ELE 520: Mathematics of Data Science

Large-scale eigenvalue problems

Yuxin Chen

Princeton University, Fall 2020

Outline

- Power method
- Lanczos algorithm

Eigendecomposition

Consider a *symmetric* matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, where n is large

How to compute the eigenvalues and eigenvectors of $\boldsymbol{A}$ efficiently?

- hopefully accomplished via a few matrix-vector products

Power method

Power iteration

$$\boldsymbol{q}_t = \underbrace{\frac{1}{\|\boldsymbol{A}\boldsymbol{q}_{t-1}\|_2}}_{\text{re-normalization}} \boldsymbol{A}\boldsymbol{q}_{t-1}, \qquad t = 1, 2, \cdots$$

- each iteration consists of a matrix-vector product
- equivalently,

$$\boldsymbol{q}_t = \frac{1}{\|\boldsymbol{A}^t\boldsymbol{q}_0\|_2}\boldsymbol{A}^t\boldsymbol{q}_0$$

Example

Consider $\boldsymbol{A} = \begin{bmatrix} 2 & \\ & 1 \end{bmatrix}$ and $\boldsymbol{q}_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, then

$$\boldsymbol{A}^t \boldsymbol{q}_0 = \begin{bmatrix} 2^t \\ 1 \end{bmatrix}$$

$$\Longrightarrow \qquad \boldsymbol{q}_t = \frac{1}{\|\boldsymbol{A}^t \boldsymbol{q}_0\|_2} \boldsymbol{A}^t \boldsymbol{q}_0 = \begin{bmatrix} \frac{2^t}{\sqrt{2^{2t}+1}} \\ \frac{1}{\sqrt{2^{2t}+1}} \end{bmatrix} \to \underbrace{\begin{bmatrix} 1 \\ 0 \end{bmatrix}}_{\text{leading eigenvector of } \boldsymbol{A}} \qquad \text{as } t \to \infty$$

Power method

Algorithm 4.1 Power method

1: **initialize** $\boldsymbol{q}_0 \leftarrow$ random unit vector
2: **for** $t = 1, 2, \cdots$ **do**
3: $\quad \boldsymbol{q}_t = \frac{1}{\|\boldsymbol{A}\boldsymbol{q}_{t-1}\|_2}\boldsymbol{A}\boldsymbol{q}_{t-1}$ (power iteration)
4: $\quad \hat{\lambda}_1^{(t)} = \boldsymbol{q}_t^\top \boldsymbol{A}\boldsymbol{q}_t$

- $\boldsymbol{q}_t$: estimate of the leading eigenvector of $\boldsymbol{A}$
- $\hat{\lambda}_1^{(t)}$: estimate of the leading eigenvalue of $\boldsymbol{A}$

Convergence of power method

- $\boldsymbol{A} \in \mathbb{R}^{n\times n}$: eigenvalues $\lambda_1 > \lambda_2 \geq \cdots \geq \lambda_n$; eigenvectors $\boldsymbol{u}_1, \cdots, \boldsymbol{u}_n$

Theorem 4.1 (Convergence of power method)

If $\lambda_1 > \lambda_2 \geq |\lambda_n|$ *and set* $\nu_1 = \boldsymbol{q}_0^\top \boldsymbol{u}_1$, *then*

$$\left|\hat{\lambda}_1^{(t)} - \lambda_1\right| \leq (\lambda_1 - \lambda_n) \frac{1-\nu_1^2}{\nu_1^2} \left(\frac{\lambda_2}{\lambda_1}\right)^{2t}$$

Proof of Theorem 4.1

Write $\boldsymbol{q}_0 = \sum_{i=1}^n \nu_i \boldsymbol{u}_i$, then

$$\boldsymbol{A}^t \boldsymbol{q}_0 = \sum_{i=1}^n \lambda_i^t \boldsymbol{u}_i \boldsymbol{u}_i^\top \boldsymbol{q}_0 = \sum_{i=1}^n \lambda_i^t \nu_i \boldsymbol{u}_i$$

$$\Longrightarrow \qquad \|\boldsymbol{A}^t \boldsymbol{q}_0\|_2 = \Big\| \sum_{i=1}^n \lambda_i^t \nu_i \boldsymbol{u}_i \Big\|_2 = \sqrt{\sum_{i=1}^n \lambda_i^{2t} \nu_i^2}$$

