When You Follow Your Heart

You Cease Having Regrets

Yao Zhang

University of Chinese Academy of Sciences

National Astronomical Observatories, Chinese Academy of Sciences

https://zhims.github.io

February 5, 2021

Overview

Our Academic Family

Our Academic Family

张凯　　张尧　　杨玉　　余怡君　　郑熹　…… ……

Figure 1: Our Academic Family

Computing Science VS Computer Science

Low Rank and Sparse based on Matrix

Lemma 1 (Johnson–Lindenstrauss Lemma)

Given $0 < \epsilon < 1$, as set of m points in $\mathbb{R}^N$, and a number $n > 8\frac{\log m}{\varepsilon^2}$, there is a linear map, $f\colon \mathbb{R}^N \to \mathbb{R}^n$ such that

$$(1-\varepsilon)\left\|u-v\right\|^2 \leqslant \left\|f(u)-f(v)\right\|^2 \leqslant (1+\varepsilon)\left\|u-v\right\|^2 \tag{1}$$

There is a recommended paper [Udell et al, 2018].

Given $X \in \mathbb{R}^{m \times n}$,

$$\|X\|_* = \|\Sigma\|_1 \tag{2}$$

where $X = U \sum V^T,\ UU^T = I_{m\times m}, VV^T = I_{n \times n}$.

There is an additional slides: http://stanford.edu/~qysun/Lecture04_SunQY.pdf which introduce sparsity in Convolutional Neural Networks.

Generalized PCA VS Representation Theory

Generalized PCA model:

$$\min_{L,S} \|L\|_{rank} + \lambda\|S\|_{sparse} \quad s.t.\ X = L + S \tag{3}$$

Representation theory model:

$$\min_{D,C,E} \|C\|_{rank\ or\ sparse} + \lambda\|E\|_{p-norm}\ s.t.\ X = DC + E \tag{4}$$

We want to get that:

$$\begin{aligned} &\min_X \tau\|X\|_* + \frac{1}{2}\|X-A\|_F^2 \\ &\min_X \tau\|X\|_1 + \frac{1}{2}\|X-A\|_F^2 \end{aligned} \tag{5}$$

Example 1

1. Robust Principal Component Analysis (RPCA) model:

$$\min_{L,S} \|L\|_* + \lambda\|S\|_1 \quad s.t.\ X = L + S \tag{6}$$

2. Low Rank Representation (LRR) model

$$\min_{Z,E} \|Z\|_* + \lambda\|E\|_1 \quad s.t.\ X = XZ + E \tag{7}$$

More details: https://zhims.github.io/doc/note/datascience/LRS/Tips.pdf.

How to Choose λ in Eq. 6 & 7

Convolutional Neural Network is powerful in many applications but it is expensive.

How about a simple Multilayer Perceptron (MLP)?

There is a recommended paper [Scetbon et.al 2020].

More details: `https://zhims.github.io/doc/note/datascience/LRS/deepksvd.pdf`.

This is also a recommended paper [Elhamifar et al, 2013].

Computational Complexity

Remark 1

It is well known that the computation complexity of thin SVD for an $m \times n$ matrix X with $m \geqslant n$ is $O\left(mn^2\right)$. The cost of computing the inverse for $d \times d$ matrix is $O\left(d^3\right)$, and the expense of multiplication for $m \times d$ matrix and $d \times n$ matrix is $O(mdn)$.

If we have that

$$\begin{aligned} & X = AB \text{ or } X = ABC \text{ or } X = \cdots \\ s.t. \quad & min(rank(A), rank(B), rank(C), \cdots) \geq rank(X) \end{aligned} \tag{8}$$

More details: https://zhims.github.io/doc/note/datascience/LRS/Schattenp.pdf.

Some Low Rank models

1. $\text{Rank}(X) \to \|X\|_*$
2. $\text{Rank}(X) \to \|X\|_* - (\sigma_1 + \cdots + \sigma_k)$
3. $\text{Rank}(X) \to$ Schatten-p norm
4. $\text{Rank}(X) \to \sum_{i=1}^{r} \log\left(1 + \sigma_i^2\right)$
5. $\text{Rank}(X) \to \log\det\left(I + X^T X\right)$
6. $\text{Rank}(X) \to \log\det\left(diag(Y, Z) + \delta I\right)$ where $\begin{bmatrix} Y & X \\ X^T & Z \end{bmatrix} \geqslant 0$
7. $\text{Rank}(X) \to \cdots$

Decomposition

[Golub et al, 2013],

Decompositions have three roles to play in matrix computations. They can be used to convert a given problem into an equivalent easy-to-solve problem, they can expose hidden relationships among the a_{ij} , and they can open the door to data-sparse approximation.The role of tensor decompositions is similar and in this section we showcase a few important examples. The matrix SVD has a prominent role to play throughout. The goal is to approximate or represent a given tensor with an illuminating (hopefully short) sum of rank-1 tensors. Optimization problems arise that are multilinear in nature and lend themselves to the alternating least squares framework. These methods work by freezing all but one of the unknowns and improving the free-to-range variable with some tractable linear optimization strategy. Interesting matrix computations arise during this process and that is the focus of our discussion. For a much more complete survey of tensor decompositions, properties, and algorithms, see

[Kolda et al, 2009].

