計算科学における情報圧縮

Information Compression in Computational Science

2021.10.28

#4:情報圧縮の数理2 （特異値分解と低ランク近似）

Singular value decomposition and low rank approximation

理学系研究科　大久保　毅

Graduate School of Science, **Tsuyoshi Okubo**

Outline

- Singular value decomposition (SVD) (cont.)
- Generalized inverse matrix (Quick review)
- Low rank approximation
  - Low rank approximation by SVD
  - Low "rank" approximation for tensor
- Application of the low rank approximations to images

Singular value decomposition

Singular value decomposition (SVD)

Singular value decomposition
(特異値分解)

$$A : M \times N$$

$$A_{ij} \in \boldsymbol{C}$$

$$A = U \Sigma V^{\dagger}$$

$U : M \times M$ **Unitary**

$V : N \times N$ **Unitary**

$$\Sigma = \begin{pmatrix} \Sigma_{r \times r} & 0_{r \times (N-r)} \\ 0_{(M-r) \times r} & 0_{(M-r) \times N-r} \end{pmatrix}$$

$$\Sigma_{r \times r} = \begin{pmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_r \end{pmatrix}$$

Diagonal matrix with
non-negative real elements

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$$

Singular values

Properties of SVD 1

1. Any matrices can be decomposed as SVD: $A = U\Sigma V^\dagger$

$A : M \times N \Rightarrow A^\dagger A : N \times N$

$*A^\dagger A$ is a Hermitian matrix.

$$(A^\dagger A)^\dagger = A^\dagger A \Rightarrow$$

It can be diagonalized by
a unitary matrix V .

$$V^\dagger (A^\dagger A) V = \mathrm{diag}\{\lambda_1, \lambda_2, \cdots, \lambda_N\}$$

$$V = (\vec{v}_1, \vec{v}_2, \cdots, \vec{v}_N)$$

$\vec{v}_i$: eigenvector

* $A^\dagger A$ is a positive semi-definite matrix.
(半正定値、準正定値)

$$\vec{x}^* \cdot (A^\dagger A \vec{x}) = \|A\vec{x}\|^2 \geq 0 \Leftrightarrow$$

Its eigenvalues are
non-negative

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N \geq 0$$

Properties of SVD 1

1. Any matrices can be decomposed as SVD: $A = U\Sigma V^\dagger$

$$V^\dagger(A^\dagger A)V = \mathrm{diag}\{\lambda_1, \lambda_2, \cdots, \lambda_N\}$$
$$V = (\vec{v}_1, \vec{v}_2, \cdots, \vec{v}_N)$$

$\Leftrightarrow$

$$(A\vec{v}_i)^* \cdot (A\vec{v}_j) = \lambda_i \delta_{ij}$$
$$(\|A\vec{v}_i\|^2 = \lambda_i)$$

Suppose $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r > 0 = \lambda_{r+1} = \cdots = \lambda_N$

(There are r positive eigenvalues.)

Make new orthonormal basis $U = (\vec{u}_1, \vec{u}_2, \cdots, \vec{u}_M)$ in $\mathbb{C}^M$

For $(i = 1, 2, \ldots, r)$ $\quad \sigma_i = \sqrt{\lambda_i}\,, \vec{u}_i = \dfrac{1}{\sigma_i} A\vec{v}$

For $(i = r+1, \ldots, M)$ Any orthonormal basis orthogonal to $\vec{u}_i \quad (i = 1, 2, \ldots, r)$

$$\vec{u}_i^* \cdot (A\vec{v}_j) = \sigma_i \delta_{ij} \quad (i = 1, \ldots, M; j = 1, \ldots, N)$$

(For simplicity, we set $\sigma_i = 0$ for $i > r$.)

Properties of SVD 1

1. Any matrices can be decomposed as SVD: $A = U\Sigma V^\dagger$

   We can perform same "proof" by using $AA^\dagger$.

   $U = (\vec{u}_1, \vec{u}_2, \cdots, \vec{u}_M)$ is the unitary matrix which diagonalize $AA^\dagger$ as

$$U^\dagger (AA^\dagger) U = \mathrm{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_r, \underbrace{0, \ldots, 0}_{M-r}\}$$

In summary,

- A matrix A can be decomposed as SVD: $A = U\Sigma V^\dagger$
- Singular values are related to the eigenvalues of $A^\dagger A$ and $AA^\dagger$ as

$$\sigma_i = \sqrt{\lambda_i} \ .$$

- V and U are eigenvectors of $A^\dagger A$ and $AA^\dagger$,respectively.

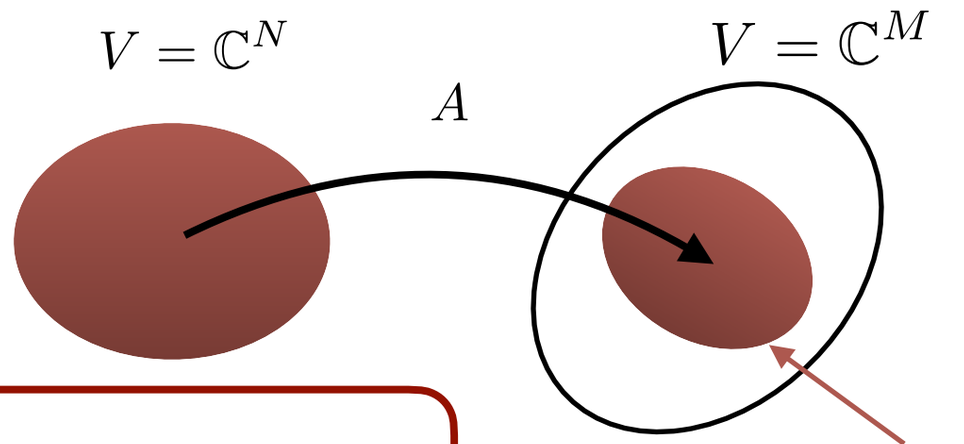
Properties of SVD 2

$$A = U\Sigma V^\dagger$$

2. # of positive singular values is identical with the rank.

$$A : M \times N \Rightarrow A : \mathbb{C}^N \to \mathbb{C}^M$$

$$\mathrm{rank}(A) \equiv \dim(\mathrm{img}(A))$$



Remember

The orthonormal basis $\{\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_N\}$ satisfies

$$(A\vec{v}_i)^* \cdot (A\vec{v}_j) = \lambda_i \delta_{ij}$$

Here, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r > 0 = \lambda_{r+1} = \cdots = \lambda_N$ and $\sigma_i = \sqrt{\lambda_i}$

$$\forall \vec{x} \in \mathbb{C}^N, \vec{x} = \sum_{i=1}^{N} C_i \vec{v}_i \Rightarrow A\vec{x} = \sum_{i=1}^{N} C_i (A\vec{v}_i) = \sum_{i=1}^{r} C_i (A\vec{v}_i)$$

$$\Rightarrow \mathrm{img}(A) = \mathrm{Span}\{A\vec{v}_1, A\vec{v}_2, \ldots, A\vec{v}_r\}$$

$$\Rightarrow \dim(\mathrm{img}(A)) = r$$ **=# of positive singular values**

Properties of SVD 3 (optional)

$$A = U\Sigma V^\dagger$$

3. Singular vectors

$A : M \times N$ $\qquad U = (\vec{u}_1, \vec{u}_2, \cdots, \vec{u}_M)\,, V = (\vec{v}_1, \vec{v}_2, \cdots, \vec{v}_N)$

For $i = 1, 2, \ldots, r$

$$A\vec{v}_i = \sigma_i \vec{u}_i \ , \ A^\dagger \vec{u}_i = \sigma_i \vec{v}_i$$

