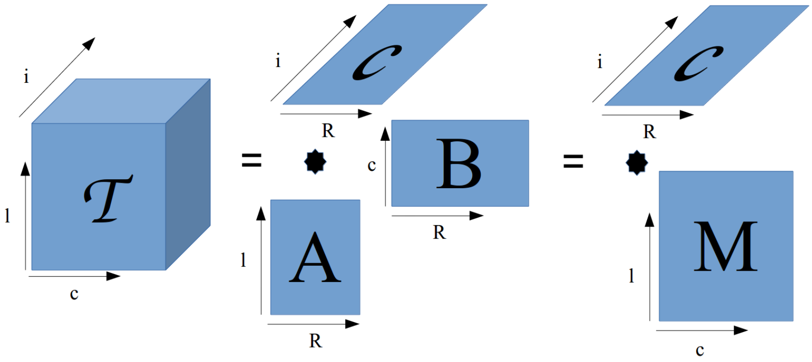
Decomposição tensorial

$$\text{PO: } \min_{A,B,C} \|\mathcal{T} - A \otimes B \otimes C\|_F$$

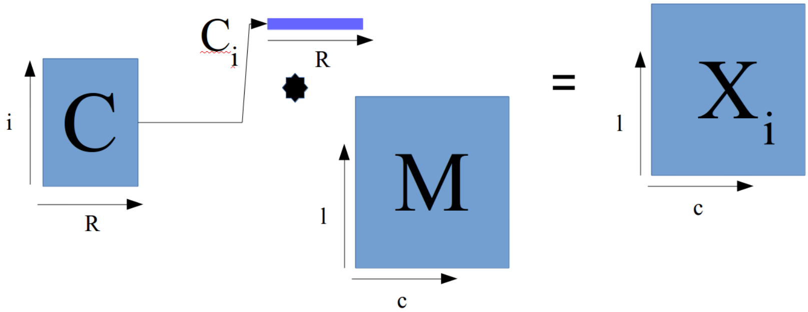


Decomposição tensorial: métodos

- Alternating algorithms:
  - alternating least squares (ALS)
  - alternating slice-wise diagonalisation (ASD)
- Algebraic algorithms:
  - simultaneous diagonalization (SD)
  - simultaneous generalized Schur decomposition (SGSD)
- Optimization algorithms:
  - Levenberg–Marquardt (LM)
  - nonlinear conjugate gradient (NCG)
  - limited memory BFGS (L-BFGS)
- Direct methods:
  - Direct multilinear decomposition (DMLD)



i
l
c
$\mathcal{T}$
=
i
R
$\mathcal{C}$
c
B
R
l
A
R
=
i
R
$\mathcal{C}$
l
M
c



C
i
R
C
i
R
M
l
c
=
X
i
l
c

Projeção tensorial

- A i-ésima amostra pode ser reconstituída individualmente, com erro relacionado ao posto (R) usado na decomposição, a partir das matrizes A e B e da i-ésima linha de C, definindo $M \triangleq A \otimes B$:

$$X_i \approx A \otimes B \otimes C_i = \sum_{r=1}^{R} C_{ir}(A \otimes B) = \sum_{r=1}^{R} C_{ir} M \qquad (2)$$

  portanto, cada amostra é uma combinação da matriz $M \in \mathbb{R}^{l \times c}$.

- Obs.: para imagens 112×92, um valor de R com erro $\epsilon < 10\%$ está acima de 5. Valor usado: $\epsilon(50) < 5\%$.

Projeção tensorial

- Considere uma base de dados de faces, ao agrupar N fotos de c_i pessoas (classes) diferentes na forma de um tensor de terceira ordem $\mathcal{T}$, pode-se, então, representar cada face X_i num espaço formado pelos fatores de posto R que compõem $\mathcal{T}$, na forma:

