Tensor Completion for Estimating Missing Values in Image Data

Adam Schmidt

December 9, 2020

\footnote{Ji Liu et al., 2009.}
Problem

- Estimate image data of tensors with low rank
Problem

- Estimate image data of tensors with low rank
- Signal: Order 1 Tensor
Problem

- Estimate image data of tensors with low rank
- Signal: Order 1 Tensor
- Image: Order 2 Tensor
Problem

- Estimate image data of tensors with low rank
- Signal: Order 1 Tensor
- Image: Order 2 Tensor
- Video: Order 3 Tensor
Problem

- Estimate image data of tensors with low rank
- Signal: Order 1 Tensor
- Image: Order 2 Tensor
- Video: Order 3 Tensor
- MRI Video: Order 4 Tensor
Solution

- Expand low-rank matrix completion to tensor completion
Figure 1: The left figure contains 80% missing entries shown as white pixels and the right figure shows its reconstruction using the low rank approximation.
Eating Our Vegetables

\[ D_{\tau}(X) = U\Sigma_{\tau}V^\top \quad (1) \]

\[ \Sigma_{\tau}(X) = \text{diag}(\max(\sigma_i - \tau, 0)) \quad (2) \]
Eating Our Vegetables

\[ D_\tau(X) = U\Sigma_\tau V^\top \]  
\[ \Sigma_\tau(X) = \text{diag}(\max(\sigma_i - \tau, 0)) \]  

Masked sample set: \( X_\Omega \)
Eating Our Vegetables

\[ D_{\tau}(X) = U\Sigma_{\tau}V^\top \]  
\[ \Sigma_{\tau}(X) = \text{diag}(\max(\sigma_i - \tau, 0)) \]  

Masked sample set: \( X_{\Omega} \)

Tensor \( X \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_n} \)
\[ D_\tau(X) = U\Sigma_\tau V^\top \] (1)

\[ \Sigma_\tau(X) = \text{diag}(\max(\sigma_i - \tau, 0)) \] (2)

Masked sample set: \( X_\Omega \)

Tensor \( \mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_n} \) and a tensor unfold operation defined as:

\[ \text{unfold}_k(\mathcal{X}) := X_{(k)} \in \mathbb{R}^{I_k \times (I_1 \times \ldots \times I_{k-1} \times I_{k+1} \times \ldots \times I_n)} \] (3)

and the reverse:

\[ \text{fold}_k(X_{(k)}) = \mathcal{X} \] (4)
Trace Norm

The rank operation is discrete and non-convex.
Trace Norm

The rank operation is discrete and non-convex. Thus trace norm is used as a method:

$$\|X\|_{tr} = \sum_{i} \sigma_i(X)$$ (5)

Tightest convex envelope for rank of matrices. (loose reason: L1 enforces sparsity)
Matrix Completion

We have a matrix $M$ at $\Omega$. We want to estimate a low-rank $X$. 
Matrix Completion

We have a matrix $M$ at $\Omega$. We want to estimate a low-rank $X$.

$$\min_X \frac{1}{2} \|X - M\|_\Omega^2$$

s.t. $\text{rank}(X) \leq r$
Matrix Completion

We have a matrix $M$ at $\Omega$. We want to estimate a low-rank $X$.

$$\min_{X} \frac{1}{2} \|X - M\|_\Omega^2$$

s.t. $\text{rank}(X) \leq r$

$$\min_{X,Y} \frac{1}{2} \|X - Y\|_F^2$$

s.t. $\text{rank}(X) \leq r$

$Y_\Omega = M_\Omega$

$$\min_{X,Y} \frac{1}{2} \|X - Y\|_F^2$$

s.t. $\|X\|_{tr} \leq c$

$Y_\Omega = M_\Omega$
Tensor Completion

$\mathcal{T}_\Omega$ is what we know

$$\min_{\mathcal{X}, \mathcal{Y}} : \frac{1}{2} \| \mathcal{X} - \mathcal{Y} \|^2_F$$

s.t. $\| \mathcal{X} \|_{tr} \leq c$

$\mathcal{Y}_\Omega = \mathcal{T}_\Omega$
Tensor Completion

\[
\min_{\mathcal{X}, Y} \quad \frac{1}{2n} \sum_{i=1}^{n} \|X^{(i)} - Y^{(i)}\|_F^2 \\
\text{s.t.} \|\mathcal{X}\|_{tr} \leq c \\
Y_\Omega = \mathcal{T}_\Omega
\]

