Autovalores e Autovetores

Prof. Afonso Paiva

Departamento de Matemática Aplicada e Estatística
Instituto de Ciências Matemáticas e de Computação
USP – São Carlos

Métodos Numéricos e Computacionais I – SME0305

Introdução

Definição (autovalor e autovetor)

Seja $\mathbf{A}$ *uma matriz em* $\mathcal{M}(n,n)$. *Um escalar* $\lambda \in \mathbb{C}$ *é um* **autovalor** *de* $\mathbf{A}$ *se existir um vetor* $\mathbf{v} \in \mathbb{R}^n$, *com* $\mathbf{v} \neq \bar{\mathbf{0}}$, *tal que:*

$$\mathbf{A}\mathbf{v} = \lambda \mathbf{v}\,.$$

O vetor $\mathbf{v}$ *é chamado de* **autovetor** *associado a* λ.

Introdução

Definição (autovalor e autovetor)

Seja $\mathbf{A}$ *uma matriz em* $\mathcal{M}(n, n)$. *Um escalar* $\lambda \in \mathbb{C}$ *é um* **autovalor** *de* $\mathbf{A}$ *se existir um vetor* $\mathbf{v} \in \mathbb{R}^n$, *com* $\mathbf{v} \neq \bar{\mathbf{0}}$, *tal que:*

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v}\,.$$

O vetor $\mathbf{v}$ *é chamado de* **autovetor** *associado a* λ.

Como calcular λ?

Introdução

Definição (autovalor e autovetor)

Seja $\mathbf{A}$ *uma matriz em* $\mathcal{M}(n,n)$. *Um escalar* $\lambda \in \mathbb{C}$ *é um* **autovalor** *de* $\mathbf{A}$ *se existir um vetor* $\mathbf{v} \in \mathbb{R}^n$, *com* $\mathbf{v} \neq \bar{\mathbf{0}}$, *tal que:*

$$\mathbf{Av} = \lambda \mathbf{v} \,.$$

O vetor $\mathbf{v}$ *é chamado de* **autovetor** *associado a* λ.

Como calcular λ? Através das raízes do **polinômio característico** $P(\lambda)$!

$$\mathbf{Av} = \lambda \mathbf{v} = \lambda \mathbf{Iv} \;\Leftrightarrow\; (\mathbf{A} - \lambda \mathbf{I})\mathbf{v} = \bar{\mathbf{0}} \;\Leftrightarrow\; \underbrace{\det(\mathbf{A} - \lambda \mathbf{I})}_{P(\lambda)} = 0$$

Introdução

Definição (autovalor e autovetor)

Seja $\mathbf{A}$ *uma matriz em* $\mathcal{M}(n,n)$. *Um escalar* $\lambda \in \mathbb{C}$ *é um* **autovalor** *de* $\mathbf{A}$ *se existir um vetor* $\mathbf{v} \in \mathbb{R}^n$, *com* $\mathbf{v} \neq \bar{\mathbf{0}}$, *tal que:*

$$\mathbf{A}\mathbf{v} = \lambda \mathbf{v}\,.$$

O vetor $\mathbf{v}$ *é chamado de* **autovetor** *associado a* λ.

Como calcular λ? Através das raízes do **polinômio característico** $P(\lambda)$!

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v} = \lambda\mathbf{I}\mathbf{v} \;\Leftrightarrow\; (\mathbf{A} - \lambda\mathbf{I})\mathbf{v} = \bar{\mathbf{0}} \;\Leftrightarrow\; \underbrace{\det(\mathbf{A} - \lambda\mathbf{I})}_{P(\lambda)} = 0$$

Definição (espectro)

O **espectro** *de* $\mathbf{A} \in \mathcal{M}(n,n)$ *é o conjunto formado pelo seus autovalores, isto é,* $\Lambda(\mathbf{A}) = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$.

Matriz Ortogonal

Definição (matriz ortogonal)

Uma matriz $\mathbf{A} \in \mathcal{M}(n,n)$ *cujas* n *colunas (e linhas) formam um conjunto ortonormal é dita* **ortogonal**.

Matriz Ortogonal

Definição (matriz ortogonal)

Uma matriz $\mathbf{A} \in \mathcal{M}(n,n)$ *cujas* n *colunas (e linhas) formam um conjunto ortonormal é dita* **ortogonal**.

Proposição

Se $\mathbf{A} \in \mathcal{M}(n,n)$ *é ortogonal então:*

1. $\mathbf{A}^\top = \mathbf{A}^{-1}$;
2. $\|\mathbf{A}\mathbf{v}\|_2 = \|\mathbf{v}\|_2, \forall\, \mathbf{v} \in \mathbb{R}^n$;
3. $\|\mathbf{A}\mathbf{v}\|_2 = \|\mathbf{v}\|_2, \forall\, \mathbf{v} \in \mathbb{R}^n$
4. *Os autovalores de* $\mathbf{A}$ *são* ± 1;
5. $\det(\mathbf{A}) = \pm 1$.

Matriz Ortogonal

1. $\mathbf{A}^\top = \mathbf{A}^{-1}$.

Matriz Ortogonal

1. $\mathbf{A}^\top = \mathbf{A}^{-1}$. Seja $\mathbf{a}_j$ a j-ésima coluna de $\mathbf{A}$. Logo,

$$\mathbf{A}^\top\mathbf{A} = \begin{bmatrix} \mathbf{a}_1^\top \\ \hline \mathbf{a}_2^\top \\ \hline \vdots \\ \hline \mathbf{a}_n^\top \end{bmatrix} \left[\begin{array}{c|c|c|c} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_n \end{array}\right] = \begin{bmatrix} \mathbf{a}_1\cdot\mathbf{a}_1 & \mathbf{a}_1\cdot\mathbf{a}_2 & \cdots & \mathbf{a}_1\cdot\mathbf{a}_n \\ \mathbf{a}_1\cdot\mathbf{a}_2 & \mathbf{a}_2\cdot\mathbf{a}_2 & \cdots & \mathbf{a}_2\cdot\mathbf{a}_n \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{a}_1\cdot\mathbf{a}_n & \mathbf{a}_2\cdot\mathbf{a}_n & \cdots & \mathbf{a}_n\cdot\mathbf{a}_n \end{bmatrix} = \mathbf{I}$$

Matriz Ortogonal

1. $\mathbf{A}^\top = \mathbf{A}^{-1}$. Seja $\mathbf{a}_j$ a j-ésima coluna de $\mathbf{A}$. Logo,

$$\mathbf{A}^\top\mathbf{A} = \begin{bmatrix} \mathbf{a}_1^\top \\ \hline \mathbf{a}_2^\top \\ \hline \vdots \\ \hline \mathbf{a}_n^\top \end{bmatrix} \left[\begin{array}{c|c|c|c} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_n \end{array}\right] = \begin{bmatrix} \mathbf{a}_1\cdot\mathbf{a}_1 & \mathbf{a}_1\cdot\mathbf{a}_2 & \cdots & \mathbf{a}_1\cdot\mathbf{a}_n \\ \mathbf{a}_1\cdot\mathbf{a}_2 & \mathbf{a}_2\cdot\mathbf{a}_2 & \cdots & \mathbf{a}_2\cdot\mathbf{a}_n \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{a}_1\cdot\mathbf{a}_n & \mathbf{a}_2\cdot\mathbf{a}_n & \cdots & \mathbf{a}_n\cdot\mathbf{a}_n \end{bmatrix} = \mathbf{I}$$

2. $\|\mathbf{A}\mathbf{v}\|_2 = \|\mathbf{v}\|_2, \forall\, \mathbf{v} \in \mathbb{R}^n$

Matriz Ortogonal

1. $\mathbf{A}^\top = \mathbf{A}^{-1}$. Seja $\mathbf{a}_j$ a j-ésima coluna de $\mathbf{A}$. Logo,

$$\mathbf{A}^\top\mathbf{A} = \left[\begin{array}{c} \mathbf{a}_1^\top \\ \hline \mathbf{a}_2^\top \\ \hline \vdots \\ \hline \mathbf{a}_n^\top \end{array}\right] \left[\begin{array}{c|c|c|c} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_n \end{array}\right] = \begin{bmatrix} \mathbf{a}_1\cdot\mathbf{a}_1 & \mathbf{a}_1\cdot\mathbf{a}_2 & \cdots & \mathbf{a}_1\cdot\mathbf{a}_n \\ \mathbf{a}_1\cdot\mathbf{a}_2 & \mathbf{a}_2\cdot\mathbf{a}_2 & \cdots & \mathbf{a}_2\cdot\mathbf{a}_n \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{a}_1\cdot\mathbf{a}_n & \mathbf{a}_2\cdot\mathbf{a}_n & \cdots & \mathbf{a}_n\cdot\mathbf{a}_n \end{bmatrix} = \mathbf{I}$$

2. $\|\mathbf{A}\mathbf{v}\|_2 = \|\mathbf{v}\|_2, \forall\, \mathbf{v} \in \mathbb{R}^n$

$$\|\mathbf{A}\mathbf{v}\|_2^2 = \mathbf{A}\mathbf{v}\cdot\mathbf{A}\mathbf{v} = \mathbf{v}\cdot\mathbf{A}^\top\mathbf{A}\mathbf{v} = \mathbf{v}\cdot\mathbf{v} = \|\mathbf{v}\|_2^2$$

Matriz Ortogonal

1. $\mathbf{A}^\top = \mathbf{A}^{-1}$. Seja $\mathbf{a}_j$ a j-ésima coluna de $\mathbf{A}$. Logo,

$$\mathbf{A}^\top\mathbf{A} = \left[\begin{array}{c} \mathbf{a}_1^\top \\ \hline \mathbf{a}_2^\top \\ \hline \vdots \\ \hline \mathbf{a}_n^\top \end{array}\right] \left[\begin{array}{c|c|c|c} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_n \end{array}\right] = \begin{bmatrix} \mathbf{a}_1 \cdot \mathbf{a}_1 & \mathbf{a}_1 \cdot \mathbf{a}_2 & \cdots & \mathbf{a}_1 \cdot \mathbf{a}_n \\ \mathbf{a}_1 \cdot \mathbf{a}_2 & \mathbf{a}_2 \cdot \mathbf{a}_2 & \cdots & \mathbf{a}_2 \cdot \mathbf{a}_n \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{a}_1 \cdot \mathbf{a}_n & \mathbf{a}_2 \cdot \mathbf{a}_n & \cdots & \mathbf{a}_n \cdot \mathbf{a}_n \end{bmatrix} = \mathbf{I}$$

2. $\|\mathbf{Av}\|_2 = \|\mathbf{v}\|_2, \forall\, \mathbf{v} \in \mathbb{R}^n$

$$\|\mathbf{Av}\|_2^2 = \mathbf{Av} \cdot \mathbf{Av} = \mathbf{v} \cdot \mathbf{A}^\top\mathbf{Av} = \mathbf{v} \cdot \mathbf{v} = \|\mathbf{v}\|_2^2$$

