All-at-once Optimization for Mining Higher-order Tensors

Evrim Acar
Faculty of Life Sciences
University of Copenhagen

Tamara G. Kolda, Sandia National Labs, Livermore, CA
Daniel M. Dunlavy, Sandia National Labs, Albuquerque, NM
Matrix Factorizations in Data Mining

Capturing underlying hidden factors

$\mathbf{X}$  
users  
terms

$\approx$

$\begin{align*}
\text{topic 1} & : \mathbf{a}_1 \cdot \mathbf{b}_1 \\
\text{topic 2} & : \mathbf{a}_2 \cdot \mathbf{b}_2 \\
\vdots & \\
\text{topic R} & : \mathbf{a}_R \cdot \mathbf{b}_R
\end{align*}$
Matrix Factorizations in Data Mining

Capturing underlying hidden factors

Incomplete Matrix Factorization/Matrix Completion

\[ \mathbf{X} \approx a_1 b_1 + a_2 b_2 + \cdots + a_R b_R \]

\[ \mathbf{X} \approx \mathbf{\hat{X}} \]
Matrix Factorizations in Data Mining

Capturing underlying hidden factors

Incomplete Matrix Factorization/
Matrix Completion

Collective Matrix Factorization
Data sets are often multi-modal...

Capturing underlying hidden factors

Incomplete Tensor Factorization / Tensor Completion

Coupled Matrix-Tensor Factorization
Tensor model of our interest: CANDECOMP/ PARAFAC (CP)

CANDECOMP/PARAFAC (CP): [Hitchcock, 1927; Harshman, 1970; Carroll & Chang, 1970]

\[ \mathbf{X} = \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \]

N-mode vector outer product

\[ \mathbf{X} \in \mathbb{R}^{I \times J \times K}, \mathbf{a} \in \mathbb{R}^{I}, \mathbf{b} \in \mathbb{R}^{J}, \mathbf{c} \in \mathbb{R}^{K} \]

\[ \mathbf{X} = \mathbf{a} \circ \mathbf{b} \circ \mathbf{c} \text{ iff } x_{ijk} = a_i b_j c_k \]

\[ \mathbf{A} \in \mathbb{R}^{I \times R} = [\mathbf{a}_1 \ldots \mathbf{a}_R] \]

\[ \mathbf{B} \in \mathbb{R}^{J \times R} = [\mathbf{b}_1 \ldots \mathbf{b}_R] \]

\[ \mathbf{C} \in \mathbb{R}^{K \times R} = [\mathbf{c}_1 \ldots \mathbf{c}_R] \]
Neurologists visually analyze EEG recordings and identify **WHERE** a seizure starts.

→ Visual analysis is time-consuming.
→ Visual analysis is subjective and error-prone due to fatigue, etc.

**GOAL:** Can we localize seizure origin automatically?
Epilepsy Tensors

Channels → Time

Time samples → CWT

Channels → Scales (freq.)

Channels
First component: Eye Artifact

Epileptic Seizure Localization:

- **Temporal**
- **Spectral**
- **Spatial**
Second component: Seizure

Epileptic Seizure Localization:

- Temporal
- Spectral
- Spatial

![Diagram showing the relationship between time samples, scales, and channels for seizure localization.](image-url)
CP is very popular!

• **Chemometrics**
  – Fluorescence Spectroscopy
  – Chromatographic Data Analysis

• **Neuroscience**
  – Epileptic Seizure Localization
  – Analysis of EEG and ERP

• **Signal Processing**
  – Blind Source Separation

• **Computer Vision**
  – Face detection

• **Social Network Analysis**
  – Web link analysis
  – Conversation detection in emails
  – Link prediction

• **Many more…**

---


Acar et al., *Bioinformatics, 2007.*

Hazan, Polak and Shashua, *ICCV 2005.*

Bader, Berry, Browne, *Survey of Text Mining: Clustering, Classification, and Retrieval, 2nd Ed., 2007.*
Fitting CP is challenging!

