Information Compression #6
Basics of Krylov subspace methods

Youhei Yamaji

Department of Applied Physics, The University of Tokyo

1. The largest and smallest eigenvalues
2. Sparse matrix generated by Hamiltonian
3. Krylov subspace method

Computational Science Alliance
The University of Tokyo

Classification of Information Compression in Linear Algebra by Memory Costs

(1) A matrix can be stored

-SVD for dense matrix
-Compressed sensing (so far)

(2) Although a matrix cannot be stored, vectors can be stored

-SVD for sparse matrix
-Krylov subspace method

(3) A vector cannot be stored

-Matrix product/tensornetwork states

This Week's Information Compression Algorithm

Main focus:

Algorithms that calculate specified eigenvalues and eigenvectors of huge* *sparse* matrices

You may not store your matrix A or you may not pay $O(L^3)$ cost

$$A \in \mathbb{R}^{L \times L}$$

Especially the largest and smallest eigenstates

Largest and Smallest Eigenvalues

1. Ground state of quantum many-body system

$$\langle O \rangle = \frac{\vec{u}^\dagger O \vec{u}}{\vec{u}^\dagger \vec{u}}$$

The ground state is important:

-Room temperature is often enough low
and well described by zero-temperature wave function

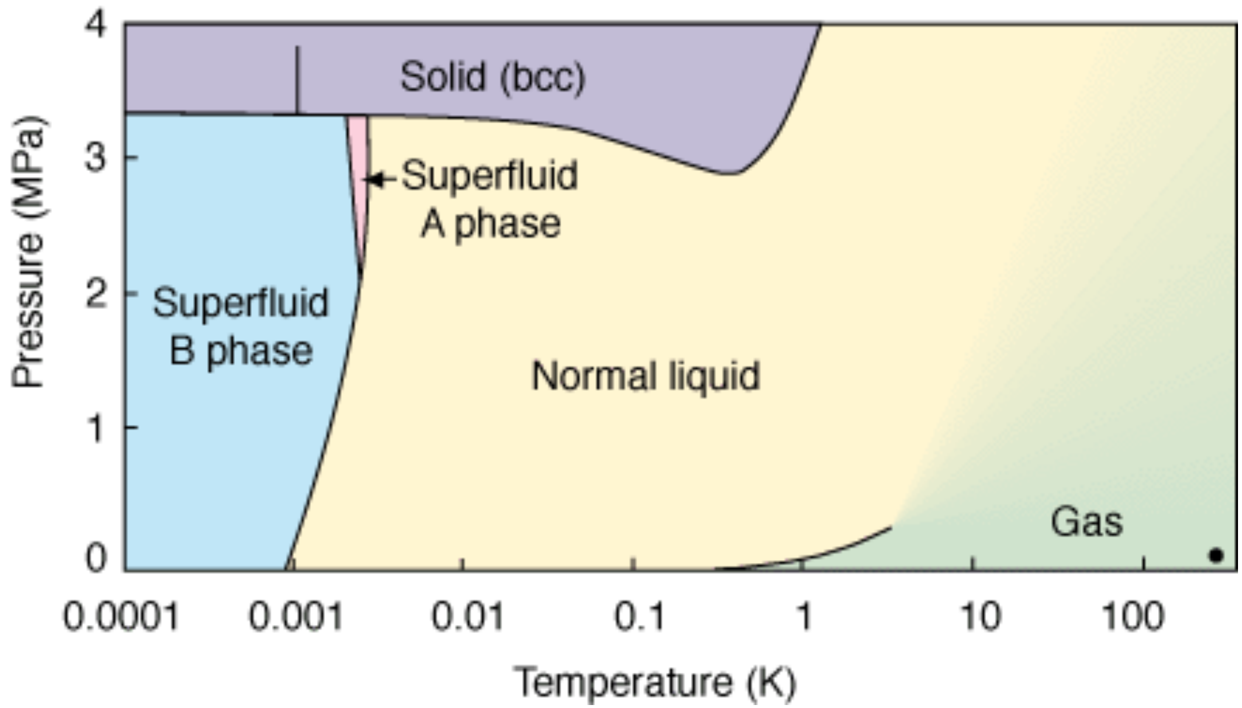
-Interest in ground states (at zero temperature)

Low-temperature phase such as superfluid phase

Zero-temperature phase transitions
(quantum phase transition)