Since $\boldsymbol{q}_t = \frac{1}{\|\boldsymbol{A}^t \boldsymbol{q}_0\|_2} \boldsymbol{A}^t \boldsymbol{q}_0$ and $\boldsymbol{A}$ is symmetric, we get

$$\begin{aligned}
\hat{\lambda}_1^{(t)} = \boldsymbol{q}_t^\top \boldsymbol{A} \boldsymbol{q}_t &= \frac{1}{\|\boldsymbol{A}^t \boldsymbol{q}_0\|_2^2} \boldsymbol{q}_0^\top \boldsymbol{A}^{2t+1} \boldsymbol{q}_0 \\
&= \frac{1}{\sum_{i=1}^n \lambda_i^{2t} \nu_i^2} \boldsymbol{q}_0^\top \left(\sum_{i=1}^n \lambda_i^{2t+1} \boldsymbol{u}_i \boldsymbol{u}_i^\top \right) \boldsymbol{q}_0 \\
&= \frac{1}{\sum_{i=1}^n \lambda_i^{2t} \nu_i^2} \sum_{i=1}^n \lambda_i^{2t+1} \nu_i^2
\end{aligned}$$

Proof of Theorem 4.1 (cont.)

As a consequence,

$$
\begin{aligned}
\left|\hat{\lambda}_1^{(t)} - \lambda_1\right| &= \frac{1}{\sum_{i=1}^n \lambda_i^{2t}\nu_i^2} \left|\sum_{i=1}^n \lambda_i^{2t+1}\nu_i^2 - \sum_{i=1}^n \lambda_1\lambda_i^{2t}\nu_i^2\right| \\
&= \frac{1}{\sum_{i=1}^n \lambda_i^{2t}\nu_i^2} \left|\sum_{i=2}^n \lambda_i^{2t}(\lambda_1 - \lambda_i)\nu_i^2\right| \\
&\le \frac{\lambda_1 - \lambda_n}{\lambda_1^{2t}\nu_1^2} \sum_{i=2}^n \lambda_i^{2t}\nu_i^2 \qquad\qquad \text{(since } \lambda_1 - \lambda_i \le \lambda_1 - \lambda_n\text{)} \\
&\le \frac{\lambda_1 - \lambda_n}{\lambda_1^{2t}\nu_1^2} \lambda_2^{2t} \sum_{i=2}^n \nu_i^2 \\
&= \frac{\lambda_1 - \lambda_n}{\lambda_1^{2t}\nu_1^2} \lambda_2^{2t}(1 - \nu_1^2) \qquad \text{(since } \sum_i \nu_i^2 = 1 \text{ (as } \|\boldsymbol{q}_0\|_2 = 1\text{))}
\end{aligned}
$$

as claimed

Block power method

Computing the top-r eigen-subspace:

Algorithm 4.2 Power method

1: **initialize** $\boldsymbol{Q}_0 \in \mathbb{R}^{n \times r} \leftarrow$ random orthonormal matrix
2: **for** $t = 1, 2, \cdots$ **do**
3: $\quad \boldsymbol{Z}_t = \boldsymbol{A}\boldsymbol{Q}_{t-1}$
4: $\quad$ compute QR decomposition $\boldsymbol{Z}_t = \boldsymbol{Q}_t \boldsymbol{R}_t$, where $\boldsymbol{Q}_t \in \mathbb{R}^{n \times r}$ has orthonormal columns and $\boldsymbol{R}_t \in \mathbb{R}^{r \times r}$ is upper-triangular

- use QR decomposition to reorthogonalize power iterates

Lanczos algorithm

Key idea 1: reduction to a tridiagonal form

Intermediate step

$$\underbrace{\begin{bmatrix} \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet \end{bmatrix}}_{\boldsymbol{A}} \xrightarrow{\text{find orthonormal } \boldsymbol{Q}} \underbrace{\begin{bmatrix} \bullet & \bullet & & & \\ \bullet & \bullet & \bullet & & \\ & \bullet & \bullet & \bullet & \\ & & \bullet & \bullet & \bullet \\ & & & \bullet & \bullet \end{bmatrix}}_{\boldsymbol{T}=\boldsymbol{Q}^\top \boldsymbol{A}\boldsymbol{Q} \text{ (tridiagonal)}}$$