- Alternating Least Squares (ALS) \cite{CarrollChang1970, Harshman1970} >> computationally efficient but not always accurate, i.e., not robust to overfactoring.

- All-at-once optimization:
  - Gauss-Newton approach \cite{Paatero1997, TomasiBro2006} >> accurate but has scalability issues

- ALS-based imputation \cite{Kiers1997, WalczakMassart2001} >> due to imputation, do not scale to large-scale data sets.

- All-at-once optimization:
  - Gauss-Newton approach \cite{Paatero1997, TomasiBro2005} >> has scalability issues

Alternating Least Squares (ALS) \cite{Wilderjans2009} >> not always accurate, i.e., not robust to overfactoring.
Fitting CP is challenging!

- Alternating Least Squares (ALS) [Carroll & Chang, 1970; Harshman, 1970]
  \>>> \textit{computationally efficient but not always accurate, i.e., not robust to overfactoring.}

- All-at-once optimization:
  - Gauss-Newton approach [Paatero, 1997; Tomasi and Bro, 2006]
    \>>> \textit{accurate but has scalability issues}

- ALS-based imputation [Kiers, 1997; Walczak & Massart, 2001]
  \>>> \textit{due to imputation, do not scale to large-scale data sets.}

- All-at-once optimization:
  - Gauss-Newton approach [Paatero, 1997; Tomasi and Bro, 2005]
    \>>> \textit{has scalability issues}

Alternating Least Squares (ALS) [Wilderjans, 2009]
\>>> \textit{not always accurate, i.e., not robust to overfactoring.}

\textbf{GOAL:} Scalable and accurate algorithms for fitting a CANDECOMP/PARAFAC model

\textit{Proposed Approach: Gradient-based optimization}
CP is a Nonlinear Optimization Problem

\[ \mathbf{X} = \sum_{i=1}^{R} \mathbf{a}_i \mathbf{b}_i^\top + \cdots + \sum_{i=1}^{R} \mathbf{a}_R \mathbf{b}_R^\top \]

Goal is to find matrices \( \mathbf{A}, \mathbf{B}, \mathbf{C} \) that solve the following nonlinear optimization problem:

\[
\min_{\mathbf{A}, \mathbf{B}, \mathbf{C}} \| \mathbf{X} - [\mathbf{A}, \mathbf{B}, \mathbf{C}] \|^2
\]

**Alternating Approach (ALS)**

Initialize \( \mathbf{B} \) and \( \mathbf{C} \)

for \( k = 1, \ldots \)

\[
\min_{\mathbf{A}} \| \mathbf{X} - [\mathbf{A}, \mathbf{B}, \mathbf{C}] \|^2
\]

\[
\min_{\mathbf{B}} \| \mathbf{X} - [\mathbf{A}, \mathbf{B}, \mathbf{C}] \|^2
\]

\[
\min_{\mathbf{C}} \| \mathbf{X} - [\mathbf{A}, \mathbf{B}, \mathbf{C}] \|^2
\]

end
CP is a Nonlinear Optimization Problem

Goal is to find matrices \( A, B, C \) that solve the following nonlinear optimization problem:

\[
\min_{A,B,C} \| X - [A, B, C] \|^2
\]

**Alternating Approach (ALS)**

Initialize \( B \) and \( C \) for \( k = 1, \ldots \)

\[
\begin{align*}
\min_A & \| X - [A, B, C] \|^2 \\
\min_B & \| X - [A, B, C] \|^2 \\
\min_C & \| X - [A, B, C] \|^2 \\
\end{align*}
\]

end

Each step can be converted to a least squares problem:

\[
\min_A \| X_{(1)} - A(C \odot B)^T \|^2
\]

\[
A = X_{(1)} \left( (C \odot B)^T \right)^+ 
\]

Very fast, but not always accurate!
CP is a Nonlinear Optimization Problem

Goal is to find matrices $A$, $B$, $C$ that solve the following nonlinear optimization problem:

$$
\min_{A,B,C} \| \mathbf{X} - [A, B, C] \|^2
$$

**Alternating Approach (ALS)**

Initialize $B$ and $C$

for $k = 1, \ldots$

$$
\min_A \| \mathbf{X} - [A, B, C] \|^2
\min_B \| \mathbf{X} - [A, B, C] \|^2
\min_C \| \mathbf{X} - [A, B, C] \|^2
$$

end

Very fast, but not always accurate!

