CS5016: Computational Methods and Applications

Eigen and Singular Value Decomposition

Albert Sunny

Department of Computer Science and Engineering
Indian Institute of Technology Palakkad

28 March, 2024

Eigenvectors and Eigenvalues

A non-zero vector $\boldsymbol{v} \in \mathbb{R}^n$ is an Eigenvector of an $n \times n$ matrix $\boldsymbol{A}$ is it satisfies

$$\boldsymbol{A}\boldsymbol{v} = \lambda \boldsymbol{v}$$

for some $\lambda \in \mathbb{R}$ (eigenvalue). Linear transformation only scale these vectors — a notion of fundamental directions

$$\boldsymbol{A}\boldsymbol{V} = \boldsymbol{A}\begin{bmatrix}\boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots & \boldsymbol{v}_n\end{bmatrix} = \begin{bmatrix}\boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots & \boldsymbol{v}_n\end{bmatrix}\begin{bmatrix}\lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \lambda_n\end{bmatrix} = \boldsymbol{V}\boldsymbol{D}$$

To learn more, visit:
`https://en.wikipedia.org/wiki/Eigenvalues_and_eigenvectors`

A very good video on Eigenvalues and Eigenvectors
`https://www.youtube.com/watch?v=PFDu9oVAE-g`

How are the Eigenvalues and Eigenvectors of $\boldsymbol{A}^{100}$ related to those of $\boldsymbol{A}$?

Power Method

Let us order the Eigen values of $\boldsymbol{A}$ as follows

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n|$$

Let us start with some non-zero vector $\boldsymbol{b}_0 \in \mathbb{R}$ and create a sequence of vectors using the following iteration

$$\boldsymbol{b}_{k+1} = \frac{\boldsymbol{A}\boldsymbol{b}_k}{\|\boldsymbol{A}\boldsymbol{b}_k\|_2}$$

where $\|\cdot\|$ is the L^2 norm. It is also known as the *Euclidean norm*. Then,

$$\boldsymbol{b}_k \xrightarrow{k\to\infty} \boldsymbol{v}_1 \quad \text{and} \quad \frac{\boldsymbol{b}_k^T\boldsymbol{A}\boldsymbol{b}_k}{\boldsymbol{b}_k^T\boldsymbol{b}_k} \xrightarrow{k\to\infty} \lambda_1$$

where $\boldsymbol{v}_1$ is the Eigenvector corresponding to Eigenvalue λ_1.

Why does it work?

$$\boldsymbol{b}_k = \frac{\boldsymbol{A}\boldsymbol{b}_{k-1}}{\|\boldsymbol{A}\boldsymbol{b}_{k-1}\|_2} = \frac{\boldsymbol{A}^2\boldsymbol{b}_{k-2}}{\|\boldsymbol{A}^2\boldsymbol{b}_{k-2}\|_2} = \cdots = \frac{\boldsymbol{z}_k}{\|\boldsymbol{z}_k\|_2} \tag{1}$$

where $\boldsymbol{z}_k = \boldsymbol{A}^k\boldsymbol{b}_0$. Consequently, we have

$$\frac{\boldsymbol{b}_k^T\boldsymbol{A}\boldsymbol{b}_k}{\boldsymbol{b}_k^T\boldsymbol{b}_k} = \frac{\boldsymbol{z}_k^T\boldsymbol{A}\boldsymbol{z}_k}{\boldsymbol{z}_k^T\boldsymbol{z}_k} \tag{2}$$

Since Eigen vectors form a basis, we have $\boldsymbol{b}_0 = \sum_{i=1}^n c_i\boldsymbol{v}_i$. Thus

$$\boldsymbol{z}_k = \boldsymbol{A}^k\boldsymbol{b}_0 = \sum_{i=1}^n c_i\lambda_i^k\boldsymbol{v}_i = \lambda_1^k\left(c_1\boldsymbol{v}_1 + \sum_{i=2}^n c_i\left(\frac{\lambda_i}{\lambda_1}\right)^k\boldsymbol{v}_i\right) \tag{3}$$

Plug in (3) in (2) and (1) and complete the proof.