- **motivation**: eigendecomposition of a tridiagonal matrix can be performed efficiently (via a number of specialized algorithms), due to its special structure

Key idea 2: tridiagonalization and Krylov subspaces

One way to tridiagonalize $\boldsymbol{A}$ is to compute an orthonormal basis of certain subspaces, defined as follows

- **Krylov subspaces** generated by $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{b} \in \mathbb{R}^{n}$ are defined as

$$\mathcal{K}_t := \mathsf{span}\{\boldsymbol{b}, \boldsymbol{A}\boldsymbol{b}, \cdots, \boldsymbol{A}^{t-1}\boldsymbol{b}\}, \qquad t = 1, \cdots, n$$

- **Krylov matrices**

$$\boldsymbol{K}_t := \left[\boldsymbol{b}, \boldsymbol{A}\boldsymbol{b}, \cdots, \boldsymbol{A}^{t-1}\boldsymbol{b}\right] \in \mathbb{R}^{n\times t}, \qquad t = 1, \cdots, n$$

Key idea 2: tridiagonalization and Krylov subspaces

Lemma 4.2

If $\boldsymbol{Q}_t := [\boldsymbol{q}_1, \cdots, \boldsymbol{q}_t] \in \mathbb{R}^{n \times t}$ *forms an orthonormal basis of* $\mathcal{K}_t$ *for all* $1 \leq t \leq n$. *Then*

$$\boldsymbol{T}_t := \boldsymbol{Q}_t^\top \boldsymbol{A} \boldsymbol{Q}_t \text{ is tridiagonal}, \qquad 1 \leq t \leq n$$

- tridiagonalization can be carried out by successively computing the orthonormal basis of Krylov subspaces $\{\mathcal{K}_t\}_{t=1,2,\cdots}$

Proof of Lemma 4.2

For any $i > j + 1$,

$$(\boldsymbol{T}_t)_{i,j} = \langle \boldsymbol{q}_i, \boldsymbol{A}\boldsymbol{q}_j \rangle$$

Since $\boldsymbol{Q}_j$ is orthonormal basis of span$\{\boldsymbol{b}, \boldsymbol{A}\boldsymbol{b}, \cdots, \boldsymbol{A}^{j-1}\boldsymbol{b}\}$, we have

$$\boldsymbol{q}_j \in \text{span}\{\boldsymbol{b}, \boldsymbol{A}\boldsymbol{b}, \cdots, \boldsymbol{A}^{j-1}\boldsymbol{b}\}$$

$$\Longrightarrow \qquad \boldsymbol{A}\boldsymbol{q}_j \in \text{span}\{\boldsymbol{A}\boldsymbol{b}, \cdots, \boldsymbol{A}^{j}\boldsymbol{b}\} \subset \text{span}\{\boldsymbol{q}_1, \cdots, \boldsymbol{q}_{j+1}\}$$

Since $i > j + 1$, one has $\boldsymbol{q}_i \perp \{\boldsymbol{q}_1, \cdots, \boldsymbol{q}_{j+1}\}$ and hence

$$(\boldsymbol{T}_t)_{i,j} = \langle \boldsymbol{q}_i, \boldsymbol{A}\boldsymbol{q}_j \rangle = 0$$

Similarly, $(\boldsymbol{T}_t)_{i,j} = 0$ if $j > i + 1$. This completes the proof

A simple formula: 3-term recurrence

Denote

$$\boldsymbol{T} = \boldsymbol{Q}_n^\top \boldsymbol{A} \boldsymbol{Q}_n = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \ddots & & \\ & \ddots & \ddots & \beta_{n-1} \\ & & \beta_{n-1} & \alpha_n \end{bmatrix} \quad \text{or} \quad \underbrace{\boldsymbol{A}\boldsymbol{Q}_n = \boldsymbol{Q}_n \boldsymbol{T}}_{\text{since } \boldsymbol{Q}_n \in \mathbb{R}^{n\times n} \text{ is orthonormal}}$$