3. $\mathbf{Av} = \lambda\mathbf{v}$

Matriz Ortogonal

1. $\mathbf{A}^\top = \mathbf{A}^{-1}$. Seja $\mathbf{a}_j$ a j-ésima coluna de $\mathbf{A}$. Logo,

$$\mathbf{A}^\top\mathbf{A} = \left[\begin{array}{c} \mathbf{a}_1^\top \\ \hline \mathbf{a}_2^\top \\ \hline \vdots \\ \hline \mathbf{a}_n^\top \end{array}\right] \left[\begin{array}{c|c|c|c} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_n \end{array}\right] = \begin{bmatrix} \mathbf{a}_1 \cdot \mathbf{a}_1 & \mathbf{a}_1 \cdot \mathbf{a}_2 & \cdots & \mathbf{a}_1 \cdot \mathbf{a}_n \\ \mathbf{a}_1 \cdot \mathbf{a}_2 & \mathbf{a}_2 \cdot \mathbf{a}_2 & \cdots & \mathbf{a}_2 \cdot \mathbf{a}_n \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{a}_1 \cdot \mathbf{a}_n & \mathbf{a}_2 \cdot \mathbf{a}_n & \cdots & \mathbf{a}_n \cdot \mathbf{a}_n \end{bmatrix} = \mathbf{I}$$

2. $\|\mathbf{A}\mathbf{v}\|_2 = \|\mathbf{v}\|_2, \forall\, \mathbf{v} \in \mathbb{R}^n$

$$\|\mathbf{A}\mathbf{v}\|_2^2 = \mathbf{A}\mathbf{v} \cdot \mathbf{A}\mathbf{v} = \mathbf{v} \cdot \mathbf{A}^\top\mathbf{A}\mathbf{v} = \mathbf{v} \cdot \mathbf{v} = \|\mathbf{v}\|_2^2$$

3. $\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$

$$\|\mathbf{v}\|_2 = \|\mathbf{A}\mathbf{v}\|_2 = \|\lambda\mathbf{v}\|_2 = |\lambda|\|\mathbf{v}\|_2 \;\Rightarrow\; \lambda = \pm 1$$

Matrizes Semelhantes

Definição (matrizes semelhantes)

As matrizes $\mathbf{A}, \mathbf{B} \in \mathcal{M}(n,n)$ *são* **semelhantes** *se existir* $\mathbf{P} \in \mathcal{M}(n,n)$ *invertível, tal que:*

$$\mathbf{B} = \mathbf{P}^{-1}\mathbf{A}\mathbf{P}\,.$$

Matrizes Semelhantes

Definição (matrizes semelhantes)

As matrizes $\mathbf{A}, \mathbf{B} \in \mathcal{M}(n,n)$ *são* **semelhantes** *se existir* $\mathbf{P} \in \mathcal{M}(n,n)$ *invertível, tal que:*

$$\mathbf{B} = \mathbf{P}^{-1}\mathbf{A}\mathbf{P}\,.$$

Proposição

Se $\mathbf{A}$ *e* $\mathbf{B}$ *são semelhantes então elas possuem os mesmos autovalores.*

Decomposição Espectral

Toda matriz simétrica $\mathbf{A} \in \mathcal{M}(n,n)$ pode ser **diagonalizada** por uma matriz **ortogonal** $\mathbf{V} \in \mathcal{M}(n,n)$, isto é:

$$\mathbf{D} = \mathbf{V}^\top \mathbf{A} \mathbf{V}$$

onde $\mathbf{D}$ é uma matriz diagonal formada pelos autovalores (todos reais) de $\mathbf{A}$ e as colunas de $\mathbf{V}$ são os seus respectivos autovetores. Logo,

$$\boxed{\mathbf{A} = \mathbf{V}\mathbf{D}\mathbf{V}^\top,}$$

Projeção Ortogonal

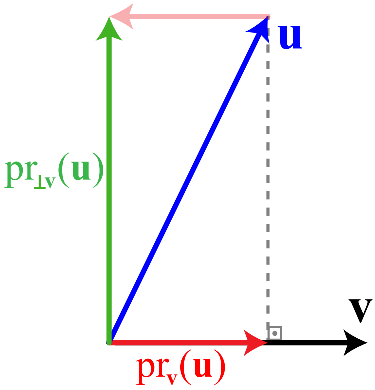
$$\boxed{\mathrm{pr}_{\mathbf{v}}(\mathbf{u}) = \frac{\mathbf{u}\cdot\mathbf{v}}{\mathbf{v}\cdot\mathbf{v}}\,\mathbf{v} = \frac{\mathbf{u}\cdot\mathbf{v}}{\|\mathbf{v}\|_2^2}\,\mathbf{v}}$$

$$\|\mathbf{v}\|_2 = 1 \;\Rightarrow\; \mathrm{pr}_{\mathbf{v}} = \mathbf{v}\mathbf{v}^\top$$

Projeção no Complemento Ortogonal

$$\boxed{\mathrm{pr}_{\perp\mathbf{v}}(\mathbf{u}) = \mathbf{u} - \mathrm{pr}_{\mathbf{v}}(\mathbf{u})}$$

$$\|\mathbf{v}\|_2 = 1 \;\Rightarrow\; \mathrm{pr}_{\perp\mathbf{v}} = \mathbf{I} - \mathbf{v}\mathbf{v}^\top$$



Processo de Ortogonalização de Gram-Schmidt

Entrada: dado um conjunto L.I. $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\} \subset \mathbb{R}^m$, com $m \geq n$.

Saída: um conjunto **ortornormal** $\{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n\}$.

Processo de Ortogonalização de Gram-Schmidt

Entrada: dado um conjunto L.I. $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\} \subset \mathbb{R}^m$, com $m \geq n$.

Saída: um conjunto **ortornormal** $\{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n\}$.

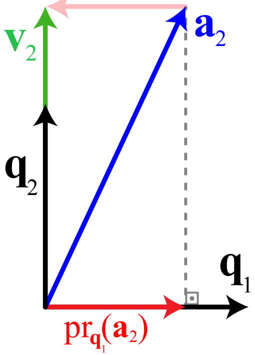
- $\mathbf{v}_1 = \mathbf{a}_1$
- $\mathbf{q}_1 = \mathbf{v}_1 / \|\mathbf{v}_1\|_2$

Processo de Ortogonalização de Gram-Schmidt

Entrada: dado um conjunto L.I. $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\} \subset \mathbb{R}^m$, com $m \geq n$.

Saída: um conjunto **ortornormal** $\{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n\}$.

- $\mathbf{v}_1 = \mathbf{a}_1$
- $\mathbf{q}_1 = {}^{\mathbf{v}_1}/\|\mathbf{v}_1\|_2$
- $\mathbf{v}_2 = \mathbf{a}_2 - \underbrace{(\mathbf{q}_1 \cdot \mathbf{a}_2)\,\mathbf{q}_1}_{\mathrm{pr}_{\mathbf{q}_1}(\mathbf{a}_2)}$
- $\mathbf{q}_2 = {}^{\mathbf{v}_2}/\|\mathbf{v}_2\|_2$

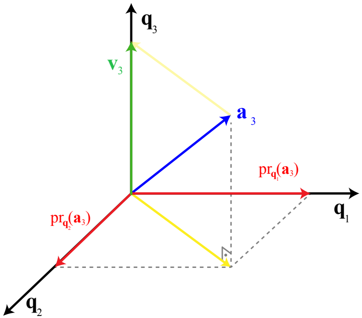


Processo de Ortogonalização de Gram-Schmidt

Entrada: dado um conjunto L.I. $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\} \subset \mathbb{R}^m$, com $m \geq n$.

Saída: um conjunto **ortornormal** $\{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n\}$.

- $\mathbf{v}_1 = \mathbf{a}_1$
- $\mathbf{q}_1 = \mathbf{v}_1 / \|\mathbf{v}_1\|_2$
- $\mathbf{v}_2 = \mathbf{a}_2 - \underbrace{(\mathbf{q}_1 \cdot \mathbf{a}_2)\,\mathbf{q}_1}_{\mathrm{pr}_{\mathbf{q}_1}(\mathbf{a}_2)}$
- $\mathbf{q}_2 = \mathbf{v}_2 / \|\mathbf{v}_2\|_2$
- $\mathbf{v}_3 = \mathbf{a}_3 - \left[\underbrace{(\mathbf{q}_1 \cdot \mathbf{a}_3)\,\mathbf{q}_1}_{\mathrm{pr}_{\mathbf{q}_1}(\mathbf{a}_3)} + \underbrace{(\mathbf{q}_2 \cdot \mathbf{a}_3)\,\mathbf{q}_2}_{\mathrm{pr}_{\mathbf{q}_2}(\mathbf{a}_3)}\right]$
- $\mathbf{q}_3 = \mathbf{v}_3 / \|\mathbf{v}_3\|_2$



Processo de Ortogonalização de Gram-Schmidt

Termo Geral

Para obter $\mathbf{v}_j$ **ortogonal** a $\mathbf{q}_1, \ldots, \mathbf{q}_{j-1}$:

$$\boxed{\mathbf{v}_j = \mathbf{a}_j - \sum_{i=1}^{j-1} (\mathbf{q}_i \cdot \mathbf{a}_j)\, \mathbf{q}_i \,.}$$

Depois **normalizar** $\mathbf{v}_j$:

$$\boxed{\mathbf{q}_j = \frac{\mathbf{v}_j}{\|\mathbf{v}_j\|_2} \,.}$$

Decomposição QR
Completa

A **Decomposição QR completa** de $\mathbf{A} \in \mathcal{M}(m,n)$ é dada por:

$$\boxed{\mathbf{A} = \mathbf{Q} \cdot \mathbf{R}\,,}$$

onde $\mathbf{Q} \in \mathcal{M}(m,m)$ é **ortogonal** e $\mathbf{R} \in \mathcal{M}(m,n)$ é **triangular superior**.

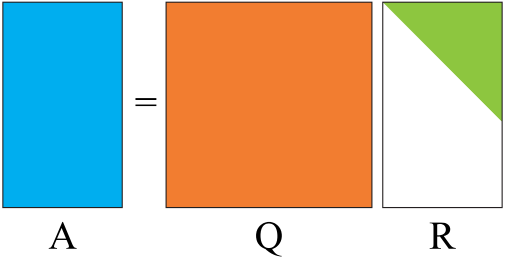
Decomposição QR

Completa

A **Decomposição QR completa** de $\mathbf{A} \in \mathcal{M}(m,n)$ é dada por:

$$\boxed{\mathbf{A} = \mathbf{Q} \cdot \mathbf{R}\,,}$$

onde $\mathbf{Q} \in \mathcal{M}(m,m)$ é **ortogonal** e $\mathbf{R} \in \mathcal{M}(m,n)$ é **triangular superior**.



Decomposição QR

Reduzida

Seja $\mathbf{A} \in \mathcal{M}(m,n)$, com $m \geq n$. Uma representação mais compacta é dada pela **Decomposição QR reduzida** de $\mathbf{A}$:

$$\boxed{\mathbf{A} = \hat{\mathbf{Q}} \cdot \hat{\mathbf{R}}\,,}$$

onde $\hat{\mathbf{Q}} \in \mathcal{M}(m,n)$ e $\hat{\mathbf{R}} \in \mathcal{M}(n,n)$.