Low Rank and Sparse based on Tensor

CANDECOMP/PARAFAC (CP)Decomposition

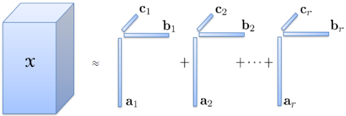

Figure 2: CP Decomposition

$$\min_{A,B,C} \|\mathcal{X} - \mathcal{M}\|^2 \; s.t. \; \mathcal{M} = [A, B, C] \Leftrightarrow \min_{A,B,C} \sum_{i,j,k} \left(x_{ijk} - \sum_{l} a_{il} b_{jl} c_{kl} \right)^2$$

unfortunately, it is nonconvex, but its subproblems are convex.

There is a recommended Chapter 12 in Book [Golub et al, 2013].

More details: https://zhims.github.io/doc/note/datascience/LRS/tensor_2009.pdf.

Tucker Decomposition

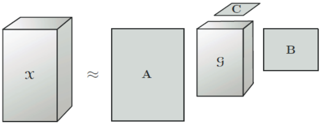

Figure 3: Tucker Decomposition

$$\min_{\mathcal{G},A,B,C} \|\mathcal{X} - \mathcal{M}\|^2 \; s.t. \;\; \mathcal{M} = [\mathcal{G}, A, B, C]$$

$$\text{where } m_{ijk} = \sum_{r_1} \sum_{r_2} \sum_{r_3} g_{r_1 r_2 r_3} a_{ir_1} b_{jr_2} c_{kr_3}$$

There is a recommended Chapter 12 in Book [Golub et al, 2013].

More details: `https://zhims.github.io/doc/note/datascience/LRS/tensor_2009.pdf`.

Definition 2

Let $\mathcal{X}$ be an $n_1 \times n_2 \times n_3$ tensor and $\mathcal{Y}$ be an $n_2 \times n_4 \times n_3$ tensor. Then the t-product, denote by $\mathcal{X} * \mathcal{Y}$, is the $n_1 \times n_4 \times n_3$ tensor given by

$$\mathcal{X} * \mathcal{Y} = fold(blockcirc(\mathcal{X}) \cdot unfold(\mathcal{Y})) \tag{9}$$

where $blockcirc(\mathcal{X}) \doteq \begin{bmatrix} X^{(1)} & X^{(n_3)} & \dots & X^{(2)} \\ X^{(2)} & X^{(1)} & \dots & X^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ X^{(n_3)} & X^{(n_3-1)} & \dots & X^{(1)} \end{bmatrix} \in \mathbb{R}^{(n_1 n_3)\times(n_2 n_3)}$, $unfold(\mathcal{Y}) \doteq \begin{bmatrix} Y^{(1)} \\ Y^{(2)} \\ \vdots \\ Y^{(n_3)} \end{bmatrix} \in \mathbb{R}^{(n_2 n_3)\times n_4}$,

and the inverse operator fold takes $unfold(\mathcal{X})$ into a tensor: $fold(unfold(\mathcal{Y})) = \mathcal{Y}$.

More details: https://zhims.github.io/doc/note/datascience/LRS/tensor.pdf.

Theorem 3 (Tensor SVD)

For any $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the t-SVD of $\mathcal{X}$ is given by

$$\mathcal{X} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T \tag{10}$$

where $\mathcal{U} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ and $\mathcal{V} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$ are orthogonal, $\mathcal{S} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is f-diagonal.

More details: https://zhims.github.io/doc/note/datascience/LRS/tensor.pdf.

How about the topics in page 12 & 13?

How about the transformations, cosine, sine, wavelet, random matrix transformation?

Application I: Subspace Clustering

Subspace Clustering I

$$\min_Z \lambda_1 \|Z\|_* + \lambda_2 \|Z\|_1 + \frac{\lambda_3}{2} \|X - XZ\|_F^2 \quad s.t. \quad diag\ Z = 0,\ 1^T Z = 1^T \quad (11)$$

In practice, $\lambda_1 + \lambda_2 = 1$, , choose λ_3 from paper [Elhamifar et al, 2013].

It is easy to show that the sequence generated by our algorithm is Cauchy sequence.

Now, how to choose λ_1, λ_2, and $\lambda_3, \cdots$ by a simple MLP or CNN?