**All-at-once Optimization**

1st order optimization

- Paatero, 1999;
- Acar, Dunlavy and Kolda, 2011
- Nonlinear CG

2nd order optimization

- Paatero, 1997;
- Tomasi & Bro, 2006
- Gauss-Newton is used to solve the nonlinear least squares problem.
- Jacobian is $IJK$ by $(I+J+K)R$

Not scalable!
Our gradient-based approach: CPOPT

CPOPT is a general gradient-based optimization approach for fitting the CP model:

\[
\min_{A,B,C} \| X - [A, B, C] \|^2
\]

Define the objective function:

\[
f(x) = \| X - [A, B, C] \|^2
\]

\[
x = \begin{bmatrix}
a_1 \\
a_R \\
b_1 \\
b_R \\
c_1 \\
c_R
\end{bmatrix}
\]

\[
\nabla f(x) =
\begin{bmatrix}
\frac{\partial f}{\partial a_1} \\
\vdots \\
\frac{\partial f}{\partial a_R} \\
\frac{\partial f}{\partial b_1} \\
\vdots \\
\frac{\partial f}{\partial b_R} \\
\frac{\partial f}{\partial c_1} \\
\vdots \\
\frac{\partial f}{\partial c_R}
\end{bmatrix}
\]

Pick a first-order optimization method

Once we have the gradient, we can use any first-order optimization method, e.g., NCG and LBFGS.

Our Algorithm: Nonlinear CG with Hestenes-Stiefel updates and Moré-Thuente line search as implemented in Poblano Toolbox [Dunlavy, Kolda, Acar, 2009]
Gradient in Matrix Form

Objective Function

\[ f(x) = \frac{1}{2} \| X - [A, B, C] \|^2 \]

Gradient

\[
\frac{\partial f}{\partial A} = -X_{(1)}(C \odot B) + A(C^T C \ast B^T B)
\]

\[
\frac{\partial f}{\partial B} = -X_{(2)}(C \odot A) + B(C^T C \ast A^T A)
\]

\[
\frac{\partial f}{\partial C} = -X_{(3)}(B \odot A) + C(B^T B \ast A^T A)
\]

Note the relation with ALS:

\[ A = X_{(1)}(C \odot B)^\dagger = X_{(1)}(C \odot B)(C^T C \ast B^T B)^\dagger \]
CPOPT is robust to overfactoring!

Amino Acid Data
http://www.models.life.ku.dk/
CPOPT is robust to overfactoring!

Amino Acid Data
http://www.models.life.ku.dk/
CPOPT is computationally efficient!

Computational time for fitting an $R$-component CP model:

Tensor Size: 50 x 50 x 50

Tensor Size: 250 x 250 x 250

Computational Cost (per iteration) of fitting an $R$-component CP model to a $K \times K \times K$ tensor:

$O(RK^3)$  $O(R^3K^3)$  $O(RK^3)$
In the presence of missing data…

Given tensor $\mathbf{X}$ and $R$ (# of components), find matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ that solve the following problem:

**Optimization Problem**

$$\min_{\mathbf{A}, \mathbf{B}, \mathbf{C}} \| \mathbf{W}^\ast (\mathbf{X} - [\mathbf{A}, \mathbf{B}, \mathbf{C}]) \|^2$$

$$w_{ijk} = \begin{cases} 1 & \text{if } x_{ijk} \text{ is known,} \\ 0 & \text{if } x_{ijk} \text{ is missing.} \end{cases}$$