$$X_i \approx \sum_{r=1}^{R} x_{ir} M = M \otimes x_i \tag{3}$$

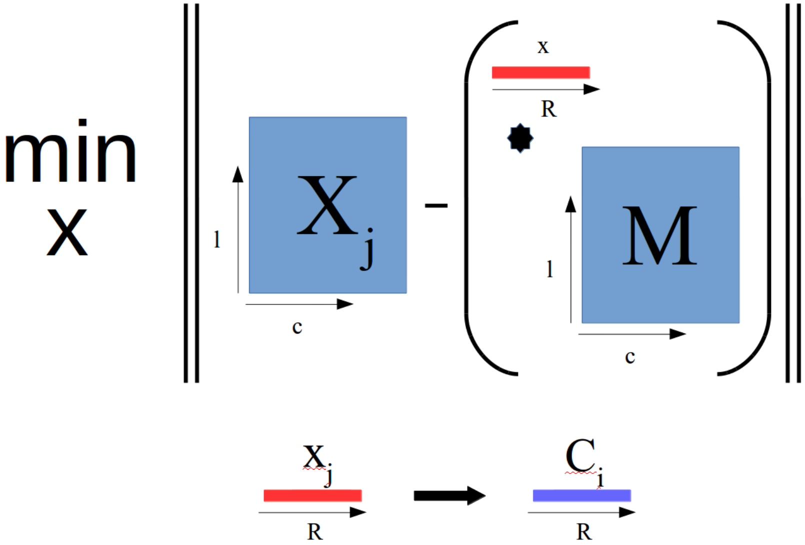
  com vetor linha $x_i \in \mathbb{R}^{1 \times R}$.
- Se as faces (amostras) conhecidas podem ser reconstituídas partir da matriz M, (com erro dependente do posto R definido *a priori*), seria possível reconhecer uma pessoa (classe) que tenha sua face dentro da base usada para obtenção de M?

- Uma face (amostra) X_j fora da base (de treino), de uma pessoa (classe) c_j que esteja na base, pode ser decomposto nos fatores M e x_j na forma:

$$X_j \approx \sum_{r=1}^{R} x_{jr} M = M \otimes x_j \tag{4}$$

com vetor linha $x_j \in \mathbb{R}^{1\times R}$. Seguindo o critério:

$$x_j = \arg\min_{x} \|X_j - M \otimes x\|_F \tag{5}$$


min
X
X
j
l
c
x
R
M
l
c
X
j
R
C
i
R

- com M a matriz fator da decomposição das faces da base (de treino). Nesse caso, um classificador por vizinho mais próximos p.e., comparando a amostra j com a amostra da base (de treino) i:

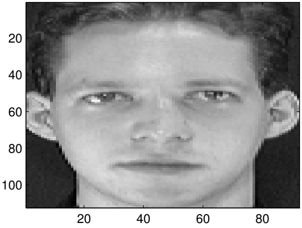
$$
\begin{aligned}
c_j &= \arg\min_{c_i} \|X_j - X_i\| && (6)\\
&= \arg\min_{c_i} \|M \otimes x_j - M \otimes x_i\| && (7)\\
&= \arg\min_{c_i} \|M \otimes (x_j - x_i)\| && (8)
\end{aligned}
$$

como $\|M \otimes (x_j - x_i)\| = \alpha \|M\| \|x_j - x_i\|$, o problema pode ser rescrito como

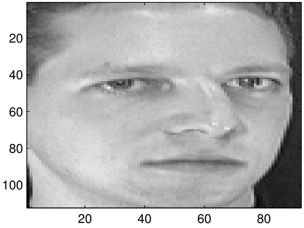
$$c_j = \arg\min_{c_i} \|x_j - x_i\| \qquad (9)$$

- Para amostras $112 \times 92 = 10.304$, para $R = 50$ ($\epsilon(50) < 5\%$) com precisão de $> 95\%$, pode-se representar a amostra com um vetor de tamanho 50. Redução de tamanho para $0{,}4\%$ do original (redução de $99{,}6\%$).
- Considere uma base com 10 imagens de mesmo tamanho, seriam necessários $10 \times 112 \times 92 = 103.040$. Usando a projeção proposta, seria necessário armazenar o fator M e 10 vetores de tamanho 50: $\times 112 \times 92 + 10 \times 50 = 10.304 + 500 = 10.804$. Redução de 90%, ou seja, utiliza-se aprox. 10% do armazenamento original.

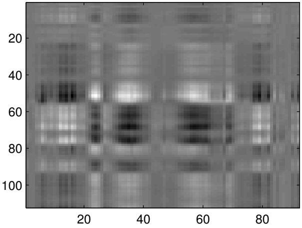
Face de treino.



