README for the Matlab Package of the LRTC Algorithm

Ji Liu[1], Przemyslaw Musialski [2], Peter Wonka[1], and Jieping Ye[1]
Arizona State University[1]
VRVis Research Center[2]
Ji.Liu@asu.edu, musialski@vrvis.at, Peter.Wonka@asu.edu, Jieping.Ye@asu.edu

1. About This Package

The matlab package implements the **LRTC** algorithm to solve the low rank tensor completion problem defined in our paper "Tensor Completion for Estimating Missing Values in Visual Data, ICCV2009".

The main function in this package minimizes the problem of eqn. (15) in this paper, i.e.,

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

2. The Main Function

The main function in this package is defined as

$$\begin{aligned}[rImg, errList, R] = &\textbf{LRTC}(\\ &inData,\\ &alpha,\\ &beta,\\ &gamma,\\ &mark,\\ &maxIter,\\ &initial)\end{aligned}$$

The output is

- $rImg$: The recovered result
- $errList$: A list of function values. The function value of the equation above is recorded once per iteration. This list can be used for checking convergence.
- R: The real rank value of each mode

The input is

- $inData$: $\mathcal{T}$
- $alpha$: $[\alpha_1, \alpha_2, ..., \alpha_i, ...]$

- $beta$: $[\beta_1, \beta_2, ..., \beta_i, ...]$
- $gamma$: $[\gamma_1, \gamma_2, ..., \gamma_i, ...]$
- $mark$: $\bar{\Omega}$, Ω is a binary tensor with the same dimensions as $\mathcal{T}$. "0" means a value is missing and "1" means a value is defined
- $initial$: The initial value for $\mathcal{Y}_{\bar{\Omega}}$
- $maxIter$: The maximum number of iterations

Please check the paper for the detailed definition of the input parameters.

3. Example

Please follow the "example.m" file in this package for an example on how to use this function.

```
%%%%%%%%%%%%%%%  Example.m  %%%%%%%%%%%%%%%
clc; clear;
inData = double(imread('testImg.png','png'));
mark = inData(:,:,:)>254;
maxIter = 600;
initial = [];
alpha = [1, 1, 1];
beta = [1, 1, 1];
gamma = [100, 100, 0];
[rImg, errList, R] = LRTC(
        inData,
        alpha,
        beta,
        gamma,
        mark,
        maxIter,
        initial);
figure; plot(errList);
figure; imshow(inData/256);
figure; imshow(rImg/256);
%%%%%%%%%%%%%%%  Example.m  %%%%%%%%%%%%%%%
```

In this example, we consider the color image as a tensor with 3 modes. Because the last mode indicates the number of color channels, the last parameter in "$gamma$" is set as "0", i.e., we do not require it to be low rank. In this case, the missing data in 3 color channels are recovered simultaneously. One can also recover respectively the missing data at each channel.

4. How to Tune Parameters?

Typically, α_i's can be fixed as "1". γ_i's are the most important parameters, which control the penalty of rank at each mode. A larger value leads to a lower rank reconstruction. Based on our experience, the reconstruction results are not very sensitive to the β_i's values. A large value for the β_i's may lead to a slightly better reconstruction, but also a slow convergence speed. A good strategy is to set the β_i's as "1" at first and in the following iterations increase its values gradually using the last reconstruction as a warm start. Of course, if the goal is to just obtain an an acceptable reconstruction, one can always set α_i's and β_i's as "1" and tune the γ_i's values only.

5. Additional Results

We also provide several additional results to facilitate a comparison with **LRTC**. In these results (all of them shown in the video accompaning the paper) we give the original data, the mask image defining the missing values, and our reconstruction. For the images the data is straightforward. For tensors we provide the data in matlab and image format. We provide the code for selected tensor examples, but not for all of them. Usually the input is in the file named (miss) and our result is in the file named (rec). We also include the original data (org) for comparison.