Low-Temperature Phases

Phase diagram of ^3He



D. D. Osheroff, R. C. Richardson, and D. M. Lee,
Phys. Rev. Lett. 28, 885 (1972).

Erkki Thuneberg
http://ltl.tkk.fi/research/theory/helium.html

Largest and Smallest Eigenvalues

2. Principle component analysis for huge data

Eigenvalue problem of covariance matrices

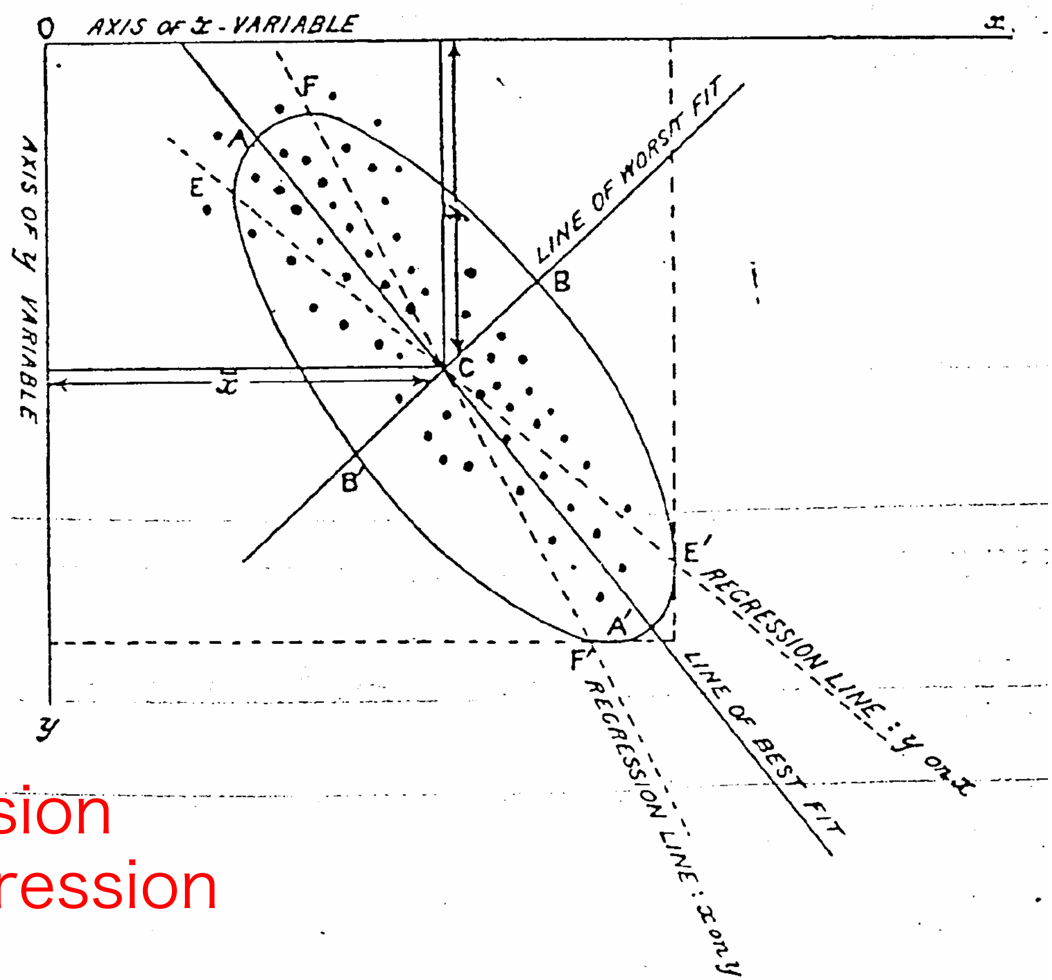
K. Pearson,
Philosophical Magazine 2, 559 (1901)

$$\begin{bmatrix} \sum_\ell (x-\overline{x})^2 & \sum_\ell (x-\overline{x})(y-\overline{y}) \\ \sum_\ell (y-\overline{y})(x-\overline{x}) & \sum_\ell (y-\overline{y})^2 \end{bmatrix}$$

Largest and Smallest Eigenvalues

2. Principle component analysis for huge data

K. Pearson, Philosophical Magazine 2, 559 (1901)



Higher dimension
→ Data compression

Category of Numerical Linear Algebra

You need to choose algorithm depending on whether

your matrix is 1) sparse/dense and

2) stored/not stored in memory

For a matrix that is dense and stored, you can find standard subroutines with $O(L^3)$* cost in LAPACK

*L is the linear dimension of your matrix A

$$A \in \mathbb{R}^{L \times L}$$

Largest and Smallest Eigenvalues

Ground state of quantum many-body system

Typically, sparse and not stored

Principle component analysis for huge data

Eigenvalue problem of covariance matrices

Dense/sparse and stored/not stored

-Partial SVD/low-rank approximation

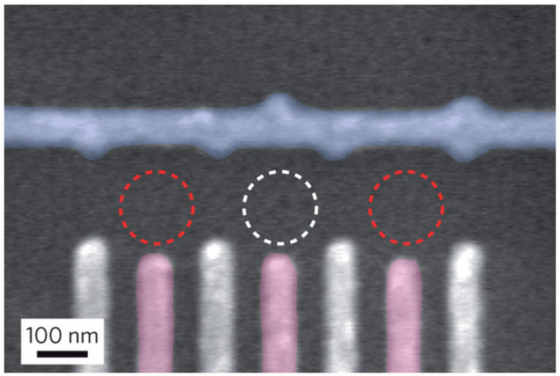
will discussed in 8th lecture

Sparse Matrix Generated by Hamiltonian

Quantum Many-Body Problems

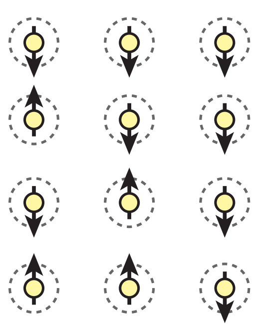
Quantum dots

F. R. Braakman, et al., Nat. Nano. 8, 432 (2013)

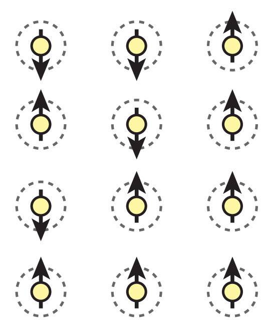


Quantum dot:
-A quantum box can confine a single electron
-Utilized for single electron transistor, quantum computers

Three-body problem:
→ Number of states = 2^3 (factor 2 from spin)



$|0\rangle \otimes |0\rangle \otimes |0\rangle$
$|1\rangle \otimes |0\rangle \otimes |0\rangle$
$|0\rangle \otimes |1\rangle \otimes |0\rangle$
$|1\rangle \otimes |1\rangle \otimes |0\rangle$