What is the tensor trace norm?
Tensor Completion

\[
\min_{\mathcal{X}, \mathcal{Y}} \frac{1}{2n} \sum_{i=1}^{n} \| X(i) - Y(i) \|_F^2
\]

s.t. \[
\frac{1}{n} \sum_{i=1}^{n} \| X(i) \|_{tr} \leq c
\]

\[\mathcal{Y}_\Omega = \mathcal{T}_\Omega\]

Average of each unfolded trace norm.
Loosening it up:

\[
\min_{\mathcal{X}, \mathcal{Y}, M_i} \quad \frac{1}{2n} \sum_{i=1}^{n} \| M_i - Y(i) \|_F^2 \\
\text{s.t.} \quad \frac{1}{n} \sum_{i=1}^{n} \| M_i \|_{tr} \leq c \\
M_i = X(i) \text{ for } i = 1, 2, \ldots, n \\
\mathcal{Y}_\Omega = \mathcal{T}_\Omega
\]
Relaxing equality:

$$\min_{x, \mathbf{Y}, \mathbf{M}_i} : \frac{1}{2n} \sum_{i=1}^{n} \|\mathbf{M}_i - Y(i)\|_F^2$$

s.t. \( \frac{1}{n} \sum_{i=1}^{n} \|\mathbf{M}_i\|_{tr} \leq c \)

\[ \|\mathbf{M}_i - X(i)\|_F^2 \leq d_i, \text{ for } i = 1, 2, \ldots, n \]

\( \mathbf{Y}_\Omega = \mathcal{T}_\Omega \)
Tensor Completion

Converting to the dual:

$$\min_{\mathcal{X}, \mathcal{Y}, M_i} : \frac{1}{2n} \sum_{i=1}^{n} \| M_i - Y(i) \|_F^2$$

$$+ \frac{\gamma}{n} \sum_{i=1}^{n} \| M_i \|_{tr}$$

$$+ \frac{1}{2n} \sum_{i=1}^{n} \alpha_i \| M_i - X(i) \|_F^2$$

s.t. $\mathcal{Y}_\Omega = \mathcal{T}_\Omega$
Tensor Completion

Tossing in more weights:

\[
\min_{X,Y,M_i} : \frac{1}{2n} \sum_{i=1}^{n} \beta_i \| M_i - Y(i) \|_F^2 \\
+ \frac{1}{n} \sum_{i=1}^{n} \gamma_i \| M_i \|_{tr} \\
+ \frac{1}{2n} \sum_{i=1}^{n} \alpha_i \| M_i - X(i) \|_F^2 \\
\text{s.t. } Y_\Omega = T_\Omega
\]
Tensor Completion

Tossing in more weights:

\[
\min_{\mathcal{X}, \mathcal{Y}, M_i} : \frac{1}{2n} \sum_{i=1}^{n} \beta_i \| M_i - Y(i) \|^2_F \\
+ \frac{1}{n} \sum_{i=1}^{n} \gamma_i \| M_i \|_{tr} \\
+ \frac{1}{2n} \sum_{i=1}^{n} \alpha_i \| M_i - X(i) \|^2_F \\
\text{s.t } \mathcal{Y}_\Omega = \mathcal{T}_\Omega
\]

\( \mathcal{X} \) is totally free
Block Coordinate Descent

- divide into $n + 2$ blocks
- break up
\begin{align*}
\min_{x} & : \frac{1}{2} \sum_{i=1}^{n} \alpha_i \| M_i - X(i) \|_F^2 \\
\text{Solution is the weighted mean.}
\end{align*}
\[ \min_{\mathcal{X}} : \frac{1}{2} \sum_{i=1}^{n} \alpha_i \| M_i - X_{(i)} \|_F^2 \]  

(6)

Solution is the weighted mean.

\[ \mathcal{X} = \frac{\sum_{i=1}^{n} \alpha_i \text{fold}_i(M_i)}{\sum_{i=1}^{n} \alpha_i} \]  

(7)
Solution is the weighted mean with a mask

\[ Y_{\bar{\Omega}} = \left( \frac{\sum_{i=1}^{n} \alpha_i \cdot fold_i(M_i)}{\sum_{i=1}^{n} \alpha_i} \right)_{\bar{\Omega}} \]
\[ \min_{M_i} : \frac{\beta_i}{2} \| M_i - Y_{(i)} \|_F^2 + \frac{\gamma_i}{n} \| M_i \|_{tr} + \frac{\alpha_i}{2n} \| M_i - X_{(i)} \|_F^2 \]  

