CSE 392: Matrix and Tensor Algorithms for Data

Instructor: Shashanka Ubaru

University of Texas, Austin
Spring 2024

Lecture 13: Krylov subspace methods

Outline

Iterative methods

- **Subspace iteration/ power method:** multiple passes over the matrix $\boldsymbol{A}$.
- With q iterations, we can achieve:

$$\|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{z}_q\boldsymbol{z}_q^\top\|_F \leq (1+\epsilon)\|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{v}_1\boldsymbol{v}_1^\top\|_F.$$

- if $q = O\left(\frac{\log d/\epsilon}{\gamma}\right)$ (if gap is large) or
- $q = O\left(\frac{\log d/\epsilon}{\epsilon}\right)$ (if gap is too small or for gap independent analysis).

Krylov subspace methods

- Given a square matrix $\boldsymbol{A}$ and a starting vector $\boldsymbol{z}_1$, the *Krylov Subspace* of dimension q is given by:
$$\boldsymbol{K}_q(\boldsymbol{A}, \boldsymbol{z}_1) = \text{span}\{\boldsymbol{z}_1, \boldsymbol{A}\boldsymbol{z}_1, \ldots, \boldsymbol{A}^q\boldsymbol{z}_1\}$$
- Important class of projection methods for solving linear systems and for eigenvalue problems.
- *Properties of* $\boldsymbol{K}_q$:
$\boldsymbol{K}_q = \{\mathbf{p}(\boldsymbol{A})\boldsymbol{z} | \mathbf{p} = \text{polynomial of degree} \leq q\}$.
$\boldsymbol{K}_q = \boldsymbol{K}_{q_1}$ for all $q \geq q_1$. Moreover, $\boldsymbol{K}_{q_1}$ is invariant under $\boldsymbol{A}$.
- For square matrix $\boldsymbol{A}$: Arnoldi's Algorithm
- For symmetric matrix $\boldsymbol{A}$: Lanczos Algorithm
- For rectangular matrix $\boldsymbol{B} \in \mathbb{R}^{n \times d}$ and SVD, we consider $\boldsymbol{A} = \boldsymbol{B}^\top \boldsymbol{B}$.

Lanczos algorithm

- Given a symmetric matrix $\boldsymbol{A} \in \mathbb{R}^{d \times d}$ and a starting vector $\boldsymbol{z}_1$, compute an orthonormal basis $\boldsymbol{Z}_q$ of $\boldsymbol{K}_q(\boldsymbol{A}, \boldsymbol{z}_1)$.

Lanczos algorithm

- Choose a starting vector $\boldsymbol{z}_1$, with unit norm. Set $\beta_1 = 0, \boldsymbol{z}_0 = 0$.
- For $l = 1, \ldots, q-1$
  - $\boldsymbol{y}_l = \boldsymbol{A}\boldsymbol{z}_l - \beta_l \boldsymbol{z}_{l-1}$
  - $\alpha_l = \langle \boldsymbol{y}_l, \boldsymbol{z}_l \rangle$
  - $\boldsymbol{y}_l = \boldsymbol{y}_l - \alpha_l \boldsymbol{z}_l$
  - $\beta_{l+1} = \|\boldsymbol{y}_l\|_2$. If $\beta_{l+1} = 0$ then stop
  - $\boldsymbol{z}_{l+1} = \boldsymbol{y}_l / \beta_{l+1}$
- Return $\boldsymbol{Z}_q = [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_q]$

In theory $\boldsymbol{z}_l$'s defined by 3-term recurrence are orthogonal. But in practice, we need reorthogonalization.