$|0\rangle \otimes |0\rangle \otimes |1\rangle$
$|1\rangle \otimes |0\rangle \otimes |1\rangle$
$|0\rangle \otimes |1\rangle \otimes |1\rangle$
$|1\rangle \otimes |1\rangle \otimes |1\rangle$

States represented by superposition

$$\mathcal{F} = \{ \sum_{n_0=0,1} \sum_{n_1=0,1} \sum_{n_2=0,1} C_{n_0 n_1 n_2} |n_0\rangle \otimes |n_1\rangle \otimes |n_2\rangle : C_{n_0 n_1 n_2} \in \mathbb{C} \}$$

Quantum Many-Body Problems

Mutual Interactions

1. Operators acting on a single qubit

A two dimensional representation of Lie algebra SU(2)

$$[\hat{S}_j^x, \hat{S}_j^y] = i\hat{S}_j^z$$
$$[\hat{S}_j^y, \hat{S}_j^z] = i\hat{S}_j^x$$
$$[\hat{S}_j^z, \hat{S}_j^x] = i\hat{S}_j^y$$

-Commutator $[\hat{A}, \hat{B}] \equiv \hat{A}\hat{B} - \hat{B}\hat{A}$

$$\hat{S}_j^x|0\rangle = \frac{1}{2}|1\rangle$$
$$\hat{S}_j^x|1\rangle = \frac{1}{2}|0\rangle$$
$$\hat{S}_j^y|0\rangle = \frac{i}{2}|1\rangle$$
$$\hat{S}_j^y|1\rangle = -\frac{i}{2}|0\rangle$$
$$\hat{S}_j^z|1\rangle = \frac{1}{2}|1\rangle$$
$$\hat{S}_j^z|0\rangle = -\frac{1}{2}|0\rangle$$

Quantum Many-Body Problems

Mutual Interactions

Fock space of N qubits:

$$\mathcal{F} = \{ \sum_{n_0=0,1} \sum_{n_1=0,1} \cdots \sum_{n_{N-1}=0,1} C_{n_0 n_1 \cdots n_{N-1}} |n_0\rangle \otimes |n_1\rangle \otimes \cdots \otimes |n_{N-1}\rangle \}$$

$$(C_{n_0 n_1 \cdots n_{N-1}} \in \mathbb{C})$$

2. Operators acting on N-quibit Fock space:

$$\boxed{\hat{S}_j^a, \hat{S}_j^a \hat{S}_{j+1}^a : \mathcal{F} \to \mathcal{F}}$$

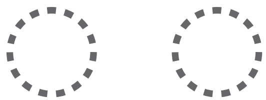
$$\hat{S}_j^a \doteq \overbrace{1 \otimes \cdots \otimes 1}^{j-1} \otimes \hat{S}_j^a \otimes \overbrace{1 \otimes \cdots \otimes 1}^{N-j}$$

$$\hat{S}_j^a \hat{S}_{j+1}^a \doteq \overbrace{1 \otimes \cdots \otimes 1}^{j-1} \otimes \hat{S}_j^a \otimes \hat{S}_{j+1}^a \otimes \overbrace{1 \otimes \cdots \otimes 1}^{N-j-1}$$

Quantum Many-Body Problems

Quantum entanglement

Example: Two qubits

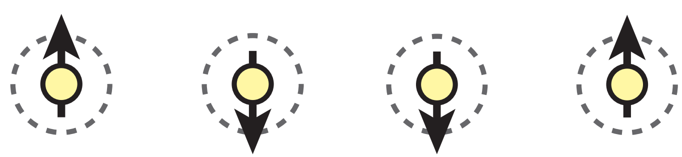


-Superposition
-Utilized for quantum teleportation
 cf.) EPR “paradox”

Mutual interactions between two qubits

$$\hat{H} = J \sum_{a=x,y,z} \hat{S}_0^a \hat{S}_1^a \qquad (J \in \mathbb{R}, J > 0)$$

→Superposition



$$|1\rangle \otimes |0\rangle - |0\rangle \otimes |1\rangle$$

Hamiltonian Matrix

Example: N qubits

N-qubit Fock space:

$$\mathcal{F} = \{ \sum_{n_0=0,1} \sum_{n_1=0,1} \cdots \sum_{n_{N-1}=0,1} C_{n_0 n_1 \cdots n_{N-1}} |n_0\rangle \otimes |n_1\rangle \otimes \cdots \otimes |n_{N-1}\rangle \}$$

$$(C_{n_0 n_1 \cdots n_{N-1}} \in \mathbb{C})$$

Mutual interactions among N qubits:
Hamiltonian operator

$$\hat{H} : \mathcal{F} \to \mathcal{F}$$

$$\hat{H} = J \sum_{j=0}^{N-1} \sum_{a=x,y,z} \hat{S}_j^a \hat{S}_{\mathrm{mod}(j+1,N)}^a$$

Vectors in Fock Space

Correspondence between spin and bit

$$|\uparrow\rangle = |1\rangle$$
$$|\downarrow\rangle = |0\rangle$$

2^N-dimensional Fock space:

$$\mathcal{F} = \{ \sum_{n_0=0,1} \sum_{n_1=0,1} \cdots \sum_{n_{N-1}=0,1} C_{n_0 n_1 \cdots n_{N-1}} |n_0\rangle \otimes |n_1\rangle \otimes \cdots \otimes |n_{N-1}\rangle \}$$
$$(C_{n_0 n_1 \cdots n_{N-1}} \in \mathbb{C})$$

Decimal representation of orthonormalized basis

$$|I\rangle_{\mathrm{d}} = |n_0\rangle \otimes |n_1\rangle \otimes |n_2\rangle \otimes \cdots \otimes |n_{N-1}\rangle \qquad I = \sum_{\nu=0}^{N-1} n_\nu \cdot 2^\nu$$