Exploiting the tridiagonal structure yields

$$\boldsymbol{A} \underbrace{[\boldsymbol{q}_1, \cdots, \boldsymbol{q}_t]}_{\boldsymbol{Q}_t} = \underbrace{[\boldsymbol{q}_1, \cdots, \boldsymbol{q}_{t+1}]}_{\boldsymbol{Q}_{t+1}} \begin{bmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_{t-1} \\ & & \beta_{t-1} & \alpha_t \\ & & & \beta_t \end{bmatrix}$$

$$\Longrightarrow \qquad \boldsymbol{A}\boldsymbol{q}_t = \beta_{t-1}\boldsymbol{q}_{t-1} + \alpha_t \boldsymbol{q}_t + \beta_t \boldsymbol{q}_{t+1}$$

Lanczos iterations

$$\boldsymbol{A}\boldsymbol{q}_t = \beta_{t-1}\boldsymbol{q}_{t-1} + \alpha_t \boldsymbol{q}_t + \beta_t \boldsymbol{q}_{t+1}$$

This 3-term recurrence says $\boldsymbol{A}\boldsymbol{q}_t \in \text{span}\{\boldsymbol{q}_{t-1}, \boldsymbol{q}_t, \boldsymbol{q}_{t+1}\}$

- this means $\alpha_t = \underbrace{\boldsymbol{q}_t^\top \boldsymbol{A}\boldsymbol{q}_t}_{\text{projection of } \boldsymbol{A}\boldsymbol{q}_t \text{ onto span}(\boldsymbol{q}_t)}$, since $\{\boldsymbol{q}_{t-1}, \boldsymbol{q}_t, \boldsymbol{q}_{t+1}\}$ are orthonormal

Since $\boldsymbol{q}_{t+1}$ needs to have unit norm, one has

- $\boldsymbol{q}_{t+1} \leftarrow \text{normalize}(\boldsymbol{A}\boldsymbol{q}_t - \beta_{t-1}\boldsymbol{q}_{t-1} - \alpha_t \boldsymbol{q}_t)$ (direction of residual)
- $\beta_t = \|\boldsymbol{A}\boldsymbol{q}_t - \beta_{t-1}\boldsymbol{q}_{t-1} - \alpha_t \boldsymbol{q}_t\|_2$ (size of residual)

Lanczos algorithm

Algorithm 4.3 Lanczos algorithm

1: **initialize** $\beta_0 = 0$, $\boldsymbol{q}_0 = \boldsymbol{0}$, $\boldsymbol{q}_1 \leftarrow$ random unit vector
2: **for** $t = 1, 2, \cdots$ **do**
3: $\quad \alpha_t = \boldsymbol{q}_t^\top \boldsymbol{A} \boldsymbol{q}_t$
4: $\quad \beta_t = \|\boldsymbol{A}\boldsymbol{q}_t - \beta_{t-1}\boldsymbol{q}_{t-1} - \alpha_t \boldsymbol{q}_t\|_2$
5: $\quad \boldsymbol{q}_{t+1} = \frac{1}{\beta_t}(\boldsymbol{A}\boldsymbol{q}_t - \beta_{t-1}\boldsymbol{q}_{t-1} - \alpha_t \boldsymbol{q}_t)$

- each iteration only requires a matrix-vector product
- systematic construction of the orthonormal bases for successive Krylov subspaces

Convergence of the Lanczos algorithm

- $\boldsymbol{A} \in \mathbb{R}^{n\times n}$: eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n$, eigenvectors $\boldsymbol{u}_1, \cdots, \boldsymbol{u}_n$
- $\boldsymbol{T}_t = \begin{bmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_{t-1} \\ & & \beta_{t-1} & \alpha_t \end{bmatrix}$: eigenvalues $\theta_1 \geq \cdots \geq \theta_t$

Theorem 4.3 (Kaniel-Paige convergence theory)

Let $\nu_1 = \boldsymbol{q}_1^\top \boldsymbol{u}_1$, $\rho = \frac{\lambda_1-\lambda_2}{\lambda_2-\lambda_n}$, *and* $\mathcal{C}_{t-1}(x)$ *be the Chebyshev polynomial of degree* $t-1$. *Then*

$$\lambda_1 \geq \theta_1 \geq \lambda_1 - (\lambda_1 - \lambda_n)\frac{1-\nu_1^2}{\nu_1^2}\frac{1}{\big(\mathcal{C}_{t-1}(1+2\rho)\big)^2}$$