**Objective Function**

$$f_\mathbf{W}(\mathbf{x}) = \| \mathbf{W}^\ast (\mathbf{X} - [\mathbf{A}, \mathbf{B}, \mathbf{C}]) \|^2$$

where the vector $\mathbf{x}$ comprises the entries of $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ stacked column-wise:

$$x = \begin{bmatrix} a_1 \\ \vdots \\ a_R \\ b_1 \\ \vdots \\ b_R \\ c_1 \\ \vdots \\ c_R \end{bmatrix}$$

$R(I + J + K)$ variables
Our gradient-based approach: CP-WOPT

**CP-WOPT (CP Weighted OPTimization):** It is a general gradient-based approach for solving the weighted least squares problem for the CP model.

Define \( y = W \cdot x \) and \( z = W \cdot [A, B, C] \)

\[
\begin{align*}
    f_W(A, B, C) &= \| W \cdot (x - [A, B, C]) \|^2 = \| y - z \|^2
\end{align*}
\]

Gradient can be computed as follows:

\[
\begin{align*}
    \frac{\partial f_W}{\partial A} &= 2(Z_{(1)} - Y_{(1)})(C \odot B) \\
    \frac{\partial f_W}{\partial B} &= 2(Z_{(2)} - Y_{(2)})(C \odot A) \\
    \frac{\partial f_W}{\partial C} &= 2(Z_{(3)} - Y_{(3)})(B \odot A)
\end{align*}
\]
If there is a lot of missing data:
Sparse $\mathcal{W}$

CP-WOPT (CP Weighted OPTimization): It is a general gradient-based approach for solving the weighted least squares problem for the CP model.

Define $\mathbf{y} = \mathcal{W} \ast \mathbf{x}$ and $\mathbf{z} = \mathcal{W} \ast [\mathbf{A}, \mathbf{B}, \mathbf{C}]$

Gradient can be computed as follows:

$$\frac{\partial f_{\mathcal{W}}}{\partial \mathbf{A}} = 2 (\mathbf{Z}(1) - \mathbf{Y}(1)) (\mathbf{C} \odot \mathbf{B})$$

$$\frac{\partial f_{\mathcal{W}}}{\partial \mathbf{B}} = 2 (\mathbf{Z}(2) - \mathbf{Y}(2)) (\mathbf{C} \odot \mathbf{A})$$

$$\frac{\partial f_{\mathcal{W}}}{\partial \mathbf{C}} = 2 (\mathbf{Z}(3) - \mathbf{Y}(3)) (\mathbf{B} \odot \mathbf{A})$$

Sparse tensors with the same sparsity pattern

Efficient Sparse Computations

Algorithm: Nonlinear CG with Hestenes-Stiefel updates and Moré-Thuente line search
Details on Alternative Techniques

\[ f_{\mathcal{W}}(\mathbf{x}) = \| \mathcal{W} \ast (\mathbf{X} - [A, B, C]) \|^2 \]

**ALS-based Imputation (EM-ALS):**

**Step 1:** Initialize matrices A, B, and C

**Step 2:** Complete the missing entries of X using the current model

\[ \tilde{\mathbf{X}} = \mathcal{W} \ast \mathbf{X} + (1 - \mathcal{W}) \ast [A, B, C] \]

**Step 3:** Compute A, B and C via alternating least squares, i.e.,

\[ A = \arg \min_A \| \tilde{\mathbf{X}} - [A, B, C] \| \]

**Step 4:** Go back to Step 2 until convergence

**NLS approach extended to missing data:**

- Developed by Paatero, 1997 and Tomasi and Bro, 2005. We use INDAFAC by Tomasi and Bro.
- Fits the model only to the known data entries just like CP-WOPT
- Uses Gauss-Newton
- Jacobian is \((1-M)IJK\) by \((I+J+K)R\), where \(M\) is the missing data rate.
**Experimental Set-up**