Wave function as a vector

$$|\phi\rangle = \sum_{n_0=0}^{1} \sum_{n_1=0}^{1} \cdots \sum_{n_{N-1}=0}^{1} C_{n_0 n_1 \cdots n_{N-1}} |n_0\rangle \otimes |n_1\rangle \otimes \cdots \otimes |n_{N-1}\rangle$$

$$v(I) = C_{n_0 n_1 \cdots n_{N-1}} \qquad \boxed{v(0:2^N-1)}$$

Vectors and Matrices in Fock Space

Inner product of vectors

$$\left(\langle n_0| \otimes \langle n_1| \otimes \cdots \otimes \langle n_{N-1}|\right) \times \left(|n_0'\rangle \otimes |n_1'\rangle \otimes \cdots \otimes |n_{N-1}'\rangle\right)$$
$$= \langle n_0|n_0'\rangle \times \langle n_1|n_1'\rangle \times \cdots \times \langle n_{N-1}|n_{N-1}'\rangle$$

$$\langle n| \times |n'\rangle = \langle n|n'\rangle = \delta_{n,n'}$$

$$\langle \phi'|\phi\rangle = \sum_{n_0=0}^{1} \sum_{n_1=0}^{1} \cdots \sum_{n_{N-1}=0}^{1} C'^{*}_{n_0 n_1 \cdots n_{N-1}} C_{n_0 n_1 \cdots n_{N-1}}$$

$$|\phi'\rangle = \sum_{n_0=0}^{1} \sum_{n_1=0}^{1} \cdots \sum_{n_{N-1}=0}^{1} C'_{n_0 n_1 \cdots n_{N-1}} |n_0\rangle \otimes |n_1\rangle \otimes \cdots \otimes |n_{N-1}\rangle$$

$$|\phi\rangle = \sum_{n_0=0}^{1} \sum_{n_1=0}^{1} \cdots \sum_{n_{N-1}=0}^{1} C_{n_0 n_1 \cdots n_{N-1}} |n_0\rangle \otimes |n_1\rangle \otimes \cdots \otimes |n_{N-1}\rangle$$

Hamiltonian matrix

$$H_{II'} = \langle I|\hat{H}|I'\rangle$$

Orthonomalized basis: $|I\rangle, |I'\rangle \in \mathcal{F} \qquad \langle I|I'\rangle = \delta_{I,I'}$

Sparse Matrix

- Particle or orbital number: N
- Fock space dimension: $\exp[N \text{ x const.}]$
- # of terms in Hamiltonian: Polynomial of N

→ # of matrix elements of Hamiltonian matrix:
(Polynomial of N) x $\exp[N \text{ x const.}]$

For sufficiently large N,
(Polynomial of N) x $\exp[N \text{ x const.}]$
$\ll (\exp[N \text{ x const.}])^2$

Then, the Hamiltonian matrix is sparse

An Example of Hamiltonian Matrix

$$\hat{H} = J \sum_{i=0}^{N-1} \hat{S}_i^z \hat{S}_{i+1}^z - \Gamma \sum_{i=0}^{N-1} \hat{S}_i^x$$

-Non-commutative

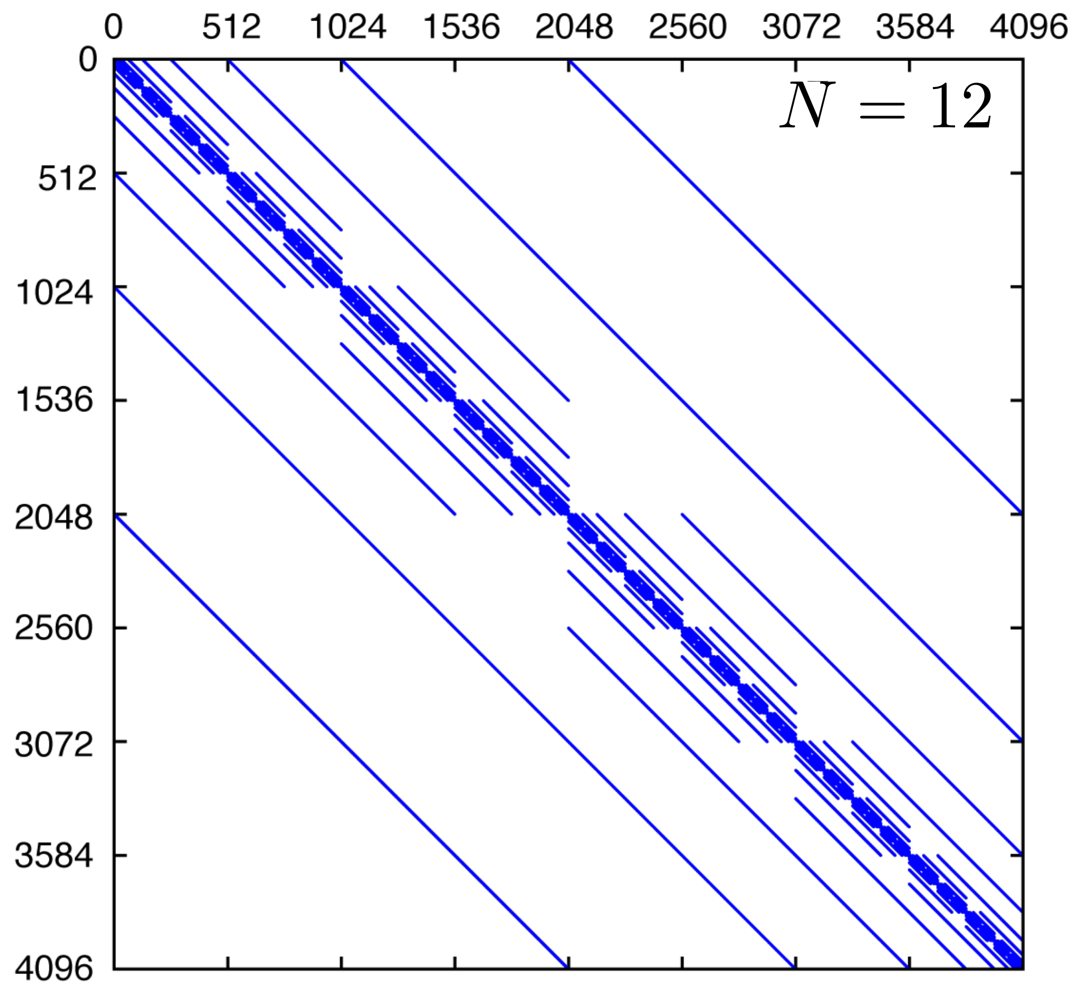
$$\left[\sum_{i=0}^{N-1} \hat{S}_i^z \hat{S}_{i+1}^z , \sum_{i=0}^{N-1} \hat{S}_i^x \right] \neq 0$$