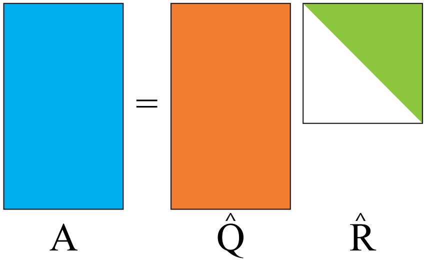
Decomposição QR

Reduzida

Seja $\mathbf{A} \in \mathcal{M}(m,n)$, com $m \geq n$. Uma representação mais compacta é dada pela **Decomposição QR reduzida** de $\mathbf{A}$:

$$\boxed{\mathbf{A} = \hat{\mathbf{Q}} \cdot \hat{\mathbf{R}}\,,}$$

onde $\hat{\mathbf{Q}} \in \mathcal{M}(m,n)$ e $\hat{\mathbf{R}} \in \mathcal{M}(n,n)$.



Decomposição QR

Reduzida

$$\underbrace{\left[\begin{array}{c|c|c|c} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_n \end{array}\right]}_{\mathbf{A}} = \underbrace{\left[\begin{array}{c|c|c|c} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_n \end{array}\right]}_{\hat{\mathbf{Q}}} \underbrace{\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ & r_{22} & \cdots & r_{2n} \\ & & \ddots & \vdots \\ & & & r_{nn} \end{bmatrix}}_{\hat{\mathbf{R}}}$$

- $\mathbf{a}_1 = r_{11}\,\mathbf{q}_1$
- $\mathbf{a}_2 = r_{12}\,\mathbf{q}_1 + r_{22}\,\mathbf{q}_2$
- $\quad\vdots$
- $\mathbf{a}_n = r_{1n}\,\mathbf{q}_1 + r_{2n}\,\mathbf{q}_2 + \cdots + r_{nn}\,\mathbf{q}_n$

Precisamos determinar r_{ij} e os vetores coluna $\mathbf{q}_j$.

Decomposição QR

Reduzida

Usando o Processo de Gram-Schmidt nos vetores coluna $\{\mathbf{a}_1, \ldots, \mathbf{a}_n\}$, obtemos $\mathbf{A} = \hat{\mathbf{Q}} \cdot \hat{\mathbf{R}}$ com

Decomposição QR

Reduzida

Usando o Processo de Gram-Schmidt nos vetores coluna $\{\mathbf{a}_1, \ldots, \mathbf{a}_n\}$, obtemos $\mathbf{A} = \hat{\mathbf{Q}} \cdot \hat{\mathbf{R}}$ com

$$\boxed{r_{ij} = \mathbf{q}_i \cdot \mathbf{a}_j \quad (i \neq j)}$$

$$\boxed{|r_{jj}| = \|\mathbf{v}_j\|_2}$$

Decomposição QR

Reduzida

Usando o Processo de Gram-Schmidt nos vetores coluna $\{\mathbf{a}_1, \ldots, \mathbf{a}_n\}$, obtemos $\mathbf{A} = \hat{\mathbf{Q}} \cdot \hat{\mathbf{R}}$ com

$$\boxed{r_{ij} = \mathbf{q}_i \cdot \mathbf{a}_j \quad (i \neq j)}$$

$$\boxed{|r_{jj}| = \|\mathbf{v}_j\|_2}$$

Teorema

Toda matriz $\mathbf{A} \in \mathcal{M}(m, n)$ (com $m \geq n$) possui Decomposição QR completa e reduzida. Além disso, se $\mathbf{A}$ tem posto completo então $\mathbf{A} = \hat{\mathbf{Q}} \cdot \hat{\mathbf{R}}$ é única com $r_{jj} > 0$.

MATLAB – Decomposição QR

Gram-Schmidt Clássico

```
function [Q,R] = clgs(A)

[m,n] = size(A);
Q = zeros(m,n);
R = zeros(n,n);

for j=1:n
  V = A(:,j);
  for i=1:j-1
    R(i,j) = Q(:,i)'*A(:,j);
    V = V - R(i,j)*Q(:,i);
  end
  R(j,j) = norm(V);
  Q(:,j) = V/R(j,j);
end
```

Decomposição QR

Gram-Schmidt Clássico

Exemplo 1

Calcule a Decomposição QR de

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 2 \\ 1 & 0 & 1 \\ 0 & 1 & 3 \end{bmatrix}.$$

Verifique o erro da decomposição $\|\mathbf{A} - \mathbf{QR}\|_F$ e o erro de ortogonalidade $\|\mathbf{I} - \mathbf{Q}^\top \mathbf{Q}\|_F$.

Decomposição QR

Gram-Schmidt Clássico

Exemplo 2

Refaça o exemplo anterior com

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \\ \varepsilon & 0 & 0 \\ 0 & \varepsilon & 0 \\ 0 & 0 & \varepsilon \end{bmatrix}.$$

Decomposição QR

Gram-Schmidt Clássico

Exemplo 2

Refaça o exemplo anterior com

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \\ \varepsilon & 0 & 0 \\ 0 & \varepsilon & 0 \\ 0 & 0 & \varepsilon \end{bmatrix}.$$

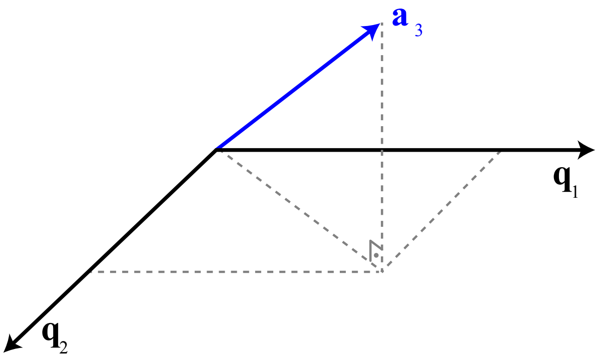
Gram-Schmidt clássico é numericamente instável!!!

Decomposição QR
Gram-Schmidt Modificado

- pequenas modificações no Gram-Schmidt clássico;
- numericamente estável (menos sensível a erros de arredondamento);
- $\mathrm{pr}_{\perp \mathbf{q}_i}$ é aplicado a todos $\mathbf{v}_j$ assim que $\mathbf{q}_i$ é determinado.

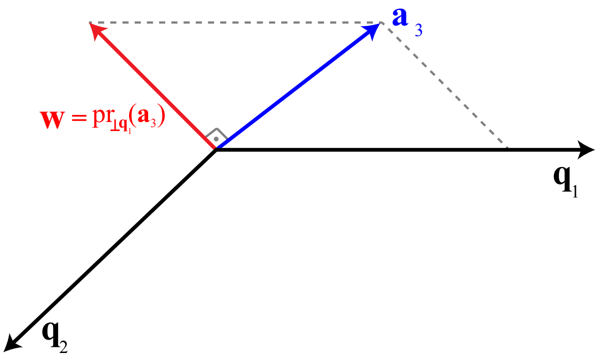
Decomposição QR

Gram-Schmidt Modificado



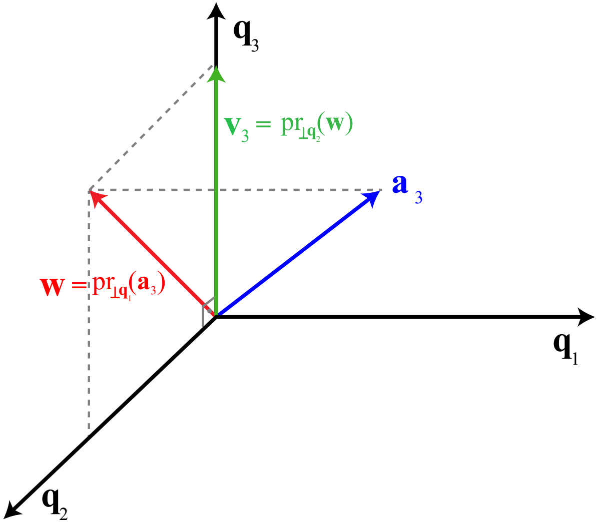
Decomposição QR

Gram-Schmidt Modificado



Decomposição QR

Gram-Schmidt Modificado



Decomposição QR

Gram-Schmidt Modificado

Clássico

- $\mathbf{v}_j \leftarrow \mathbf{a}_j$
- $\mathbf{v}_j \leftarrow \mathbf{v}_j - \left(\mathbf{q}_1 \cdot \mathbf{a}_j\right) \mathbf{q}_1$
- $\mathbf{v}_j \leftarrow \mathbf{v}_j - \left(\mathbf{q}_2 \cdot \mathbf{a}_j\right) \mathbf{q}_2$
- $\vdots$
- $\mathbf{v}_j \leftarrow \mathbf{v}_j - \left(\mathbf{q}_{j-1} \cdot \mathbf{a}_j\right) \mathbf{q}_{j-1}$
- $\mathbf{q}_j \leftarrow \mathbf{v}_j / \|\mathbf{v}_j\|_2$

Modificado

- $\mathbf{v}_j \leftarrow \mathbf{a}_j$
- $\mathbf{v}_j \leftarrow \mathbf{v}_j - \left(\mathbf{q}_1 \cdot \mathbf{v}_j\right) \mathbf{q}_1$
- $\mathbf{v}_j \leftarrow \mathbf{v}_j - \left(\mathbf{q}_2 \cdot \mathbf{v}_j\right) \mathbf{q}_2$
- $\vdots$
- $\mathbf{v}_j \leftarrow \mathbf{v}_j - \left(\mathbf{q}_{j-1} \cdot \mathbf{v}_j\right) \mathbf{q}_{j-1}$
- $\mathbf{q}_j \leftarrow \mathbf{v}_j / \|\mathbf{v}_j\|_2$

Decomposição QR

Gram-Schmidt Modificado

Clássico

- $\mathbf{v}_j \leftarrow \mathbf{a}_j$
- $\mathbf{v}_j \leftarrow \mathbf{v}_j - (\mathbf{q}_1 \cdot \mathbf{a}_j)\, \mathbf{q}_1$
- $\mathbf{v}_j \leftarrow \mathbf{v}_j - (\mathbf{q}_2 \cdot \mathbf{a}_j)\, \mathbf{q}_2$
- $\vdots$
- $\mathbf{v}_j \leftarrow \mathbf{v}_j - (\mathbf{q}_{j-1} \cdot \mathbf{a}_j)\, \mathbf{q}_{j-1}$
- $\mathbf{q}_j \leftarrow \mathbf{v}_j / \|\mathbf{v}_j\|_2$

Modificado

- $\mathbf{v}_j \leftarrow \mathbf{a}_j$
- $\mathbf{v}_j \leftarrow \mathbf{v}_j - (\mathbf{q}_1 \cdot \mathbf{v}_j)\, \mathbf{q}_1$
- $\mathbf{v}_j \leftarrow \mathbf{v}_j - (\mathbf{q}_2 \cdot \mathbf{v}_j)\, \mathbf{q}_2$
- $\vdots$
- $\mathbf{v}_j \leftarrow \mathbf{v}_j - (\mathbf{q}_{j-1} \cdot \mathbf{v}_j)\, \mathbf{q}_{j-1}$
- $\mathbf{q}_j \leftarrow \mathbf{v}_j / \|\mathbf{v}_j\|_2$

Complexidade: $2mn^2$ flops.

MATLAB – Decomposição QR

Gram-Schmidt Modificado

```
function [Q,R] = mgs(A)

[m,n] = size(A);
V = A;
Q = zeros(m,n);
R = zeros(n,n);

for i=1:n
  R(i,i) = norm(V(:,i));
  Q(:,i) = V(:,i)/R(i,i);
  for j=i+1:n
    R(i,j) = Q(:,i)'*V(:,j);
    V(:,j) = V(:,j)- R(i,j)*Q(:,i);
  end
end
```

Método de Francis

Seja $\mathbf{A} \in \mathcal{M}(n,n)$ **simétrica**, vamos usar a Decomposição QR para calcular **todos** os seus autovalores e autovetores.

