The Power Method

Wednesday, February 27, 2013
3:28 PM

Problem: Given a $A \in \underbrace{\mathbb{R}^{n\times n} \text{ symmetric}}_{S^n}$

find its top (and bottom) eigenvalues and eigenvectors.

Power Method (Power Iteration)

Algorithm:

Input: A, $q^{(0)}$ ← typically, simply chosen at random.

for $k = 1, 2, \ldots$

$$z^{(k)} = A q^{(k-1)}$$

$$q^{(k)} = z^{(k)} / \|z^{(k)}\|_2$$

$$\lambda^{(k)} = q^{(k)} A q^{(k)}$$

(1) Why would we expect this to work?

(2) How fast do we expect it to converge.

Example: $A = \begin{bmatrix} 3 & & \\ & 2 & \\ & & 1 \end{bmatrix}$ Leading e-vector: $e_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$, $\lambda_1 = 3$.

—

The Power Method ~ Cont'd

Wednesday, February 27, 2013
7:02 PM

$$A = \begin{bmatrix} 3 & & \\ & 2 & \\ & & 1 \end{bmatrix} \qquad q^{(0)} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

$$q^{(1)} = z^{(1)} / \|z^{(1)}\|,$$

$$z^{(1)} = A q^{(0)} = \begin{pmatrix} 3 \\ 2 \\ 1 \end{pmatrix}$$

$$q^{(k)} = \frac{A^k q^{(0)}}{\|A^k q^{(0)}\|_2} = \frac{\begin{pmatrix} 3^k \\ 2^k \\ 1^k \end{pmatrix}}{\left\| \begin{pmatrix} 3^k \\ 2^k \\ 1 \end{pmatrix} \right\|_2} \longrightarrow \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

$$\boxed{\left(3/2\right)^k}$$

The Power Method ~ cont'd

Wednesday, February 27, 2013
7:08 PM

Algorithms: Input: A, $q^{(0)}$

For $k = 1, 2, \ldots$

$z^{(k)} = A q^{(k-1)}$

$q^{(k)} = z^{(k)} / \|z^{(k)}\|_2$

$\lambda^{(k)} = q^{(k)T} A q^{(k)}$

Thm: The Power Method

Suppose $A \in S^n$,

$Q^T A Q = \Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n)$

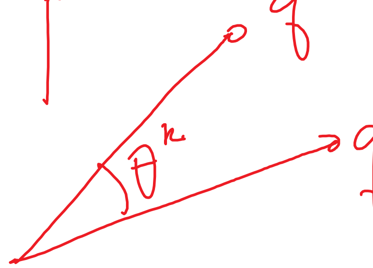
$Q = [q_1, \cdots, q_n]$ orthonormal, also assume:

$\lambda_1 > \lambda_2 \geq \cdots \geq \lambda_n$

this strict inequality.

Recall: Columns of Q, i.e., $\{q_i\}$ are e-vectors of A.

Let $\cos(\theta_k) = |\langle q_1, q^{(k)} \rangle|$



If $\cos(\theta_0) \neq 0$, then:

ⓐ $|\sin(\theta_k)| \leq \tan(\theta_0) \cdot \left|\frac{\lambda_2}{\lambda_1}\right|^k$

geometric convergence w/ rate (λ_2 / λ_1)

ⓑ $|\lambda^{(k)} - \lambda_1| \leq |\lambda_1 - \lambda_n| \cdot \tan(\theta_0)^2 \cdot |\lambda_2 / \lambda_1|^{2k}$

$\cdot \left|\frac{\lambda_2}{\lambda_1}\right|^{2k}$

Key points: (i) Initialization vector cannot be $\perp$ to q_1

(ii) Convergence is geometric.

The Power Method ~ cont'd

Wednesday, February 27, 2013
7:15 PM

<u>Setting</u>: $A \in S^n$, $Q = [q_1, \dots, q_n]$ e-vectors with

e-values $\lambda_1 > \lambda_2 \geq \dots \geq \lambda_n$.