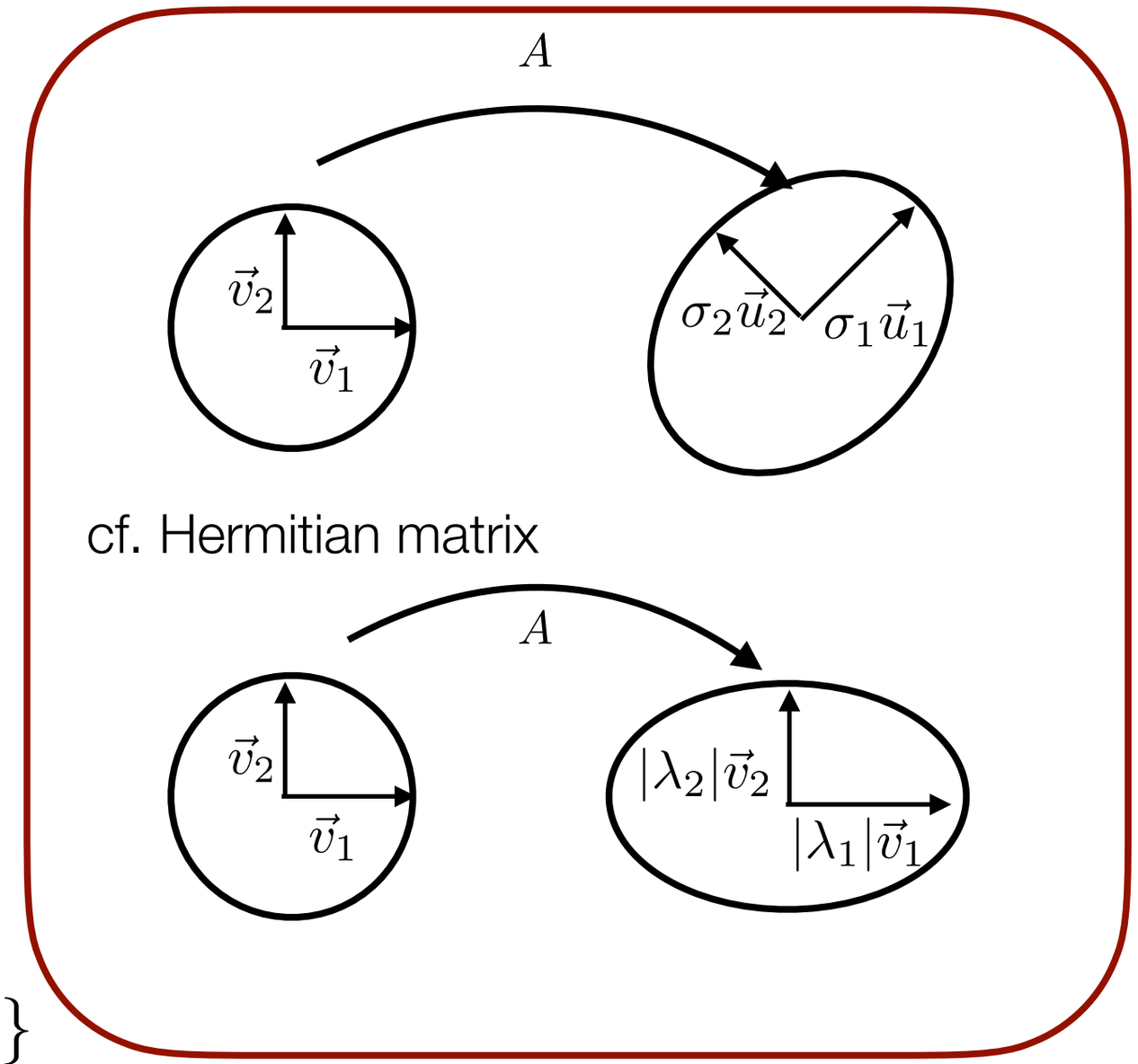
$\vec{v}_i$: right singular vector

$\vec{u}_i$: left singular vector

Relation to image and kernel:

$$\mathrm{img}(A) = \mathrm{Span}\{\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_r\}$$
$$\ker(A) = \mathrm{Span}\{\vec{v}_{r+1}, \vec{v}_{r+2}, \ldots, \vec{v}_N\}$$

$$\mathrm{img}(A^\dagger) = \mathrm{Span}\{\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_r\}$$
$$\ker(A^\dagger) = \mathrm{Span}\{\vec{u}_{r+1}, \vec{u}_{r+2}, \ldots, \vec{u}_M\}$$



Properties of SVD 4 (optional)

$$A = U\Sigma V^\dagger$$

4. Min-max theorem (Courant-Fischer theorem)

$A : N \times N$, Hermitian matrix

Suppose its eigenvalues are $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N$.

$$\lambda_k = \min_{\boldsymbol{S};\dim(\boldsymbol{S})\leq k-1} \max_{\vec{x}\in\boldsymbol{S}^\perp;\|\vec{x}\|=1} \vec{x}^* \cdot A\vec{x}$$

$$\boldsymbol{S}^\perp = \{\vec{x} : \vec{x}^* \cdot \vec{y} = 0, \vec{y} \in \boldsymbol{S}\}$$

Orthonormal complement（直交補空間）

We can prove this by considering vector subspace spanned by eigenvectors. (see references)

Intuitive examples:

Maximum appears for the eigenvector.

$$A\vec{u}_i = \lambda_i \vec{u}_i$$

$$\lambda_1 = \max_{\vec{x}\in\boldsymbol{C}^N;\|\vec{x}\|=1} \vec{x}^* \cdot (A\vec{x}) \quad \Rightarrow \quad \vec{x} = \vec{u}_1$$

$$\lambda_2 = \max_{\vec{x}\in\boldsymbol{C}^N;\vec{x}\perp\vec{u}_1,\|\vec{x}\|=1} \vec{x}^* \cdot (A\vec{x}) = \min_{\boldsymbol{S};\dim(\boldsymbol{S})\leq 1} \max_{\vec{x}\in\boldsymbol{S}^\perp;\|\vec{x}\|=1} \vec{x}^* \cdot (A\vec{x}) \quad \Rightarrow \quad \vec{x} = \vec{u}_2$$

Properties of SVD 4 (optional)

$$A = U\Sigma V^\dagger$$

4. Min-max theorem (Courant-Fischer theorem)

$A : M \times N$

Suppose its singular values are $\sigma_1 \geq \sigma_2 \geq \cdots$.

$$\sigma_k = \min_{\boldsymbol{S};\dim(\boldsymbol{S})\leq k-1} \max_{\vec{x}\in\boldsymbol{S}^\perp;\|\vec{x}\|=1} \|A\vec{x}\|$$

By setting k=1,

$$\sigma_1 = \max_{\vec{x}\in\boldsymbol{C}^N,\|\vec{x}\|=1} \|A\vec{x}\|$$

which means

$$\|A\vec{x}\| \leq \sigma_1 \|\vec{x}\|$$

for $\vec{x} \in \boldsymbol{C}^N$

We can easily prove this by using

$A^\dagger A$: Hermitian

$$A^\dagger A \vec{v}_i = \lambda_i$$

$$\sigma_i = \sqrt{\lambda_i}$$

Properties of SVD 5 (optional)

$$A = U\Sigma V^\dagger$$

5. Singular values for multiplication and addition

$\sigma_i(A)$: singular value of matrix A

(for $i > \mathrm{rank}(A)$, we set $\sigma_i = 0$)

*Following properties can be proven by using min-max theorem.

Multiplication: $A : M \times L, B : L \times N$

$$\sigma_k(AB) \leq \sigma_1(A)\sigma_k(B) \qquad (k = 1, 2, \dots)$$

$$(\sigma_k(AB) \leq \sigma_k(A)\sigma_1(B))$$

$$\Rightarrow \quad \mathrm{rank}(AB) \leq \min(\mathrm{rank}(A), \mathrm{rank}(B))$$

Addition: $A, B : M \times N$

$$\sigma_{k+j-1}(A+B) \leq \sigma_k(A) + \sigma_j(B) \qquad (k, j = 1, 2, \dots)$$

$$(\sigma_{k+j-1}(A+B) \leq \sigma_j(A) + \sigma_k(B))$$

$$\Rightarrow \quad \text{If } \mathrm{rank}(B) \leq r \text{ ,}$$

$$\sigma_{k+r}(A+B) \leq \sigma_k(A)$$

Libraries for SVD

There are **LAPACK** routines for SVD.

DGESDD, ZGESDD

DGESVD, ZGESVD

(For dense matrices)

*Linear Algebra PACKage

At *netlib.org* (reference implementations)

+

A lot of vender implementations

- Intel MKL
- Apple Accelerate Framework
- Fujitsu SSLII
- ...

numpy and **scipy** modules in python have routines for SVD.

numpy.linalg.svd

scipy.linalg.svd

(For dense matrices)

scipy.sparse.linalg.svds

(For sparse matrices or
calculation of partial singular values)

Computation cost

For a $M \times N$ matrix ($M \leqq N$):

Full SVD: $O(NM^2)$

Partial SVD: $O(NMk)$

k : # of singular values
to be calculated

Generalized inverse matrix (Quick review)

Regular matrix and its inverse matrix

A square matrix *A* is a **regular matrix** （正則）if a matrix *X* satisfying

$$AX = XA = I$$

exists. The matrix *X* is called inverse matrix（逆行列） of *A* and it is written as $X = A^{-1}$.