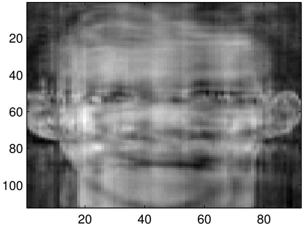
Face de teste.

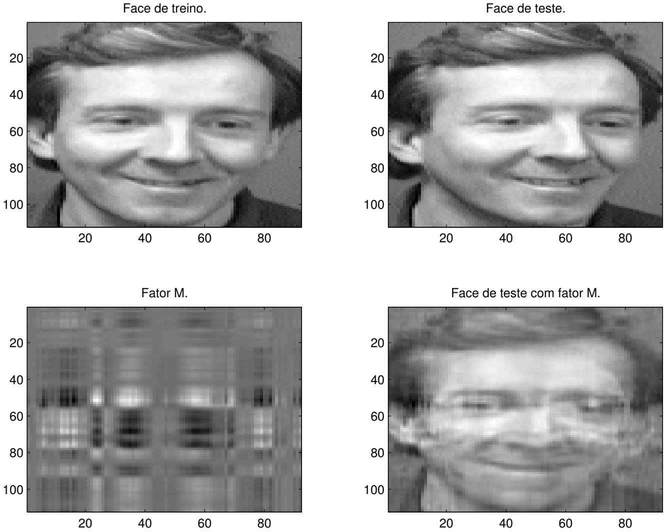


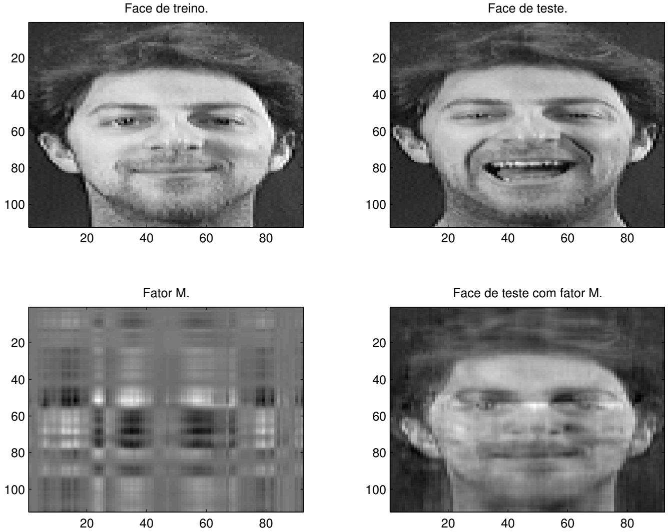
Fator M.



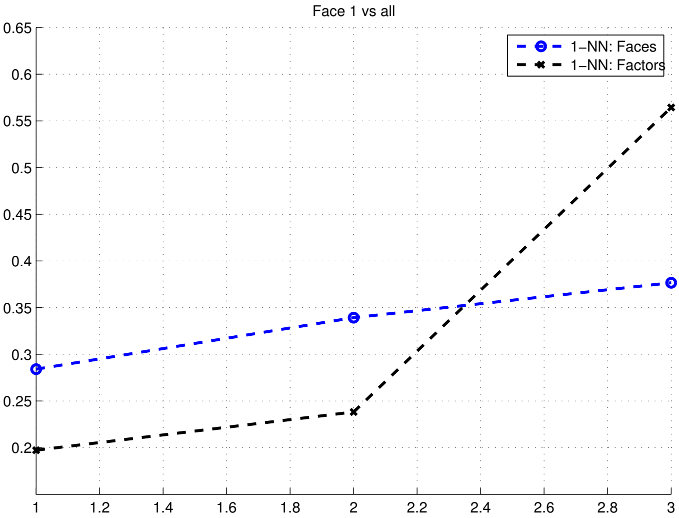
Face de teste com fator M.