From definition of Power Method Iteration:

$$q^{(k)} = \frac{A^k q^{(0)}}{\|A^k q^{(0)}\|_2}.$$

$$|\sin(\theta_k)|^2 = \underbrace{1 - \langle q_1, q^{(k)} \rangle^2}$$

Pythagorean theorem
+ definition of <u>sine</u>.

$$= 1 - \left(\frac{q_1^T A^k q^{(0)}}{\|A^k q^{(0)}\|_2} \right)^2.$$

$q^{(0)} \in \mathbb{R}^n$, $\{q_1, q_2, \dots, q_n\}$ onb for $\mathbb{R}^n$.

$$q^{(0)} = a_1 q_1 + \dots + a_n q_n$$

$\neq 0 \iff a_1 \neq 0$

assumption: $q^{(0)} \theta_0 \neq 0 \iff a_1 \neq 0$

$\|q^{(0)}\|_2 = 1 \iff \sum a_i^2 = 1.$

q_i e-vector for A

$Aq_i = \lambda_i q_i$

$A^k q^{(0)} = \sum a_i A^k q_i$

$= \sum a_i \lambda_i^k q_i$

Wednesday, February 27, 2013
7:21 PM

$$|\sin(\theta_k)|^2 = 1 - \langle q_1, q^{(k)}\rangle^2 = 1 - \left(\frac{q_1^T A^k q^{(0)}}{\|A^k q^{(0)}\|_2}\right)^2$$

and $q^{(0)} = \sum a_i q_i$, $\sum a_i^2 = 1$

$$\Rightarrow A^k q^{(0)} = \sum a_i \lambda_i^k q_i$$

$$|\sin(\theta_k)|^2 = 1 - \frac{a_1^2 \lambda_1^{2k}}{\sum a_i^2 \lambda_i^{2k}} = \frac{\sum_{i=2}^n a_i^2 \lambda_i^{2k}}{\sum_{i=1}^n a_i^2 \lambda_i^{2k}}$$

$$\leq \frac{\sum_{i=2}^n a_i^2 \lambda_i^{2k}}{a_1^2 \lambda_1^{2k}} = \left(\frac{1}{a_1^2}\right) \sum_{i=2}^n a_i^2 \left(\lambda_i / \lambda_1\right)^{2k}$$

$$\leq \left(\frac{1}{a_1^2}\right) \cdot \left(\frac{\lambda_2}{\lambda_1}\right)^{2k} \cdot \sum_{i=2}^n a_i^2$$

$$= \underbrace{\frac{1-a_1^2}{a_1^2}}_{} \cdot \left(\lambda_2 / \lambda_1\right)^{2k}$$

$$= (\tan\theta_0)^2 \left(\lambda_2 / \lambda_1\right)^{2k}$$

$$\Rightarrow \quad |\sin\theta_k| \leq \tan(\theta_0) \cdot |\lambda_2/\lambda_1|^k.$$

This is the first part of the Theorem

Wednesday, February 27, 2013
7:26 PM

Eigenvalue convergence:

Recall: $\lambda^{(k)} = q^{(k)T} A q^{(k)}$

$$q^{(k)} = \frac{A^k q^{(0)}}{\|A^k q^{(0)}\|_2}$$

$$\lambda^{(k)} = q^{(k)T} A q^{(k)} = \frac{q^{(0)T} A^{2k+1} q^{(0)}}{q^{(0)T} A^{2k} q^{(0)}} = \frac{\sum_{i=1}^{n} a_i^2 \lambda_i^{2k+1}}{\sum_{i=1}^{n} a_i^2 \lambda_i^{2k}}$$

$$\Rightarrow |\lambda^{(k)} - \lambda_1| = \left| \frac{\sum_{i=2}^{n} a_i^2 \lambda_i^{2k} (\lambda_i - \lambda_1)}{\sum_{i=1}^{n} a_i^2 \lambda_i^{2k}} \right|$$

$$\leq |\lambda_1 - \lambda_n| \cdot \frac{1}{a_1^2} \sum_{i=2}^{n} a_i^2 \left(\lambda_i / \lambda_1\right)^{2k}$$

$$\leq |\lambda_1 - \lambda_n| \cdot \tan(\theta_0)^2 \left(\lambda_2 / \lambda_1\right)^{2k}$$

$\blacksquare$

Wednesday, February 27, 2013
7:41 PM

Inverse Iteration:

Suppose we have some $\bar{\lambda}$ close to λ_i

Then: $(A - \bar{\lambda} I)^{-1}$ has a very large eigenvalue at $\lambda = \lambda_i$ b/c $(A - \bar{\lambda} I)$ is nearly rank defficient.