→Quantum fluctuations
or Zero point motion

-Sparse
# of elements ∝ $O(2^N)$

-Solvable

-Hierarchical matrix?



Computational and Memory Costs

Matrix-vector product of dense matrix

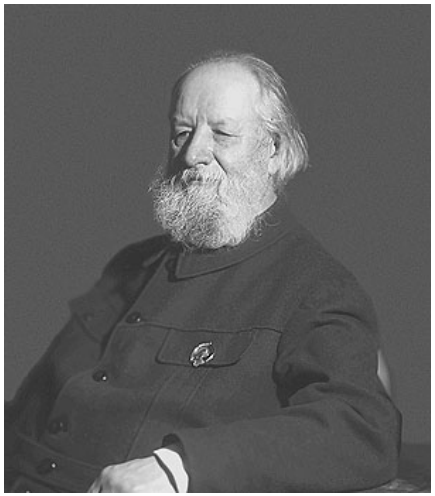
$$v_i = \sum_{j=0}^{N_{\mathrm{H}}-1} A_{ij} u_j$$

Computational: O((Fock space dimension)^2)
Memory: O((Fock space dimension)^2)

Matrix-vector product of large and sparse matrix

Computational: O(Fock space dimension)
Memory: O(Fock space dimension)
Hamiltonian is not stored in memory

Krylov Subspace Method for Sparse and Huge Matrices



Alexey Krylov

Aleksey Nikolaevich Krylov
1863-1945
Russian naval engineer and applied mathematician

Krylov subspace $A \in \mathbb{C}^{L \times L}$

$$\mathcal{K}_n(A, \vec{b}) = \mathrm{span}\{\vec{b}, A\vec{b}, \ldots, A^{n-1}\vec{b}\}$$

Numerical cost to construct K_n: $\mathcal{O}(\mathrm{nnz}(A) \times n)$

Numerical cost to orthogonalize K_n: $\mathcal{O}(L \times n^2)$

Cornelius Lanczos 1950
Walter Edwin Arnoldi 1951

*nnz: Number of non-zero entries/elements

An Algorithm for Eigenvalue Problems of Large & Sparse Matrix: Power Method

Min. Eigenvalue of hermitian

$$\hat{H}|n\rangle = E_n|n\rangle$$

$$\langle n'|n\rangle = \delta_{n',n}$$

$$E_0 \leq E_1 \leq \cdots$$

Initial vector: $|v_1\rangle = \sum_{n=0} c_n |n\rangle$

Prameter: $\max_n \{E_n\} \leq \Lambda$

$$\lim_{m\to+\infty} \frac{(\Lambda - \hat{H})^m |v_1\rangle}{\sqrt{\langle v_1|(\Lambda - \hat{H})^{2m}|v_1\rangle}} = |0\rangle$$

$$(\Lambda - \hat{H})^m |v_1\rangle = \sum_n (\Lambda - E_n)^m c_n |n\rangle$$

$$\lim_{m\to+\infty} \frac{\sum_{n>0} (\Lambda - E_n)^{2m} |c_n|^2}{(\Lambda - E_0)^{2m} |c_0|^2} = 0$$

Advanced Algorithm: Krylov Subspace Method

Krylov subspace method:
Finding approximate eigenstates in a Krylov subspace

$$\mathcal{K}_m(\hat{H}, |v_1\rangle) = \mathrm{span}\{|v_1\rangle, \hat{H}|v_1\rangle, \ldots, \hat{H}^{m-1}|v_1\rangle\}$$

Construction and orthogonalization of Krylov subspaces

Shift invariance:

$$\mathcal{K}_m(\hat{H}, |v_1\rangle) = \mathcal{K}_m(\hat{H} + z\mathbf{1}, |v_1\rangle)$$

Krylov subspace method:
-Lanczos method (symmetric/hermitian),
Arnoldi method (general matrix)
-Conjugate gradient method (CG method)
(many variation)

Lanczos Method

$$\textbf{Initial}: \beta_1 = 0,\ |v_0\rangle = 0$$

$$\textbf{for}\ j = 1, 2, \ldots, m\ \textbf{do}$$

$$|w_j\rangle = \hat{H}|v_j\rangle - \beta_j|v_{j-1}\rangle$$

$$\alpha_j = \langle w_j|v_j\rangle$$

$$|w_j\rangle \leftarrow |w_j\rangle - \alpha_j|v_j\rangle$$

$$\beta_{j+1} = \sqrt{\langle w_j|w_j\rangle}$$

$$|v_{j+1}\rangle = |w_j\rangle/\beta_{j+1}$$

Lanczos Method

$$\alpha_j = \langle v_j | \hat{H} | v_j \rangle$$

$$\beta_j = \langle v_{j-1} | \hat{H} | v_j \rangle = \langle v_j | \hat{H} | v_{j-1} \rangle$$

