Summer@ICERM 2020
June 8th - July 31st

Efficient Eigensolvers and Their Applications

Final Presentations

Erica Liu, Kelly Rivera, & Jordan Fox

Friday, July 31st

Table of Contents

Basics of Page Rank

Page Rank: Introductory Example

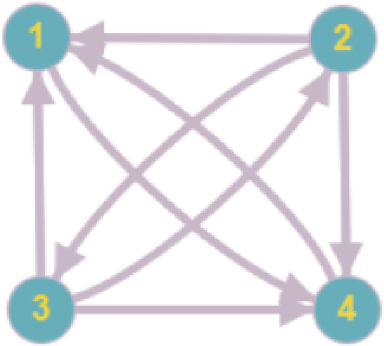


Figure 1

Page Rank: Importance Score Equation

$$x_k = \sum_{j \in L_k} \frac{x_j}{n_j} \tag{1}$$

where x_k represents the importance score of page k, L_k is the set of pages with links to page k, and n_j is the number of links from page j.

Page Rank: System of Equations

$$x_k = \sum_{j \in L_k} \frac{x_j}{n_j} \tag{2}$$

x_k: importance score of page k, L_k: set of pages with links to page k, and n_j: number of links from page j

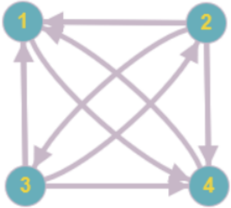


Figure 2

$$x_1 = \frac{x_2}{3} + \frac{x_3}{3} + \frac{x_4}{1}$$
$$x_2 = \frac{x_3}{3}$$
$$x_3 = \frac{x_2}{3}$$
$$x_4 = \frac{x_1}{1} + \frac{x_2}{3} + \frac{x_3}{3}$$

Page Rank: Adjacency Matrix

System of Equations:

$$x_1 = \frac{x_2}{3} + \frac{x_3}{3} + \frac{x_4}{1}$$
$$x_2 = \frac{x_3}{3}$$
$$x_3 = \frac{x_2}{3}$$
$$x_4 = \frac{x_1}{1} + \frac{x_2}{3} + \frac{x_3}{3}$$

$$A = \begin{bmatrix} 0 & \frac{1}{3} & \frac{1}{3} & 1 \\ 0 & 0 & \frac{1}{3} & 0 \\ 0 & \frac{1}{3} & 0 & 0 \\ 1 & \frac{1}{3} & \frac{1}{3} & 0 \end{bmatrix}$$

Page Rank: Continued

$$Ax = \lambda x; \text{where } \lambda = 1 \tag{3}$$

Page Rank: Continued

$$Ax = \lambda x; \text{where } \lambda = 1 \quad (3)$$

$$\begin{bmatrix} 0 & \frac{1}{3} & \frac{1}{3} & 1 \\ 0 & 0 & \frac{1}{3} & 0 \\ 0 & \frac{1}{3} & 0 & 0 \\ 1 & \frac{1}{3} & \frac{1}{3} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

Page Rank: Eigenvector Example

Consider a case where this was the resulting eigenvector:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 12 \\ 6 \\ 3 \\ 9 \end{bmatrix}$$

Page Rank: Eigenvector Example

Consider a case where this was the resulting eigenvector:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 12 \\ 6 \\ 3 \\ 9 \end{bmatrix}$$

$$\text{Probabilities: } \frac{1}{30} * \begin{bmatrix} 12 \\ 6 \\ 3 \\ 9 \end{bmatrix} = \begin{bmatrix} 0.4 \\ 0.2 \\ 0.1 \\ 0.3 \end{bmatrix}$$

Eigensolvers

Before 1961: An Intuition

One thought: "Let's leveraging the simple definitions and elegant characterizations of eigenvalues and eigenvectors!"

Before 1961: Characteristic Polynomial

$$p(\lambda) = det(A - \lambda I) = 0$$

The accuracy of this approach depends on the basis employed for the polynomial as well the eigenstructure of A.

Finding eigenvalues of matrices via roots of polynomials is — regardless of the algorithmic details — a numerically unstable eigensolver.