Properties: A^{-1} is unique.

$$(A^{-1})^{-1} = A$$

$$(AB)^{-1} = B^{-1}A^{-1}$$

A is a regular matrix $\Leftrightarrow$ $\mathrm{rank}(A) = N$

Can we consider an "inverse matrix" of a non-regular matrix (including a rectangular matrix) ?

Generalized inverse matrix

Generalized inverse matrix（一般化逆行列）：

For $A : M \times N$, a matrix $A^- : N \times M$ satisfying

$$AA^-A = A$$

is called generalized inverse matrix.

Properties:

- Generalized inverse matrix is not unique.
  - At least one generalized matrix exists.

- If A is a regular matrix, $A^- = A^{-1}$

  → A^- is a generalization of inverse matrix.

Moore-Penrose pseudo inverse

Moore-Penrose pseudo inverse matrix（擬似逆行列）:

For $A : M \times N$, a matrix $A^+ : N \times M$ satisfying

(1) $AA^+A = A$ (2) $A^+AA^+ = A^+$

(3) $(AA^+)^\dagger = AA^+$ (4) $(A^+A)^\dagger = A^+A$

is called (Moore-Penrose) pseudo inverse matrix.

Relation to SVD

- Pseudo inverse is unique and calculated from SVD.

$$A = U\Sigma V^\dagger = U \begin{pmatrix} \Sigma_{r\times r} & 0_{r\times(N-r)} \\ 0_{(M-r)\times r} & 0_{(M-r)\times N-r} \end{pmatrix} V^\dagger$$

$$\Rightarrow A^+ = V \begin{pmatrix} \Sigma^{-1}_{r\times r} & 0_{r\times(M-r)} \\ 0_{(N-r)\times r} & 0_{(N-r)\times M-r} \end{pmatrix} U^\dagger$$

$$A^+A = V \begin{pmatrix} \Sigma^{-1}_{r\times r} & 0_{r\times(M-r)} \\ 0_{(N-r)\times r} & 0_{(N-r)\times M-r} \end{pmatrix} U^\dagger U \begin{pmatrix} \Sigma_{r\times r} & 0_{r\times(N-r)} \\ 0_{(M-r)\times r} & 0_{(M-r)\times N-r} \end{pmatrix} V^\dagger$$

$$= \sum_{i=1}^{r} \vec{v}_i \vec{v}_i^\dagger \left(= \sum_{i=1}^{r} |v_i\rangle\langle v_i|\right)$$

A^+A is a projector onto img($A^\dagger$).

(AA^+ is a projector onto img(A).)

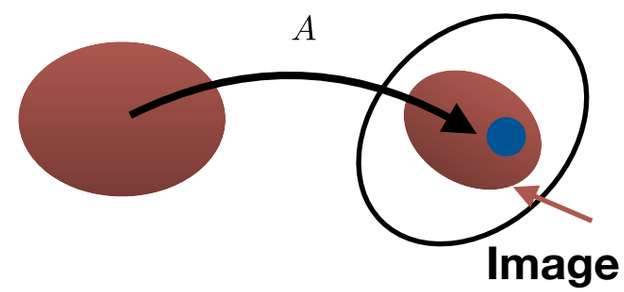
Simultaneous linear equation

Simultaneous linear equation（連立一次方程式）

$$A\vec{x} = \vec{b} \qquad A : M \times N, \vec{x} \in \mathbb{C}^N, \vec{b} \in \mathbb{C}^M \qquad \vec{b} :$$

Two situations:

(1) There are solutions. $\Leftrightarrow$ $\vec{b} \in \mathrm{img}(A)$



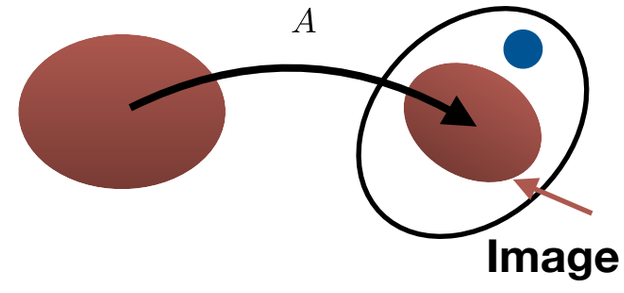
(i) There is the unique solution.

$$\mathrm{rank}(A) = N$$

(ii) There are infinite solutions (underdetermined).

$$\mathrm{rank}(A) < N$$ (We can add any vector $A\vec{y} = \vec{0}$.)

(2) There is no solution. $\Leftrightarrow$ $\vec{b} \notin \mathrm{img}(A)$
(overdetermined)



Pseudo inverse and simultaneous linear equation

Simultaneous linear equation $A\vec{x} = \vec{b}$ $A : M \times N, \vec{x} \in \mathbb{C}^N, \vec{b} \in \mathbb{C}^M$

(1) There are solutions. $\Leftrightarrow$ $\vec{b} \in \mathrm{img}(A)$

- A vector defined by the pseudo inverse as

$$\vec{x}' \equiv A^+ \vec{b}$$

is one of the solutions.

Because $\vec{b} \in \mathrm{img}(A)$,there exists $\vec{v} : A\vec{v} = \vec{b}$.

$$A\vec{x}' = AA^+\vec{b} = AA^+A\vec{v} = A\vec{v} = \vec{b}$$

- $\vec{x}'$ has the smallest norm $\|\vec{x}'\|$ among the solutions.

$$\|\vec{x}\| \geq \|A^+A\vec{x}\| = \|A^+\vec{b}\| = \|\vec{x}'\|$$

$\because A^+A$ is a projector.

The pseudo inverse gives us the smallest norm solution.

Pseudo inverse and simultaneous linear equation

Simultaneous linear equation $A\vec{x} = \vec{b}$ $A : M \times N, \vec{x} \in \mathbb{C}^N, \vec{b} \in \mathbb{C}^M$

(2) There is no solution. $\Leftrightarrow$ $\vec{b} \notin \mathrm{img}(A)$

- A vector defined by the pseudo inverse as

$$\vec{x}' \equiv A^+\vec{b}$$

minimizes the "distance" $\|b - A\vec{x}\|$.

$$\vec{y} = A\vec{c} \in \mathrm{img}(A), \vec{c} \in \mathbb{C}^N$$

$$\|\vec{y} - \vec{b}\|^2 = \|\underbrace{\vec{y} - AA^+\vec{b}}_{\mathrm{img}(A)} - \underbrace{(I - AA^+)\vec{b}}_{\mathrm{img}(A)^\perp}\|^2$$

$$= \|\vec{y} - AA^+\vec{b}\|^2 + \|\vec{b} - AA^+\vec{b}\|^2$$

$$\geq \|\vec{b} - AA^+\vec{b}\|^2 = \|\vec{b} - A\vec{x}'\|^2$$

The pseudo inverse gives us approximate "least square solution".

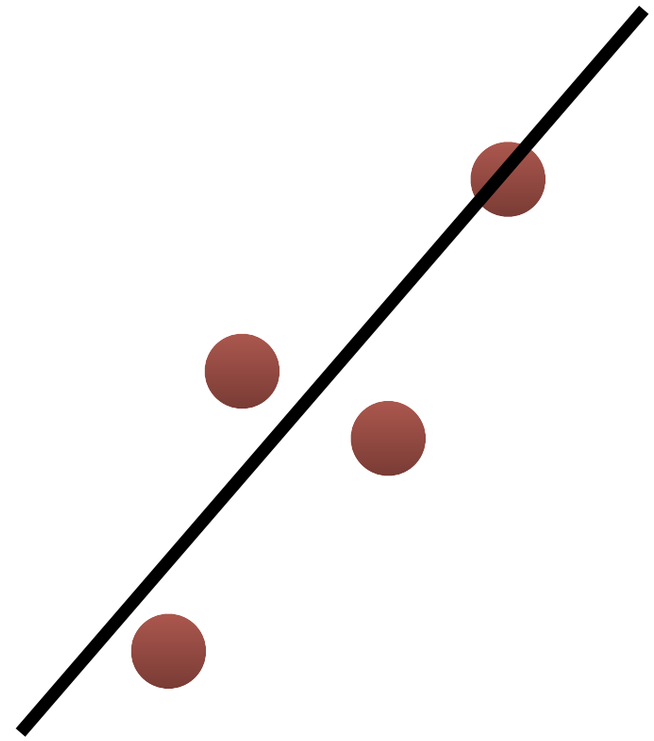
Example of Least square solution problem

Fitting of a line to data points

$$y = ax + b$$

Data poins:

$$(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$$



$$\begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ x_3 & 1 \\ x_4 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix}$$

$$A\vec{x} = \vec{b}$$

Least square fitting（最小二乗法）

$$\begin{pmatrix} a \\ b \end{pmatrix} = A^{+} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix}$$

Low rank approximation

Amount of data in SVD representation

$A : M \times N$

$$A = U\Sigma V^\dagger = U \begin{pmatrix} \Sigma_{r\times r} & 0_{r\times(N-r)} \\ 0_{(M-r)\times r} & 0_{(M-r)\times N-r} \end{pmatrix} V^\dagger$$

neglect zero singular values →

$$= \bar{U}\Sigma_{r\times r}\bar{V}^\dagger$$

$$\bar{U} : M \times r, \bar{V}^\dagger : r \times N$$

If rank(A) is much smaller than M and N,

$$r \ll M, N$$

we can reduce the data to represent A.