Método de Francis

Seja $\mathbf{A} \in \mathcal{M}(n,n)$ **simétrica**, vamos usar a Decomposição QR para calcular **todos** os seus autovalores e autovetores.

Processo:

- $\mathbf{A}_1 = \mathbf{A} \;\Rightarrow\; \mathbf{A}_1 = \mathbf{Q}_1\mathbf{R}_1$

Método de Francis

Seja $\mathbf{A} \in \mathcal{M}(n, n)$ **simétrica**, vamos usar a Decomposição QR para calcular **todos** os seus autovalores e autovetores.

Processo:

- $\mathbf{A}_1 = \mathbf{A} \;\Rightarrow\; \mathbf{A}_1 = \mathbf{Q}_1\mathbf{R}_1$
- $\mathbf{A}_2 = \mathbf{R}_1\mathbf{Q}_1 \;\Rightarrow\; \mathbf{A}_2 = \mathbf{Q}_2\mathbf{R}_2$

Método de Francis

Seja $\mathbf{A} \in \mathcal{M}(n,n)$ **simétrica**, vamos usar a Decomposição QR para calcular **todos** os seus autovalores e autovetores.

Processo:

- $\mathbf{A}_1 = \mathbf{A} \;\Rightarrow\; \mathbf{A}_1 = \mathbf{Q}_1\mathbf{R}_1$
- $\mathbf{A}_2 = \mathbf{R}_1\mathbf{Q}_1 \;\Rightarrow\; \mathbf{A}_2 = \mathbf{Q}_2\mathbf{R}_2$
- $\mathbf{A}_3 = \mathbf{R}_2\mathbf{Q}_2 \;\Rightarrow\; \mathbf{A}_3 = \mathbf{Q}_3\mathbf{R}_3$

Método de Francis

Seja $\mathbf{A} \in \mathcal{M}(n,n)$ **simétrica**, vamos usar a Decomposição QR para calcular **todos** os seus autovalores e autovetores.

Processo:

- $\mathbf{A}_1 = \mathbf{A} \;\Rightarrow\; \mathbf{A}_1 = \mathbf{Q}_1\mathbf{R}_1$
- $\mathbf{A}_2 = \mathbf{R}_1\mathbf{Q}_1 \;\Rightarrow\; \mathbf{A}_2 = \mathbf{Q}_2\mathbf{R}_2$
- $\mathbf{A}_3 = \mathbf{R}_2\mathbf{Q}_2 \;\Rightarrow\; \mathbf{A}_3 = \mathbf{Q}_3\mathbf{R}_3$
- $\vdots$
- $\mathbf{A}_{k-1} = \mathbf{R}_{k-2}\mathbf{Q}_{k-2} \;\Rightarrow\; \mathbf{A}_{k-1} = \mathbf{Q}_{k-1}\mathbf{R}_{k-1}$
- $\mathbf{A}_k = \mathbf{R}_{k-1}\mathbf{Q}_{k-1}$

Método de Francis

Proposição

As matrizes $\mathbf{A}$ *e* $\mathbf{A}_k$ *possuem os mesmos autovalores.*

Método de Francis

Proposição

As matrizes $\mathbf{A}$ *e* $\mathbf{A}_k$ *possuem os mesmos autovalores.*

Como $\mathbf{A}_{k-1} = \mathbf{Q}_{k-1}\mathbf{R}_{k-1} \Rightarrow \mathbf{R}_{k-1} = \mathbf{Q}_{k-1}^{\top}\mathbf{A}_{k-1}$ e $\mathbf{A}_k = \mathbf{R}_{k-1}\mathbf{Q}_{k-1}$.

Método de Francis

Proposição

As matrizes $\mathbf{A}$ *e* $\mathbf{A}_k$ *possuem os mesmos autovalores.*

Como $\mathbf{A}_{k-1} = \mathbf{Q}_{k-1}\mathbf{R}_{k-1} \Rightarrow \mathbf{R}_{k-1} = \mathbf{Q}_{k-1}^{\top}\mathbf{A}_{k-1}$ e $\mathbf{A}_k = \mathbf{R}_{k-1}\mathbf{Q}_{k-1}$.

Temos que, $\mathbf{A}_k = \mathbf{Q}_{k-1}^{\top}\mathbf{A}_{k-1}\mathbf{Q}_{k-1}$.

Método de Francis

Proposição

As matrizes $\mathbf{A}$ *e* $\mathbf{A}_k$ *possuem os mesmos autovalores.*

Como $\mathbf{A}_{k-1} = \mathbf{Q}_{k-1}\mathbf{R}_{k-1} \Rightarrow \mathbf{R}_{k-1} = \mathbf{Q}_{k-1}^{\top}\mathbf{A}_{k-1}$ e $\mathbf{A}_k = \mathbf{R}_{k-1}\mathbf{Q}_{k-1}$.

Temos que, $\mathbf{A}_k = \mathbf{Q}_{k-1}^{\top}\mathbf{A}_{k-1}\mathbf{Q}_{k-1}$. Assim,

$$
\begin{aligned}
\mathbf{A}_k &= \mathbf{Q}_{k-1}^{\top}\,\mathbf{A}_{k-1}\,\mathbf{Q}_{k-1} \\
&= \mathbf{Q}_{k-1}^{\top}\mathbf{Q}_{k-2}^{\top}\,\mathbf{A}_{k-2}\,\mathbf{Q}_{k-2}\mathbf{Q}_{k-1} \\
&\ \vdots \\
&= \mathbf{Q}_{k-1}^{\top}\mathbf{Q}_{k-2}^{\top}\cdots\mathbf{Q}_{1}^{\top}\,\mathbf{A}_1\,\mathbf{Q}_1\cdots\mathbf{Q}_{k-2}\mathbf{Q}_{k-1} \\
&= \left(\mathbf{Q}_1\cdots\mathbf{Q}_{k-2}\mathbf{Q}_{k-1}\right)^{\top}\mathbf{A}\left(\mathbf{Q}_1\cdots\mathbf{Q}_{k-2}\mathbf{Q}_{k-1}\right)
\end{aligned}
$$

Método de Francis

Proposição

As matrizes $\mathbf{A}$ e $\mathbf{A}_k$ possuem os mesmos autovalores.

Como $\mathbf{A}_{k-1} = \mathbf{Q}_{k-1}\mathbf{R}_{k-1} \Rightarrow \mathbf{R}_{k-1} = \mathbf{Q}_{k-1}^{\top}\mathbf{A}_{k-1}$ e $\mathbf{A}_k = \mathbf{R}_{k-1}\mathbf{Q}_{k-1}$.

Temos que, $\mathbf{A}_k = \mathbf{Q}_{k-1}^{\top}\mathbf{A}_{k-1}\mathbf{Q}_{k-1}$. Assim,

$$\begin{aligned}
\mathbf{A}_k &= \mathbf{Q}_{k-1}^{\top}\,\mathbf{A}_{k-1}\,\mathbf{Q}_{k-1} \\
&= \mathbf{Q}_{k-1}^{\top}\mathbf{Q}_{k-2}^{\top}\,\mathbf{A}_{k-2}\,\mathbf{Q}_{k-2}\mathbf{Q}_{k-1} \\
&\;\;\vdots \\
&= \mathbf{Q}_{k-1}^{\top}\mathbf{Q}_{k-2}^{\top}\cdots\mathbf{Q}_{1}^{\top}\,\mathbf{A}_1\,\mathbf{Q}_1\cdots\mathbf{Q}_{k-2}\mathbf{Q}_{k-1} \\
&= \left(\mathbf{Q}_1\cdots\mathbf{Q}_{k-2}\mathbf{Q}_{k-1}\right)^{\top}\,\mathbf{A}\,\left(\mathbf{Q}_1\cdots\mathbf{Q}_{k-2}\mathbf{Q}_{k-1}\right)
\end{aligned}$$

Fazendo $\mathbf{V} = \mathbf{Q}_1\mathbf{Q}_2\cdots\mathbf{Q}_{k-1}$ temos que $\mathbf{A}$ e $\mathbf{A}_k = \mathbf{V}^{\top}\mathbf{A}\mathbf{V}$ são semelhantes.

Método de Francis

Proposição

As matrizes $\mathbf{A}$ *e* $\mathbf{A}_k$ *possuem os mesmos autovalores.*

Método de Francis

Proposição

As matrizes $\mathbf{A}$ *e* $\mathbf{A}_k$ *possuem os mesmos autovalores.*

Proposição

A sequência $\mathbf{A}_k$ *converge para uma matriz diagonal.*

Método de Francis

Proposição

As matrizes $\mathbf{A}$ *e* $\mathbf{A}_k$ *possuem os mesmos autovalores.*

Proposição

A sequência $\mathbf{A}_k$ *converge para uma matriz diagonal.*

Logo, os elementos da **diagonal de $\mathbf{A}_k$** fornecem uma aproximação para os **autovalores** de $\mathbf{A}$, enquanto que as **colunas da matriz**:

$$\boxed{\mathbf{V} = \mathbf{Q}_1 \mathbf{Q}_2 \cdots \mathbf{Q}_{k-1}}$$

uma aproximação dos seus respectivos **autovetores**.

Método de Francis

Critérios de Parada

Dados $\varepsilon > 0$ e $k_{max} \in \mathbb{N}$, temos:

1. $k = k_{max}$

Método de Francis

Critérios de Parada

Dados $\varepsilon > 0$ e $k_{max} \in \mathbb{N}$, temos:

1. $k = k_{max}$
2. $\max_{i<j} \{|a_{ij}|\} < \varepsilon$

Método de Francis

Critérios de Parada

Dados $\varepsilon > 0$ e $k_{max} \in \mathbb{N}$, temos:

1. $k = k_{max}$
2. $\max_{i<j} \{|a_{ij}|\} < \varepsilon$
3. $off(\mathbf{A}) < \varepsilon$, com

$$\boxed{off(\mathbf{A}) = \sqrt{\|\mathbf{A}\|_F^2 - \sum_{i=1}^{n} a_{ii}^2}}$$

MATLAB – Método de Francis

```
function [V,D] = francis(A,tol)

n = size(A,1);
V = eye(n);
erro = inf;

while erro>tol
    [Q,R] = mgs(A);
    A = R*Q;
    V = V*Q;
    erro = max(max(abs(tril(A,-1))));
end

D = diag(A);
```

Decomposição SVD

Completa

A **Decomposição SVD**[1] de $\mathbf{A} \in \mathcal{M}(m,n)$ é dada por:

$$\boxed{\mathbf{A} = \mathbf{U} \cdot \Sigma \cdot \mathbf{V}^\top\,,}$$

onde as matrizes $\mathbf{U} \in \mathcal{M}(m,m)$ é **ortogonal**, $\Sigma \in \mathcal{M}(m,n)$ é **diagonal** e $\mathbf{V} \in \mathcal{M}(n,n)$ é **ortogonal**.

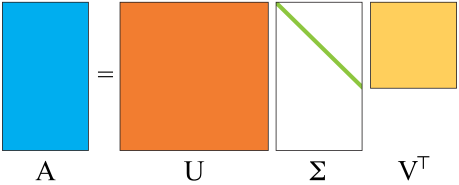
[1] Do inglês, Singular Value Decomposition.