Lanczos algorithm

- The Rayleigh Ritz-projection is given by:

$$\boldsymbol{T}_q = \boldsymbol{Z}_q^\top \boldsymbol{A} \boldsymbol{Z}_q.$$

- The Ritz matrix is a tridiagonal matrix:

$$\boldsymbol{T}_q = \begin{bmatrix} \alpha_1 & \beta_2 & & & & \\ \beta_2 & \alpha_2 & \beta_3 & & & \\ & \beta_3 & \alpha_3 & \beta_4 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & \ddots \\ & & & & \beta_q & \alpha_q \end{bmatrix}.$$

- Let $\boldsymbol{u}$ be the top eigenvector of $\boldsymbol{T}_q$.
- Eigenvector estimate of $\boldsymbol{A}$ will be $\boldsymbol{w} = \boldsymbol{Z}_q \boldsymbol{u}$.
- If non-symmetric, *Arnoldi*'s algorithm. $\boldsymbol{T}_q$ will be Upper Hessenberg matrix.

Convergence

Theorem (Lanczos algorithm Convergence)

Let $\gamma = \frac{\lambda_1 - \lambda_2}{\lambda_1}$ be the gap between the first and second largest eigenvalues of a matrix $\boldsymbol{A} \in \mathbb{R}^{d \times d}$. If Lanczos algorithm is initialized with a random Gaussian vector then, with high probability, after $q = O\left(\frac{\log d/\epsilon}{\sqrt{\gamma}}\right)$ steps, we have for the estimate $\boldsymbol{w} = \boldsymbol{Z}_q \boldsymbol{u}$:

$$\|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{w}\boldsymbol{w}^\top\|_F^2 \leq (1+\epsilon)\|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{v}_1\boldsymbol{v}_1^\top\|_F^2.$$

- **Gapless:** For $q = O\left(\frac{\log d/\epsilon}{\sqrt{\epsilon}}\right)$ steps, we obtain a $\boldsymbol{w}$ satisfying:

$$\|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{w}\boldsymbol{w}^\top\|_F^2 \leq (1+\epsilon)\|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{v}_1\boldsymbol{v}_1^\top\|_F^2.$$

- **Total runtime:** $O(\mathrm{nnz}(\boldsymbol{A})q) = O\left(\mathrm{nnz}(\boldsymbol{A}) \cdot \frac{\log d/\epsilon}{\sqrt{\epsilon}}\right)$.

Proof:
First, we have

Claim: Amongst all vectors in the span of the Krylov subspace (which are given by $\boldsymbol{w} = \boldsymbol{Z}_q \boldsymbol{x}$), $\boldsymbol{w} = \boldsymbol{Z}_q \boldsymbol{u}$ minimizes the error $\|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{w}\boldsymbol{w}^\top\|_F^2$.

Proof:
First, we have

Claim: Amongst all vectors in the span of the Krylov subspace (which are given by $\boldsymbol{w} = \boldsymbol{Z}_q\boldsymbol{x}$), $\boldsymbol{w} = \boldsymbol{Z}_q\boldsymbol{u}$ minimizes the error $\|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{w}\boldsymbol{w}^\top\|_F^2$.

We know that, this is equivalent to maximizing $\|\boldsymbol{A}\boldsymbol{w}\boldsymbol{w}^\top\|_F^2$.
Next, $\boldsymbol{u}$ is the top eigenvector of $\boldsymbol{T}_q = \boldsymbol{Z}_q^\top\boldsymbol{A}\boldsymbol{Z}_q$.

Proof:

First, we have

Claim: Amongst all vectors in the span of the Krylov subspace (which are given by $\boldsymbol{w} = \boldsymbol{Z}_q \boldsymbol{x}$), $\boldsymbol{w} = \boldsymbol{Z}_q \boldsymbol{u}$ minimizes the error $\|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{w}\boldsymbol{w}^\top\|_F^2$.

We know that, this is equivalent to maximizing $\|\boldsymbol{A}\boldsymbol{w}\boldsymbol{w}^\top\|_F^2$.
Next, $\boldsymbol{u}$ is the top eigenvector of $\boldsymbol{T}_q = \boldsymbol{Z}_q^\top \boldsymbol{A} \boldsymbol{Z}_q$.

