(7) Iterative ideas for eigenvalues / eigenvector searches → the prelude to Step 2

In all that follows, we will look @ real, symmetric matrices, though generalizations exist.

(a) Rayleigh Quotient

- Suppose we have a matrix A; let's construct the scalar function $\mathbb{R}^n \to \mathbb{R}$

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

$r(x)$ is called the Rayleigh Quotient

- If $\underline{x}$ is an eigenvector with eigenvalue λ then $r(\underline{x}) = \lambda$.

- What happens when $\underline{x}$ is close to being an eigenvector?

let's calculate ∇r

$$\nabla r = \begin{pmatrix} \partial r / \partial x_1 \\ \vdots \\ \partial r / \partial x_n \end{pmatrix} \qquad \text{if } x^T = (x_1 \ldots x_n)$$

$$\frac{\partial r}{\partial x_i} = \frac{\partial}{\partial x_i} \left(\frac{\sum_{j,k} x_j a_{jk} x_k}{\sum_{k=1}^{n} x_k^2} \right)$$

$$= \frac{1}{x^T x} \left(\sum_k a_{ik} x_k + \sum_j x_j a_{ji} \right)$$

$$- \frac{x^T A x}{(x^T x)^2} \cdot 2 x_i$$

$$= \frac{1}{x^T x} \cdot \left[2 (Ax)_i - 2 \frac{x^T A x}{x^T x} x_i \right]$$

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

If $\underline{x}$ is an eigenvector so $A\underline{x} = r(x)\underline{x} = \lambda x$

$$\underline{\nabla r} = 0$$

$\Rightarrow$ [The eigenvectors are the stationary points of the Rayleigh Quotient, and the eigenvalues are the value of $r(x)$ at these points.

Consequence

Consider the Taylor expansion of $r(\underline{x})$ near the stationary pt.

let q be the eigenvector then

$$r(\underline{x}) - \lambda = r(\underline{x}) - r(q) = \cancel{(\underline{x}-q)\cdot\nabla r\big|_q} + O(\|\underline{x}-q\|)^2$$

$\rightarrow$ if $\underline{x}$ is an "estimate" of q then $r(\underline{x})$ is a quadratically accurate estimate of the eigenvalue λ.

(b) The Power iteration

Suppose we write any vector $\underline{V}$ in the basis of the (unknown) eigenvectors of A.

$$\underline{V}^{(0)} = a_1 q_1 + a_2 q_2 + \cdots \; a_n q_n$$

let $\underline{V}^{(1)} = AV^{(0)} = a_1\lambda_1 q_1 + \cdots \; a_n\lambda_n q_n$

so that $\underline{V}^{(k)} = a_1\lambda_1^k q_1 + \cdots \; a_n\lambda_n^k q_n$

$\Rightarrow$ if λ_1 is the largest eigenvalue, then $V^{(k)}$ is getting closer and closer to

$$a_i \lambda_i^k q_i$$

The difference between $v^{(k)}$ and q_i is $O\left(\left|\frac{\lambda_i}{\lambda_j}\right|^k\right)$ where λ_j is the <u>second</u> largest Evalue.

This method could provide a way of getting the evector corresponding to the largest evalue.

<u>Problem</u>
- we want all the evalues
- the convergence is slow if $\frac{\lambda_i}{\lambda_j}$ is not v. large.

(c) <u>Inverse Iteration</u>

The inverse iteration method can solve both problems.

<u>Idea</u>: For any $\mu \in \mathbb{R}$ where μ is <u>NOT</u> an Evalue of A then

- the Evectors of A are the Evectors of $(A-\mu I)^{-1}$
- the corresponding Evalues are $\dfrac{1}{\lambda_j - \mu}$ where $\{\lambda_j\}$ are Evalues of A.

<u>Proof</u>: let $\underline{v}_j$ be an evector of A with Evalue λ_j then

$$A\underline{v}_j = \lambda_j \underline{v}_j$$

$$(A-\mu I)\underline{v}_j = A v_j - \mu \underline{v}_j I$$

$$= (\lambda_j - \mu_j)\underline{v}_j$$

$$\Rightarrow \quad \underline{v}_j = (A-\mu I)^{-1}(\lambda_j - \mu_j)\underline{v}_j$$

$$\Rightarrow \quad \frac{1}{\lambda_j - \mu_j} \underline{v}_j = (A - \mu I)^{-1} \underline{v}_j$$

Now suppose μ is close to a certain Evalue λ_j
then $\dfrac{1}{\lambda_j - \mu}$ is very large, and much larger than $\dfrac{1}{\lambda_k - \mu}$ for all other evalues $\lambda_k \neq \lambda_j$.

$\Rightarrow$ ① the power iteration on $(A-\mu I)^{-1}$ can be chosen to converge to any v_j provided μ is chosen close to λ_j.

② the iterations will be fast provided μ is v. close to λ_j

ⓓ Combining the Rayleigh quotient method & the inverse power method

Rayleigh quotient : starting from a vector close to q_j yields an approximation to λ_j

Inverse method : Starting from a evalue close to λ_j finds a better approximation to q_j

Idea : combine them!

$$\text{better approximate value of } \lambda_j \; \underset{\text{Rayleigh quotient}}{\overset{\text{INV. POWER}}{\rightleftarrows}} \; \text{better approximate evector } q_j$$

Rayleigh quotient iteration algorithm (starting with A in Hessenberg form)

① Start with $\begin{cases} v^{(0)} = \text{a vector with unit norm} \\ \lambda^{(0)} = v^{(0)T} A v^{(0)} \end{cases}$

② for $k = 1, \ldots$

- Solve $(A - \lambda^{(k-1)} I) w = v^{(k-1)}$ using method desired

  ← equivalent to apply $(A - \lambda^{(k-1)} I)^{-1}$ to $v^{(k-1)}$

- Set $v^{(k)} = \dfrac{w}{\|w\|}$ ← normalize w and use it as next guess vector

- $\lambda^{(k)} = v^{(k)T} A v^{(k)}$ ← next shift is R.Q of vector $v^{(k)}$

Note

- The convergence of this algorithm is phenomenally fast.
- The algorithm converges to the Evector closest in direction to $v^{(0)}$. To find the others, start with other $v^{(0)}$ on the unit hypersphere.
- The algorithm is of course much faster if A is put in Hessenberg form first; since A is assume real & symmetric, H is actually tridiagonal
- Note that E-vectors of $H \neq$ Evectors of A so if we want the E-vectors of A we need to keep the information on the Hessenberg transformation to transform the E-vectors of H back into the E-vectors of A. In practise, this is rarely done...