Eigensolvers

Two ideologies:

1. A converged sequence
2. The computation of an eigenvalue-revealing factorizations of A, where the eigenvalues appear as entries of one of the factors:
   - diagonalization

     $$\text{nondefective } A = X\Lambda X^{-1}$$

   - unitary diagonalization

     $$\text{normal } A = Q\Lambda Q^*$$

   - Schur diagonalization

     $$A = QTQ^*$$

Power Iteration Method

Algorithm 1 Power Iteration Method

Initialize a unit vector $x^{(0)}$
$\lambda_{\mathbf{0}} \leftarrow r(x^{(0)})$
for k = 1, 2 ... till converge **do**
    $x^k \leftarrow Ax^{(k-1)} / \left\| Ax^{(k-1)} \right\|$
    $\lambda_k \leftarrow r(x^k)$
return x^k, λ_k

Why does this sequence converge to the desired eigenvector?

Theorem of Convergence If the matrix has a unique dominant eigenvalue, the sequence

$$\frac{x}{\|x\|}, \frac{Ax}{\|Ax\|}, \frac{A^2x}{\|A^2x\|}, \frac{A^3x}{\|A^3x\|}, \dots$$

converges to the eigenvector corresponding to the dominant eigenvalue of A.

Closer look

Claim: When the dominant eigenvalue is simple, Power Iteration method could converge to the desired eigenvector.

Run time: Power Iteration is not efficient for general use. It's convergence rate is linear, $O(|\frac{\lambda_2}{\lambda_1}|)$.

Inverse Iteration

If an invertible matrix $A \in R^{n \times n}$ has eigenvalues $\{\lambda_1, \lambda_2, \ldots \lambda_n\}$ such that $|\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_n|$, then the matrix A^{-1} has eigenvalues $\{\frac{1}{\lambda_1}, \frac{1}{\lambda_2}, \ldots \frac{1}{\lambda_n}\}$ such that $|\frac{1}{\lambda_1}| \leq |\frac{1}{\lambda_2}| \leq \ldots \leq |\frac{1}{\lambda_n}|$. The eigenvectors are unchanged.

Algorithm 2 Inverse Iteration Method

Initialize a unit vector $x^{(0)}$
$\lambda_0 \leftarrow (x^{(0)})A(x^{(0)})$
while termination condition not satisfied **do**
    Solve $A(x^{(k)}) = v^{(k-1)}$ for w
    $x^k \leftarrow x^{(k)} / \left\| x^{(k)} \right\|$
    $\lambda_k \leftarrow (x^k)A(x^k)$
    if *Condition* **then**
        break
return x^k, λ_k

Note that for any $\mu \in R$, the matrix $B = A - \mu I$ has eigenvalues $\{\lambda_1 - \mu, \lambda_2 - \mu, \ldots \lambda_n - \mu\}$. The eigenvectors are unchanged.

Algorithm 3 Inverse Iteration With Shift

Initialize a unit vector $x^{(0)}$
$\lambda_{\mathbf{0}} \leftarrow (x^{(0)})A(x^{(0)})$
$B \leftarrow$(A - μI)
while termination condition not satisfied **do**
  Solve $B(x^{(k)}) = x^{(k-1)}$ for x$^{(k)}$
  $x^{(k)} \leftarrow x^{(k)} / \left\|x^{(k)}\right\|$
  $\lambda_k \leftarrow (x^{(k)})A(x^{(k)})$
  if *Condition* **then**
    break
return $x^{(k)}, \lambda_k$

QR Algorithm

Algorithm 4 QR Algorithm

1: Set $A = A_0$
2: **for** k = 1, 2 ... till convergence [1] **do**
3: $Q_k R_k = A_k$
4: $A_{k+1} = R_k Q_k$
return x, λ

- The QR algorithm computes all eigenvalues and eigenvectors of a matrix A
- The main diagonal entries of matrix A^k equal eigenvalues
- The columns of Q^k equal the eigenvectors

Why does this Algorithm work?

