Roadmap

- Motivation
- Matrix tools
- Tensor tools
- Case studies

- Basics
- Tucker
- PARAFAC
- Incrementalization
Incrementalization
Incremental Tensor Decomposition

• Dynamic data model
  – Tensor Streams
• Dynamic Tensor Decomposition (DTA)
• Streaming Tensor Decomposition (STA)
• Window-based Tensor Decomposition (WTA)
Dynamic Tensor Stream

- Streams come with structure
  - (time, source, destination, port)
  - (time, author, keyword)
- How to summarize tensor streams effectively and incrementally?
**Dynamic Data model**

- Tensor Streams
  - A sequence of Mth order tensor
    \[ \chi_1 \ldots \chi_n \text{ where } \chi_i \in \mathbb{R}^{N_1 \times \ldots \times N_M} \]

n is increasing over time

<table>
<thead>
<tr>
<th>Order</th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correspondence</td>
<td>Multiple streams</td>
<td>Time evolving graphs</td>
<td>3D arrays</td>
</tr>
</tbody>
</table>

**Example**

[Diagram showing time, sensors, authors, keywords, sources, destinations, and a 3D array labeled \( \chi \)]

Faloutsos, Kolda, Sun
Incremental Tensor Decomposition

Dynamic data model

- Tensor Streams
- Dynamic Tensor Decomposition (DTA)
- Streaming Tensor Decomposition (STA)
- Window-based Tensor Decomposition (WTA)

2. Jimeng Sun, Dacheng Tao, Christos Faloutsos. Beyond Streams and Graphs: Dynamic Tensor Analysis, *KDD 2006*
Incremental Tensor Decomposition

Old Tensors  New Tensor

Old cores

Faloutsos, Kolda, Sun
1st order DTA - problem

Given \( x_1 \ldots x_n \) where each \( x_i \in \mathbb{R}^N \), find \( U \in \mathbb{R}^{N \times R} \) such that the error \( e \) is small:

\[
e = \sum_{i=1}^{n} \left\| x_i - x_iUU^T \right\|_F^2
\]

Note that \( Y = XU \)
**1\textsuperscript{st} order Dynamic Tensor Analysis**

**Input:** new data vector $x \in \mathbb{R}^N$, old variance matrix $C \in \mathbb{R}^{N \times N}$

**Output:** new projection matrix $U \in \mathbb{R}^{N \times R}$

**Algorithm:**
1. update variance matrix $C_{\text{new}} = x^T x + C$
2. Diagonalize $U \Lambda U^T = C_{\text{new}}$
3. Determine the rank $R$ and return $U$

Diagonalization has to be done for every new $x$!

Faloutsos, Kolda, Sun

4-9
**M**\(^{th}\) order DTA

- **Construct Variance Matrix of Incremental Tensor**
  - \(\mathbf{X}_{(d)} \times \mathbf{X}_{(d)}^T = \mathbf{X}_{(d)}^T \mathbf{X}_{(d)}\)

- **Reconstruct Variance Matrix**
  - \(\mathbf{U}_d^T \mathbf{S}_d \mathbf{U}_d = \mathbf{C}_d\)

- **Diagonalize Variance Matrix**

- **Update Variance Matrix**
M\textsuperscript{th} order DTA – complexity

Storage:
\( O(\prod N_i) \), i.e., size of an input tensor at a single timestamp

Computation:
\[ \sum N_i^3 \text{ (or } \sum N_i^2) \text{ diagonalization of } C \]
\[ + \sum N_i \prod N_i \text{ matrix multiplication } X_{(d)\!}^T X_{(d)} \]

For low order tensor (<3), diagonalization is the main cost
For high order tensor, matrix multiplication is the main cost
Incremental Tensor Decomposition

😊 Dynamic data model
  • Tensor Streams

😊 Dynamic Tensor Decomposition (DTA)
  • Streaming Tensor Decomposition (STA)
  • Window-based Tensor Decomposition (WTA)

2. Jimeng Sun, Dacheng Tao, Christos Faloutsos. Beyond Streams and Graphs: Dynamic Tensor Analysis, *KDD 2006*
1st order Streaming Tensor Analysis (STA)

- Adjust U smoothly when new data arrive without diagonalization [VLDB05]

- For each new point \( x \)
  - Project onto current line
  - Estimate error
  - Rotate line in the direction of the error and in proportion to its magnitude

For each new point \( x \) and for \( i = 1, \ldots, k \):

- \( y_i := U_i^T x \) (proj. onto \( U_i \))
- \( d_i \leftarrow \lambda d_i + y_i^2 \) (energy \( \propto i\)-th eigenval.)
- \( e_i := x - y_i U_i \) (error)
- \( U_i \leftarrow U_i + \frac{1}{d_i} y_i e_i \) (update estimate)
- \( x \leftarrow x - y_i U_i \) (repeat with remainder)
M\textsuperscript{th} order STA

- Run 1\textsuperscript{st} order STA along each mode
- Complexity:
  - Storage: $O(\prod N_i)$
  - Computation: $\sum R_i \prod N_i$ which is smaller than DTA
Incremental Tensor Decomposition