Decomposição SVD
Completa

A **Decomposição SVD**[1] de $\mathbf{A} \in \mathcal{M}(m,n)$ é dada por:

$$\boxed{\mathbf{A} = \mathbf{U} \cdot \Sigma \cdot \mathbf{V}^\top ,}$$

onde as matrizes $\mathbf{U} \in \mathcal{M}(m,m)$ é **ortogonal**, $\Sigma \in \mathcal{M}(m,n)$ é **diagonal** e $\mathbf{V} \in \mathcal{M}(n,n)$ é **ortogonal**.



Os coeficientes σ_i da diagonal de Σ são chamados de **valores singulares** de $\mathbf{A}$.

[1] Do inglês, Singular Value Decomposition.

Decomposição SVD

Reduzida

Seja $\mathbf{A} \in \mathcal{M}(m,n)$, com $m \geq n$. Uma representação mais compacta da Decomposição SVD é dada na forma **reduzida**:

$$\boxed{\mathbf{A} = \hat{\mathbf{U}} \cdot \hat{\Sigma} \cdot \mathbf{V}^{\top}\,,}$$

com $\hat{\mathbf{U}} \in \mathcal{M}(m,n)$, $\hat{\Sigma} \in \mathcal{M}(n,n)$ e $\mathbf{V} \in \mathcal{M}(n,n)$.

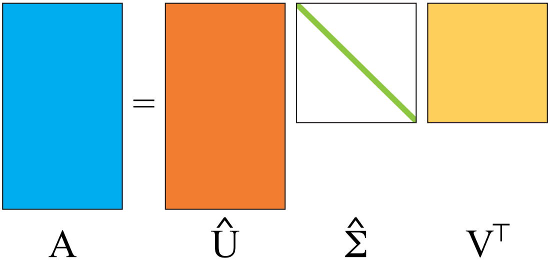
Decomposição SVD

Reduzida

Seja $\mathbf{A} \in \mathcal{M}(m,n)$, com $m \geq n$. Uma representação mais compacta da Decomposição SVD é dada na forma **reduzida**:

$$\mathbf{A} = \hat{\mathbf{U}} \cdot \hat{\Sigma} \cdot \mathbf{V}^\top ,$$

com $\hat{\mathbf{U}} \in \mathcal{M}(m,n)$, $\hat{\Sigma} \in \mathcal{M}(n,n)$ e $\mathbf{V} \in \mathcal{M}(n,n)$.



Decomposição SVD

Teorema (Decomposição SVD)

Toda matriz $\mathbf{A} \in \mathcal{M}(m, n)$ *de posto* $p > 0$ *possui decomposição SVD, isto é,* $\mathbf{A} = \mathbf{U}\,\Sigma\,\mathbf{V}^\top$ *com* $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p > 0$.

[2] Algoritmos mais eficientes para SVD são encontrados no livro "Golub and Van Loan, *Matrix Computations*, 1996".

Decomposição SVD

Teorema (Decomposição SVD)

Toda matriz $\mathbf{A} \in \mathcal{M}(m,n)$ *de posto* $p > 0$ *possui decomposição SVD, isto é,* $\mathbf{A} = \mathbf{U}\,\Sigma\,\mathbf{V}^\top$ *com* $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p > 0$.

Uma forma *ingênua*[2] de calcular a Decomposição SVD de $\mathbf{A}$ é utilizando o **Método de Francis** nas matrizes simétricas $\mathbf{A}^\top\mathbf{A}$ e $\mathbf{A}\mathbf{A}^\top$, pois:

- $\mathbf{A}^\top\mathbf{A} = (\mathbf{U}\Sigma\mathbf{V}^\top)^\top(\mathbf{U}\Sigma\mathbf{V}^\top) = \mathbf{V}\Sigma(\mathbf{U}^\top\mathbf{U})\Sigma\mathbf{V}^\top = \mathbf{V}\,\Sigma^2\,\mathbf{V}^\top$
- $\mathbf{A}\mathbf{A}^\top = (\mathbf{U}\Sigma\mathbf{V}^\top)(\mathbf{U}\Sigma\mathbf{V}^\top)^\top = \mathbf{U}\Sigma(\mathbf{V}^\top\mathbf{V})\Sigma\mathbf{U}^\top = \mathbf{U}\,\Sigma^2\,\mathbf{U}^\top$

[2] Algoritmos mais eficientes para SVD são encontrados no livro "Golub and Van Loan, *Matrix Computations*, 1996".

Decomposição SVD

Teorema (Decomposição SVD)

Toda matriz $\mathbf{A} \in \mathcal{M}(m,n)$ *de posto* $p > 0$ *possui decomposição SVD, isto é,* $\mathbf{A} = \mathbf{U}\,\Sigma\,\mathbf{V}^\top$ *com* $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p > 0$.

Uma forma *ingênua*[2] de calcular a Decomposição SVD de $\mathbf{A}$ é utilizando o **Método de Francis** nas matrizes simétricas $\mathbf{A}^\top\mathbf{A}$ e $\mathbf{A}\mathbf{A}^\top$, pois:

- $\mathbf{A}^\top\mathbf{A} = (\mathbf{U}\Sigma\mathbf{V}^\top)^\top(\mathbf{U}\Sigma\mathbf{V}^\top) = \mathbf{V}\Sigma(\mathbf{U}^\top\mathbf{U})\Sigma\mathbf{V}^\top = \mathbf{V}\,\Sigma^2\,\mathbf{V}^\top$
- $\mathbf{A}\mathbf{A}^\top = (\mathbf{U}\Sigma\mathbf{V}^\top)(\mathbf{U}\Sigma\mathbf{V}^\top)^\top = \mathbf{U}\Sigma(\mathbf{V}^\top\mathbf{V})\Sigma\mathbf{U}^\top = \mathbf{U}\,\Sigma^2\,\mathbf{U}^\top$

Note que a diagonal de Σ^2 é formada pelo quadrado dos valores singulares de $\mathbf{A}$, isto é, σ_i^2.

[2] Algoritmos mais eficientes para SVD são encontrados no livro "Golub and Van Loan, *Matrix Computations*, 1996".

MATLAB – Decomposição SVD

```
function [U,S,V] = my_svd(A)

tol = 1e-5;
[m,n] = size(A);
k = min(m,n);
S = zeros(m,n);

[U,~] = francis(A*A',tol);
[V,D] = francis(A'*A,tol);
S(1:k,1:k) = diag(sqrt(D));
```

Resumo em MATLAB

- `[Q,R] = qr(A)`: decomposição QR completa de $\mathbf{A}$;

- `[V,D] = eig(A)`: calcula os autovalores e autovetores de $\mathbf{A}$;
  `[V,D] = eigs(A,k)`: calcula k autovalores e autovetores de $\mathbf{A}$;

- `[U,S,V] = svd(A)`: decomposição SVD de $\mathbf{A}$;
  `[U,S,V] = svds(A,k)`: SVD com apenas k valores singulares de $\mathbf{A}$;

Aplicação de SVD

Compressão de Imagem

Entrada: uma imagem $\mathcal{I}$ de $m \times n$ pixels (*grayscale*)

- representar $\mathcal{I}$ como $\mathbf{A} \in \mathcal{M}(m,n)\,, 0 \leq a_{ij} \leq 1$;

Aplicação de SVD

Compressão de Imagem

Entrada: uma imagem $\mathcal{I}$ de $m \times n$ pixels (*grayscale*)

- representar $\mathcal{I}$ como $\mathbf{A} \in \mathcal{M}(m,n)$, $0 \leq a_{ij} \leq 1$;
- compressão via SVD: trocar $\mathbf{A}$ por $\mathbf{A}_k$;

Aplicação de SVD

Compressão de Imagem

Entrada: uma imagem $\mathcal{I}$ de $m \times n$ pixels (*grayscale*)

- representar $\mathcal{I}$ como $\mathbf{A} \in \mathcal{M}(m,n)$, $0 \leq a_{ij} \leq 1$;
- compressão via SVD: trocar $\mathbf{A}$ por $\mathbf{A}_k$;
- $\mathbf{A}_k$ possui apenas os k primeiros valores singulares de $\mathbf{A}$;

Aplicação de SVD

Compressão de Imagem

Entrada: uma imagem $\mathcal{I}$ de $m \times n$ pixels (*grayscale*)

- representar $\mathcal{I}$ como $\mathbf{A} \in \mathcal{M}(m,n)$, $0 \leq a_{ij} \leq 1$;
- compressão via SVD: trocar $\mathbf{A}$ por $\mathbf{A}_k$;
- $\mathbf{A}_k$ possui apenas os k primeiros valores singulares de $\mathbf{A}$;
- armazenamos $k(m+n+1)$ números ao invés de mn.

Aplicação de SVD

Compressão de Imagem

Entrada: uma imagem $\mathcal{I}$ de $m \times n$ pixels (*grayscale*)

- representar $\mathcal{I}$ como $\mathbf{A} \in \mathcal{M}(m,n)$, $0 \leq a_{ij} \leq 1$;
- compressão via SVD: trocar $\mathbf{A}$ por $\mathbf{A}_k$;
- $\mathbf{A}_k$ possui apenas os k primeiros valores singulares de $\mathbf{A}$;
- armazenamos $k(m+n+1)$ números ao invés de mn.

imagem original
$k = 374$

Aplicação de SVD

Compressão de Imagem

Entrada: uma imagem $\mathcal{I}$ de $m \times n$ pixels (*grayscale*)

- representar $\mathcal{I}$ como $\mathbf{A} \in \mathcal{M}(m,n)$, $0 \leq a_{ij} \leq 1$;
- compressão via SVD: trocar $\mathbf{A}$ por $\mathbf{A}_k$;
- $\mathbf{A}_k$ possui apenas os k primeiros valores singulares de $\mathbf{A}$;
- armazenamos $k(m+n+1)$ números ao invés de mn.

imagem original
$k = 374$

compress. $\approx 44\%$
$k = 120$

Aplicação de SVD

Compressão de Imagem

Entrada: uma imagem $\mathcal{I}$ de $m \times n$ pixels (*grayscale*)

- representar $\mathcal{I}$ como $\mathbf{A} \in \mathcal{M}(m,n)$, $0 \leq a_{ij} \leq 1$;
- compressão via SVD: trocar $\mathbf{A}$ por $\mathbf{A}_k$;
- $\mathbf{A}_k$ possui apenas os k primeiros valores singulares de $\mathbf{A}$;
- armazenamos $k(m+n+1)$ números ao invés de mn.

imagem original
$k = 374$

compress. $\approx 44\%$
$k = 120$

compress. $\approx 77\%$
$k = 50$

MATLAB – Aplicação de SVD

Compressão de Imagem

```
A = imread('flor.png'); % carrega uma imagem indexada [0,255]
A = im2double(A); % transforma para valores reais em [0,1]
figure, imshow(A); % mostra imagem original

[U,S,V] = svd(A);
k = 120; % numero de valores singulares < min(size(A))
Ak = U(:,1:k)*S(1:k,1:k)*V(:,1:k)'; % imagem comprimida
figure, imshow(Ak);
imwrite(Ak,'flor_svd.png'); % salva imagem
```

Método das Potências

Seja $\mathbf{A} \in \mathcal{M}(n,n)$ diagonalizável, isto é, $\mathbf{A}$ possui autovalores $\lambda_1, \ldots, \lambda_n$ associados a um conjunto de autovetores $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\} \subset \mathbb{R}^n$ L.I. (com $\|\mathbf{v}_i\|_2 = 1\,, \forall i$).