(At this stage, no data loss)

$$U = (\vec{u}_1, \vec{u}_2, \cdots, \vec{u}_M)$$
$$V = (\vec{v}_1, \vec{v}_2, \cdots, \vec{v}_N)$$

↓

$$\bar{U} = (\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_r)$$
$$\bar{V} = (\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_r)$$

Low rank approximation

Low rank approximation（低ランク近似）

Find an approximate matrix

$$A \simeq \tilde{A}$$

with lower rank:

$$\mathrm{rank}(\tilde{A}) > \mathrm{rank}(A)$$

Through the low rank approximation,
we can reduce amount of the data.

An example of information compressions.

Notice! In order to quantify accuracy of the approximation,
we need a measure of distance between matrices.

Low rank approximation by SVD

Consider a matrix obtained by neglecting smaller singular values

$$A = \bar{U}\Sigma_{r\times r}\bar{V}^\dagger \quad \Rightarrow \quad \tilde{A} = \tilde{U}\Sigma_{k\times k}\tilde{V}^\dagger \qquad (k < r)$$

$$\Sigma_{r\times r} = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r)$$
$$\bar{U} = (\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_r)$$
$$\bar{V} = (\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_r)$$

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$$

$$\mathrm{rank}(A) = r$$

$$\Sigma_{k\times k} = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_k)$$
$$\tilde{U} = (\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_k)$$
$$\tilde{V} = (\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_k)$$

Keep the largest k singular values
(and corresponding singular vectors).

$$\mathrm{rank}(\tilde{A}) = k < r$$

This approximation is one of the low rank approximation.

* For this approximation, we need $\mathrm{O}(MNk)$ calculations for SVD of a $M\times N$ matrix.

Norm of matrices $\|A\|$

There are two popular norms:

(1) **Frobenius norm** （フロベニウス　ノルム）

$$\|A\|_F = \sqrt{\sum_{i,j} |A_{ij}|^2} = \sqrt{\mathrm{Tr}(A^\dagger A)}$$

*Trace（対角和）

$$\mathrm{Tr}(X) = \sum_i X_{ii}$$

(2) **Operator norm**（作用素ノルム）

$$\|A\|_O = \inf\{c \geq 0; \|A\vec{x}\| \leq c\|\vec{x}\|\}$$
$$= \sigma_1(A)$$

*inf =infimum（下限）

*We define the norm for a vector as

$$\|\vec{x}\| = \sqrt{\sum_i |x_i|^2}$$

By using these norms, we define the distance between matrices:

$$\|A - \tilde{A}\|$$

Accuracy of low rank approximation by SVD

(A part of) Eckart-Young-Mirsky Theorem

C. Eckart, and G. Young, Psychometrika **1**, 211 (1936).
L. Mirsky, Q. J. Math. **11**, 50 (1960).

For $A : M \times N$

$$\min\{\|A - B\|_F : \mathrm{rank}(B) = k\} = \sqrt{\sum_{i=k+1}^{\min(N,M)} \sigma_i^2}$$

$$\min\{\|A - B\|_O : \mathrm{rank}(B) = k\} = \sigma_{k+1}$$

Because the k-rank approximation by SVD gives

$$\|A - \tilde{A}\|_F = \sqrt{\sum_{i=k+1}^{\min(N,M)} \sigma_i^2}, \qquad \|A - \tilde{A}\|_O = \sigma_{k+1}$$

it is an "optimal" approximation with rank k.

Short proof of the theorem: Frobenius norm (optional)

*This proof is based on
"システム制御のための数学（１）"太田快人 著

From the inequality of singular values for matrix addition (property 5),

for j=1,... ,$\min(M,N)$ $\quad (\mathrm{rank}(B) = k)$

$$\sigma_{j+k}(A) = \sigma_{j+k}((A-B)+B) \leq \sigma_j(A-B)$$

Property 5

By taking a square and summing up them

$$\sum_{i=k+1}^{\min(M,N)} \sigma_i^2(A) \leq \sum_{j=1}^{\min(M,N)} \sigma_j^2(A-B) = \|A-B\|_F^2$$

*Note $\sigma_j(A) = 0 \quad (j > \mathrm{rank}(A))$

Short proof of the theorem: operator norm (optional)

*This proof is based on
"システム制御のための数学（１）"太田快人 著

From the min-max theorem of singular values (property 4),

$$(\mathrm{rank}(B) = k)$$

$$\sigma_{k+1}(A) \leq \max_{\vec{x}\in\ker(B),\|\vec{x}\|=1} \|A\vec{x}\| = \max_{\vec{x}\in\ker(B),\|\vec{x}\|=1} \|(A-B)\vec{x}\|$$

Property4 with
$\boldsymbol{S} = \mathrm{img}(B^\dagger)$
$\boldsymbol{S}^\perp = \ker(B)$

$B\vec{x} = 0 \quad (\vec{x} \in \ker(B))$

$$\leq \max_{\|\vec{x}\|=1} \|(A-B)\vec{x}\| = \|A-B\|_O$$

Expand the vector space

Definition of the operator norm

Relation to **principal component analysis** （主成分分析）

Data set $\{X_{ij}\}$: X: $N \times M$ matrix

i = index for data, j = data type (coordinates, momentum, ...)

* Suppose the mean of data is 0: $\sum_i X_{ij} = 0$

Covariance matrix（共分散行列）: $C = X^T X$

Principal component analysis (PCA):

Data compression through the spectrum decomposition of C.

$C = V \Lambda V^T$ Λ: diagonal matrix, $\Lambda_{ii} = \lambda_i \geq 0$

$V = (\vec{v}_1, \vec{v}_2, \cdots, \vec{v}_N)$

$\vec{v}_i$ corresponding to large λ_i contains important information.

By construction, λ and V are related to SVD of X!

$X = U \Sigma V^T$ $, \sigma_i = \sqrt{\lambda_i}$

PCA can be regarded as the low rank approximation X.

Generalization to tensors

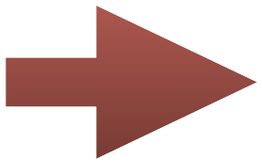
Scalar, Vector, Matrix, Tensor,...