**Step 1:** Generate random factor matrices $A, B, C$ with $R = 5$ each.

**Step 2:** Construct tensor from factor matrices and add noise ($\eta = 10\%$)

$$x_{\text{full}} = [A, B, C] + \eta \frac{\|X\|}{\|N\|} N$$

30 triplets

**Step 3:** Randomly remove some entries ($M = 60\%, 80\%, 90\%, 95\%$)

**Step 4:** Factorize incomplete tensor with EM-ALS, INDAFAC and CP-WOPT (dense and sparse)
**Experimental Set-up**

**Step 1**: Generate random factor matrices $A$, $B$, $C$ with $R = 5$ each.

Compute factor match score (FMS):

$$\frac{1}{R} \sum_{r=1}^{R} \left( 1 - \frac{|\lambda_r - \hat{\lambda}_r|}{\max\{\lambda_r, \hat{\lambda}_r\}} \right) a_r^T \hat{a}_r b_r^T \hat{b}_r c_r^T \hat{c}_r$$

**Step 2**: Construct tensor from factor matrices and add noise ($\eta = 10\%$)

$$X_{\text{full}} = [A, B, C] + \eta \frac{\|X\|}{\|N\|} N$$

**Step 3**: Randomly remove some entries ($M = 60\%, 80\%, 90\%, 95\%$)

**Step 4**: Factorize incomplete tensor with **EM-ALS**, **INDAFAC** and **CP-WOPT** (dense and sparse)
CP-WOPT is as accurate as other algorithms!

- **Data set size:** 50 x 40 x 30
- **Number of components (R):** 5
- **Missing data pattern:** randomly missing entries
- **CP-WOPT** dense (D) and sparse (S) versions compared with INDAFAC and EM-ALS
- **Initializations** based on n-mode singular vectors plus 4 random starting points
CP-WOPT is as accurate as other algorithms!

- Data set size: 50 x 40 x 30
- Number of components (R): 5
- Missing data pattern: randomly missing entries
- CP-WOPT dense (D) and sparse (S) versions compared with INDAFAC and EM-ALS
- Initializations based on n-mode singular vectors plus 4 random starting points
Larger problems are easier!

- **Data set size:** $100 \times 80 \times 60$
- **Number of components** $(R)$: 5
- **Missing data pattern:** randomly missing entries
- **Smaller values of ratio** $\rho$ indicate more difficult problems:

$$\rho = \frac{\text{Number of known tensor entries}}{\text{Number of variables}}$$
CP-WOPT is Fast!

- Data set size:
  - 50 x 40 x 30
  - 100 x 80 x 60
  - 150 x 120 x 90
- Number of components (R): 5
- Missing data pattern: randomly missing entries
- CP-WOPT dense (D) and sparse (S) versions compared with INDAFAC and EM-ALS
- Timing = sum of the total computation time for all starting points

As data gets sparse, sparse version of CP-WOPT is the fastest!
CP-WOPT scales to larger problems!

500 x 500 x 500 with M=99%  
(1.25 million nonzeros)

1000 x 1000 x 1000 with M = 99.5%  
(5 million nonzeros)

Dense storage = 1GB  
Sparse storage = 40MB

Dense storage = 8GB  
Sparse storage = 160MB
GOAL: To differentiate between left and right hand stimulation

CP-WOPT Application: EEG Analysis with Missing Channels

[Acar, Dunlavy, Kolda and Mørup, 2011]
CP-WOPT Application: EEG Analysis with Missing Channels

[Acarr, Dunlavy, Kolda and Morup, 2011]

GOAL: To differentiate between left and right hand stimulation

No Missing Data

30 Channels/slice Missing
**GOAL:** Joint analysis of data from multiple sources in order to capture the underlying latent structures

**OUR APPROACH:** All-at-once optimization solving for all factor matrices simultaneously (CMTF-OPT)

\[
f(A, B, C, D) = \frac{1}{2} \| X - [A, B, C] \|^2 + \frac{1}{2} \| Y - AD^T \|^2
\]

\[
\frac{\partial f}{\partial A} = -X(1)(C \odot B) + A(C^T C \ast B^T B) - YD + AD^T D
\]

\[
\frac{\partial f}{\partial B} = -X(2)(C \odot A) + B(C^T C \ast A^T A)
\]

\[
\frac{\partial f}{\partial C} = -X(3)(B \odot A) + C(B^T B \ast A^T A)
\]

\[
\frac{\partial f}{\partial D} = -Y^T A + DA^T A
\]

We can use first-order optimization once we have the gradient.