Idea: Power iteration using $(A - \bar{\lambda} I)^{-1}$

Rayleigh Quotient Iteration

$A \in S^n$: if x e-vector $Ax = \lambda x$.

But what if x is not?

$r(x)$ = "approximate e-value corresponding to vector x"

x e-vector: $Ax = \lambda x$ $\quad \|(A - \lambda I)x\|_2 = 0$

$\arg\min_\lambda \|(A - \lambda I)x\|_2$

$$r(x) = \arg\min_{\lambda} \underbrace{\|(A - \lambda I)x\|_2}_{\text{convex}}$$

Exercise: $$= \boxed{\frac{x^T A x}{x^T x} = r(x)}$$

Wednesday, February 27, 2013
7:46 PM

Algorithm: Input x_0, $\|x_0\|_2 = 1$

For $k = 0, 1, 2, \ldots$

$$\mu_k = r(x_k)$$

solve: $(A - \mu_k I) z_{k+1} = x_k$

$$x_{k+1} = z_{k+1} / \|z_{k+1}\|_2$$

Power iteration using $(A - \mu_k I)^{-1}$.

What about more e-vectors / e-values?

Orthogonal Iteration.

for $k = 1, 2, \ldots$

$$Z_k = A Q_{k-1}$$

$$Q_k R_k = Z_k$$

$Q_0, Q_1, Q_2, \ldots$

Q_i is $n \times m$ orthonormal matrix.

If $m = 1$: check that this is

If $m=1$: check that this is just power iteration.

→ normalizes and orthogonalizes

"QR-Factorization".

A $n \times m$

QR-factorization

$A = QR$

orthogonal

upper triangular.

Lanczos Iteration

Wednesday, February 27, 2013
9:11 PM

Problem: Given $A \in S^n$, find its eigenvalues & eigenvectors.

* In particular: we want to find its top & bottom e-values / e-vectors quickly.

Preliminaries

① Rayleigh quotient: "approximate e-value" of a vector x

$$r(x) \doteq \frac{x^T A x}{x^T x}.$$

② Krylov subspaces

Given a vector $q \in \mathbb{R}^n$, $A \in S^n$

define the Krylov subspace as:

$$\mathcal{K}(A, q, k) \doteq \operatorname{span}\{q, Aq, \dots, A^{k-1} q\}$$

$$K(A, q, k) \doteq [\, q \;\; Aq \;\; \cdots \;\; A^{k-1} q \,]$$

Range $(K) = K$.

Finding onb. for $K(A, f, k)$.

Lanczos Iteration

Wednesday, February 27, 2013
9:19 PM

③ Tri-diagonal Matrices.

If $A \in S^n$, $\exists\, Q$ orthogonal, $Q^T A Q = T$

tridiagonal

[Tri-diagonalization often an intermediate step to diagonalization

Recall: $Q^T A Q = \Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$

ⓐ e-vectors of A are columns of Q
ⓑ e-values of A are $\lambda_1, \ldots, \lambda_n$.

⊛ Please make sure you are comfortable w/ ⓐ & ⓑ.

④ Connection to Krylov subspaces.

Suppose that $Q_k = \begin{bmatrix} | & & | \\ q_1, & \ldots, & q_k \\ | & & | \end{bmatrix}$

... onb for $K(A, q_1, k)$, $k = 1, 2, \ldots, n$

Claim: $Q_n^T A Q_n = T$ is tridiagonal,

pf: $T_{ij} = ? = q_i^T A q_j = 0$ for $i > j+1$

$Aq_j \in \text{span}\{q_1, \ldots, q_j, q_{j+1}\} \perp q_i, \; i > j+1$

The result follows (symmetry used).

Lanczos Iteration

Wednesday, February 27, 2013
9:26 PM

Key ideas & outline

① $\lambda_1 \triangleq \max_{y \neq 0} : \frac{y^T A y}{y^T y}$, $\lambda_n \triangleq \min_{y \neq 0} : \frac{y^T A y}{y^T y}$

max/min over $\mathbb{R}^n$

Find and augment iteratively a k-dim'l subspace V_k, so that

$$M_k = \max_{0 \neq y \in V_k} : \frac{y^T A y}{y^T y} , \quad m_k = \min_{0 \neq y \in V_k} : \frac{y^T A y}{y^T y}$$

good approxim's to λ_1, λ_n respectively.