Scalar: c Number

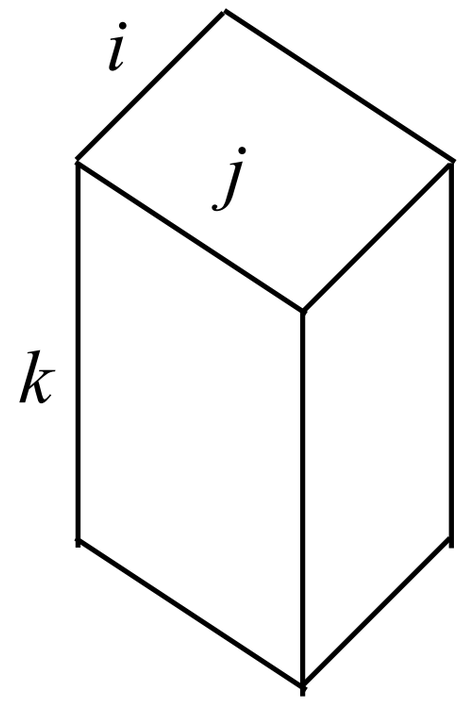
Vector: v_i One dimensional array of numbers

Matrix: M_{ij} Two dimensional array of numbers

Tensor: $T_{ijk\ldots}$ Higher dimensional array of numbers



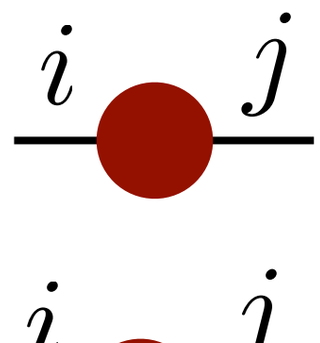
Scalar: 0-dim. tensor
Vector: 1-dim. tensor
Matrix: 2-dim. tensor



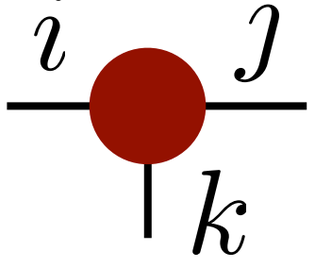
Graphical representations for tensor network

- Vector $\vec{v} : v_i$

- Matrix $M : M_{i,j}$

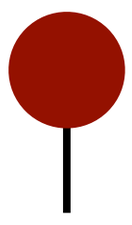


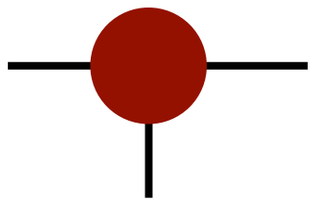
- Tensor $T : T_{i,j,k}$



*** n-rank tensor = n-leg object**

When indices are not presented in a graph, it represent a tensor itself.

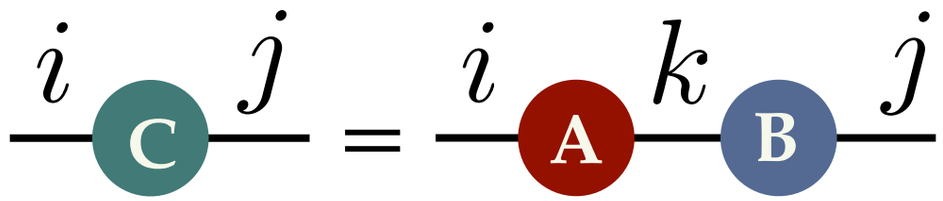
$\vec{v} =$ 

$T =$ 

Graphical representations for tensor network

Matrix product

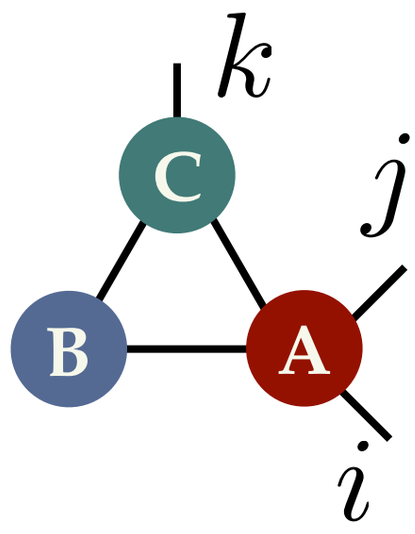
$$C_{i,j} = (AB)_{i,j} = \sum_k A_{i,k} B_{k,j}$$



$$C = AB$$

Generalization to tensors

$$\sum_{\alpha,\beta,\gamma} A_{i,j,\alpha,\beta} B_{\beta,\gamma} C_{\gamma,k,\alpha}$$



Contraction of a network = Calculation of a lot of multiplications

Low rank approximation: generalization to tensor

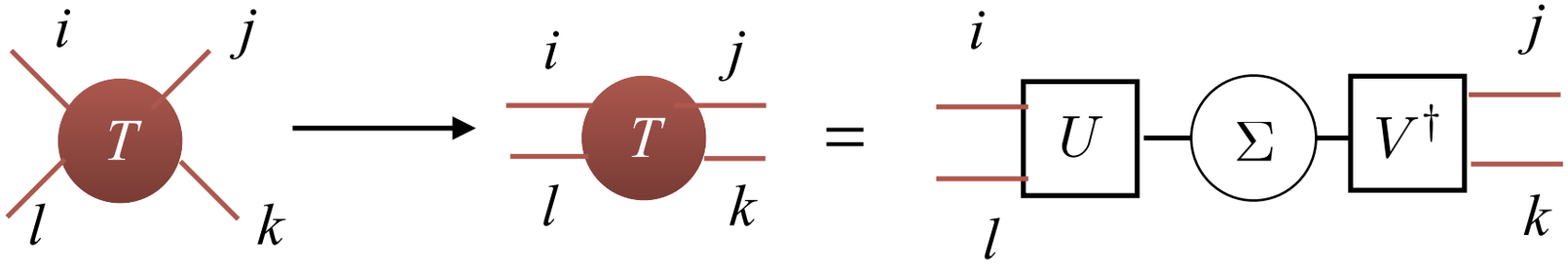
Tensor: $T_{ijk\ldots}$

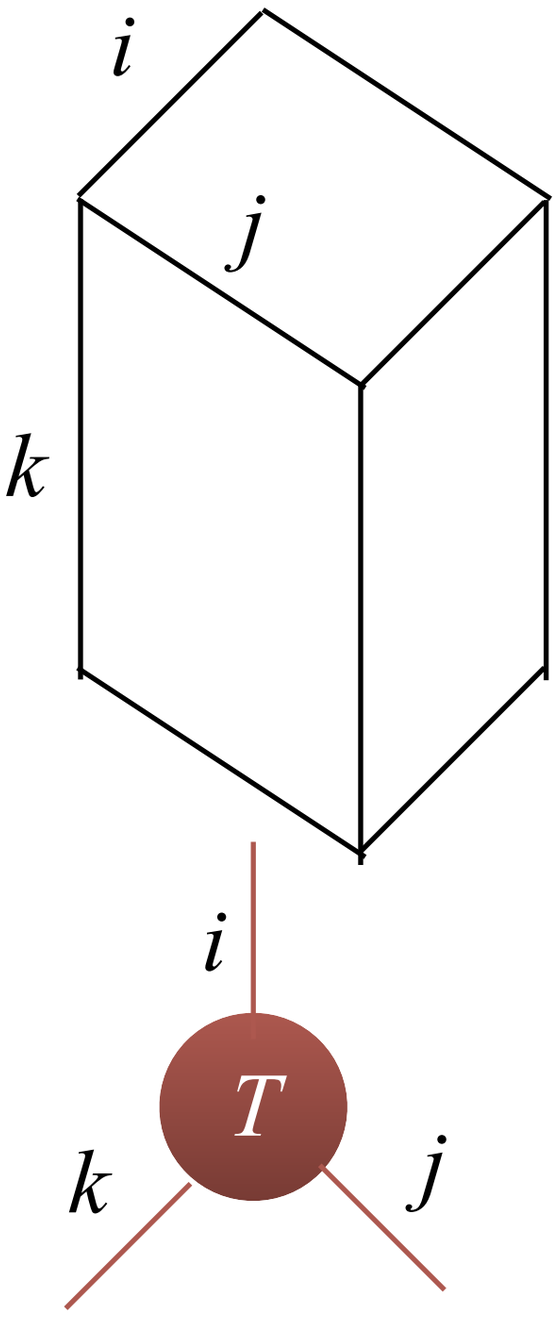
Naive application of SVD:

Make a matrix by dividing indices into two parts.

$$T_{ijkl} \to T_{(il),(jk)}$$

Then apply SVD (and low rank approximation).



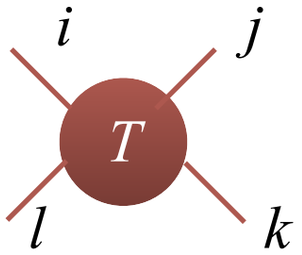


Note: The result depends on the initial mapping to a matrix.

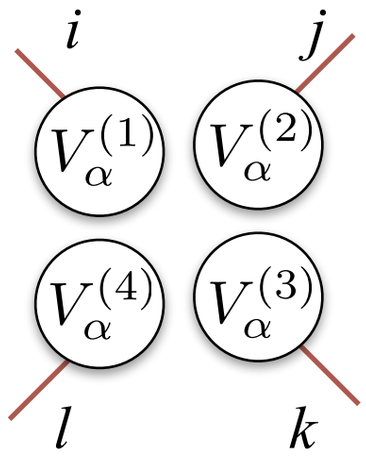
CP decomposition

Review: T. G. Kolda et al, SIAM Review **51**, 455 (2009)

CP (Canonical Polyadic) decomposition: Hitchcock (1927)



$$= \sum_{\alpha=1}^{r}$$



$$T_{ijkl} = \sum_{\alpha=1}^{r} (V_\alpha^{(1)})_i (V_\alpha^{(2)})_j (V_\alpha^{(3)})_k (V_\alpha^{(4)})_l$$

min (r) = tensor rank

*Determining tensor rank is NP-hard problem.