Next, we show that, if we set $q = O\left(\frac{\log d/\epsilon}{\sqrt{\gamma}}\right)$ and compute $\boldsymbol{Z}_q$, then there a vector $\boldsymbol{w} = \boldsymbol{Z}_q \boldsymbol{x}$ such that $\langle \boldsymbol{v}_1, \boldsymbol{w} \rangle \geq 1 - \epsilon$.
I.e., there is a $\boldsymbol{w}$ in the Krylov subspace that has a large inner product with the top eigenvector $\boldsymbol{v}_1$.

The vector $\boldsymbol{w}$ can be written as

$$\boldsymbol{w} = p_q(\boldsymbol{A})\boldsymbol{z}_1,$$

where $p_q(\cdot)$ is called the Lanczos polynomial and has degree q.
For any q degree polynomial p_q, there is some $\boldsymbol{x}$ such that $\boldsymbol{Z}_q\boldsymbol{x} = p_q(\boldsymbol{A})\boldsymbol{z}_1$, because any linear combinations of $\boldsymbol{z}_1, \boldsymbol{A}\boldsymbol{z}_1, \ldots, \boldsymbol{A}^q\boldsymbol{z}_1$ lie in the span of $\boldsymbol{Z}_q$.

The vector $\boldsymbol{w}$ can be written as

$$\boldsymbol{w} = p_q(\boldsymbol{A})\boldsymbol{z}_1,$$

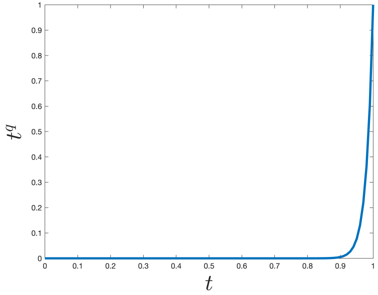
where $p_q(\cdot)$ is called the Lanczos polynomial and has degree q.
For any q degree polynomial p_q, there is some $\boldsymbol{x}$ such that $\boldsymbol{Z}_q\boldsymbol{x} = p_q(\boldsymbol{A})\boldsymbol{z}_1$, because any linear combinations of $\boldsymbol{z}_1, \boldsymbol{A}\boldsymbol{z}_1, \ldots, \boldsymbol{A}^q\boldsymbol{z}_1$ lie in the span of $\boldsymbol{Z}_q$.

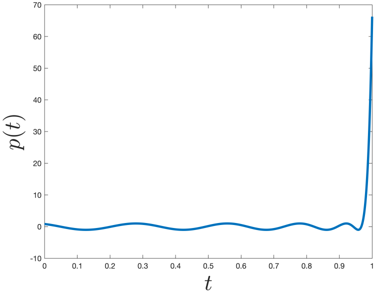
Let us write $\boldsymbol{z}_1 = \sum_{i=1}^d \mu_i \boldsymbol{v}_i$ and $p_q(\boldsymbol{A})\boldsymbol{z}_1 = \sum_{i=1}^d \rho_i \boldsymbol{v}_i$, then we have

$$\rho_i = \mu_i p_q(\lambda_i)$$

Claim: There is a $O\left(\sqrt{\frac{1}{\gamma}}\log(1/\epsilon')\right)$ degree polynomial $\hat{p}$ such that $\hat{p}(1) = 1$ and $|\hat{p}(t)| \leq \epsilon'$ for $0 \leq t \leq 1 - \gamma$.

Polynomials





Plots are from `https://www.chrismusco.com/amlds2023/notes/lecture11.html`.

We set $p_q(t) = \hat{p}(t/\lambda_1)$, and we have $\rho_i = \mu_i p_q(\lambda_i)$.
We follow similar steps as the power method proof.

$$\frac{|\rho_j|}{|\rho_1|} = \frac{p_q(\lambda_i)|\mu_i|}{p_q(\lambda_1)|\mu_1|} = \frac{\hat{p}_q(\lambda_i/\lambda_1)|\mu_i|}{|\mu_1|} \leq \sqrt{\epsilon/d}.$$

For $O\left(\sqrt{\frac{1}{\gamma}}\log(1/\epsilon')\right)$ with $\epsilon' = \sqrt{\epsilon/d}/d^3$.