Orthogonalization

$$|v_j\rangle = \frac{\hat{H}|v_{j-1}\rangle - \sum_{\ell=1}^{j-1} |v_\ell\rangle\langle v_\ell|\hat{H}|v_{j-1}\rangle}{\langle v_j|\hat{H}|v_{j-1}\rangle}$$

$$\langle v_\ell|\hat{H}|v_{j-1}\rangle = \begin{cases} 0 & (\ell \leq j-3) \\ \beta_{j-1} & (\ell = j-2) \\ \alpha_{j-1} & (\ell = j-1) \end{cases}$$

Lanczos Method

$$\textbf{Initial}: \beta_1 = 0, \ |v_0\rangle = 0$$

$$\textbf{for } j = 1, 2, \ldots, m \textbf{ do}$$

$$|w_j\rangle = \hat{H}|v_j\rangle - \beta_j|v_{j-1}\rangle$$

$$\alpha_j = \langle w_j|v_j\rangle$$

$$|w_j\rangle \leftarrow |w_j\rangle - \alpha_j|v_j\rangle$$

$$\beta_{j+1} = \sqrt{\langle w_j|w_j\rangle}$$

$$|v_{j+1}\rangle = |w_j\rangle/\beta_{j+1}$$

Lanczos Method

$$\alpha_j = \langle v_j|\hat{H}|v_j\rangle$$
$$\langle v_j|v_k\rangle = \delta_{j,k}$$
$$\beta_j = \langle v_{j-1}|\hat{H}|v_j\rangle = \langle v_j|\hat{H}|v_{j-1}\rangle$$

Hamiltonian projected onto *m* D Krylov subsace

$$H_m = \begin{pmatrix} \alpha_1 & \beta_2 & & & & 0 \\ \beta_2 & \alpha_2 & \beta_3 & & & \\ & \beta_3 & \alpha_3 & \ddots & & \\ & & \ddots & \ddots & \beta_{m-1} & \\ & & & \beta_{m-1} & \alpha_{m-1} & \beta_m \\ 0 & & & & \beta_m & \alpha_m \end{pmatrix}$$

Eigenvalues of projected Hamiltonian
→ Approximate eigenvalues of original Hamiltonian

Lanczos Method: # of Vectors Required

$$\textbf{Initial}: \beta_1 = 0,\ |v_0\rangle = 0$$

$$\textbf{for } j = 1, 2, \ldots, m \textbf{ do}$$

$$|w_j\rangle \leftarrow \hat{H}|v_j\rangle - \beta_j|v_{j-1}\rangle \qquad |v_{j-1}\rangle \to |w_j\rangle,\ |v_j\rangle$$

$$\alpha_j = \langle w_j|v_j\rangle \qquad |w_j\rangle,\ |v_j\rangle$$

$$|w_j\rangle \leftarrow |w_j\rangle - \alpha_j|v_j\rangle \qquad |w_j\rangle,\ |v_j\rangle$$

$$\beta_{j+1} = \sqrt{\langle w_j|w_j\rangle} \qquad |w_j\rangle,\ |v_j\rangle$$

$$|v_{j+1}\rangle = |w_j\rangle/\beta_{j+1} \qquad |w_j\rangle \to |v_{j+1}\rangle,\ |v_j\rangle$$

Convergence of Lanczos Method

Yousef Saad,
Numerical Methods for Large Eigenvalue Problems (2nd ed)
The Society for Industrial and Applied Mathematics 2011

Eigenvalue: λ_n

Eigenvector: $|n\rangle$

Assumption: $\lambda_1 > \lambda_2 > \cdots > \lambda_n$

Convergence theorem for the largest eigenvalue

$$0 \leq \lambda_1 - \lambda_1^{(m)} \leq (\lambda_1 - \lambda_n) \left[\frac{\tan\theta(|v_1\rangle, |1\rangle)}{C_{m-1}(1+2\gamma_1)} \right]^2$$

$$\sim 4(\lambda_1 - \lambda_n) \left[\tan\theta(|v_1\rangle, |1\rangle)\right]^2 e^{-4\sqrt{\gamma_1} m}$$

$$\gamma_1 = \frac{\lambda_1 - \lambda_2}{\lambda_2 - \lambda_n}$$

$$C_k(t) = \frac{1}{2}\left[\left(t + \sqrt{t^2 - 1}\right)^k + \left(t + \sqrt{t^2 - 1}\right)^{-k} \right]$$