Convergence of the Lanczos algorithm

Corollary 4.4

Let $R = 1 + 2\rho + 2\sqrt{\rho^2 + \rho}$ *with* $\rho = \frac{\lambda_1 - \lambda_2}{\lambda_2 - \lambda_n}$. *We have*

$$|\lambda_1 - \theta_1| \leq \underbrace{\frac{4(1-\nu_1^2)}{\nu_1^2}(\lambda_1 - \lambda_n)}_{\textit{prefactor}} \quad \underbrace{R^{-2(t-1)}}_{\textit{convergence rate}}$$

- this follows immediately from the following fact

$$\underbrace{\mathcal{C}_{t-1}^2(1+2\rho) = \frac{\left(R^{t-1} + R^{-(t-1)}\right)^2}{4}}_{\text{properties of Chebyshev polynomials}} \geq \frac{R^{2(t-1)}}{4}$$

Power method vs. Lanczos algorithm

Consider a case where $\lambda_2 = -\lambda_n$. Recall that $\rho = \frac{\lambda_1 - \lambda_2}{\lambda_2 - \lambda_n} = \frac{\lambda_1 - \lambda_2}{2\lambda_2}$

- **power method:** convergence rate

$$\left(\frac{\lambda_2}{\lambda_1}\right)^{2t} = \frac{1}{(1 + 2\rho)^{2t}}$$

- **Lanczos algorithm:** convergence rate

$$\frac{1}{\left(1 + 2\rho + 2\sqrt{\rho^2 + \rho}\right)^{2t}}$$

  - if $\rho \gg 1$, then $1 + 2\rho + 2\sqrt{\rho^2 + \rho} \approx 1 + 4\rho \approx 2(1 + 2\rho)$
  - if $\rho \ll 1$, then $1 + 2\rho + 2\sqrt{\rho^2 + \rho} \approx 1 + 2\sqrt{\rho} > 1 + 2\rho$
  - outperforms the power method

Proof of Theorem 4.3

It sufficies to prove the 2nd inequality. Recalling that $\boldsymbol{T}_t = \boldsymbol{Q}_t^\top \boldsymbol{A} \boldsymbol{Q}_t$, we have

$$\theta_1 = \max_{\boldsymbol{v}:\boldsymbol{v}\neq\boldsymbol{0}} \frac{\boldsymbol{v}^\top \boldsymbol{T}_t \boldsymbol{v}}{\boldsymbol{v}^\top \boldsymbol{v}} = \max_{\boldsymbol{v}:\boldsymbol{v}\neq\boldsymbol{0}} \frac{(\boldsymbol{Q}_t\boldsymbol{v})^\top \boldsymbol{A} (\boldsymbol{Q}_t\boldsymbol{v})}{(\boldsymbol{Q}_t\boldsymbol{v})^\top (\boldsymbol{Q}_t\boldsymbol{v})} = \max_{\boldsymbol{w}\in\mathcal{K}_t:\boldsymbol{w}\neq\boldsymbol{0}} \frac{\boldsymbol{w}^\top \boldsymbol{A} \boldsymbol{w}}{\boldsymbol{w}^\top \boldsymbol{w}}$$

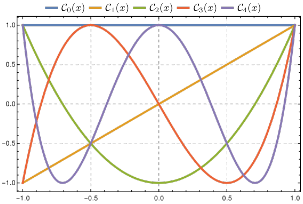
For any $\boldsymbol{w} \in \mathcal{K}_t := \{\boldsymbol{q}_1, \boldsymbol{A}\boldsymbol{q}_1, \cdots, \boldsymbol{A}^{t-1}\boldsymbol{q}_1\}$, one can write it as $\mathcal{P}(\boldsymbol{A})\boldsymbol{q}_1$ for some polynomial $\mathcal{P}(\cdot)$ of degree $t-1$. This means

$$\theta_1 = \max_{\mathcal{P}(\cdot)\in\mathcal{P}_{t-1}} \frac{(\mathcal{P}(\boldsymbol{A})\boldsymbol{q}_1)^\top \boldsymbol{A} (\mathcal{P}(\boldsymbol{A})\boldsymbol{q}_1)}{(\mathcal{P}(\boldsymbol{A})\boldsymbol{q}_1)^\top (\mathcal{P}(\boldsymbol{A})\boldsymbol{q}_1)}$$

where $\mathcal{P}_{t-1}$ is set of polynomials of degree $t-1$. If $\boldsymbol{q}_1 = \sum_{i=1}^n \nu_i \boldsymbol{u}_i$, then