② Krylov: we will show that an excellent choice for $V_k = K(A, q_1, k)$,

③ Suppose $Q_k = [q_1 \cdots q_k]$ onb $K(A, q, k)$

$$M_k = \max_{0 \neq y \in K} : \frac{y^T A y}{y^T y} = \max_{y \neq 0} : \frac{(Q_k y)^T A (Q_k y)}{(Q_k y)^T (Q_k y)}$$

why: By def'n, Range$(Q_k) = K$

Will show: M_k close to λ_1

m_k " λ_n

m_k

④ Computationally? $(Q_k y)^T A (Q_k y) = y^T \underbrace{[Q_k^T A Q_k]}_{} y$

→ Tri-diagonal ! (when $k=n$)

k=n: $\underbrace{Q^T A Q}_{} = \underbrace{T}_{}$ = tridiagonal

e-vals/e-vec of A ~~given~~ by e-vals - evec of T.

For $k<n$, turns out: $\boxed{\text{very close!}}$

Lanczos Iteration

Wednesday, February 27, 2013
9:37 PM

Problem: Given $A \in S^n$ symmetric, sparse, find top/bottom eigenvalues of A.

Recall: Rayleigh quotient: $r(x) = \frac{x^T A x}{x^T x}$ $(x \neq 0)$

Exercise: show $\lambda_n \leq r(x) \leq \lambda_1$

For any sequence of o.n. vectors $q_1, \ldots, q_n \in \mathbb{R}^n$

$$Q_k = \begin{bmatrix} | & & | \\ q_1 & \cdots & q_k \\ | & & | \end{bmatrix} \quad n \times k \text{ matrix}$$

$$M_k = \lambda_1(Q_k^T A Q_k) = \max_{y \neq 0} \frac{y^T (Q_k^T A Q_k) y}{y^T y}$$

$$= \max_{\|y\|_2 = 1} : r(Q_k y) \leq \underline{\lambda_1(A)}$$

$$m_k = \min_{\|y\|_2 = 1} : r(Q_k y) \geq \lambda_n(A)$$

Idea: design - build in iterative fashion sequence $\{q_1, q_2, \ldots\}$ so that M_k, m_k increasingly good estimates of λ_1, λ_n resp'ly.

$$\{q_1, \ldots, q_k\} \cup \{q_{k+1}\}$$

?

needs to improve both

M_k & m_k.

Lanczos Iteration

Wednesday, February 27, 2013
9:48 PM

$$M_k = \max_{\|y\|_2=1} : r(Q_k y) \qquad \bar{y}$$

$$m_k = \min_{\|y\|_2=1} : r(Q_k y) \qquad \underline{y}$$

$$u_k = Q_k \bar{y}, \qquad r(u_k) = r(Q_k \bar{y}) = M_k$$

$$v_k = Q_k \underline{y}, \qquad r(v_k) = r(Q_k \underline{y}) = m_k$$

$$u_k, v_k \in \text{span}\{q_1, \ldots, q_k\}$$

Add q_{k+1} so that $u_{k+1}, v_{k+1} \in \text{span}\{q_1, \ldots, q_{k+1}\}$
new

<u>both</u> improved.

How to choose q_{k+1}?

Optimization-based idea:

add q_{k+1} so that $\nabla r(u_k), \nabla r(v_k)$

<u>both</u> contained in $\{q_1, \ldots, q_k, q_{k+1}\}$.

$$r(x) = \frac{x^T A x}{\quad}$$

Recall: $r(x) = \frac{x^T A x}{x^T x}$

$$\nabla r(x) = \frac{2}{x^T x}\left[A x - r(x) x \right]$$

$$\nabla r(x) \in \operatorname{span}\{x, Ax\}$$

Lanczos Iteration

Wednesday, February 27, 2013
9:53 PM

$\nabla r(x) \in \text{span}\{x, Ax\}$

$u_k, v_k \in \text{span}\{q_1, \dots, q_k\}$

$\nabla r(v_k), \nabla r(u_k) \in \text{span}\{q_1, \dots, q_k, Aq_1, \dots Aq_k\}$

<u>Krylov</u>: $\text{span}\{q_1, \dots, q_k\} = \text{span}\{q_1, Aq_1, \dots, A^{k-1}q_1\}$

i.e. $q_1, \dots, q_k$ is onb for $K(A, q, k)$

<u>Then</u>: $u_k, v_k \in \text{span}\{q_1, \dots, q_k\}$

(by Krylov) $= \text{span}\{q_1, Aq_1, \dots, A^{k-1}q_1\}$

$\Rightarrow \nabla r(u_k), \nabla r(v_k) \in \text{span}\{q_1, Aq_1, \dots, A^{k-1}q_1, A^k q_1\}$.