Distribution of Eigenvalues of Hermitian Matrices

An important relationship between distribution or density of states and statistical mechanics

$$P(E) = \frac{\rho(E)e^{-\beta E}}{\int dE'\rho(E')e^{-\beta E'}} \sim \frac{\exp[-(E-\langle E\rangle)^2/2CT^2]}{\sqrt{2\pi CT^2}}$$

$$k_{\mathrm{B}} = 1$$

$$C = \frac{\langle \hat{H}^2\rangle - \langle \hat{H}\rangle^2}{T^2}$$

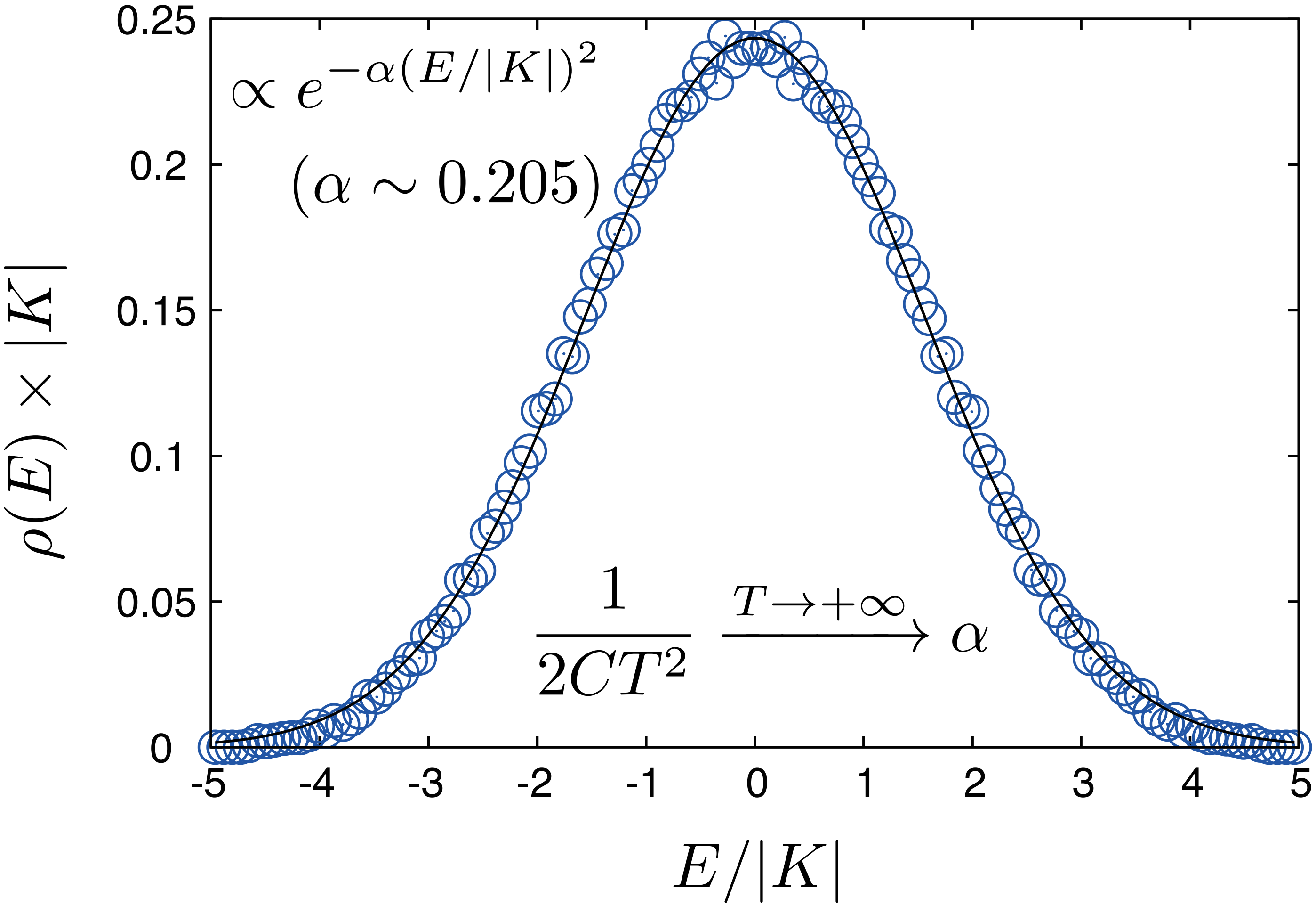
$$\langle \hat{H}^m\rangle \sim \int E^m P(E) dE$$

An Example of Density of State

24 site cluster of Kitaev model
(frustrated S =1/2 spins)

A. Kitaev, Annals Phys. 321, 2 (2006).

2^{24} = 16,777,216



Example of Dense Matrix: Random Symmetric Matrices

Eugene P. Wigner, Annals of Mathematics, 2nd Series, 67, 325 (1958)

Wigner's random matrix $(A)_{ij} = a_{ij}$

$$a_{ij} = a_{ji}$$

(Not necessarily sparse)

$$\int p_{ij}(a)da = 1$$

$$p_{ij}(+a) = p_{ij}(-a)$$

$$\langle a_{ij}^n \rangle = \int p_{ij}(a)a^n da \leq B_n$$

$$\langle a_{ij}^2 \rangle = \int p_{ij}(a)a^2 da = 1$$

Example of Dense Matrix: Random Symmetric Matrices

Eugene P. Wigner, Annals of Mathematics, 2nd Series, 67, 325 (1958)

Density of states of L x L symmetric random matirx

$$A\vec{v} = E\vec{v}$$

$$\sigma(E) = \begin{cases} \dfrac{\sqrt{4L - E^2}}{2\pi L} & (E^2 < 4L) \\ 0 & (E^2 > 4L) \end{cases}$$

Comment:
Sparse matrices in quantum many-body problems show smaller density of states than random matrices around the both ends of the distribution
→ Sparse around maximum/minimum eigenvalues
→ Lanczos method may work well