Easily extends to many data sets!
CMTF is also useful for missing data recovery

In many applications such as recommendation systems, the goal is to recover missing entries.

\[
f(A, B, C, D) = \frac{1}{2} \| \mathbf{W} \ast (\mathbf{X} - [A, B, C]) \|^2 + \frac{1}{2} \| Y - AD^T \|^2
\]

\[
\omega_{ijk} = \begin{cases} 
1 & \text{if } x_{ijk} \text{ is known,} \\
0 & \text{if } x_{ijk} \text{ is missing.}
\end{cases}
\]

CMTF can deal with larger amounts of missing data!

Tensor Completion Score (TCS)

\[
\text{TCS} = \frac{\| (1 - \mathbf{W}) \ast (\mathbf{X} - \hat{\mathbf{X}}) \|}{\| (1 - \mathbf{W}) \ast \mathbf{X} \|}
\]

\[\hat{\mathbf{X}} = [\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}], \text{ where } \hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}} \text{ are the extracted factor matrices.}\]
Experimental Set-up

Data Generation:
- Generate factor matrices

Construct tensor $\mathbf{X}$ and matrix $\mathbf{Y}$ and add noise

$\mathbf{X} = [\mathbf{A}, \mathbf{B}, \mathbf{C}] + \eta \mathbf{N}$

$\mathbf{Y} = \mathbf{A}\mathbf{D}^\top + \eta \mathbf{N}$

Solve $f$ using CMTF-ALS and CMTF-OPT

$$f(\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}, \hat{\mathbf{D}}) = \frac{1}{2} \| \mathbf{x} - [\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}] \|^2 + \frac{1}{2} \| \mathbf{Y} - \hat{\mathbf{A}}\hat{\mathbf{D}}^\top \|^2$$

Accuracy as the performance metric: The algorithm is accurate if original factors, i.e., $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$, match with the extracted factors, i.e., $\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}, \hat{\mathbf{D}}$

$$\text{FMS} = \min_r (1 - \frac{|\xi_r - \hat{\xi}_r|}{\max(|\xi_r|, |\hat{\xi}_r|)}) a_r^\top \hat{\alpha}_r b_r^\top \hat{\beta}_r c_r^\top \hat{\gamma}_r d_r^\top \hat{\delta}_r$$

where $\xi_r = \lambda_r + \alpha_r$, for $r = 1, 2, \ldots, R$

$\mathbf{X} = [\mathbf{A}, \mathbf{B}, \mathbf{C}] = \sum_{r=1}^{R} \lambda_r \mathbf{a}_r \odot \mathbf{b}_r \odot \mathbf{c}_r$

$\mathbf{Y} = \mathbf{A}\mathbf{D}^\top = \sum_{r=1}^{R} \alpha_r \mathbf{a}_r \odot \mathbf{d}_r$
CMTF-OPT is more robust to overfactoring!

\( \lambda_r = \alpha_r = 1 \)