😊 Dynamic data model
  • Tensor Streams

😊 Dynamic Tensor Decomposition (DTA)

😊 Streaming Tensor Decomposition (STA)
  • Window-based Tensor Decomposition (WTA)

2. Jimeng Sun, Dacheng Tao, Christos Faloutsos. Beyond Streams and Graphs: Dynamic Tensor Analysis, *KDD 2006*
Window-based Tensor Analysis (WTA)

\[
\lambda_1(u_1 \circ v_1 \circ w_1)
\]

1st factor

Faloutsos, Kolda, Sun

4-16
Meta-algorithm for window-based tensor analysis

Input:
The tensor window $\mathcal{D} \in \mathbb{R}^{W \times R_0 \times \cdots \times R_M}$

Output:
The projection matrix $\mathbf{U}_0 \in \mathbb{R}^{W \times R_0}$, $\mathbf{U}_i |_{i=1}^M \in \mathbb{R}^{N_i \times R_i}$ and the core tensor $\mathcal{Y}$.

Algorithm:
1. Initialize $\mathbf{U}_i |_{i=0}^M$
2. Conduct 3 - 5 iteratively
3. For $k = 0$ to $M$
4. Fix $\mathbf{U}_i$ for $i \neq k$ and find the $\mathbf{U}_k$ that minimizes $d(\mathcal{D}, \mathcal{D} \prod_{i=0}^M x_i (\mathbf{U}_i \mathbf{U}_i^T))$
5. Check convergence
6. Calculate the core tensor $\mathcal{Y} = \mathcal{D} \prod_{i=0}^M x_i \mathbf{U}_i$

Faloutsos, Kolda, Sun
Moving Window scheme (MW)

- Update the variance matrix $C_{(i)}$ incrementally
- Diagonalize $C(i)$ to find $U(i)$

A good and efficient initialization
Roadmap

- Motivation
- Matrix tools
- Tensor tools
- Case studies
  - Sensors
  - Social networks
  - Web mining

Faloutsos, Kolda, Sun
P1: Environmental sensor monitoring

Temperature

Humidity

Light

Voltage

Faloutsos, Kolda, Sun

CMU SCS
**P1: sensor monitoring**

- **1st factor** consists of the main trends:
  - Daily periodicity on time
  - Uniform on all locations
  - Temp, Light and Volt are positively correlated while negatively correlated with Humid
P1: sensor monitoring

2nd factor captures an atypical trend:
- Uniformly across all time
- Concentrating on 3 locations
- Mainly due to voltage

Interpretation: two sensors have low battery, and the other one has high battery.

Faloutsos, Kolda, Sun 4-22
P3: Social network analysis

- Multiway latent semantic indexing (LSI)
  - Monitor the change of the community structure over time
### P3: Social network analysis (cont.)

<table>
<thead>
<tr>
<th>Authors</th>
<th>Keywords</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>michael carey, michael stonebreaker, h. jagadish, hector garcia-molina</td>
<td>queri, parallel, optimization, concurr, objectorient</td>
<td>1995</td>
</tr>
<tr>
<td>surajit chaudhuri, mitch cherniack, michael stonebreaker, ugur etintemel</td>
<td>distrib systems, view, storage, servic, process, cache</td>
<td>2004</td>
</tr>
<tr>
<td>jiawei han, jian pei, philip s. yu, jianyong wang, charu c. aggarwal</td>
<td>pattern, support, cluster, gener, queri</td>
<td>2004</td>
</tr>
</tbody>
</table>

- Two groups are correctly identified: Databases and Data mining
- People and concepts are drifting over time
P4: Network anomaly detection

- Reconstruction error gives indication of anomalies.
- Prominent difference between normal and abnormal ones is mainly due to the unusual scanning activity (confirmed by the campus admin).
P5: Web graph mining

• How to order the importance of web pages?
  – Kleinberg’s algorithm HITS
  – PageRank
  – Tensor extension on HITS (TOPHITS)
Kleinberg’s Hubs and Authorities (the HITS method)

Sparse adjacency matrix and its SVD:

\[ x_{ij} = \begin{cases} 1 & \text{if page } i \text{ links to page } j \\ 0 & \text{otherwise} \end{cases} \]

\[ X \approx \sum_r \sigma_r h_r o a_r \]

authority scores for 1\(^{st}\) topic + authority scores for 2\(^{nd}\) topic

hub scores for 1\(^{st}\) topic + hub scores for 2\(^{nd}\) topic
HITS Authorities on Sample Data

We started our crawl from http://www-neos.mcs.anl.gov/neos, and crawled 4700 pages, resulting in 560 cross-linked hosts.

authority scores for 1st topic
authority scores for 2nd topic

hub scores for 1st topic
hub scores for 2nd topic

Faloutsos, Kolda, Sun
Three-Dimensional View of the Web

Observe that this tensor is very sparse!

