Journal of Yichao Li for Fall semester 2016

December 20, 2016

Contents

1 Tensor Decomposition is a Feature Learning Method, 10-11-2016

Check out my website for this post.

I also try to copy everything to this journal, but it is not pretty as I thought.

There are no good python libraries for tensor analysis, see this Quora post.

TensorFlow has some basic operations on Tensors, see this introduction

I havent tried TensorFlow nor tensor-analysis, nor TensorLib by CMU. The last one was developped in 2009 and there is no more updates.

I used scikit-tensor to analysis a sensory bread data. The description of the data is as follows:

The data are arranged in a three-way array (10 breads 11 attributes 8 judges). Also available is the salt content of the ten samples. A PARAFAC model can be seen as a reasonable approximate model incorporating saliencies for each assessor, but there is no hard theory stating the nature of the data. The data are quite noisy compared to, e.g., spectral data. The data are usually centered across the sample mode before fitting the model. This centering removes the assessor specific offsets on the attribute scales. Scaling of the data is an issue where no consensus has yet been achieved.

The code is :

```
import logging
from scipy.io.matlab import loadmat
from sktensor import dtensor, cp_als

logging.basicConfig(level=logging.DEBUG)

mat = loadmat('brod.mat')
T = dtensor(mat['X'])

P, fit, itr, exectimes = cp_als(T, 3, init='random')
P.U[0].shape
P.U[1].shape
P.shape
```

This slide has a good introduction on tensors with python code. It also explains the tensor rank.

TENSOR FACTORIZATION: HOW TO

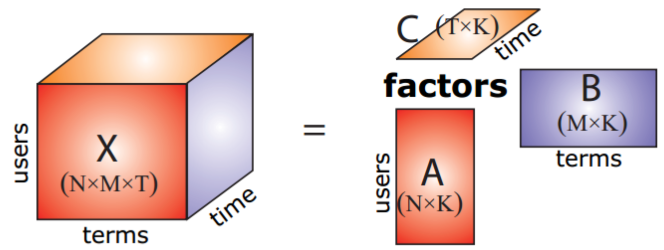
Unfolded Tensor on the kth mode

$$\mathbf{T}_{(1)} = \hat{\mathbf{A}}(\hat{\mathbf{C}} \odot \hat{\mathbf{B}})^T$$
$$\mathbf{T}_{(2)} = \hat{\mathbf{B}}(\hat{\mathbf{C}} \odot \hat{\mathbf{A}})^T$$
$$\mathbf{T}_{(3)} = \hat{\mathbf{C}}(\hat{\mathbf{B}} \odot \hat{\mathbf{A}})^T$$

Alternating Least Squares(ALS):
Fix all but one factor matrix to which LS is applied

$$\min_{\mathbf{A} \geq 0} \|\mathbf{T}_{(1)} - \mathbf{A}(\mathbf{C} \odot \mathbf{B})^T\|$$
$$\min_{\mathbf{B} \geq 0} \|\mathbf{T}_{(2)} - \mathbf{B}(\mathbf{C} \odot \mathbf{A})^T\|$$
$$\min_{\mathbf{C} \geq 0} \|\mathbf{T}_{(3)} - \mathbf{C}(\mathbf{B} \odot \mathbf{A})^T\|$$

HOW TO INTERPRET: USER X TERM X TIME



X is a 3-way tensor in which x_{nmt} is 1 if the term m was used by user n at interval t, 0 otherwise

A^{NxK} is the the association of each user n to a factor k

B^{MxK} is the association of each term m to a factor k

C^{TxK} shows the time activity of each factor ref: http://www.slideshare.net/panisson/exploring-temporal-graph-data-with-python-a-study-on-tensor-decomposition-of-wearable-sensor-data

2 Tensor Basics, 10-18-2016

The article I read is "The Poor Mans Introduction to Tensors".

2.1 Einstein Summation convention

The motivation for this idea is that explicitly showing all the summations, even with $\sum$, is very ugly. So we want to use Einstein summation convention.

Now, when you see a pair of index, e.g. j, in an equation, and one is raised and the other is lowered, then a summation over all j is implied.

$$T_{ij}v^j = \sum_j T_{ij}v^j$$

2.2 Kronecker delta

δ^a_b is called *Kronecker delta* if

$$\delta^a_b = 1, a = b$$
$$\delta^a_b = 0, a \neq b$$

2.3 Tensor

The tensor defined in this article might be different from what I see in the journals.

Tensor is a linear map that takes vectors or dual vectors as input and outputs a scalar.

It seems that a tensor is a function. I think tensor is a data structure, like a matrix. It is just storing the data, like a multi-indexed array.

I can understand the following equation:

$$S(w, z, v, u, q) = S^{ij}_{klm} w_i z_j v^k u^l q^m$$

But I still don't known this:

$$f : \mathcal{X} \to \mathbb{R} : \mathbf{x} = (\mathbf{A}, \mathbf{B}, \mathbf{C}) \mapsto \frac{1}{2} \left\| \mathcal{T} - \sum_{j=1}^{r} a_j \otimes b_j \otimes c_j \right\|^2,$$

3 Tensor Basics, 10-27-2016

"What is a Tensor" video lectures at Youtube

4 Survey of low-rank tensor approximation techniques, 11-8-2016

I found a survey paper by Lars Grasedyck, 2010, which was a survey of low-rank tensor approximation methods.

They mentioned several decomposition methods.

The first is CP decomposition. "More recent developments of CP decomposition are increasing the efficiency and robustness of gradient-based and Newton-like methods, modifying and improving ALS, and studying the convergence of ALS, etc."

The rest methods are: Tucker decomposition, Tensor train decomposition, Hierarchical Tucker Decomposition.

Several MATLAB toolboxes are available for dealing with tensors in CP and Tucker decomposition,including the Tensor Toolbox, the N-way toolbox, the PLS Toolbox, and the Tensorlab.

Also a C++ library, TensorCalculus, is allowing for computations with general tensor networks.

A literature survey of low-rank tensor approximation techniques

Lars Grasedyck[1,*], **Daniel Kressner**[2,**], and **Christine Tobler**[2,***]

[1] Institut für Geometrie und Praktische Mathematik, RWTH Aachen, Templergraben 55, 52056 Aachen, Germany

[2] ANCHP, MATHICSE, EPF Lausanne, Switzerland

Key words tensor, low rank, multivariate functions, linear systems, eigenvalue problems
MSC (2010) 15A69, 65F10, 65F15

During the last years, low-rank tensor approximation has been established as a new tool in scientific computing to address large-scale linear and multilinear algebra problems, which would be intractable by classical techniques. This survey attempts to give a literature overview of current developments in this area, with an emphasis on function-related tensors.