[1] Convergence is determined when A becomes an upper triangular matrix

QR Algorithm

Algorithm 5 QR Algorithm

1: Set $A = A_0$
2: **for** k = 1, 2 ... till convergence [1] **do**
3: $Q_k R_k = A_k$
4: $A_{k+1} = R_k Q_k$
return x, λ

- The QR algorithm computes all eigenvalues and eigenvectors of a matrix A
- The main diagonal entries of matrix A^k equal eigenvalues
- The columns of Q^k equal the eigenvectors

Why does this Algorithm work? Schur Decomposition theorem

[1] Convergence is determined when A becomes an upper triangular matrix

QR Algorithm Variations

Householder transformations

- $Q_k = I - 2v_k v_k^T$
- $R = (Q_n \cdots Q_2 Q_1)A$
- $Q = Q_n \cdots Q_2 Q_1$

QR Algorithm Variations

Householder transformations

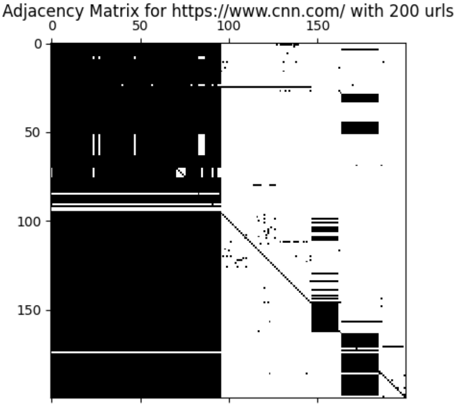
- $Q_k = I - 2v_k v_k^T$
- $R = (Q_n \cdots Q_2 Q_1)A$
- $Q = Q_n \cdots Q_2 Q_1$

Gram-Schmidt Process with Shift

- $v_{k+1} = a_{k+1} - (a_{k+1} \cdot e_1)e_1 \cdots (a_{k+1} \cdot e_k)e_k \Longleftarrow e_{k+1} = \frac{u_{\mathbf{k+1}}}{||u_{\mathbf{k+1}}||}$
- $Q = [e_1, e_2, ..., e_k]$
- $R = Q^T A$
- $\mu_k = a_{nn}^k$
- $QR = A_k - \mu I$
- $A_{k+1} = RQ + \mu I$

Numerical Results: Performance Comparisons

https://CNN.com/



CNN Trends

```
/2020/07/03/cnn-underscored/hamilton-dolby-vision-atmos-watching-guide/index.html,
/2020/06/29/entertainment/jennifer-hudson-aretha-franklin-trnd/index.html,0.011837
/2020/07/29/entertainment/machine-gun-kelly-megan-fox-instagram-post-intl-scli/ind
/2020/07/10/entertainment/palm-springs-review/index.html,0.011837871417136525
/2020/07/28/entertainment/emmy-nominations-2020/index.html,0.011426692039646236
feedback,0.01127909203324837
/2020/07/30/entertainment/entertainment-newsletter/index.html,0.011231401549143951
/2020/06/11/entertainment/watchmen-second-look/index.html,0.011230410220265179
/2020/07/28/entertainment/emmy-nominations-analysis/index.html,0.0112302312238486
/2020/07/28/entertainment/ramy-youssef-emmy-nomination-trnd/index.html,0.011228407
/2020/05/06/cnn-underscored/disney-face-mask-outbrain/index.html,0.011146690426331
/2020/07/30/entertainment/netflix-moesha-girlfriends-sister/index.html,0.011134101
```