Block Krylov method

- For larger $k \geq 1$ (finding the top-k singular vectors/values).

Block Lanczos Method

- Choose $\boldsymbol{S} \in \mathbb{R}^{d \times k}$ a random Gaussian matrix .
- Set $\boldsymbol{K} = [\boldsymbol{S}, \boldsymbol{AS}, \ldots, \boldsymbol{A}^{q-1}\boldsymbol{S}]$.
- $\boldsymbol{Z} = \text{orth}(\boldsymbol{K})$
- Compute $\boldsymbol{T} = \boldsymbol{Z}^\top \boldsymbol{AZ}$
- Set $\tilde{\boldsymbol{U}}_k$ to top k eigenvectors of $\boldsymbol{T}$
- Return $\boldsymbol{Z}_q \tilde{\boldsymbol{U}}_k$

Total runtime: $O(\text{nnz}(\boldsymbol{A})kq)$. With $q = O\left(\frac{\log d/\epsilon}{\sqrt{\epsilon}}\right)$.

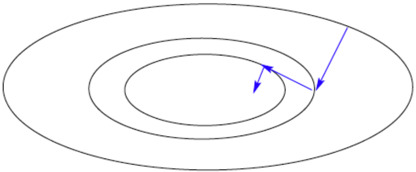
Krylov methods

Further Reading:

- *Randomized Block Krylov Methods for Stronger and Faster Approximate Singular Value Decomposition* by Cameron Musco, Christopher Musco.
- *Structural Convergence Results for Approximation of Dominant Subspaces from Block Krylov Spaces* by Petros Drineas, Ilse Ipsen, Eugenia-Maria Kontopoulou, Malik Magdon-Ismail.
- `https://www.chrismusco.com/amlds2022/lectures/lanczos_method.html`

Linear system solvers

- Given a square matrix $\boldsymbol{A} \in \mathbb{R}^{d\times d}$ and a vector $\boldsymbol{b} \in \mathbb{R}^d$, solve:
$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}.$$
- *Iterative methods:* Solve for $\boldsymbol{x}$ iteratively as:
$$\boldsymbol{x}_{l+1} = \boldsymbol{x}_l + \alpha \boldsymbol{r}$$
$\boldsymbol{r}$ = a certain direction given some starting vector $\boldsymbol{x}_0$.
- *Minimum residual methods:* $\boldsymbol{x}(\alpha) = \boldsymbol{x} + \alpha\boldsymbol{r}$, with $\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}$. $\min_\alpha \|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}(\alpha)\|_2$ with some orthogonal condition.
- Steepest Descent:



$$\begin{aligned}\boldsymbol{r}_l &= \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_l \\ \alpha &= \langle \boldsymbol{r}_l, \boldsymbol{r}_l\rangle / \langle \boldsymbol{A}\boldsymbol{r}_l, \boldsymbol{r}_l\rangle \\ \boldsymbol{x}_{l+1} &= \boldsymbol{x}_l + \alpha \boldsymbol{r}_l\end{aligned}$$

Krylov subspace methods

- *Lanczos Algorithm:* For symmetric matrix $\boldsymbol{A}$, orthonormal basis $\boldsymbol{Z}_q$ and tridiagonal matrix $\boldsymbol{T}_q$. (Arnoldi's method for non-symmetric)
- From Petrov-Galerkin condition, we get:

$$\boldsymbol{x}_q = \boldsymbol{x}_0 + \boldsymbol{Z}_q \boldsymbol{T}_q^{-1} \boldsymbol{Z}_q^\top \boldsymbol{r}_0$$

- Select $\boldsymbol{z}_1 = \boldsymbol{r}_0 / \|\boldsymbol{r}_0\|$, then

$$\boldsymbol{x}_q = \boldsymbol{x}_0 + \boldsymbol{Z}_q \boldsymbol{T}_q^{-1} \boldsymbol{e}_1$$

- Several algorithms **mathematically equivalent/similar** to this approach: Full Orthogonalization method (FOM), Incomplete OM (IOM), GMRES, Orthmin, Axelsson's CGLS, Conjugate Gradient (CG), and others.