Our model is

$$\min_{C,D,F,E} \alpha \|C\|_{l_{p_1}}^{p_1} + \beta \|C\|_{S_{p_2}}^{p_2} + \gamma \|\Phi(Y) - \Phi(Y) C\|_F^2 \tag{12}$$

How about

$$\begin{aligned} \min_{B,C} \|B\|_{S_{p_1}}^{p_1} + \lambda_1 \|B\|_{l_{p_2}}^{p_2} + \lambda_2 \|C\|_{S_{p_3}}^{p_3} + \lambda_3 \|C\|_{l_{p_4}}^{p_4} \\ + \frac{\lambda_4}{2} \|\phi(X) - \phi(X) C\|_F^2 + \frac{\lambda_5}{2} \left\|K_G - B^T B\right\|_F^2 \end{aligned} \tag{13}$$

There is a recommended paper: https://arxiv.org/pdf/1707.04974v4.pdf.

And, more details: https://zhims.github.io/doc/note/datascience/LRS/kernel.pdf.

But there is a subproblem when we solve the Eq 13:

$$\min_{A} \|A + B\|_F^2 + \left\|A^T A + C\right\|_F^2 \tag{14}$$

Problem 14 can be written as

$$\min_{A} Tr(A + B)^T (A + B) + Tr\left(A^T A + C\right)^T \left(A^T A + C\right) \tag{15}$$

denote

$$J = Tr(A + B)^T (A + B) + Tr\left(A^T A + C\right)^T \left(A^T A + C\right) \tag{16}$$

then

$$\frac{\partial J}{\partial A} = 2\,(A + B) + 4AA^T A + 4ACC^T = 0 \tag{17}$$

i.e.

$$A + 2AA^T A + 2ACC^T = -B \tag{18}$$

Subspace Clustering III

$$\min_{C,\mathcal{R}} \|X - XC\|_F^2 + \lambda \mathcal{R}(C) \tag{19}$$

How to solve 19?

Deep proximal gradient.

Application II: Salient Object Detection

$$\min_{C,X,\mathcal{R}} \|Y - XC\|_F^2 + \mathcal{R}(C) \tag{20}$$

or

$$\min_{f,\mathcal{R}} \|Y - f(X)\|_F^2 + \lambda\mathcal{R}(X) \tag{21}$$

or

$$\begin{aligned}\min_{f,L.S,\mathcal{R}} \|Y - f(X)\| + \lambda_1\|L\|_* + \lambda_2\|S\|_1 + \langle Y_1, X - (L+S)\rangle \\ + \frac{\lambda_3}{2}\|X - (L+S)\|_F^2 + \frac{\lambda_4}{2}\mathcal{R}(X)\end{aligned} \tag{22}$$

Application III: Linear Inverse Problems

Linear Inverse Problems I

It is a typical ill-posed linear inverse problem and can be generally formulated as:

$$\boldsymbol{y} = H\boldsymbol{x} + \boldsymbol{n} \tag{23}$$

where $\boldsymbol{x}, \boldsymbol{y}$ are lexicographically stacked representations of the original image and the degraded image, respectively, H is a matrix representing a non-invertible linear degradation operator and $\boldsymbol{n}$ is usually additive Gaussian white noise.

Our model:

$$\arg\min_{\mathbf{x}_i} \frac{1}{2}\sum_{i=1}^{3} \|Hx_i - y_i\|_2^2 + \lambda\|(ten\left[mat\left(x_1\right), mat\left(x_2\right), mat\left(x_3\right)\right])\|_* \tag{24}$$

More details: https://zhims.github.io/doc/note/datascience/LRS/GSR.pdf.

Linear Inverse Problems II

There is a recommended slides https://imaging-in-paris.github.io/semester2019/slides/w1/willett.pdf.

And, there are two recommended papers:

1. IEEE Transactions on Computational Imaging: https://arxiv.org/pdf/1901.03707.pdf.
2. IEEE Journal on Selected Areas in Information Theory: https://arxiv.org/pdf/2005.06001.pdf.

More details: https://zhims.github.io/doc/note/datascience/LRS/NeumannNetworks.pdf.

References

M. Udell and A. Townsend (2018)
Why Are Big Data Matrices Approximately Low Rank?
SIAM Journal on Mathematics of Data Science, Vol1(1):144-160, 2018.

M. Scetbon, M. Elad and P. Milanfar(2020)
Deep K-SVD Denoising
IEEE Selected Topics in Signal Processing (Special Issue on Deep Learning for Image/Video Restoration and Compression)

E. Elhamifar and R. Vidal (2013)
Sparse Subspace Clustering: Algorithm, Theory, and Applications
IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol35(11):2765–2781, 2013.

G. Golub and C. Van Loan (2013)
Matrix Computations
Johns Hopkins University Press, 2013.

T. Kolda and B. Bader (2009)
Tensor Decompositions and Applications
SIAM Review, Vol51(3): 455-500, 2009.

High-Dimensional Data Analysis with Low-Dimensional Models:

Principles, Computation, and Applications

Authors: John Wright and Yi Ma

https://book-wright-ma.github.io/Book-WM-20201206.pdf.

Thank you!

Any comments or questions?

Welcome to visit my personal homepage: https://zhims.github.io/