Método das Potências

Seja $\mathbf{A} \in \mathcal{M}(n,n)$ diagonalizável, isto é, $\mathbf{A}$ possui autovalores $\lambda_1, \ldots, \lambda_n$ associados a um conjunto de autovetores $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\} \subset \mathbb{R}^n$ L.I. (com $\|\mathbf{v}_i\|_2 = 1\,, \forall i$).

Além disso, vamos assumir que os autovalores estão ordenados da seguinte forma:

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

Método das Potências

Seja $\mathbf{A} \in \mathcal{M}(n,n)$ diagonalizável, isto é, $\mathbf{A}$ possui autovalores $\lambda_1, \ldots, \lambda_n$ associados a um conjunto de autovetores $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\} \subset \mathbb{R}^n$ L.I. (com $\|\mathbf{v}_i\|_2 = 1\,, \forall i$).

Além disso, vamos assumir que os autovalores estão ordenados da seguinte forma:

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

Objetivo:

- Calcular o autovalor *dominante* λ_1 e o seu autovetor associado $\mathbf{v}_1$.

Método das Potências

Quociente de Rayleigh

$$\mu(\mathbf{x}) = \frac{\mathbf{x} \cdot \mathbf{A}\mathbf{x}}{\|\mathbf{x}\|_2^2}$$

Método das Potências

Quociente de Rayleigh

$$\boxed{\mu(\mathbf{x}) = \frac{\mathbf{x} \cdot \mathbf{A}\mathbf{x}}{\|\mathbf{x}\|_2^2}}$$

O que acontece quando $\mathbf{x}$ é autovetor de $\mathbf{A}$ cujo autovalor é λ?

Método das Potências

Quociente de Rayleigh

$$\boxed{\mu(\mathbf{x}) = \frac{\mathbf{x}\cdot\mathbf{A}\mathbf{x}}{\|\mathbf{x}\|_2^2}}$$

O que acontece quando $\mathbf{x}$ é autovetor de $\mathbf{A}$ cujo autovalor é λ?

$$\mu(\mathbf{x}) = \frac{\mathbf{x}\cdot\mathbf{A}\mathbf{x}}{\|\mathbf{x}\|_2^2} = \mu(\mathbf{x}) = \frac{\mathbf{x}\cdot\lambda\mathbf{x}}{\|\mathbf{x}\|_2^2} = \lambda\frac{\|\mathbf{x}\|_2^2}{\|\mathbf{x}\|_2^2} = \lambda$$

Método das Potências

Quociente de Rayleigh

$$\boxed{\mu(\mathbf{x}) = \frac{\mathbf{x} \cdot \mathbf{A}\mathbf{x}}{\|\mathbf{x}\|_2^2}}$$

O que acontece quando $\mathbf{x}$ é autovetor de $\mathbf{A}$ cujo autovalor é λ?

$$\mu(\mathbf{x}) = \frac{\mathbf{x} \cdot \mathbf{A}\mathbf{x}}{\|\mathbf{x}\|_2^2} = \mu(\mathbf{x}) = \frac{\mathbf{x} \cdot \lambda\mathbf{x}}{\|\mathbf{x}\|_2^2} = \lambda\frac{\|\mathbf{x}\|_2^2}{\|\mathbf{x}\|_2^2} = \lambda$$

$\mu(\mathbf{x})$ fornece o autovalor associado a $\mathbf{x}$!

Método das Potências

Processo Iterativo

Dado um chute inicial $\mathbf{x}^{(0)} \in \mathbb{R}^n$ (não-nulo), calcule $\mathbf{y}^{(0)} = {}^{\mathbf{x}^{(0)}}/_{\|\mathbf{x}^{(0)}\|_2}$.

Método das Potências

Processo Iterativo

Dado um chute inicial $\mathbf{x}^{(0)} \in \mathbb{R}^n$ (não-nulo), calcule $\mathbf{y}^{(0)} = {}^{\mathbf{x}^{(0)}}/_{\|\mathbf{x}^{(0)}\|_2}$.

Para $k = 1, 2, \ldots$ faça:

$$\boxed{\mathbf{x}^{(k)} = \mathbf{A}\mathbf{y}^{(k-1)}\,, \quad \mathbf{y}^{(k)} = \frac{\mathbf{x}^{(k)}}{\|\mathbf{x}^{(k)}\|_2}\,, \quad \lambda_1^{(k)} = \mathbf{y}^{(k)} \cdot \mathbf{A}\mathbf{y}^{(k)}}$$

Método das Potências

Processo Iterativo

Dado um chute inicial $\mathbf{x}^{(0)} \in \mathbb{R}^n$ (não-nulo), calcule $\mathbf{y}^{(0)} = \mathbf{x}^{(0)}/\|\mathbf{x}^{(0)}\|_2$.

Para $k = 1, 2, \ldots$ faça:

$$\mathbf{x}^{(k)} = \mathbf{A}\mathbf{y}^{(k-1)}, \quad \mathbf{y}^{(k)} = \frac{\mathbf{x}^{(k)}}{\|\mathbf{x}^{(k)}\|_2}, \quad \lambda_1^{(k)} = \mathbf{y}^{(k)} \cdot \mathbf{A}\mathbf{y}^{(k)}$$

Note que pela recursão acima temos:

$$\mathbf{y}^{(k)} = \beta^{(k)} \mathbf{A}^k \mathbf{x}^{(0)} \quad \text{com} \quad \beta^{(k)} = \frac{1}{\prod_{i=0}^{k} \|\mathbf{x}^{(i)}\|_2} = \frac{1}{\|\mathbf{A}^k \mathbf{x}^{(0)}\|_2}$$

Método das Potências

Convergência

Afirmação: $\mathbf{y}^{(k)} \to \pm\mathbf{v}_1$

Método das Potências

Convergência

Afirmação: $\mathbf{y}^{(k)} \to \pm\mathbf{v}_1$

$$\mathbf{x}^{(0)} = \sum_{i=1}^{n} \alpha_i \mathbf{v}_i \;\Rightarrow\; \mathbf{y}^{(0)} = \beta^{(0)} \sum_{i=1}^{n} \alpha_i \mathbf{v}_i\,, \quad \text{com} \quad \beta^{(0)} = \frac{1}{\|\mathbf{x}^{(0)}\|_2}$$

$$\mathbf{x}^{(1)} = \mathbf{A}\mathbf{y}^{(0)} = \beta^{(0)}\mathbf{A} \sum_{i=1}^{n} \alpha_i \mathbf{v}_i = \beta^{(0)} \sum_{i=1}^{n} \alpha_i \mathbf{A}\mathbf{v}_i = \beta^{(0)} \sum_{i=1}^{n} \alpha_i \lambda_i \mathbf{v}_i$$

$$\mathbf{y}^{(1)} = \beta^{(1)} \sum_{i=1}^{n} \alpha_i \lambda_i \mathbf{v}_i\,, \quad \text{com} \quad \beta^{(1)} = \frac{1}{\|\mathbf{x}^{(0)}\|_2 \|\mathbf{x}^{(1)}\|_2}$$

Assim por diante... até o passo k:

$$\mathbf{y}^{(k)} = \beta^{(k)} \sum_{i=1}^{n} \alpha_i \lambda_i^k \mathbf{v}_i\,, \quad \text{com} \quad \beta^{(k)} = \frac{1}{\|\mathbf{x}^{(0)}\|_2 \cdots \|\mathbf{x}^{(k)}\|_2}$$

Método das Potências

Convergência

$$\mathbf{y}^{(k)} = \beta^{(k)} \sum_{i=1}^{n} \alpha_i \lambda_i^k \mathbf{v}_i = \beta^{(k)} \lambda_1^k \left(\alpha_1 \mathbf{v}_1 + \underbrace{\sum_{i=2}^{n} \alpha_i \frac{\lambda_i^k}{\lambda_1^k} \mathbf{v}_i}_{\mathbf{r}^{(k)}} \right)$$

Como $|\lambda_1| > |\lambda_i| \;\Rightarrow\; \lim_{k\to\infty} \left(\lambda_i/\lambda_1\right)^k = 0 \;\Rightarrow\; \lim_{k\to\infty} \mathbf{r}^{(k)} = \bar{\mathbf{0}}$, logo

$$\lim_{k\to\infty} \mathbf{y}^{(k)} = \lim_{k\to\infty} \beta^{(k)} \lambda_1^k \alpha_1 \mathbf{v}_1 \,.$$

Agora precisamos mostrar que $\beta^{(k)} \lambda_1^k \alpha_1 \to \pm 1$. Por outro lado,

$$\beta^{(k)} = \frac{1}{\|\mathbf{A}^k \mathbf{x}^{(0)}\|_2} = \frac{1}{\|\lambda_1^k(\alpha_1 \mathbf{v}_1 + \mathbf{r}^{(k)})\|_2} \quad \text{e} \quad \|\mathbf{v}_1\|_2 = 1$$

$$\text{Portanto,} \quad \beta^{(k)} \lambda_1^k \alpha_1 = \frac{\lambda_1^k \alpha_1}{\|\lambda_1^k(\alpha_1 \mathbf{v}_1 + \mathbf{r}^{(k)})\|_2} \to \pm 1 \,.$$

Método das Potências

Critérios de Parada

Dados $\varepsilon > 0$ e $k_{max} \in \mathbb{N}$, temos:

1. $k = k_{max}$

Método das Potências
Critérios de Parada

Dados $\varepsilon > 0$ e $k_{max} \in \mathbb{N}$, temos:

1. $k = k_{max}$
2. $|\lambda_1^{(k+1)} - \lambda_1^{(k)}| < \varepsilon$

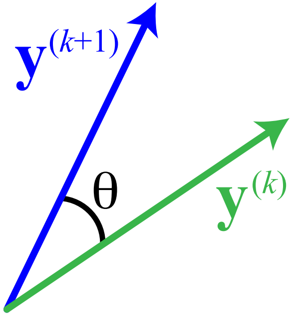
Método das Potências

Critérios de Parada

Dados $\varepsilon > 0$ e $k_{max} \in \mathbb{N}$, temos:

1. $k = k_{max}$
2. $|\lambda_1^{(k+1)} - \lambda_1^{(k)}| < \varepsilon$
3. teste de alinhamento ($|\cos\theta| \approx 1$):

$$\big|\, |\mathbf{y}^{(k+1)} \cdot \mathbf{y}^{(k)}| - 1 \,\big| < \varepsilon$$



MATLAB – Método das Potências

```
function [lambda,y,k] = potencias(A,tol)

k = 0; kmax = 1000; erro = inf;
n = size(A,1); y0 = zeros(n,1); y0(1) = 1;

while (erro>tol && k<kmax)
    x = A*y0;
    y = x/norm(x);
    erro = abs(abs(y0'*y)-1);
    y0 = y; k = k+1;
end

lambda = y'*A*y;
```

Método das Potências

Potência Inversa

Seja $\mathbf{A} \in \mathcal{M}(n,n)$ invertível. Agora, vamos assumir que os seus autovalores estão ordenados da seguinte forma:

$$|\lambda_1| \geq |\lambda_2| \geq |\lambda_3| \geq \cdots > |\lambda_n| > 0$$

Método das Potências

Potência Inversa

Seja $\mathbf{A} \in \mathcal{M}(n,n)$ invertível. Agora, vamos assumir que os seus autovalores estão ordenados da seguinte forma:

$$|\lambda_1| \geq |\lambda_2| \geq |\lambda_3| \geq \cdots > |\lambda_n| > 0$$

Objetivo:

- Calcular o **menor** autovalor em módulo λ_n e o seu autovetor associado $\mathbf{v}_n$. Note que,

$$\mathbf{A}\mathbf{v}_j = \lambda_j \mathbf{v}_j \iff \mathbf{A}^{-1}\mathbf{v}_j = \frac{1}{\lambda_j}\mathbf{v}_j \iff \Lambda(\mathbf{A}^{-1}) = \{1/\lambda_1, \ldots, 1/\lambda_n\}.$$

Logo, $1/\lambda_n$ é autovalor dominante de $\mathbf{A}^{-1}$.