Approximate SVD by Krylov Subspace Method

Low-rank approximation by *block* Krylov subspace

C. Musco & C. Musco,
NIPS'15 Proceedings of 28th International Conference on Neural Information Processing Systems 1, 1396 (2015)

$$\|A - ZZ^T A\|_2 \leq (1+\epsilon)\|A - A_k\|_2$$

Operator norm defined by 2-norm (Spectral norm)

$$A \in \mathbb{R}^{L\times M} \quad Z \in \mathbb{R}^{L\times k} \quad \text{rank } k \leq L, M$$

$$q = \mathcal{O}(\ln d/\sqrt{\epsilon})$$

$$\text{random matrix } \Pi \in \mathbb{R}^{M\times k}$$

$$\mathcal{K}_{q+1} = \text{span}\{A\Pi, (AA^T)A\Pi, \ldots, (AA^T)^q A\Pi\}$$

$$Q \in \mathbb{R}^{N\times qk}$$

Orthogonalized basis set of the block Krylov subspace

$$M = Q^T AA^T Q \in \mathbb{R}^{qk\times qk}$$

$$U_k: \text{ the top } k\text{singular vectors of } M$$

$$Z = QU_k$$

$$(\Pi)_{ij}: \text{ Random number generated by } e^{-x^2/2}/\sqrt{\pi}$$

Important References

Yousef Saad,
Numerical Methods for Large Eigenvalue Problems (2nd ed)
The Society for Industrial and Applied Mathematics 2011

Report problem 1-1

Perform SVD for a matrix (or matrices)

(i) Prepare two $M \times N$ matrices A *and* B. ($M, N > 100$ is better)

It is encouraged to prepare matrices related to your research field.

If it is difficult, prepare two pictures. (It should be different from the examples in the lecture.)

In the following, we compare the low rank approximations of A and B.

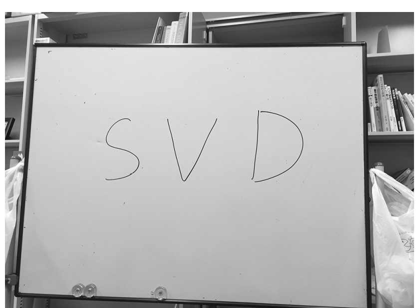
So, it is better that A and B have different properties (or they are very different pictures).

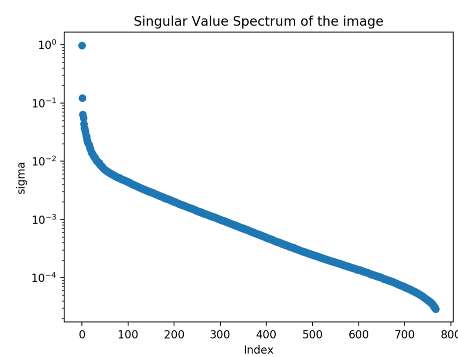
(In the report, please include the explanation (meaning) of the matrices.)

(ii) Perform SVD and plot the singular values for A and B.

You can use any libraries. (LAPACK, numpy or scipy in python, matlab, ...)

Please normalize the singular values as $\tilde{\sigma}_i = \sigma_i / \sqrt{\sum_j \sigma_j^2}$





Report problem 1-1 (cont.)

(iii) Perform low rank approximations of the matrices with ranks r_1,r_2, ...

- Calculate the distances between the original and approximate matrices. Please use Frobenius norm $\|A - \tilde{A}\|_F$ as the distance. It is better to show a normalized distance: $\|A - \tilde{A}\|_F / \|A\|_F$
- Try at least two ranks (r_1 and r_2) both for *A* and *B.*

(iv) Discuss characteristics of the low rank approximations (for your matrices) based on the singular value spectra. (This part is the most important!)

- Please include "explanation" of the relation between the distance and singular values. (You can find the relation in the lecture slides. Note that here we consider normalized distances.)
- Please discuss difference of characteristics between the low rank approximations of *A* and *B.*
- *(optional)* From the relation discussed above, we can determine a necessary rank for a give accuracy (normalized distance). Determine the ranks which give normalized distances 10^{-2} and 10^{-3} for your matrices. (You may find a hint in the sample code for "--plot_error" option.)

Report problem 1-1 (cont.)

Sample python code for Image SVD: Report_1-1.zip
(Run with python3. You need PIL, numpy, matplotlib)
It works at least on ECCS.

Usage of sample python code for Image SVD:

python image_svd.py -c chi -f filename

[Example of output]

```
Input file: sample.jpg
Array shape: (768, 1024)
Low rank approximation with chi=10
Normalized distance:0.10087303978176487
```

(In addition, the singular value spectrum and the approximated image appear.)

You can see help message: *python image_svd.py -h*

```
usage: image_svd.py [-h] [-c chi] [-f filename]

Low rank approximation of an image

optional arguments:
  -h, --help            show this help message and exit
  -c chi, --chi chi     rank of the approximated matrix. (default: chi = 10)
  -f filename, --file filename
                        filename of the image. (default: sample.jpg)
  --plot_error          plot expected normalized distance. (default: false)
```

Report problem 1-2

Minimize the cost function with L_1-regularization

$$f(\vec{x}) = \frac{1}{2\sigma^2}\|\vec{y} - A\vec{x}\|_2^2 + \lambda\|\vec{x}\|_1$$

(i) (Elementary exercise)
Obtain x that minimizes the following cost function f for given y, a, σ^2, and λ

$$f(x) = \frac{1}{2\sigma^2}(y - ax)^2 + \lambda|x|$$

(ii) Obtain x_1, x_2 that minimizes the following cost function f for given y_1, a_1, a_2, σ^2, and λ

$$f(x_1, x_2) = \frac{1}{2\sigma^2}(y_1 - a_1x_1 - a_2x_2)^2 + \lambda(|x_1| + |x_2|)$$

*(i), (ii) Depending on a, a_1, a_2, σ^2, and λ, you may have an unique solution or you may not.
**Solutions of (i) and (ii) may not satisfy y=Ax.

Next Week

1st: Huge data in modern physics
2nd: Information compression in modern physics
3rd: Review of linear algebra
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