$$\begin{aligned}
\frac{(\mathcal{P}(\boldsymbol{A})\boldsymbol{q}_1)^\top \boldsymbol{A} (\mathcal{P}(\boldsymbol{A})\boldsymbol{q}_1)}{(\mathcal{P}(\boldsymbol{A})\boldsymbol{q}_1)^\top (\mathcal{P}(\boldsymbol{A})\boldsymbol{q}_1)} &= \frac{\sum_{i=1}^n \nu_i^2 \mathcal{P}^2(\lambda_i)\lambda_i}{\sum_{i=1}^n \nu_i^2 \mathcal{P}^2(\lambda_i)} \qquad \text{(check)} \\
&= \lambda_1 - \frac{\sum_{i=2}^n \nu_i^2 (\lambda_1 - \lambda_i)\mathcal{P}^2(\lambda_i)}{\nu_1^2 \mathcal{P}^2(\lambda_1) + \sum_{i=2}^n \nu_i^2 \mathcal{P}^2(\lambda_i)} \\
&\geq \lambda_1 - (\lambda_1 - \lambda_n) \frac{\sum_{i=2}^n \nu_i^2 \mathcal{P}^2(\lambda_i)}{\nu_1^2 \mathcal{P}^2(\lambda_1) + \sum_{i=2}^n \nu_i^2 \mathcal{P}^2(\lambda_i)}
\end{aligned}$$

Proof of Theorem 4.3 (cont.)



Pick a polynomial $\mathcal{P}(x)$ that is large at $x = \lambda_1$. One choice is

$$\mathcal{P}(x) = \mathcal{C}_{t-1}\left(\frac{2x - \lambda_2 - \lambda_n}{\lambda_2 - \lambda_n}\right)$$

where $\mathcal{C}_{t-1}(\cdot)$ is the $(t-1)$-th Chebyshev polynomial generated by

$$\mathcal{C}_t(x) = 2x\mathcal{C}_{t-1}(x) - \mathcal{C}_{t-2}(x), \quad \mathcal{C}_0(x) = 1, \quad \mathcal{C}_1(x) = x$$

These polynomials are bounded by 1 on $[-1, 1]$, but grow rapidly outside

Proof of Theorem 4.3 (cont.)

Using boundedness of Chebyshev polynomial in $[-1,1]$, we have

$$
\begin{aligned}
(\lambda_1-\lambda_n)\frac{\sum_{i=2}^{n}\nu_i^2\mathcal{P}^2(\lambda_i)}{\nu_1^2\mathcal{P}^2(\lambda_1)+\sum_{i=2}^{n}\nu_i^2\mathcal{P}^2(\lambda_i)} &\leq (\lambda_1-\lambda_n)\frac{\sum_{i=2}^{n}\nu_i^2}{\nu_1^2\mathcal{P}^2(\lambda_1)} \\
&= (\lambda_1-\lambda_n)\frac{1-\nu_1^2}{\nu_1^2\mathcal{P}^2(\lambda_1)}
\end{aligned}
$$

where the last identity follows since $\sum_i \nu_i^2 = 1$ (given $\|\boldsymbol{q}_1\|_2 = 1$). This yields

$$
\theta_1 \geq \lambda_1 - (\lambda_1-\lambda_n)\frac{1-\nu_1^2}{\nu_1^2}\frac{1}{\mathcal{C}_{t-1}^2(1+2\rho)}
$$

as claimed

Warning: numerical instability

The vanilla Lanczos algorithm (which is efficient with exact arithmetic) is very sensitive to round-off issues

- orthogonality of $\{\boldsymbol{q}_1, \cdots, \boldsymbol{q}_t\}$ might be lost quickly
- eigenvalues might be duplicated

Many variations have been proposed to prevent loss of orthogonality, and to remove spurious eigenvalues

Reference

- "*Numerical linear algebra*," L. Trefethen, D. Bau, *SIAM*, 1997.
- "*An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*," C. Lanczos, 1950.
- "EE381V lecture notes," S. Sanghavi, C. Caramanis, UT Austin.
- "*Matrix computations*," G. Golub, C. Van Loan, *JHU Press*, 2012.
- "*Estimates for some computational techniques in linear algebra*," S. Kaniel, *Mathematics of Computation*, 1966.
- "*The computation of eigenvalues and eigenvectors of very large sparse matrices*," C. Paige, 1971.