$\Rightarrow$ <u>Choose</u>: q_{k+1} so that $\{q_1, \dots, q_{k+1}\}$

is onb for $K(A, q_1, k+1)$.

Main Problem: Finding onb for

Krylov subspaces.

Key for Algorithm: Tridiagonalization.

Lanczos Iteration

Wednesday, February 27, 2013
11:30 PM

Insight: We need to find an iterative (algorithmic) way to find a o.n.b. for $K(A, q, k)$.

$\boxed{\text{Tridiagonalization}}$

Suppose that $Q^T A Q = T \leftarrow$ tridiagonal
((Claim: always possible, and "easy" $\rightarrow$ this lecture).

where $q_1 = Q e_1$.

Then: $K(A, q_1, n) = Q\,[e_1, Te_1, \ldots, T^{n-1} e_1]$

that is: T, Q give a QR-factorization of the Krylov matrix K, which also gives an onb for K.

$\Rightarrow$ Goal: Tridiagonalize A with orthogonal matrix Q, s.t. $Qe_1 = q_1$.

Summary: If $\exists Q$, w/ $Qe_1 = q_1$, orthogonal and $Q' A Q = I$

Then, this Q is indeed what we want: the columns of Q give us the onb. for K.

Do This: Iteratively.

Lanczos Iteration

Wednesday, February 27, 2013
11:36 PM

Question: How do we tridiagonalize a symmetric matrix using orthogonal transformation?

One method: Householder reflections

Issue: If A is sparse, then using Householder reflections may have dense intermediate steps

$\longrightarrow$ no good.

Instead: direct algorithm that can exploit sparsity.

$Q = [q_1 \cdots q_n]$, $\boxed{T = Q^T A Q}$

$$T = \begin{bmatrix} \alpha_1 & \beta_1 & & & 0 \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \ddots & \ddots & \\ & & \ddots & \ddots & \beta_{n-1} \\ \end{bmatrix}$$

$$\begin{bmatrix} \beta_2 & \cdots & \cdots & \beta_{n-1} & \\ 0 & & \cdots & \beta_{n-1} & \alpha_n \end{bmatrix}$$

$$\boxed{AQ = QT}$$

<u>Solve</u>.

Lanczos Iteration

Wednesday, February 27, 2013
11:41 PM

$$AQ = QT$$

$$T = \begin{bmatrix} \alpha_1 & \beta_1 & & & 0 \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \ddots & \ddots & \\ & & \ddots & \ddots & \beta_{n-1} \\ 0 & & & \beta_{n-1} & \alpha_n \end{bmatrix}$$

q

k^{th} column

$$Aq_k = QT_k$$

$$= \beta_{k-1} q_{k-1} + \alpha_k q_k + \beta_k q_{k+1}, \quad \boxed{\beta_0 q_0 = 0}$$

q_i orthonormal

$$\rightarrow \boxed{q_k^T A q_k = \alpha_k \|q_k\|^2 = \alpha_k}$$

$$\beta_k q_{k+1} = \underbrace{(A - \alpha_k I) q_k - \beta_{k-1} q_{k-1}}_{r_k}$$

If $r_k \neq 0$ then $\boxed{\begin{aligned} q_{k+1} &= r_k / \beta_k, \\ \beta_k &= \pm \|r_k\|_2 \end{aligned}}$

Algorithm: Lanczos Iteration

Initialize: $r_0 = q_1, \ \beta_0 = 1, \ q_0 = 0, \ k = 0$

while: $(\beta_k \neq 0)$

$$q_{k+1} = r_k / \beta_k$$

$$k = k+1$$

$$\alpha_k = q_k^T A q_k$$

$$r_k = (A - \alpha_k I) q_k - \beta_{k-1} q_{k-1}$$

$$\beta_k = \|r_k\|_2 .$$

Lanczos Iteration

Wednesday, February 27, 2013
11:46 PM

If process terminates: $\boxed{k = n}$ then it produces: $Q^T A Q = T$.