(9)

Solution:

\[ M = D_\tau(Z), \]

\[ \tau = \frac{\gamma_i}{\alpha_i + \beta_i}, Z_i = \frac{\alpha_i X_{(i)} + \beta_i Y_{(i)}}{\alpha_i + \beta_i} \]  

(10)

\[ D_\tau(X) = U \Sigma_\tau V^\top \]  

(11)
Compared Methods

- CP/Parafac
- Tucker
- SVD
Table 1: The RSE comparison on the synthetic data of size $40 \times 40 \times 40 \times 40$. P: Parafac model based heuristic algorithm; T: Tucker model heuristic algorithm; SVD: the heuristic algorithm based on the SVD; $\alpha_0$, $\alpha_{10}$ and $\alpha_{50}$ denote the proposed LRTC algorithm with three different values of the parameter: $\alpha = 0$, $\alpha = 10$ and $\alpha = 50$, respectively. The top, middle and bottom parts of the table respond to the sample percentage: 3%, 20% and 80%, respectively.

<table>
<thead>
<tr>
<th>Rank</th>
<th>T</th>
<th>P</th>
<th>SVD</th>
<th>$\alpha_0$</th>
<th>$\alpha_{10}$</th>
<th>$\alpha_{50}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10,11,10,9</td>
<td>725</td>
<td>677</td>
<td>759</td>
<td>321</td>
<td>302</td>
<td><strong>257</strong></td>
</tr>
<tr>
<td>14,16,15,14</td>
<td>892</td>
<td>901</td>
<td>863</td>
<td>72.5</td>
<td>70.2</td>
<td><strong>65.3</strong></td>
</tr>
<tr>
<td>20,22,21,19</td>
<td>1665</td>
<td>1302</td>
<td>1474</td>
<td>50.1</td>
<td>46.9</td>
<td><strong>36.2</strong></td>
</tr>
<tr>
<td>24,25,25,26</td>
<td>2367</td>
<td>1987</td>
<td>2115</td>
<td>40.6</td>
<td><strong>38.7</strong></td>
<td><strong>38.7</strong></td>
</tr>
<tr>
<td>10,11,9,11</td>
<td>371</td>
<td>234</td>
<td>347</td>
<td>16.3</td>
<td>14.2</td>
<td><strong>12.7</strong></td>
</tr>
<tr>
<td>15,15,16,14</td>
<td>728</td>
<td>530</td>
<td>611</td>
<td>8.92</td>
<td>8.41</td>
<td><strong>8.23</strong></td>
</tr>
<tr>
<td>21,19,21,20</td>
<td>1093</td>
<td>982</td>
<td>895</td>
<td><strong>8.48</strong></td>
<td>8.56</td>
<td><strong>8.48</strong></td>
</tr>
<tr>
<td>24,25,26,26</td>
<td>1395</td>
<td>1202</td>
<td>1260</td>
<td>40.7</td>
<td>34.3</td>
<td><strong>13.7</strong></td>
</tr>
<tr>
<td>10,9,11,9</td>
<td>145</td>
<td>45</td>
<td>136</td>
<td><strong>3.08</strong></td>
<td>4.01</td>
<td>3.12</td>
</tr>
<tr>
<td>15,14,14,16</td>
<td>326</td>
<td>65</td>
<td>217</td>
<td>2.17</td>
<td><strong>2.05</strong></td>
<td>2.35</td>
</tr>
<tr>
<td>21,20,19,21</td>
<td>518</td>
<td>307</td>
<td>402</td>
<td>1.36</td>
<td>2.06</td>
<td><strong>1.27</strong></td>
</tr>
<tr>
<td>24,25,25,26</td>
<td>685</td>
<td>509</td>
<td>551</td>
<td>1.41</td>
<td>1.59</td>
<td><strong>1.04</strong></td>
</tr>
</tbody>
</table>

Table 2: The RSE comparison on the brain MRI data of size $181 \times 217 \times 181$. P: Parafac model based heuristic algorithm; T: Tucker model heuristic algorithm; SVD: the heuristic algorithm based on the SVD; $\alpha_0$, $\alpha_{10}$ and $\alpha_{50}$ denote the proposed LRTC algorithm with three different values of the parameter: $\alpha = 0$, $\alpha = 10$ and $\alpha = 50$, respectively. The top and bottom parts respond to the sample percentage: 20% and 80%, respectively.