Faloutsos, Kolda, Sun
Kolda, Bader, Kenny, ICDM05
Topical HITS (TOPHITS)

**Main Idea:** Extend the idea behind the HITS model to incorporate term (i.e., topical) information.

\[ x \approx \sum_{r=1}^{R} \lambda_r \ h_r \circ a_r \]

Faloutsos, Kolda, Sun
Main Idea: Extend the idea behind the HITS model to incorporate term (i.e., topical) information.

\[ x \approx \sum_{r=1}^{R} \lambda_r \ h_r \circ a_r \circ t_r \]

Faloutsos, Kolda, Sun 4-31
### TPHITS Terms & Authorities on Sample Data

<table>
<thead>
<tr>
<th>1st Principal Factor</th>
<th>2nd Principal Factor</th>
<th>3rd Principal Factor</th>
<th>4th Principal Factor</th>
<th>5th Principal Factor</th>
<th>6th Principal Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>JAVA .23</td>
<td>SUN .18</td>
<td>PLATF .17</td>
<td>SOLAR .16</td>
<td>DEVEL .16</td>
<td>NO-READABLE-TEXT .20</td>
</tr>
<tr>
<td>SUN .18</td>
<td>JAVA .23</td>
<td>SUN .17</td>
<td>DLL .14</td>
<td>SOFTV .12</td>
<td>NO-READABLE-TEXT .99</td>
</tr>
<tr>
<td>PLATF .17</td>
<td>JAVA .23</td>
<td>SUN .17</td>
<td>DLL .14</td>
<td>SOFTV .12</td>
<td>NO-READABLE-TEXT .99</td>
</tr>
<tr>
<td>SOLAR .16</td>
<td>JAVA .23</td>
<td>SUN .17</td>
<td>DLL .14</td>
<td>SOFTV .12</td>
<td>NO-READABLE-TEXT .99</td>
</tr>
<tr>
<td>DEVEL .16</td>
<td>JAVA .23</td>
<td>SUN .17</td>
<td>DLL .14</td>
<td>SOFTV .12</td>
<td>NO-READABLE-TEXT .99</td>
</tr>
<tr>
<td>NO-READABLE-TEXT .20</td>
<td>JAVA .23</td>
<td>SUN .17</td>
<td>DLL .14</td>
<td>SOFTV .12</td>
<td>NO-READABLE-TEXT .99</td>
</tr>
<tr>
<td>NO-READABLE-TEXT .99</td>
<td>JAVA .23</td>
<td>SUN .17</td>
<td>DLL .14</td>
<td>SOFTV .12</td>
<td>NO-READABLE-TEXT .99</td>
</tr>
</tbody>
</table>

**Tensor PARAFAC**

\[ \mathbf{X} = \sum_{k=1}^{n} \mathbf{H}_k \mathbf{T}_k \mathbf{A}_k \]

**TPHITS uses 3D analysis to find the dominant groupings of web pages and terms.**

\[ x_{ijk} = \begin{cases} 
    \frac{1}{\log(w_k)+1} & \text{if } i \rightarrow j \text{ with term } k \\
    0 & \text{otherwise}
\end{cases} 
\]

\[ w_k = \# \text{ unique links using term } k \]
## Summary

<table>
<thead>
<tr>
<th>Methods</th>
<th>Pros</th>
<th>Cons</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVD, PCA</td>
<td>Optimal in L2 and Frobenius</td>
<td>Dense representation, Negative entries</td>
<td>LSI, PageRank, HITS</td>
</tr>
<tr>
<td>CUR, CMD</td>
<td>Interpretability, sparse bases</td>
<td>Not optimal like SVD, dense core</td>
<td>DNA SNP data, network forensics</td>
</tr>
<tr>
<td>Co-clustering</td>
<td>Interpretability</td>
<td>Local minimum</td>
<td>Social networks, microarray data</td>
</tr>
<tr>
<td>Tucker</td>
<td>Flexible representation</td>
<td>Interpretability, non-uniqueness, dense core</td>
<td>TensorFaces</td>
</tr>
<tr>
<td>PARAFAC</td>
<td>Interpretability, efficient parse computation</td>
<td>Slow convergence</td>
<td>TOPHISTS</td>
</tr>
<tr>
<td>Incrementalization</td>
<td>Efficiency</td>
<td>Non-optimal</td>
<td>Tensor Streams</td>
</tr>
</tbody>
</table>
Conclusions

• Real data are often in high dimensions with multiple aspects (modes)
• Matrices and tensors provide elegant theory and algorithms
• Several research problems are still open
  – skewed distribution, anomaly detection, streaming algorithms, distributed/parallel algorithms, efficient out-of-core processing
Acknowledgements

• Cross-association foils:
  – from Dr. Deepayan Chakrabarti (Yahoo!)

• Funding:

PITA (PA Inf. Tech. Alliance)
Thank you!

- Christos Faloutsos
  www.cs.cmu.edu/~christos

- Tamara Kolda
  csmr.ca.sandia.gov/~tgkolda

- Jimeng Sun
  www.cs.cmu.edu/~jimeng

www.cs.cmu.edu/~christos/TALKS/SIGMOD-07-tutorial/
3h version: www.cs.cmu.edu/~christos/TALKS/SDM-tut-07/