Método das Potências

Potência Inversa

Dado um chute inicial $\mathbf{x}^{(0)} \in \mathbb{R}^n$ (não-nulo), calcule $\mathbf{y}^{(0)} = {}^{\mathbf{x}^{(0)}}/_{\|\mathbf{x}^{(0)}\|_2}$.

Método das Potências

Potência Inversa

Dado um chute inicial $\mathbf{x}^{(0)} \in \mathbb{R}^n$ (não-nulo), calcule $\mathbf{y}^{(0)} = \mathbf{x}^{(0)}/\|\mathbf{x}^{(0)}\|_2$.

Para $k = 1, 2, \ldots$ faça:

$$\mathbf{x}^{(k)} = \mathbf{A}^{-1}\mathbf{y}^{(k-1)}, \quad \mathbf{y}^{(k)} = \frac{\mathbf{x}^{(k)}}{\|\mathbf{x}^{(k)}\|_2}, \quad \lambda_n^{(k)} = \mathbf{y}^{(k)} \cdot \mathbf{A}\mathbf{y}^{(k)}$$

Considerações:

- Devemos resolver o sistema linear $\mathbf{A}\mathbf{x}^{(k)} = \mathbf{y}^{(k-1)}$ a cada iteração;
- Calcule a **Decomposição LU** de $\mathbf{A}$ e armazene $\mathbf{L}$ e $\mathbf{U}$;
- Use os algoritmos de substituição para resolver o sistema.

Método das Potências

Potência Inversa com Deslocamento

Pergunta: como calcular um autovalor arbitrário λ_j e o seu autovetor associado $\mathbf{v}_j$?

Método das Potências

Potência Inversa com Deslocamento

Pergunta: como calcular um autovalor arbitrário λ_j e o seu autovetor associado $\mathbf{v}_j$? A ideia é a seguinte, seja $\alpha \in \mathbb{C} \setminus \Lambda(\mathbf{A})$, logo

$$\Lambda(\mathbf{A}) = \{\lambda_1, \ldots, \lambda_n\} \implies \Lambda(\mathbf{A} - \alpha\mathbf{I}) = \{\lambda_1 - \alpha, \ldots, \lambda_n - \alpha\}.$$

Método das Potências

Potência Inversa com Deslocamento

Pergunta: como calcular um autovalor arbitrário λ_j e o seu autovetor associado $\mathbf{v}_j$? A ideia é a seguinte, seja $\alpha \in \mathbb{C} \setminus \Lambda(\mathbf{A})$, logo

$$\Lambda(\mathbf{A}) = \{\lambda_1, \ldots, \lambda_n\} \implies \Lambda(\mathbf{A} - \alpha\mathbf{I}) = \{\lambda_1 - \alpha, \ldots, \lambda_n - \alpha\}.$$

Assim, os autovalores de $\mathbf{B} = (\mathbf{A} - \alpha\mathbf{I})^{-1}$ são da forma

$$\boxed{\gamma_j = \frac{1}{\lambda_j - \alpha}}$$

Portanto, quanto mais próximo α estiver de λ_j, mais dominante será γ_j. O número α é chamado de **deslocamento**.

Resposta: agora basta aplicar o Método da Potência Inversa na matriz deslocada $\mathbf{B}$.

Método das Potências

Potência Inversa com Deslocamento

Pergunta: como calcular um autovalor arbitrário λ_j e o seu autovetor associado $\mathbf{v}_j$? A ideia é a seguinte, seja $\alpha \in \mathbb{C} \setminus \Lambda(\mathbf{A})$, logo

$$\Lambda(\mathbf{A}) = \{\lambda_1, \ldots, \lambda_n\} \implies \Lambda(\mathbf{A} - \alpha\mathbf{I}) = \{\lambda_1 - \alpha, \ldots, \lambda_n - \alpha\}\,.$$

Assim, os autovalores de $\mathbf{B} = (\mathbf{A} - \alpha\mathbf{I})^{-1}$ são da forma

$$\boxed{\gamma_j = \frac{1}{\lambda_j - \alpha}}$$

Portanto, quanto mais próximo α estiver de λ_j, mais dominante será γ_j. O número α é chamado de **deslocamento**.

Resposta: agora basta aplicar o Método da Potência Inversa na matriz deslocada $\mathbf{B}$. **Mas, como escolher α?**

MATLAB – Potência Inversa com Deslocamento

```
function [lambda,y,k] = potencia_inv(A,tol,alpha)

if(nargin==2) alpha = 0;  end

k = 0; kmax = 1000; erro = inf;
n = size(A,1); y0 = zeros(n,1); y0(1) = 1;
B = A - alpha*eye(n);
[L,U] = lu(B);

while (erro>tol && k<kmax)
    x = U\(L\y0);
    y = x/norm(x);
    erro = abs(abs(y0'*y)-1);
    y0 = y; k = k+1;
end

lambda = y'*A*y;
```

Discos de Gershgorin

Definição (discos de Gershgorin)

Seja $\mathbf{A} \in \mathcal{M}(n,n)$*. O* **disco de Gershgorin** *de centro* a_{ii} *e raio* r_i *associado a i-ésima linha de* $\mathbf{A}$ *é definido como*

$$D_{\mathbf{A}}(a_{ii}, r_i) = \{z \in \mathbb{C} \mid |z - a_{ii}| \leq r_i\} \quad \textit{com} \quad r_i = \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}|$$

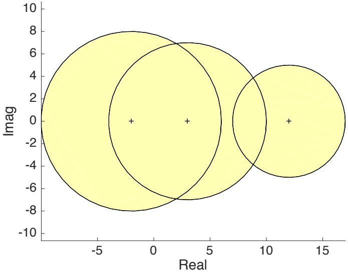
Discos de Gershgorin

Definição (discos de Gershgorin)

Seja $\mathbf{A} \in \mathcal{M}(n,n)$*. O* **disco de Gershgorin** *de centro* a_{ii} *e raio* r_i *associado a i-ésima linha de* $\mathbf{A}$ *é definido como*

$$D_{\mathbf{A}}(a_{ii}, r_i) = \{z \in \mathbb{C} \mid |z - a_{ii}| \leq r_i\} \quad com \quad r_i = \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}|$$

$$\mathbf{A} = \begin{bmatrix} 12 & 2 & 3 \\ 2 & 3 & 5 \\ 3 & 5 & -2 \end{bmatrix}$$



Discos de Gershgorin

Proposição

Seja $\mathbf{A} \in \mathcal{M}(n,n)$. *Temos que*

$$\Lambda(\mathbf{A}) \subset \left(\bigcup_{i=1}^{n} D_{\mathbf{A}}(a_{ii}, r_i) \right) \bigcap \left(\bigcup_{i=1}^{n} D_{\mathbf{A}^\top}(a_{ii}, r_i) \right).$$

Discos de Gershgorin

Proposição

Seja $\mathbf{A} \in \mathcal{M}(n,n)$*. Temos que*

$$\Lambda(\mathbf{A}) \subset \left(\bigcup_{i=1}^{n} D_{\mathbf{A}}(a_{ii}, r_i) \right) \bigcap \left(\bigcup_{i=1}^{n} D_{\mathbf{A}^\top}(a_{ii}, r_i) \right) .$$

Note que a proposição acima fornece um teste de rejeição no cálculo de autovalores e uma boa região (apesar de grande) para escolha do deslocamento α no Método da Potência Inversa.

Discos de Gershgorin

Proposição

Seja $\mathbf{A} \in \mathcal{M}(n,n)$*. Temos que*

$$\Lambda(\mathbf{A}) \subset \left(\bigcup_{i=1}^{n} D_{\mathbf{A}}(a_{ii}, r_i) \right) \bigcap \left(\bigcup_{i=1}^{n} D_{\mathbf{A}^\top}(a_{ii}, r_i) \right) .$$

Note que a proposição acima fornece um teste de rejeição no cálculo de autovalores e uma boa região (apesar de grande) para escolha do deslocamento α no Método da Potência Inversa.

Para restringir ainda mais essa região, não se esqueça do **Teorema Espectral**: se $\mathbf{A} = \mathbf{A}^\top \Rightarrow \Lambda(\mathbf{A}) \subset \mathbb{R}$.

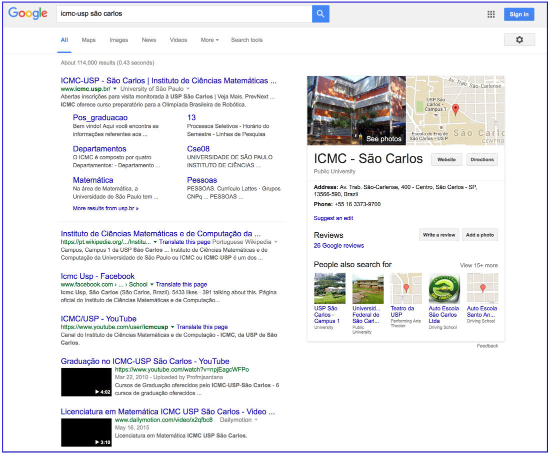
MATLAB – Discos de Gershgorin

```
function discos_gersh(A)

ctr = diag(A); raio = sum(abs(A-diag(ctr)),2);
theta = linspace(0,2*pi,50);
n = length(ctr);

figure; set(gca,'Fontsize',18); cor = [1, 1, 0.7];
axis equal; xlabel('Real'); ylabel('Imag');
hold on
for k=1:n
    D = ctr(k) + raio(k)*exp(i*theta);
    patch(real(D),imag(D),cor);
end
for k=1:n
    D = ctr(k) + raio(k)*exp(i*theta);
    plot(real(D),imag(D),'k-',real(ctr(k)),imag(ctr(k)),'k+');
end
hold off
```

Aplicação: Pagerank

Medida de relevância de uma página web (Brin & Page, 1998).

- ordenação dos resultados de uma busca no Google



Aplicação: Pagerank

Processo Estocástico

Definição (vetor estocástico)

Um vetor $\mathbf{p} \in \mathbb{R}^n$ *com* $p_i \geq 0$ *e* $\sum_{i=1}^{n} p_i = 1$ *é chamado de* **estocástico**.