Lanczos Method

Lanczos Method for Linear Systems

- Compute $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0, \beta_1 = \|\boldsymbol{r}_0\|$ and , $\boldsymbol{z}_1 = \boldsymbol{r}_0/\beta_1$.
- For $l = 1, \ldots, q$
  - $\boldsymbol{y}_l = \boldsymbol{A}\boldsymbol{z}_l - \beta_l \boldsymbol{z}_{l-1}$
  - $\alpha_l = \langle \boldsymbol{y}_l, \boldsymbol{z}_l \rangle$
  - $\boldsymbol{y}_l = \boldsymbol{y}_l - \alpha_l \boldsymbol{z}_l$
  - $\beta_{l+1} = \|\boldsymbol{y}_l\|_2$. If $\beta_{l+1} = 0$ then stop
  - $\boldsymbol{z}_{l+1} = \boldsymbol{y}_l/\beta_{l+1}$
- Set $\boldsymbol{Z}_q = [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_q]$ and $\boldsymbol{T}_q = \text{tridiag}(\beta_j, \alpha_j, \beta_{j+1})$.
- Compute $\boldsymbol{w}_q = \beta \boldsymbol{T}_q^{-1} \boldsymbol{e}_1$ and $\boldsymbol{x}_q = \boldsymbol{x}_0 + \boldsymbol{Z}_q \boldsymbol{w}_q$.

Conjugate Gradient Method

Popular variant of the Krylov subspace methods when the input matrix is S.P.D.

Conjugate Gradient Algorithm

- Compute $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0, \boldsymbol{p}_0 = \boldsymbol{r}_0$.
- Iterate: Until Convergence
  - $\alpha_l = \langle \boldsymbol{r}_l, \boldsymbol{r}_l \rangle / \langle \boldsymbol{A}\boldsymbol{p}_l, \boldsymbol{p}_l \rangle$
  - $\boldsymbol{x}_{l+1} = \boldsymbol{x}_l + \alpha_l \boldsymbol{p}_l$
  - $\boldsymbol{r}_{l+1} = \boldsymbol{r}_l - \alpha_l \boldsymbol{A}\boldsymbol{p}_l$
  - $\beta_l = \langle \boldsymbol{r}_{l+1}, \boldsymbol{r}_{l+1} \rangle / \langle \boldsymbol{r}_l, \boldsymbol{r}_l \rangle$
  - $\boldsymbol{p}_{l+1} = \boldsymbol{r}_{l+1} + \beta_l \boldsymbol{p}_l$

The $\boldsymbol{p}_l$'s are $\boldsymbol{A}$-conjugate with $\langle \boldsymbol{A}\boldsymbol{p}_l, \boldsymbol{p}_j \rangle = 0$ for $l \neq j$.
Convergence: with condition number $\kappa = \lambda_{\max}/\lambda_{\min}$.

$$\|\boldsymbol{x}^* - \boldsymbol{x}_q\|_{\boldsymbol{A}} \leq 2 \left[\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right]^q \|\boldsymbol{x}^* - \boldsymbol{x}_0\|_{\boldsymbol{A}}$$

Iterative methods

Further Reading:

- *Iterative methods for sparse linear systems* by Yousef Saad.
- *Numerical Methods for Large Eigenvalue Problems* by Yousef Saad.
- *Iterative Methods for Optimization* by C.T. Kelly.

Matlab Demo