Figure 3: CNN pages on trend

Hessenberg, shifts

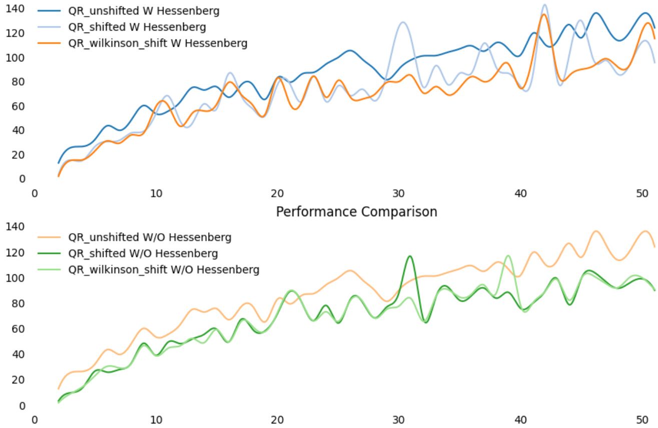


Figure 4: Hessenberg and shift comparison

Concluding remarks

Abel, Galois, and Others

Abel proved that: For any $m \geq 5$, there is a polynomial $p(z)$ of degree m with rational coefficients that has a real root $p(r) = 0$ with that property that r cannot be written using any expression involving rational numbers, addition, substraction, multiplication, division ,and kth roots.

i.e. No formula exists for expressing the roots of an arbitrary polynomials of degree 5 or more.

[2] Trefethen, L. N., Bau, D. (1997). Numerical Linear Algebra. ISBN: 0898713617

Abel, Galois, and Others

Abel proved that: For any $m \geq 5$, there is a polynomial $p(z)$ of degree m with rational coefficients that has a real root $p(r) = 0$ with that property that r cannot be written using any expression involving rational numbers, addition, substraction, multiplication, division ,and kth roots.

i.e. No formula exists for expressing the roots of an arbitrary polynomials of degree 5 or more.

Any eigenvalue solver must be iterative.[2]

[2] Trefethen, L. N., Bau, D. (1997). Numerical Linear Algebra. ISBN: 0898713617

Hilbert Matrix

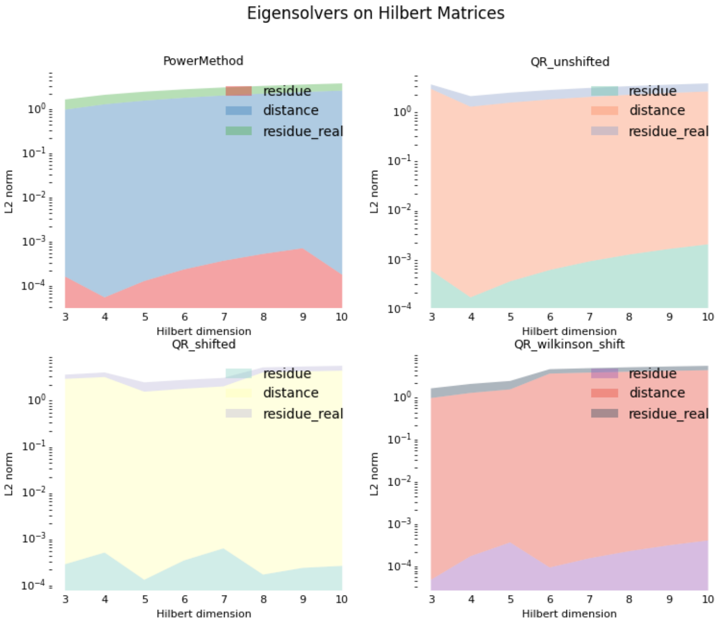
In 1894, David Hilbert introduces a special type of square matrix, with entries being the unit fractions.

$$H_{ij} = \frac{1}{i+j-1}.$$

Hilbert matrix is notoriously ill-conditioned for scientific computation. Specifically, for eigensolvers, we have residue much smaller than actual distance,

$$\|Hx - \lambda x\|_2 << \left\| \frac{x}{\|x\|} - \frac{v}{\|v\|} \right\|_2,$$

where x, λ is the estimated eigenvector- eigenvalue pair, v is the actual eigenvector.


Eigensolvers on Hilbert Matrices
PowerMethod
residue
distance
residue_real
L2 norm
Hilbert dimension
QR_unshifted
residue
distance
residue_real
L2 norm
Hilbert dimension
QR_shifted
residue
distance
residue_real
L2 norm
Hilbert dimension
QR_wilkinson_shift
residue
distance
residue_real
L2 norm
Hilbert dimension

Acknowledgements

Many thanks to:

Dr. Yanlai Chen and Dr. Minah Oh

Justin Baker and Liu Yang

All the ICERM organizers