What would happen if $\boldsymbol{b}_0^T\boldsymbol{v}_1 = 0$?

Simultaneous orthogonalization[1]

Can we simultaneously run power iteration on different initial point?
Let

$$\boldsymbol{B}_0 = \begin{bmatrix} \boldsymbol{b}_0^{(1)} & \boldsymbol{b}_0^{(2)} & \dots & \boldsymbol{b}_0^{(n)} \end{bmatrix} \in \mathbb{R}^{n\times n}$$

Then, we can produce a new set of n vectors by performing the operation $\boldsymbol{A}\boldsymbol{B}_0$. How will $\boldsymbol{A}^k\boldsymbol{B}_0$ look like? Will orthogonalizing the vectors $\boldsymbol{A}\boldsymbol{B}_0$ help?
Finding two matrices $\boldsymbol{Q}_1$ (orthogonal) and $\boldsymbol{R}_1$ (upper triangular) such that

$$\boldsymbol{A}\boldsymbol{B}_0 = \boldsymbol{Q}_1\boldsymbol{R}_1$$

Choose $\boldsymbol{B}_0 = \boldsymbol{Q}_0 = \boldsymbol{I}$ and proceed as follows

$$\begin{aligned} \boldsymbol{A}\boldsymbol{Q}_0 &= \boldsymbol{Q}_1\boldsymbol{R}_1 \\ \boldsymbol{A}\boldsymbol{Q}_1 &= \boldsymbol{Q}_2\boldsymbol{R}_2 \\ \boldsymbol{A}\boldsymbol{Q}_2 &= \boldsymbol{Q}_3\boldsymbol{R}_3 \\ &\vdots \end{aligned}$$

[1] http://madrury.github.io/jekyll/update/statistics/2017/10/04/

If convergence occurs

We have the steady state equation

$$\boldsymbol{AQ} = \boldsymbol{QR} \Rightarrow \boldsymbol{A} \text{ and } \boldsymbol{R} \text{ are similar}$$

Similar matrices have same eigenvalues. What about their eigenvectors?

Eigenvalues of triangular matrices lie on their diagonal.

If $\boldsymbol{R}$ is a diagonal matrix, then the columns of $\boldsymbol{Q}$ are the eigenvectors of $\boldsymbol{A}$.

Not possible to improve convergence and rate of convergence.

The unshifted QR algorithm (John Francis, 1961)

Matrices $\tilde{\boldsymbol{Q}}_i$ (orthogonal) and $\tilde{\boldsymbol{R}}_i$ (upper triangular)

$$\begin{aligned}
\boldsymbol{A} = \tilde{\boldsymbol{A}}_1 &= \tilde{\boldsymbol{Q}}_1 \tilde{\boldsymbol{R}}_1 \\
\tilde{\boldsymbol{R}}_1 \tilde{\boldsymbol{Q}}_1 = \tilde{\boldsymbol{A}}_2 &= \tilde{\boldsymbol{Q}}_2 \tilde{\boldsymbol{R}}_2 \\
\tilde{\boldsymbol{R}}_2 \tilde{\boldsymbol{Q}}_2 = \tilde{\boldsymbol{A}}_3 &= \tilde{\boldsymbol{Q}}_3 \tilde{\boldsymbol{R}}_3 \\
&\vdots \\
\tilde{\boldsymbol{R}}_k \tilde{\boldsymbol{Q}}_k = \tilde{\boldsymbol{A}}_{k+1} &= \tilde{\boldsymbol{Q}}_{k+1} \tilde{\boldsymbol{R}}_{k+1}
\end{aligned}$$

Note that

$$\tilde{\boldsymbol{A}}_k = \tilde{\boldsymbol{R}}_k \tilde{\boldsymbol{Q}}_k = \tilde{\boldsymbol{Q}}_k^T \tilde{\boldsymbol{Q}}_k \tilde{\boldsymbol{R}}_k \tilde{\boldsymbol{Q}}_k = \tilde{\boldsymbol{Q}}_k^T \tilde{\boldsymbol{A}}_{k-1} \tilde{\boldsymbol{Q}}_k = \prod_{i=1}^{k} \tilde{\boldsymbol{Q}}_{k+1-i}^T \boldsymbol{A} \prod_{i=1}^{k} \tilde{\boldsymbol{Q}}_i$$

Matrices $\tilde{\boldsymbol{A}}_k$ (for any $k \geq 1$) and $\boldsymbol{A}$ are similar and share the same eigenvalues.