1. What if terminates early b/c $\beta_k = 0$?
2. What if we terminate it early? Do we get good estimates of λ_1, λ_n e-values?

"good" $\leftrightarrow$ compared to, e.g., The Power Method.

What happens if algorithm terminates? i.e., $r_k = 0$. This will only happen if q_1 is contained in an invariant subspace of dim'n $< n$.

$$\{q_1, Aq_1, A^2 q_1, \ldots\ldots\} \subseteq \{q_1, \ldots, A^{m-1} q_1\},$$
$m < n$.

Lanczos Iteration

Wednesday, February 27, 2013
11:49 PM

<u>Thm</u>: Let $A \in S^n$, $q_1 \in \mathbb{R}^n$, $\|q_1\| = 1$

Then, the Lanczos Alg runs until iteration $m = \text{rank}\left(K(A, q_1, n)\right)$.

Moreover, $k = 1, \ldots, m$, $AQ_k = Q_k T_k + r_k e_k^T$

where $T_k = \begin{bmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \ddots & \ddots & \\ & \ddots & \ddots & \beta_{k-1} \\ & & \beta_{k-1} & \alpha_k \end{bmatrix}$ $\qquad \dfrac{Q_k = [q_1 \cdots q_k]}{\text{span } K(A, q_1, k)}$

<u>pf</u>: Induction on k.

Base case: $k = 1$ — immediate (<u>check!</u>)

Suppose that at iteration k, we have:

$Q_k = [q_1 \cdots q_k]$, s.t. $\text{range}(Q_k) = K(A, q_1, k)$
$Q_k^T Q_k = I_k$.

From the construction, we know

$AQ_k = Q_k T_k + r_k e_k^T \longleftarrow$ <u>Check!</u>

$$\underbrace{Q_k^T A Q_k}_{\text{symmetric}} = T_k + Q_k^T r_k e_k^T \quad (\star)$$

by algorithm

$$(Q_k^T A Q_k)_{ii} = q_i^T A q_i = \alpha_i$$

and: $q_{i+1}^T A q_i = \beta_i$

What does this and $(\star)$ tell us?

Lanczos Iteration

Wednesday, February 27, 2013
11:55 PM

This implies: $\boxed{Q_k^T A Q_k = T_k}$

$\Rightarrow \quad Q_k^T A Q_k = T_k + \underbrace{Q_k^T r_k e_k^T}_{=0}$

$\Rightarrow \quad Q_k^T r_k = 0.$

Two cases

(1) $r_k \neq 0 \Rightarrow q_{k+1} = r_k / \|r_k\|_2$

$Q_k^T r_k = 0 \rightarrow q_{k+1} \perp \{q_1, \dots, q_k\}$

and $\quad q_{k+1} \in \text{span}\{A q_k, q_k, q_{k-1}\}$

$\Rightarrow \quad Q_{k+1}^T Q_{k+1} = I_{k+1}$

and $\quad \text{range}(Q_{k+1}) = K(A, q_1, k+1)$

That is: The algorithm is working

.. claimed by the

as claimed by the induction.

② $r_k = 0 \Rightarrow \boxed{AQ_k = Q_k T_k} + \cancel{r_k e_m^T}$

$\Rightarrow k = m = \operatorname{rank}(K(A, q_1, n))$.

∎

Lanczos Iteration

Wednesday, February 27, 2013
11:59 PM

If alg. terminates at $m < n$, then we have found an invariant subspace, and hence we can compute e-values, and reduce the size of problem.

Q: When will this actually happen if we choose q_1 at random?

A: This will never happen.

$\Rightarrow$ Alg will NOT terminate early leaves question: What if we stop it early? Do we get anything?

Does T_k give any good estimates of e-values & e-vectors of A?

(we know Tn does?)

Lanczos Iteration

Thursday, February 28, 2013
12:03 AM

<u>Thm</u>: Suppose we have run Lanczos for k steps, w/o termination.