Low "rank" approximation

$$T_{ijkl} \approx \sum_{\alpha=1}^{r'} (\tilde{V}_\alpha^{(1)})_i (\tilde{V}_\alpha^{(2)})_j (\tilde{V}_\alpha^{(3)})_k (\tilde{V}_\alpha^{(4)})_l$$

$r' < r$

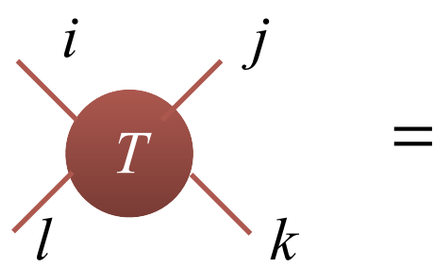
rank-r' approximation

Tucker decomposition: generalization of SVD

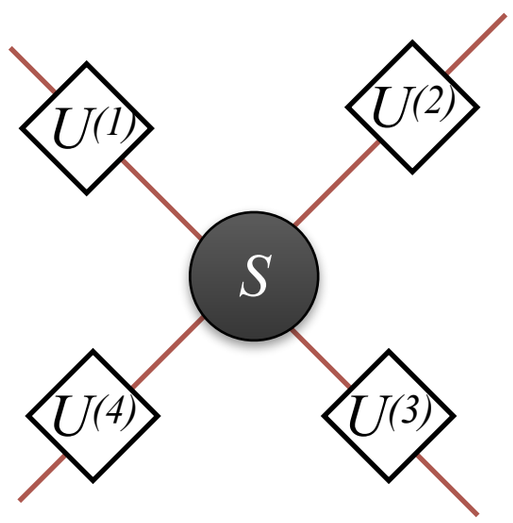
Review: T. G. Kolda et al, SIAM Review **51**, 455 (2009)

Tucker decomposition:

(Tucker (1963))



=



$U^{(i)}$: Factor matrix (usually unitary)

S :Core tensor

$$T_{ijkl} = \sum_{i'=1}^{I}\sum_{j'=1}^{J}\sum_{k'=1}^{K}\sum_{l'=1}^{L} S_{i'j'k'l'} U^{(1)}_{ii'} U^{(2)}_{jj'} U^{(3)}_{kk'} U^{(4)}_{ll'}$$

*If S is "diagonal", Tucker decomposition becomes CP decomposition.

Low "rank" approximation

$$T_{ijkl} = \sum_{i'=1}^{I'}\sum_{j'=1}^{J'}\sum_{k'=1}^{K'}\sum_{l'=1}^{L'} \tilde{S}_{i'j'k'l'} \tilde{U}^{(1)}_{ii'} \tilde{U}^{(2)}_{jj'} \tilde{U}^{(3)}_{kk'} \tilde{U}^{(4)}_{ll'}$$

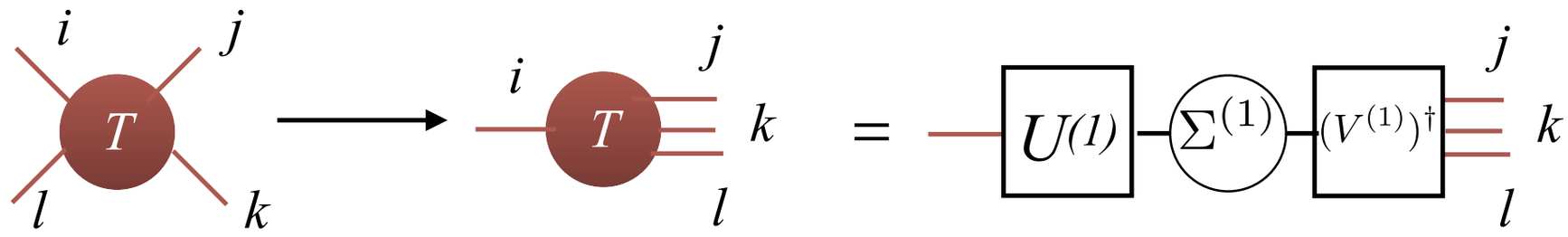
$I' < I, \quad J' < J, \quad K' < K, \quad L' < L$

rank-(I',J', K', L') approximation

Higher order SVD (HOSVD)

L. De Lathauwer et al, SIAM J. Matrix Anal. & Appl., **21**, 1253 (2000)

Define a factor matrix from matrix SVD:



Core tensor is calculated as

$$S_{i'j'k'l'} \equiv \sum_{ijkl} T_{ijkl} (U^{(1)})^{\dagger}_{i'i} (U^{(2)})^{\dagger}_{j'j} (U^{(3)})^{\dagger}_{k'k} (U^{(4)})^{\dagger}_{l'l}$$

Properties of the core tensor

$$S^{*}_{:,i_n=\alpha,:,:} \cdot S_{:,i_n=\beta,:,:} = \begin{cases} 0 & (\alpha \neq \beta) \\ (\sigma^{(n)}_{\alpha})^2 & (\alpha = \beta) \end{cases}$$

Dot product

$$A \cdot B \equiv \sum_{i,j,k,l} A_{ijkl} B_{ijkl}$$

Generalization of the diagonal matrix Σ in matrix SVD.

* Low-rank approximation based on HOSVD is not optimal.

Application of low rank approximation

Sample codes are uploaded on github and ITC-LMS.

Image_SVD.zip

(python and jupyter notebook codes)

Image compression: grayscale image

Image: 1024 × 768 pixels

768 x1024 matrix A

$$\mathrm{rank}(A) = 768$$

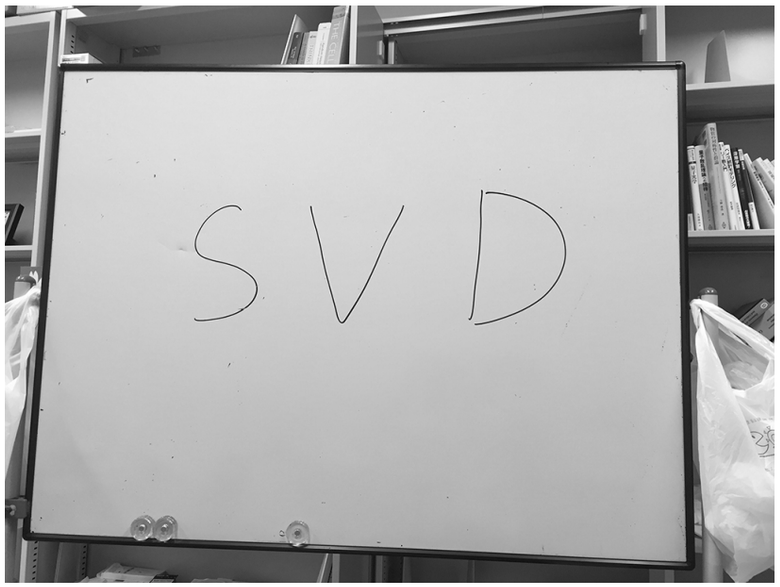
Amount of data=786,432

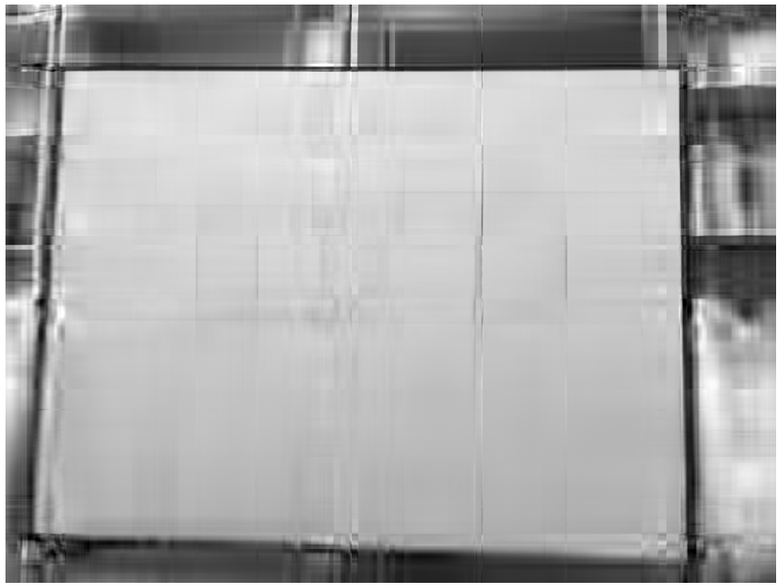
Perform SVD of A: $A = U\Sigma V^{\dagger}$