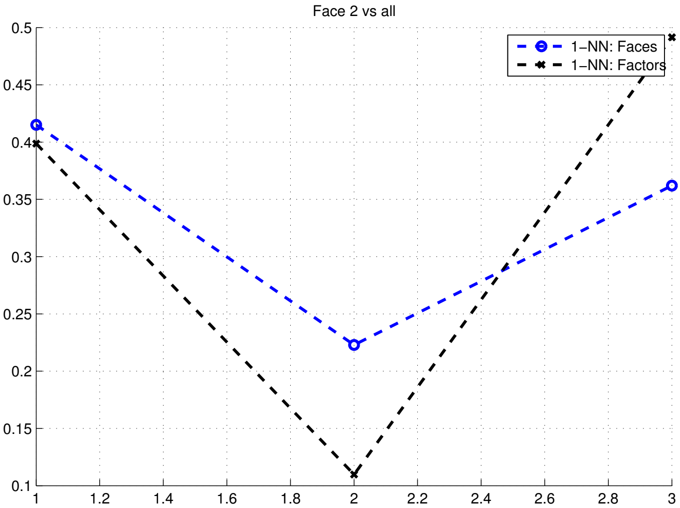



Face de treino.
Face de teste.
Fator M.
Face de teste com fator M.

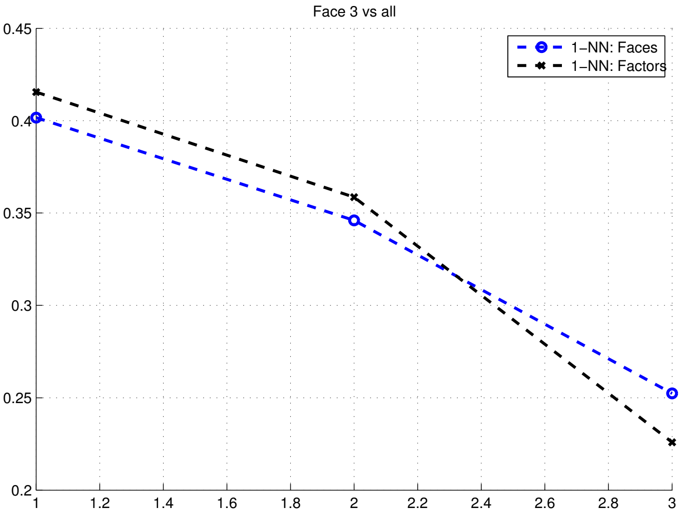

Face de treino.
Face de teste.
Fator M.
Face de teste com fator M.
20
40
60
80
100
20
40
60
80

Resultados parciais





Face 2 vs all
1–NN: Faces
1–NN: Factors
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
1
1.2
1.4
1.6
1.8
2
2.2
2.4
2.6
2.8
3

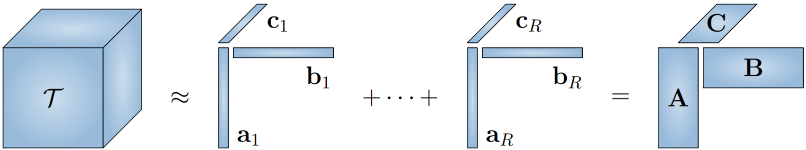

Face 3 vs all
0.45
0.4
0.35
0.3
0.25
0.2
1
1.2
1.4
1.6
1.8
2
2.2
2.4
2.6
2.8
3
1–NN: Faces
1–NN: Factors

Conclusões parciais

- Perspectiva de redução no custo de armazenamento da base projetada.
- Passo de teste exige baixo custo computacional.
- Performance de classificação sensível ao rank estimado.
- Quanto maior a base ou quanto maior o rank maior o custo computacional no passo de treino.
- Em teoria, o passo de treino não precisaria ser refeito com a adição de novas amostras confirmadas (classificação certa).
- Em teoria, caso o método de decomposição tensorial garanta unicidade de solução (que na prática não é possível) a performance de classificação seria máxima. Quanto melhor o método de decomposição, melhor será a performance de classificação.

$$\mathcal{T} = \sum_{r=1}^{R} a_r \otimes b_r \otimes c_r + \epsilon(R) = A \otimes B \otimes C + \epsilon(R) \tag{10}$$

com $a_r \in \mathbb{R}^l$, $b_r \in \mathbb{R}^c$, $c_r \in \mathbb{R}^N$, $A \in \mathbb{R}^{l \times R}$, $B \in \mathbb{R}^{c \times R}$ e $C \in \mathbb{R}^{N \times R}$.



Critério: $\min\limits_{A,B,C} \|\mathcal{T} - A \otimes B \otimes C\|_F$

Considerações finais

Considerações finais

- Na modalidade de redução dimensional (MDA), é altamente necessário se ter uma metodologia eficiente e automatizada de busca dos parâmetros ótimos. Métodos como as CNN apresentam melhor relação entre complexidade de implementação e performance.
- *Apesar do potencial discriminatório e da representação mais próxima do natural, a produção científica na área e baixa em relação a outros métodos na área de reconhecimento de padrões e aprendizado de máquina.
- *É necessário uma melhoria nos métodos de decomposição tensorial com vistas à área de aprendizado de máquina.