<table>
<thead>
<tr>
<th>Rank</th>
<th>T</th>
<th>P</th>
<th>SVD</th>
<th>$\alpha_0$</th>
<th>$\alpha_{10}$</th>
<th>$\alpha_{50}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>21,24,23</td>
<td>311</td>
<td>234</td>
<td>274</td>
<td>210</td>
<td>193</td>
<td><strong>177</strong></td>
</tr>
<tr>
<td>38,41,37</td>
<td>1259</td>
<td>1001</td>
<td>1322</td>
<td>148</td>
<td>141</td>
<td><strong>121</strong></td>
</tr>
<tr>
<td>90,93,87</td>
<td>4982</td>
<td>3982</td>
<td>5025</td>
<td>61.0</td>
<td>53.7</td>
<td><strong>42.8</strong></td>
</tr>
<tr>
<td>21,24,23</td>
<td>12.3</td>
<td>8.64</td>
<td><strong>5.48</strong></td>
<td>29.4</td>
<td>12.8</td>
<td>13.5</td>
</tr>
<tr>
<td>35,42,36</td>
<td>179</td>
<td>153</td>
<td>99</td>
<td><strong>4.41</strong></td>
<td>5.32</td>
<td>5.69</td>
</tr>
<tr>
<td>39,48,41</td>
<td>279</td>
<td>345</td>
<td>199</td>
<td><strong>0.72</strong></td>
<td>1.05</td>
<td>1.26</td>
</tr>
<tr>
<td>45,55,47</td>
<td>606</td>
<td>523</td>
<td>513</td>
<td>1.22</td>
<td>1.35</td>
<td><strong>1.06</strong></td>
</tr>
</tbody>
</table>
Pretty Results

Figure 2: The left image (one slice of the MRI) is the original; we randomly select pixels for removal shown in white in the middle image; the right image is the result of the proposed completion algorithm.

Figure 3: Facade in-painting. The top image is the original image; we select the lamp and satellite dishes together with a large set of randomly positioned squares as the missing parts shown in white in the middle image; the bottom image is the result of the proposed completion algorithm.

Figure 4: Video completion. The left image (one frame of the video) is the original; we randomly select pixels for removal shown in white in the middle image; the right image is the result of the proposed LTRC algorithm.
Figure 5: The top left image is a rendering of an original phong BRDF; we randomly select 90% of the pixels for removal shown in white in the top right image; the bottom image is the result of the proposed LRTC algorithm.
Where is this useful?

\[ J. \text{ Liu et al., 2013.} \]
Where is this useful?

Images?

\(^2\text{J. Liu et al., 2013.}\)
Where is this useful?

Images?
follow-up paper\textsuperscript{2}

\textsuperscript{2}J. Liu et al., 2013.
Pathological Cases

\[ \begin{bmatrix}
N \\
\end{bmatrix} = \text{rank } n \]

\[ \begin{bmatrix}
\begin{array}{c}
\vdots \\
\end{array}
\end{bmatrix} = \text{rank } 1 \]

\[ \text{rot } \frac{\pi}{4} \]

\[ \text{Noise} \]

[Graphs and matrices with annotations]

19 / 24
Shopping for alternatives

\(^3\)H. Liu et al., 2019.
CoSTCo\textsuperscript{3} Takes in indices, spits out value at those indices

Figure 2: Model architecture of CoSTCo.

Uses relus and 2d convs. Seems interesting for non-spatially dependent data.

\textsuperscript{3}H. Liu et al., 2019.
Implicit Functions

Fourier Features\textsuperscript{4}, NeRF\textsuperscript{5} and its many children, some conditioned children\textsuperscript{6}.

Allow interpolation for free. Implicit neural functions for continuous images\textsuperscript{7} and sinusoidal activation functions Implicit Neural Representations with Periodic Activation Functions\textsuperscript{8}

\textsuperscript{4}Tancik et al., n.d.
\textsuperscript{5}Mildenhall et al., 2020.
\textsuperscript{6}Yu et al., 2020.
\textsuperscript{7}Skorokhodov, Ignatyev, and Elhoseiny, 2020.
\textsuperscript{8}Sitzmann et al., 2020.
References I


Tancik, Matthew et al. (n.d.). “Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains”. In: (), p. 11.