<table>
<thead>
<tr>
<th>Noise ( \eta )</th>
<th>( \bar{R} )</th>
<th>ALG.</th>
<th>Success (%)</th>
<th>Mean FMS</th>
<th>Success (%)</th>
<th>Mean FMS</th>
<th>Success (%)</th>
<th>Mean FMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \eta = 0.10 )</td>
<td>R ( R )</td>
<td>OPT</td>
<td>100.0</td>
<td>1.00</td>
<td>96.7</td>
<td>0.97</td>
<td>100.0</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ALS</td>
<td>100.0</td>
<td>1.00</td>
<td>96.7</td>
<td>0.97</td>
<td>96.7</td>
<td>0.96</td>
</tr>
<tr>
<td></td>
<td>R ( R + 1 )</td>
<td>OPT</td>
<td>96.7</td>
<td>0.97</td>
<td>100.0</td>
<td>1.00</td>
<td>96.7</td>
<td>0.97</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ALS</td>
<td>3.3</td>
<td>0.29</td>
<td>6.7</td>
<td>0.71</td>
<td>0.0</td>
<td>0.06</td>
</tr>
<tr>
<td>( \eta = 0.25 )</td>
<td>R ( R )</td>
<td>OPT</td>
<td>100.0</td>
<td>0.99</td>
<td>100.0</td>
<td>1.00</td>
<td>100.0</td>
<td>0.96</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ALS</td>
<td>100.0</td>
<td>0.99</td>
<td>100.0</td>
<td>1.00</td>
<td>100.0</td>
<td>0.99</td>
</tr>
<tr>
<td></td>
<td>R ( R + 1 )</td>
<td>OPT</td>
<td>100.0</td>
<td>0.99</td>
<td>100.0</td>
<td>1.00</td>
<td>100.0</td>
<td>0.99</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ALS</td>
<td>6.7</td>
<td>0.34</td>
<td>16.7</td>
<td>0.70</td>
<td>10.0</td>
<td>0.14</td>
</tr>
<tr>
<td>( \eta = 0.35 )</td>
<td>R ( R )</td>
<td>OPT</td>
<td>100.0</td>
<td>0.99</td>
<td>100.0</td>
<td>1.00</td>
<td>100.0</td>
<td>0.99</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ALS</td>
<td>100.0</td>
<td>0.99</td>
<td>96.7</td>
<td>0.97</td>
<td>100.0</td>
<td>0.99</td>
</tr>
<tr>
<td></td>
<td>R ( R + 1 )</td>
<td>OPT</td>
<td>90.0</td>
<td>0.92</td>
<td>100.0</td>
<td>1.00</td>
<td>86.7</td>
<td>0.88</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ALS</td>
<td>33.3</td>
<td>0.55</td>
<td>13.3</td>
<td>0.71</td>
<td>36.7</td>
<td>0.39</td>
</tr>
</tbody>
</table>

\( \lambda_r, \alpha_r \geq 1 \)

Harder set of problems: Accuracies for both algorithms go down but the relative performance is the same. [Acar, Dunlavy and Kolda, 2011]
GOAL: To compare In Vitro (i.e., engineered cultures of human cells grown in vitro) and Native tissue samples and identify the structural features that can differentiate between healthy and diseased tissues.

**DATA:**
- **Native** tissue samples
- **In Vitro** tissue samples

**Construction of cell graphs**
- healthy
- cancer

**Feature Extraction**
- E.g., average degree, diameter, radius, average path length, number of connected components, …

[Acar, Plopper and Yener, 2011]
2-comp. CP Model of In Vitro Tissue Samples
Coupled Analysis of Native and In Vitro Tissue Samples

First component differentiates between cancer and normal!
Coupled Analysis of Native and In Vitro Tissue Samples

First component differentiates between cancer and normal!
Challenge: Scalable and accurate algorithms for
- Fitting a CP model
- Fitting a CP model to incomplete data
- For coupled matrix and tensor factorizations

Our Approach: All-at-once optimization using gradient-based algorithms
- CPOPT/CP-WOPT for fitting a CP model
  - As accurate as alternative techniques
  - Scales to larger data sets compared to alternative methods.
- CMTF-OPT for modeling coupled data sets
  - Builds onto CPOPT/CP-WOPT and has the same nice properties

Future Work:
- Coupled Factorizations:
  - Constraints for better interpretability – sparsity and non-negativity
  - Applications in biomedical informatics
Thank you!

Evrim Acar

evrim@life.ku.dk

http://www.models.life.ku.dk/users/acare/

SURVEY


ALGORITHMS


APPLICATIONS