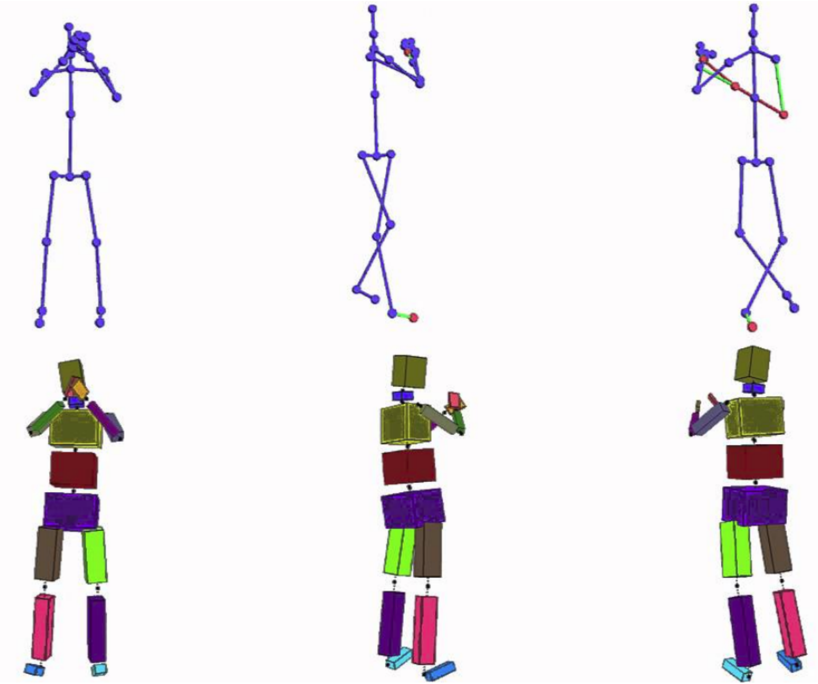
Invariâncias e unicidade

Invariâncias e unicidade

- Captura do mesmo movimento por duas câmeras, gerando duas amostras X_A e X_B. Uma possível decomposição de ambas as amostras:

$$X_A = J_A \times F_A \times P_A \tag{11}$$

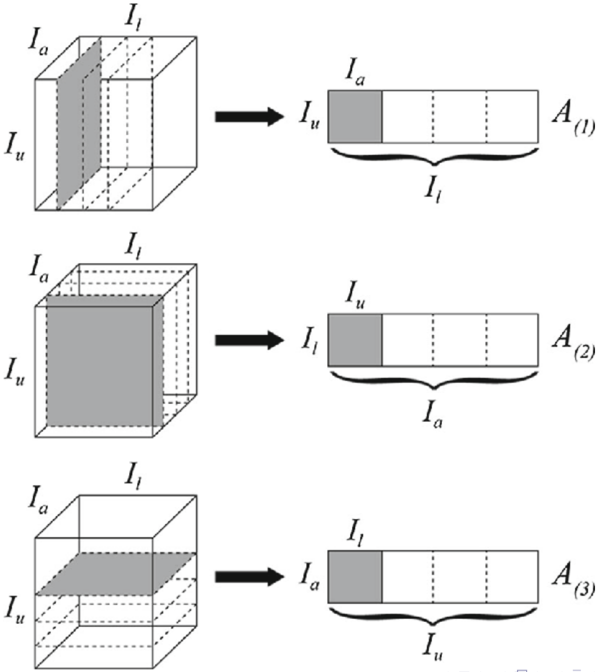
$$X_B = J_B \times F_B \times P_B \tag{12}$$