$\Rightarrow T_k$

look at e-values: $T_k \in S^n$

$$S_k^T T_k S_k = \Theta_k = \begin{pmatrix} \theta_1 & & \\ & \ddots & \\ & & \theta_k \end{pmatrix}$$

Let $\boxed{Y_k = Q_k S_k} \in \mathbb{R}^{n \times k}$ $\quad Y_k = [y_1 \cdots y_k]$

<u>Then</u>: y_i are "close" to e-vectors of A, and θ_i are "close" to e-values of A.

$$\| A y_i - \theta_i y_i \|_2 = |\beta_k| \cdot |s_{ki}|$$

k,i elements of S.

pf: Recall: $A Q_k = Q_k T_k + r_k e_k^T$

$\Rightarrow A Q_k S_k = Q_k T_k S_k + r_k e_k^T S_k$

look at i^{th} column of

$$\Rightarrow AY_k = Q_k S_k S_k^T T_k S_k + r_k e_k^T S_k$$

$$= Y_k \Theta_k + r_k e_k S_k$$

$$\Rightarrow A y_i = \theta_i y_i + r_k e_k^T S_k e_i$$

$$\Rightarrow \|A y_i - \theta_i y_i\|_2 = |\beta_k| \cdot |s_{ki}|. \quad \blacksquare$$

Lanczos Iteration

Thursday, February 28, 2013
12:08 AM

Kaniel - Paige Convergence Theory

Thm: $A \in S^n$, $\lambda_1 \geq \cdots \geq \lambda_n$, e-vectors $z_1, \ldots, z_n$.

Let T_k the $k \times k$ tridiagonal matrix obtained after k steps of Lanczos — denote e-values of T_k by $\theta_1, \ldots, \theta_k$.

Then: $$\lambda_1 \geq \theta_1 \geq \lambda_1 - \frac{(\lambda_1 - \lambda_n)\tan(\phi_1)^2}{(C_{k-1}(1+2\rho_1))^2}$$

where: $\cos\phi_1 = |\langle q_1, z_1 \rangle|$

$\rho_1 = (\lambda_1 - \lambda_2)/(\lambda_2 - \lambda_n)$

$C_{k-1}(x)$ is the Chebyshev polynomial of degree $k-1$,

pf: Upper bound is immediate. $w^T A w$

pf: Upper bound is immediate,

$$\lambda_1 = \max_{\|w\|_2=1} w^T A w = \max_{w \neq 0} \frac{w^T A w}{w^T w}$$

$$\theta_1 = \max_{y \neq 0} \frac{y^T T_k y}{y^T y} \longrightarrow \text{let } T_k = Q_k^T A Q_k$$

$$= \max_{y \neq 0} \frac{y^T Q_k^T A Q_k y}{y^T y} = \max_{0 \neq w \in \mathcal{K}(A, q_1, k)} \frac{w^T A w}{w^T w}.$$

$$\Rightarrow \theta_1 \le \lambda_1$$

Lanczos Iteration

Thursday, February 28, 2013
12:14 AM

<u>Lower bound</u>: $\theta_1 = \max\limits_{0 \neq w \in K} : \frac{w^T A w}{w^T w}$

$$K = \text{span}\{q_1, Aq_1, \ldots, A^{k-1} q_1\} \longleftarrow$$

$$= \{p(A) q_1 : p_1 \text{ a degree } k-1 \text{ poly}\}$$

Rewrite θ_1:

$$\theta_1 = \max_{p \in \mathcal{P}_{k-1}} : \frac{q_1^T p(A) A p(A) q_1}{q_1^T p(A)^2 q_1}$$

<u>Write</u>: q_1 = write in the basis of e-vectors of A: $z_1, \ldots, z_n$

$$\underline{q_1 = \sum d_i z_i}, \quad \|q_1\| = 1 \Rightarrow \sum d_i^2 = 1,$$

$$\Rightarrow \frac{q_1^T p(A) A p(A) q_1}{q^T p(A)^2 q_1} = \frac{\sum_{i=1}^{n} d_i^2 p(\lambda_i)^2 \lambda_i}{\sum d_i^2 p(\lambda_i)^2}$$

$$\geq \lambda_1 - (\lambda_1 - \lambda_n) \cdot \frac{\sum_{i=2}^{n} d_i^2 \; p(\lambda_i)^2}{d_1^2 \, p(\lambda_1)^2 + \sum_{i=2}^{n} d_i^2 \, p(\lambda_i)^2}$$

check!