Does convergence occurs?

Let

$$\boldsymbol{P}_k = \prod_{i=1}^{k} \tilde{\boldsymbol{Q}}_k \qquad \text{— an orthogonal matrix}$$

Then,

$$\tilde{\boldsymbol{A}}_k = \boldsymbol{P}_k^T \boldsymbol{A} \boldsymbol{P}_k = \boldsymbol{P}_k^T \boldsymbol{V} \boldsymbol{D} \boldsymbol{V}^{-1} \boldsymbol{P}_k = \boldsymbol{P}_k^T \hat{\boldsymbol{Q}} \hat{\boldsymbol{R}} \boldsymbol{D} \hat{\boldsymbol{R}}^{-1} \hat{\boldsymbol{Q}}^{-1} \boldsymbol{P}_k$$

Since $\boldsymbol{V}$ is invertible, it can be written as the product of an orthogonal matrix $\hat{\boldsymbol{Q}}$ and upper triangular matrix $\hat{\boldsymbol{R}}$

$$\lim_{k \to \infty} \boldsymbol{P}_k = \hat{\boldsymbol{Q}} \quad \Rightarrow \quad \lim_{k \to \infty} \tilde{\boldsymbol{A}}_k = \hat{\boldsymbol{R}} \boldsymbol{D} \hat{\boldsymbol{R}}^{-1} \quad \text{an upper triangular matrix}$$

Eigenvalues of $\boldsymbol{A}$ would lie on the diagonal of limiting matrix $\tilde{\boldsymbol{A}}_\infty$.

Does convergence occurs?

Let

$$\boldsymbol{U}_k = \prod_{i=1}^{k} \tilde{\boldsymbol{R}}_{k+1-i} \qquad \text{— an upper triangular matrix}$$

Then,

$$\boldsymbol{A}^k = \boldsymbol{P}_k \boldsymbol{U}_k$$

$$\boldsymbol{A}^k = \boldsymbol{V}\boldsymbol{D}^k \boldsymbol{V}^{-1} = \hat{\boldsymbol{Q}}\hat{\boldsymbol{R}}\boldsymbol{D}^k \boldsymbol{V}^{-1} = \hat{\boldsymbol{Q}}\hat{\boldsymbol{R}}\boldsymbol{D}^k \boldsymbol{V}^{-1}\boldsymbol{D}^{-k}\boldsymbol{D}^k$$

But,

$$[\boldsymbol{D}^k \boldsymbol{V}^{-1} \boldsymbol{D}^{-k}]_{ij} = \left(\frac{\lambda_i}{\lambda_j}\right)^k v_{ij}^{-1}$$

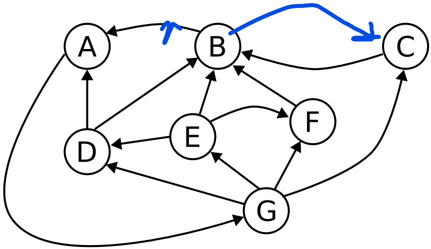
where $v_{ij}^{-1} = [\boldsymbol{V}^{-1}]_{ij}$. If $|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n| > 0$, then $\boldsymbol{D}^k \boldsymbol{V}^{-1} \boldsymbol{D}^{-k}$ tends to an upper triangular matrix. Consequently, $\boldsymbol{P}_k \boldsymbol{U}_k$ tend to the product of an $\tilde{\boldsymbol{Q}}$ (orthogonal) and a triangular matrix.