$\mathrm{rank}(\chi)$ approximation

Amount of data=(768 +1024 + 1)×χ

Image compression: grayscale image





Rank: $\chi = 768$ $\chi = 100$ $\chi = 10$

Data: **786,432** **179,300** **179,30**

(Original)

Image compression: color image

Image: 1024 × 768 pixels

768 x1024x3 tensor T

Amount of data=2,359,296

* Sub matrices for RGB colors

$$R_{ij} = T_{ij1}, \quad G_{ij} = T_{ij2}, \quad B_{ij} = T_{ij3}$$

Two image compressions:

Perform SVD for R, G, B → $\mathrm{rank}(\chi)$ approximation for RGB matrices

Amount of data=3×(768 +1024 + 1)×χ

Perform HOSVD for T → $\mathrm{rank}-(\chi', \chi', 3)$ approximation

Amount of data= (768 +1024 + 3χ)×χ

Image compression: color image1

Original

Data: **2,359,296**

~10%
Compression

SVD

$\chi = 50$

268,950

HOSVD

$\chi' = 100$

209,200

Image compression: color image2

Original

Data: **2,359,296**

~10%
Compression

SVD

$\chi = 50$

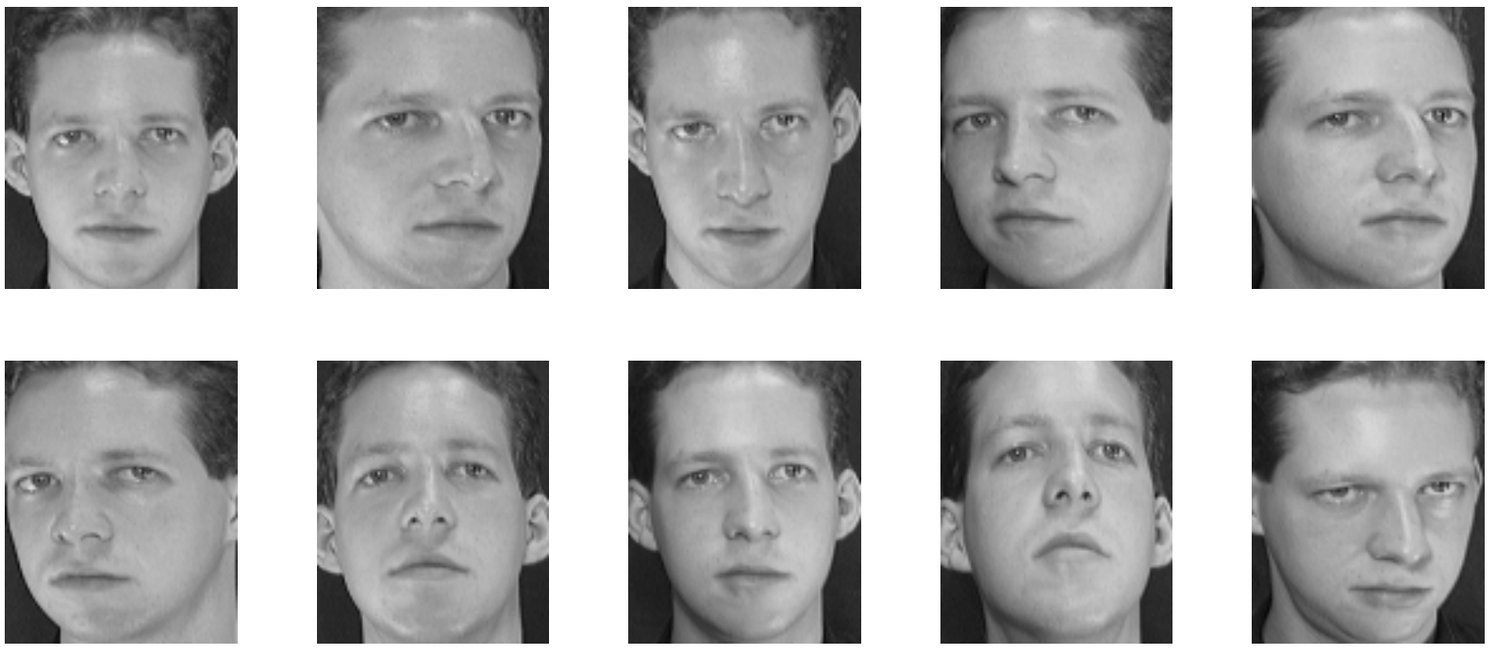
268,950

HOSVD

$\chi' = 100$

209,200

Image compression: multi images



92×122 pixels 10 images

Images were taken from ORL Database of Faces, AT&T Laboratories Cambridge

92 x122 x 10 tensor T

Amount of data=112,240

by HOSVD

$\mathrm{rank}-(\chi, \chi, \chi')$ approximation

Amount of data=
$(92 + 122) \times \chi + 10 \times \chi' + \chi^2 \chi'$

Image compression: multi images

		Data
Original		112,240
$\chi = 30$ $\chi' = 10$		15,520
$\chi = 30$ $\chi' = 9$		14,610
$\chi = 30$ $\chi' = 5$		10,970

Image compression: multi images (color)

From COIL-100 dataset = 128 x 128 x 3 x 20 x 72 tensor

Pixel color object direction

Objects:

Several directions:

Amount of data=70,778,880

by HOSVD

$$\mathrm{rank}\text{–}(\chi_p, \chi_p, \chi_c, \chi_o, \chi_r)$$

approximation

Image compression: multi images (color)