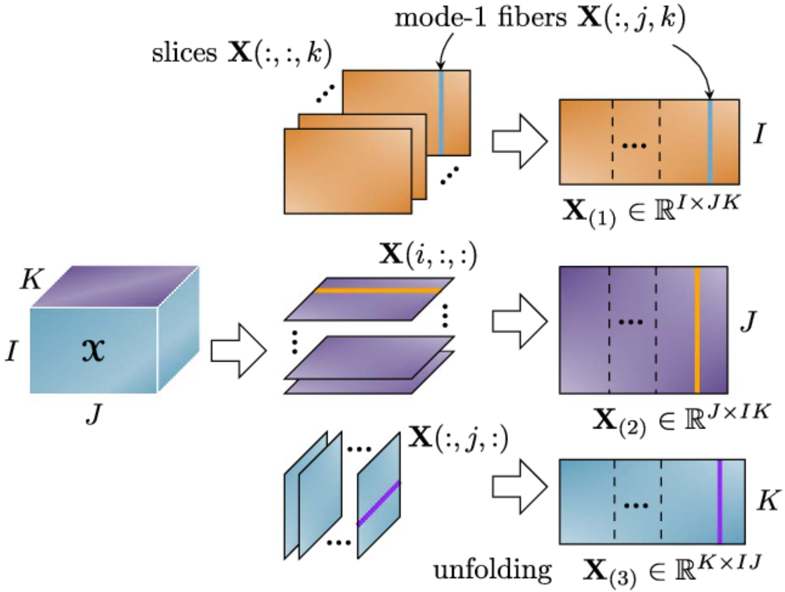
Invariâncias e unicidade

- Sabendo a matriz de transformada homogênea entre as câmeras $M_{AB} = M_{BA}^T$

$$X_{A(3)} = M_{BA} X_{B(3)} \tag{13}$$



I_l
I_a
I_u
I_a
I_u
$A_{(1)}$
I_l
I_l
I_a
I_u
I_u
I_l
$A_{(2)}$
I_a
I_l
I_a
I_u
I_l
I_a
$A_{(3)}$
I_u

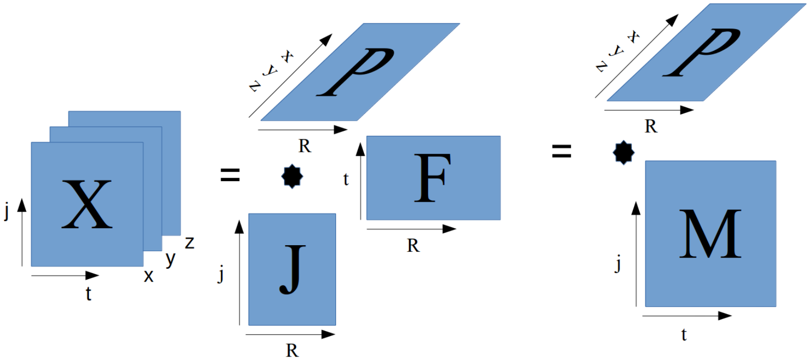

mode-1 fibers $\mathbf{X}(:, j, k)$
slices $\mathbf{X}(:, :, k)$
I
$\mathbf{X}_{(1)} \in \mathbb{R}^{I \times JK}$
$\mathbf{X}(i, :, :)$
K
I
$\mathcal{X}$
J
J
$\mathbf{X}_{(2)} \in \mathbb{R}^{J \times IK}$
$\mathbf{X}(:, j, :)$
K
unfolding
$\mathbf{X}_{(3)} \in \mathbb{R}^{K \times IJ}$

- Usando a propriedade tensorial:

$$
\begin{aligned}
X &= U_1 \times U_2 \times \cdots \times U_N && (14) \\
MX_{(n)} &= (U_1 \times U_2 \times \cdots \times (MU_n) \times \cdots \times U_N)_{(n)} && (15)
\end{aligned}
$$

Exemplo:

$$
\begin{aligned}
X &= U_1 \times U_2 \times U_3 && (16) \\
MX_{(2)} &= (U_1 \times (MU_2) \times U_3)_{(2)} && (17)
\end{aligned}
$$


X
j
t
x
y
z
=
J
j
R
P
z y x
R
F
t
R
=
P
z y x
R
M
j
t

- Aplicando às amostras de mocap.

$$\begin{aligned} X_{A(3)} &= M_{BA} X_{B(3)} && (18) \\ (J_A \times F_A \times P_A)_{(3)} &= M_{BA} \left(J_B \times F_B \times P_B \right)_{(3)} && (19) \\ (J_A \times F_A \times P_A)_{(3)} &= \left(J_B \times F_B \times (M_{BA} P_B) \right)_{(3)} && (20) \end{aligned}$$

desfazendo a matricialização:

$$\begin{aligned} J_A \times F_A \times P_A &= J_B \times F_B \times (M_{BA} P_B) && (21) \\ P_A &= M_{BA} P_B && (22) \\ M &\triangleq J_A \times F_A = J_B \times F_B && (23) \end{aligned}$$