Performance can be improved by considering the Hessenberg form and using shifts.

PageRank — an application of Eigenvalues

Consider a collection of 7 websites with the following link count matrix

$$\boldsymbol{L} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



Note that $l_{uv} = 1$ if and only if there is a link from v to u.

How would you rank the above website?

distance from most popular node
based on number of incoming nodes

PageRank — an application of Eigenvalues

Let p_u be the rank assigned to node u. Then, we have

$$p_v = \sum_u l_{uv} \frac{p_u}{\sum_w l_{uw}} \Rightarrow \boldsymbol{p} = \tilde{\boldsymbol{L}}\boldsymbol{p}$$

For the above example, we have

stochastic matices

$$\tilde{\boldsymbol{L}} = \begin{bmatrix} 0 & 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1/2 & 1/3 & 1 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 0 & 1/4 \\ 0 & 0 & 0 & 0 & 1/3 & 0 & 1/4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/4 \\ 0 & 0 & 0 & 0 & 1/3 & 0 & 1/4 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad \boldsymbol{p} = \begin{bmatrix} 0.176 \\ 0.249 \\ 0.191 \\ 0.058 \\ 0.044 \\ 0.058 \\ 0.176 \end{bmatrix}$$

What happens to the rank if the link from B to A is moved to C?

For a deeper understanding, look into the theory of Markov chains.

Singular value decomposition

The singular value decomposition of an $m \times n$ matrix $\boldsymbol{M}$ is[2]

$$\boldsymbol{M} = \boldsymbol{U}\Sigma\boldsymbol{V}^T$$

$\boldsymbol{U}$: $m \times m$ orthogonal matrix
Σ: $m \times n$ rectangular diagonal matrix
$\boldsymbol{V}$: $n \times n$ orthogonal matrix

Note that

$$(\boldsymbol{M}\boldsymbol{M}^T)\boldsymbol{U} = \boldsymbol{U}\Sigma\boldsymbol{V}^T\boldsymbol{V}\Sigma^T\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{U}(\Sigma\Sigma^T)$$
$$(\boldsymbol{M}^T\boldsymbol{M})\boldsymbol{V} = \boldsymbol{V}\Sigma^T\boldsymbol{U}^T\boldsymbol{U}\Sigma\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{V}(\Sigma^T\Sigma)$$
$$\boldsymbol{U}^T\boldsymbol{M}\boldsymbol{V} = \boldsymbol{U}^T\boldsymbol{U}\Sigma\boldsymbol{V}^T\boldsymbol{V} = \Sigma$$

Thus
$\boldsymbol{U}$ and $\boldsymbol{V}$: collection of eigenvectors of $\boldsymbol{M}\boldsymbol{M}^T$ and $\boldsymbol{M}^T\boldsymbol{M}$
Σ: square root of non-zero eigenvalues of $\boldsymbol{M}\boldsymbol{M}^T$ or $\boldsymbol{M}^T\boldsymbol{M}$

[2] https://en.wikipedia.org/wiki/Singular_value_decomposition

Low-rank matrix approximation

Approximating $\boldsymbol{M}$ with another matrix $\tilde{\boldsymbol{M}}$ which has a lower rank.

$$\min_{\tilde{\boldsymbol{M}}} \quad \|\boldsymbol{M} - \tilde{\boldsymbol{M}}\|_F \quad \text{such that} \quad rank(\tilde{\boldsymbol{M}}) \leq r$$

where $\|\cdot\|_F = \|\cdot\|_{2,2}$ is the *Frobenius* norm. The solution is

$$\tilde{\boldsymbol{M}} = \boldsymbol{U}\tilde{\boldsymbol{\Sigma}}\boldsymbol{V}^T$$

where $\tilde{\boldsymbol{\Sigma}}$ contains only the r-largest singular values.

The above result is know as the *Eckart–Young–Mirsky* theorem

Python functions

The NumPy module `numpy.linalg` offers methods to find Eigen and singular value decomposition. To know more visit `https://numpy.org/doc/stable/reference/routines.linalg.html`

Thank You