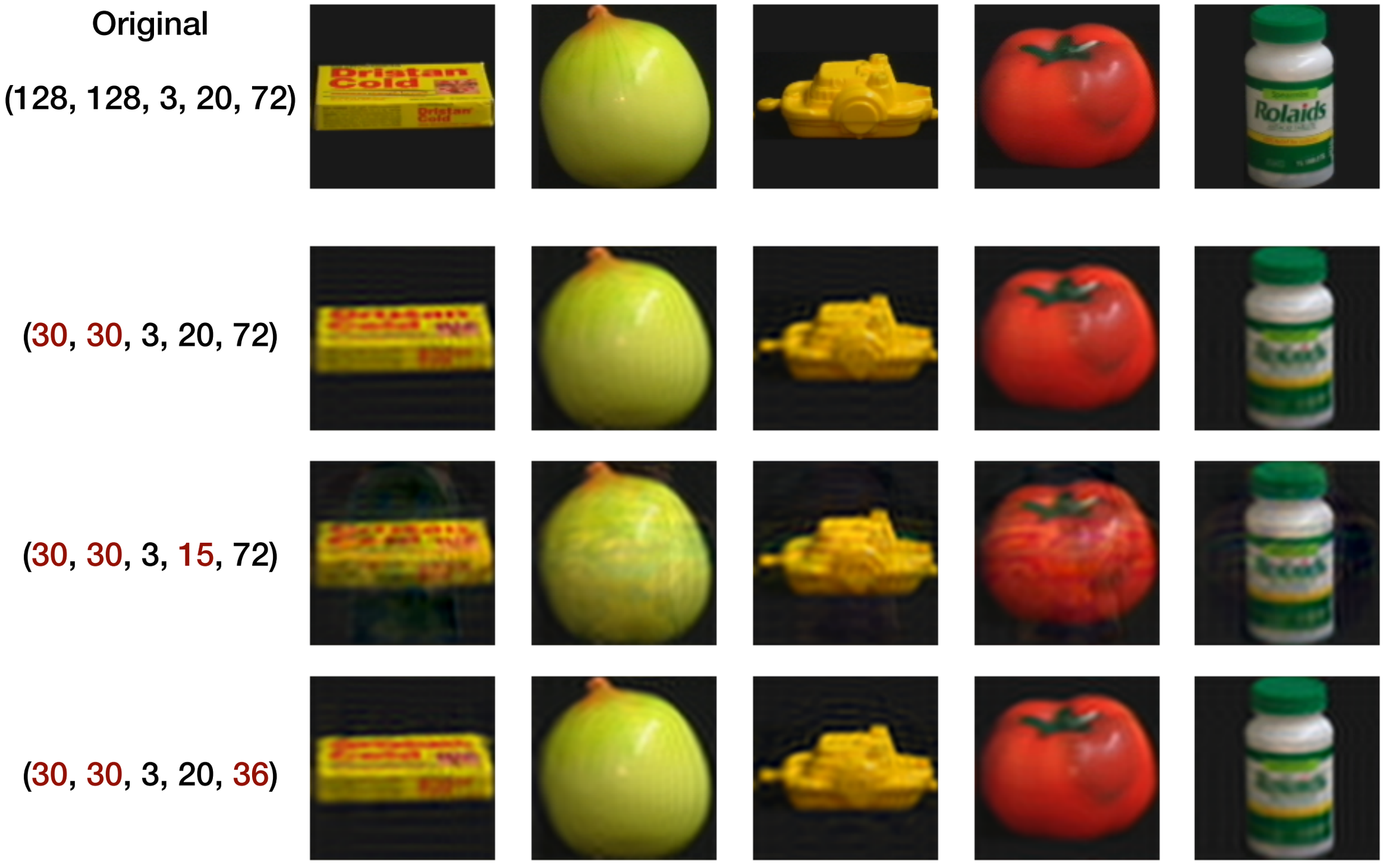
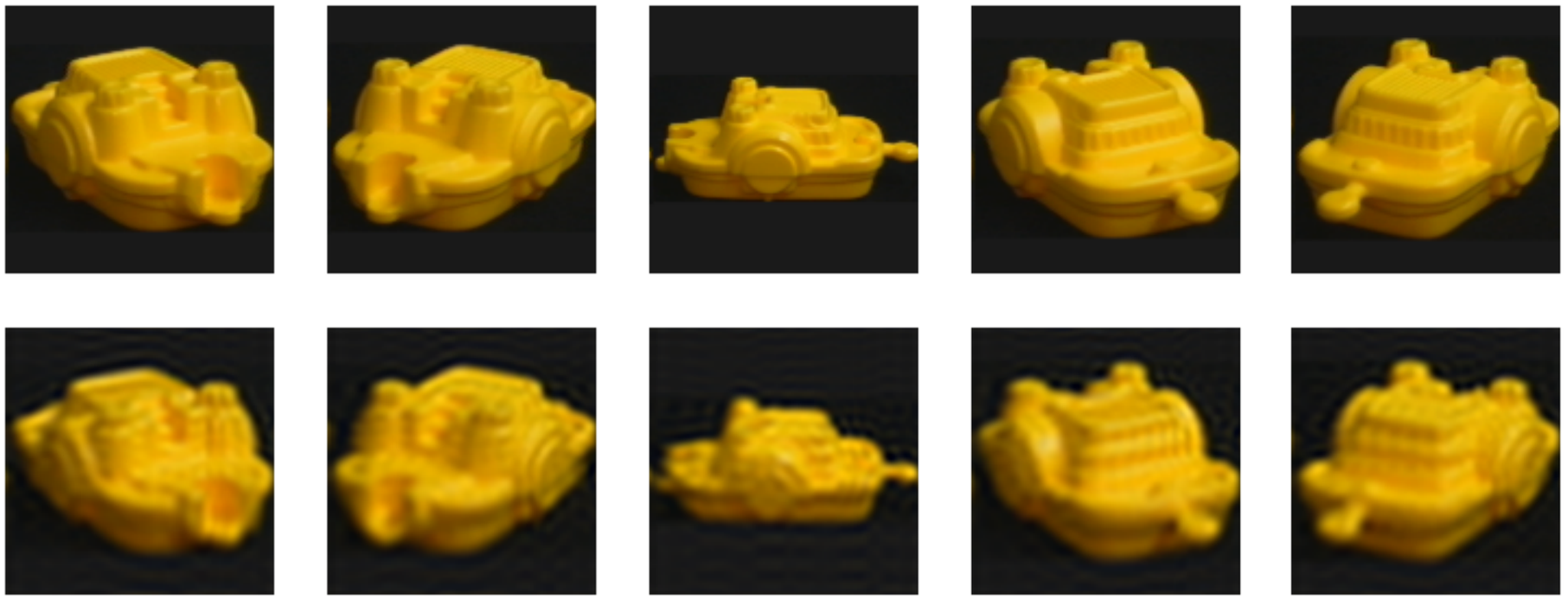


Image compression: multi images (color)

Original

(128, 128, 3, 20, 72)



(30, 30, 3, 20, 72)

(30, 30, 3, 15, 72)

(30, 30, 3, 20, 36)

Sample code: Low rank approximation for pictures

You can try low rank approximations of images through sample codes in **"image_svd.zip".**

In the zip file you find four directories:

- image_gray
- image_color
- multi_image
- coil100_image

They correspond to the examples I showed in this lecture.

Each directory contains jupyter notebook (.ipynb) and python (.py) files.
(In addition, there are sample images.)

You can run them with **PIL, numpy, matplotlib**, modules.

I recommend you to use google colaboratory,
https://colab.research.google.com
where you can run .ipynb from your web browser.

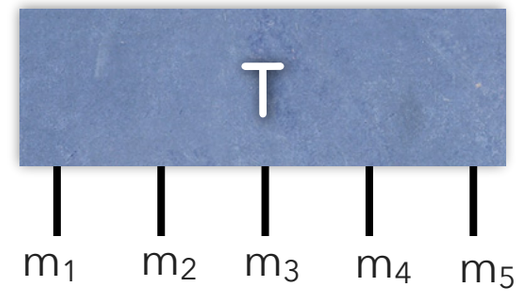
(When you use google colab, don't forget to upload image files.)

References:

- 齋藤正彦、「線形代数入門」東京大学出版会
- 太田快人、「システム制御のための数学（１）ー線形代数編ー」、コロナ社
- T. G. Kolda et al, SIAM Review **51**, 455 (2006).

Another "generalization" of SVD to tensors.

$T_{m_1,m_2,\cdots,m_N}$:N-leg tensor (or Vector)

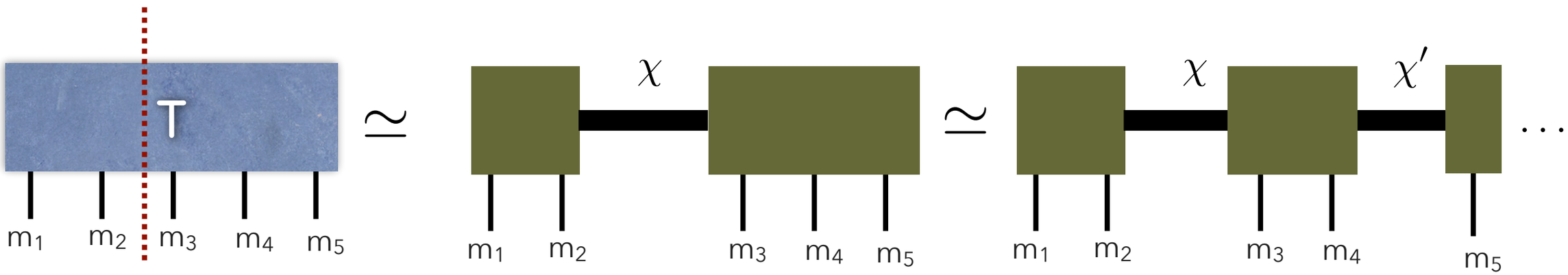


Cf. wave function: $|\Psi\rangle = \sum_{\{m_i=0,1\}} T_{m_1,m_2,\cdots,m_N} |m_1, m_2, \cdots, m_N\rangle$

We can consider it as a matrix by making two groups:

$T_{\{m_1,m_2,\cdots,m_M\},\{m_{M+1},\cdots,m_N\}}$

➡ We can perform the low rank approximation of T.



*obtained two objects are again tensors.

What does it mean? ➡ Lectures #10-#13

Next week (Yamaji sensei)

1st: Huge data in modern physics

2nd: Information compression in modern physics

(+review of linear algebra)

3rd: Review of linear algebra (+ singular value decomposition)

4th: Singular value decomposition and low rank approximation

5th: Basics of sparse modeling

6th: Basics of Krylov subspace methods

7th: Information compression in materials science

8th: Accelerating data analysis: Application of sparse modeling

9th: Data compression: Application of Krylov subspace method

10th: Entanglement of information and matrix product states

11th: Application of MPS to eigenvalue problems

12th: Tensor network representation

13th: Information compression by tensor network renormalization