Lanczos Iteration

Thursday, February 28, 2013
12:19 AM

$$\theta_1 = \max_{p \in \mathcal{P}_{k-1}} \frac{q_1^T p(A) A p(A) q_1}{q_1^T p(A)^2 q_1}$$

$$\geq \lambda_1 - (\lambda_1 - \lambda_n) \cdot \frac{\sum_{i=2}^{n} d_i^2 \hat{p}(\lambda_i)^2}{d_1^2 \hat{p}(\lambda_1)^2 + \sum_{i=2}^{n} d_i^2 \hat{p}(\lambda_i)^2}$$

Holds $\forall \hat{p}$.

Idea: design a good one — design a poly to maximize RHS.

RHS is big: $\hat{p}(\lambda_1)$ is big but $\hat{p}(\lambda_i)$ is small.

Chebyshev poly: designed to be ☆

"small" → bdd by 1 — in $[-1, 1]$, and grow rapidly outside.

Recall:

$$C_0 = 1$$

$$C_1(x) = x$$

$$C_k(x) = 2x\, C_{k-1}(x) - C_{k-2}(x)$$

Exercise: plot in Matlab to see property $\star$

Lanczos Iteration

Thursday, February 28, 2013
12:25 AM

In order to exploit: we translate and rescale so that value is bdd by 1 not for $x \in [-1, 1]$, but for $x \in (\lambda_n, \lambda_2)$

$$p(\lambda) = C_{k-1}\left(1 + 2\frac{\lambda - \lambda_n}{\lambda_2 - \lambda_n}\right)$$

Check: $|p(\lambda_i)| \leq 1 \quad i = 2, \dots, n$

$$p(\lambda_1) = C_{k-1}\left(1 + 2\frac{\lambda_1 - \lambda_2}{\lambda_2 - \lambda_n}\right).$$

Putting it together:

$$\theta_1 \geq \lambda_1 - (\lambda_1 - \lambda_n) \cdot \frac{\sum_{i=2}^{n} d_i^2 \, p(\lambda_i)^2}{d_1^2 \, p(\lambda_1)^2 + \sum_{i=2}^{n} d_i^2 \, p(\lambda_i)^2}$$

$$\geq \lambda_1 - (\lambda_1 - \lambda_n) \cdot \frac{\sum_{i=2}^{n} d_i^2}{d_1^2 \, p(\lambda_1)^2}$$ [How??]

$$\lambda_1 - (\lambda_1 - \lambda_n) \cdot \frac{1 - d_1^2}{\dots} \cdot \frac{1}{\dots}$$

$d_1^2 \qquad C_{k-1}(1+2\rho_1)$

Since: $\tan(\phi_1)^2 = \frac{1-d_1^2}{d_1^2}$, the result follows. $\blacksquare$

Lanczos Iteration

Thursday, February 28, 2013
12:30 AM

Thm: $\lambda_n \le \theta_k \le \lambda_n + \dfrac{(\lambda_1 - \lambda_n)\tan(\phi_n)^2}{C_{k-1}(1+2\rho_n)^2}$

where $\rho_n = (\lambda_{n-1} - \lambda_n) / (\lambda_1 - \lambda_{n-1})$

$\cos \phi_n = \langle q_n, z_n \rangle$

proof: Follows analogously, but replace A by $-A$.

Homework: Implement Chebyshev in Matlab, and compare

bounds: Lanczos to Power Method.

Much better!!

Lanczos Iteration

Thursday, February 28, 2013
12:33 AM

Why is Lanczos better?

Lanczos approximates λ_1 by

$$\mu_k = \max_{0 \neq w \in \mathcal{K}(A, q, k)} : \frac{w^T A w}{w^T w}$$

Power method?

$$\hat{\mu}_k = \text{"}\max\text{"}_{0 \neq w \in \operatorname{Span}\{A^{k-1} q_1\}} : \frac{w^T A w}{w^T w}$$

Clear: $\hat{\mu}_k \leq \mu_k$

Simulations will show this is (dramatically) the case!

Another way: Through proof of Lanczos —

... using Chebyshev,

if instead of using Chebyshev, we use: $p(x) = x^{k-1}$, then instead of Lanczos bound, we get Power Method bound.