Invariâncias e unicidade II

Teoricamente, para qualquer movimento i capturado por quaisquer câmeras $c = 1,...,C$, as amostras apresentam invariância espacial no espaço tensorial das juntas e dos frames (dimensões $j \times t$):

$$
\begin{aligned}
X_{i1} &= M_i \times_3 P_1 && (24)\\
X_{i2} &= M_i \times_3 P_2 && (25)\\
&\vdots && (26)\\
X_{iC} &= M_i \times_3 P_C && (27)\\
& && (28)
\end{aligned}
$$

- Na prática, pela dificuldade de obtenção de unicidade de solução, não funciona.

Invariâncias e unicidade III

- Duas possíveis soluções: modificar o problema de otimização usado na decomposição adicionando restrições tais que limitam o número de soluções, eliminando (ou reduzindo) mínimos locais; modificando a função objetivo usando no p.o. da decomposição.

Projeto de pesquisa

Projeto de pesquisa

- Diagnóstico de doença de Alzheimer e autismo. Base de dados massiva e multimodal (fornecido por instituição da área de neurociências).
- Novo modelo de aprendizado de máquina que engloba: fusão de dados (*data fusion*), análise multilinear e classificador baseado em redes neurais.
- Publicação do modelo matemático e algoritmo base em revistas e conferências especializadas.
- Instituições da área de tecnologia estão convidadas a formação de parceria.
- Desenvolvimento de duas APIs (com IP e patente de software). Num primeiro momento, baseados em Python, Google TensorFLow (usando Cloud TPU) e Docker.

Obrigado pela atenção.

- E-mail: prof.israel@gmail.com
- Linkedin: https://www.linkedin.com/in/prof-igo/

(a) e-mail

(b) Linkedin

A. Novikov, D. Podoprikhin, A. Osokin, and D. P. Vetrov, "Tensorizing neural networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 442–450.

K. Sato, C. Young, and D. Patterson, "An in-depth look at google's first tensor processing unit (tpu)," *Google Cloud Big Data and Machine Learning Blog*, vol. 12, 2017.

The MathWorks Inc., "Matlab, version 8.3.0.532 (r2014a)," Natick, Massachusetts, 2014.

Q. Li and D. Schonfeld, "Multilinear discriminant analysis for higher-order tensor data classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 12, pp. 2524–2537, Dec 2014.

F. Song, D. Zhang, D. Mei, and Z. Guo, "A multiple maximum scatter difference discriminant criterion for facial feature extraction," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 6, pp. 1599–1606, Dec 2007.

S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang, and H. J. Zhang, "Multilinear discriminant analysis for face recognition," *IEEE Transactions on Image Processing*, vol. 16, no. 1, pp. 212–220, Jan 2007.

H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "A survey of multilinear subspace learning for tensor data," *Pattern Recognition*, vol. 44, no. 7, pp. 1540 – 1551, 2011.

N. Vervliet, O. Debals, L. Sorber, M. Van Barel, and L. De Lathauwer, "Tensorlab user guide," Available online, 2016.

G. Guennebaud, B. Jacob *et al.*, "Eigen v3," http://eigen.tuxfamily.org, 2010.

B. W. Bader, T. G. Kolda, J. Sun, D. M. Dunlavy, E. Acar, J. Mayo, E. C. Chi, and S. Hansen, "Matlab tensor toolbox version 2.6," Available online, February 2015. [Online]. Available: http://www.sandia.gov/~tgkolda/TensorToolbox/

- Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 1110–1118.
- D. Shah, P. Falco, M. Saveriano, and D. Lee, "Encoding human actions with a frequency domain approach," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 5304–5311.
- F. Zhou and F. D. la Torre, "Generalized canonical time warping," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 279–294, Feb 2016.
- M. Balazia and P. Sojka, "Learning robust features for gait recognition by maximum margin criterion," in *2016 23rd International Conference on Pattern Recognition (ICPR)*, Dec 2016, pp. 901–906.