Aplicação: Pagerank

Processo Estocástico

Definição (vetor estocástico)

Um vetor $\mathbf{p} \in \mathbb{R}^n$ *com* $p_i \geq 0$ *e* $\sum_{i=1}^{n} p_i = 1$ *é chamado de* **estocástico**.

Definição (matriz estocástica)

Uma matriz $\mathbf{A} \in \mathcal{M}(n,n)$ *cujos vetores coluna são estocásticos é chamada de* **matriz estocástica**.

Aplicação: Pagerank

Processo Estocástico

Definição (vetor estocástico)

Um vetor $\mathbf{p} \in \mathbb{R}^n$ *com* $p_i \geq 0$ *e* $\sum_{i=1}^{n} p_i = 1$ *é chamado de* **estocástico**.

Definição (matriz estocástica)

Uma matriz $\mathbf{A} \in \mathcal{M}(n,n)$ *cujos vetores coluna são estocásticos é chamada de* **matriz estocástica**.

Proposição

Se $\mathbf{A} \in \mathcal{M}(n,n)$ *e* $\mathbf{p} \in \mathbb{R}^n$ *são estocásticos então:*

1. *o vetor* $\mathbf{Ap} \in \mathbb{R}^n$ *é estocástico;*
2. $\lambda = 1$ *é um autovalor de* $\mathbf{A}$.

Aplicação: Pagerank

Matriz Estocástica

Processo de Markov

- Dada uma matriz estocástica $\mathbf{A}$ e um distribuição de probabilidade $\mathbf{p}^{(t)}$ no tempo t , podemos descobrir a distribuição no tempo $t+k$:

$$\boxed{\mathbf{p}^{(t+k)} = \mathbf{A}^k \mathbf{p}^{(t)} \,.}$$

Aplicação: Pagerank

Matriz Estocástica

Processo de Markov

- Dada uma matriz estocástica $\mathbf{A}$ e um distribuição de probabilidade $\mathbf{p}^{(t)}$ no tempo t , podemos descobrir a distribuição no tempo $t+k$:

$$\boxed{\mathbf{p}^{(t+k)} = \mathbf{A}^k\mathbf{p}^{(t)}\,.}$$

Exemplo 3

Em uma cidade têm pessoas que só ficam em 3 estados: 1 (magra), 2 (gorda) e 3 (sarada). A distribuição inicial é dada por $\mathbf{p}^{(0)} = (1/3, 1/3, 1/3)^\top$, suponha que a matriz estocástica seja

$$\mathbf{A} = \begin{bmatrix} 1/2 & 0 & 1/4 \\ 0 & 1/2 & 1/4 \\ 1/2 & 1/2 & 1/2 \end{bmatrix}$$

Aplicação: Pagerank

Matriz Estocástica

Processo de Markov

- Dada uma matriz estocástica $\mathbf{A}$ e um distribuição de probabilidade $\mathbf{p}^{(t)}$ no tempo t , podemos descobrir a distribuição no tempo $t+k$:

$$\boxed{\mathbf{p}^{(t+k)} = \mathbf{A}^k \mathbf{p}^{(t)}\,.}$$

Exemplo 3

Em uma cidade têm pessoas que só ficam em 3 estados: 1 (magra), 2 (gorda) e 3 (sarada). A distribuição inicial é dada por $\mathbf{p}^{(0)} = (1/3, 1/3, 1/3)^\top$, suponha que a matriz estocástica seja

$$\mathbf{A} = \begin{bmatrix} 1/2 & 0 & 1/4 \\ 0 & 1/2 & 1/4 \\ 1/2 & 1/2 & 1/2 \end{bmatrix} \implies \mathbf{p}^{(1)} = \mathbf{A}\mathbf{p}^{(0)} = \begin{bmatrix} 1/4 \\ 1/4 \\ 1/2 \end{bmatrix}$$

Aplicação: Pagerank

Teorema de Perron-Frobenius

Teorema (Perron-Frobenius)

Seja $\mathbf{A} \in \mathcal{M}(n,n)$ *uma matriz estocástica, então:*

1. $\lambda = 1$ *é o autovalor dominante de* $\mathbf{A}$*;*
2. *O autovetor* $\mathbf{v}$ *associado a* λ *possui todas entradas positivas ou negativas. Em particular, para* λ *existe um único autovetor que é estocástico.*

Aplicação: Pagerank

Teorema de Perron-Frobenius

Teorema (Perron-Frobenius)

Seja $\mathbf{A} \in \mathcal{M}(n,n)$ *uma matriz estocástica, então:*

1. $\lambda = 1$ *é o autovalor dominante de* $\mathbf{A}$*;*
2. *O autovetor* $\mathbf{v}$ *associado a* λ *possui todas entradas positivas ou negativas. Em particular, para* λ *existe um único autovetor que é estocástico.*

Portanto, a convergência do Processo de Markov é assegurada graças ao **Método das Potências**, isto é,

$$\mathbf{p}^{(k)} \to \mathbf{v}\,.$$

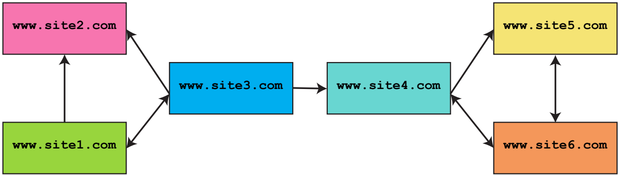
O autovetor $\mathbf{v}$ é chamado de **vetor estacionário** de $\mathbf{A}$.

Aplicação: Pagerank

MATLAB – Método das Potências *Revisitado*

```
function [lambda,y,k] = potencias_markov(A,tol)

k = 0; kmax = 1000; erro = inf;
n = size(A,1); y0 = ones(n,1)/n;

while (erro>tol && k<kmax)
    y = A*y0;
    erro = abs(abs(y0'*y)-1);
    y0 = y; k = k+1;
end

lambda = y'*A*y;
```

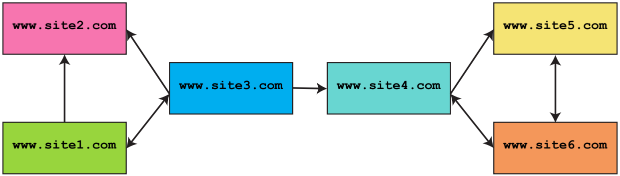
Aplicação: Pagerank
WWW



Páginas da web podem ser representadas como um grafo direcionado.

- uma página é importante se páginas importantes têm *link* para ela;
- eleição: um link de $A \to B$ é um voto de A para B;

Aplicação: Pagerank
WWW

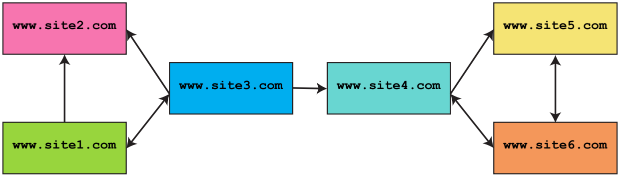


Páginas da web podem ser representadas como um grafo direcionado.

- uma página é importante se páginas importantes têm *link* para ela;
- eleição: um link de $A \to B$ é um voto de A para B;
- **visão probabilística do Pagerank:** é a probabilidade de uma página web ser visitada em certo instante de tempo durante um passeio aleatório infinito.

Aplicação: Pagerank

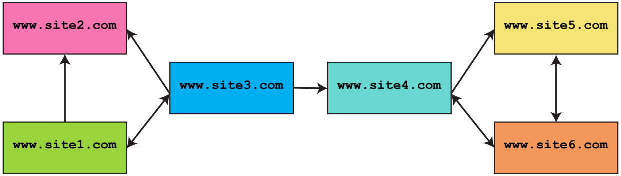
WWW



Um grafo direcionado pode ser representado por uma matriz de conectividade:

$$a_{ij} = \begin{cases} 1 & \text{se tem link } j \to i \\ 0 & \text{caso contrário} \end{cases} \qquad \mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

Aplicação: Pagerank
WWW



Agora precisamos transformar **A** em uma "matriz estocástica" **P**:

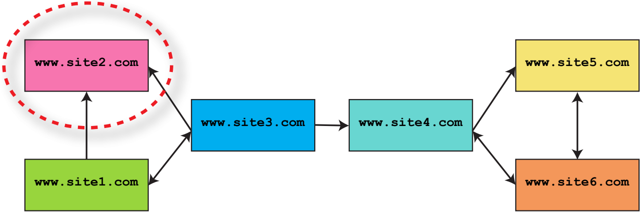
$$p_{ij} = \begin{cases} {}^{a_{ij}}/_{c_j} & \text{se } c_j \neq 0 \\ 0 & \text{caso contrário} \end{cases}$$

$$\text{com} \quad c_j = \sum_{i=1}^{n} a_{ij}$$

$$\mathbf{P} = \begin{bmatrix} 0 & 0 & 1/3 & 0 & 0 & 0 \\ 1/2 & 0 & 1/3 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1/2 & 1 & 0 \end{bmatrix}$$

Aplicação: Pagerank

WWW



Para resolver os "becos sem saída" precisamos de um vetor $\mathbf{d} \in \mathbb{R}^n$:

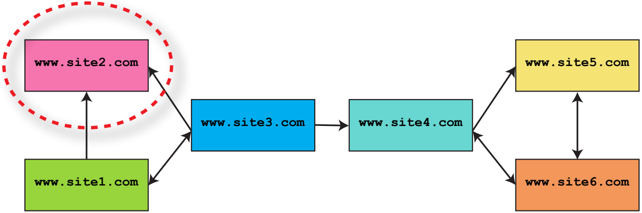
$$d_j = \begin{cases} 1/n & \text{se } c_j = 0 \\ 0 & \text{caso contrário} \end{cases}$$

$$\mathbb{1} = [1, \ldots, 1]^\top \in \mathbb{R}^n$$

$$\mathbf{S} = \begin{bmatrix} 0 & 0 & 1/3 & 0 & 0 & 0 \\ 1/2 & 0 & 1/3 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1/2 & 1 & 0 \end{bmatrix} + \mathbb{1}\mathbf{d}^\top$$

Aplicação: Pagerank

WWW



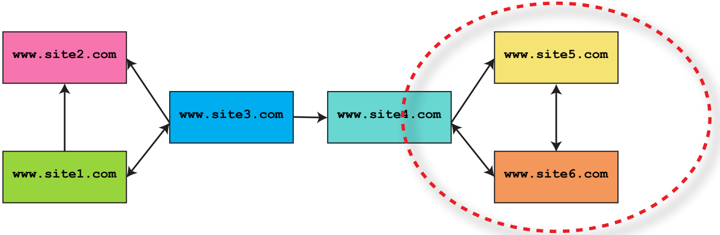
Para resolver os "becos sem saída" precisamos de um vetor $\mathbf{d} \in \mathbb{R}^n$:

$$d_j = \begin{cases} 1/n & \text{se } c_j = 0 \\ 0 & \text{caso contrário} \end{cases}$$

$$\mathbb{1} = [1, \ldots, 1]^\top \in \mathbb{R}^n$$

$$\mathbf{S} = \begin{bmatrix} 0 & 1/6 & 1/3 & 0 & 0 & 0 \\ 1/2 & 1/6 & 1/3 & 0 & 0 & 0 \\ 1/2 & 1/6 & 0 & 0 & 0 & 0 \\ 0 & 1/6 & 1/3 & 0 & 0 & 1/2 \\ 0 & 1/6 & 0 & 1/2 & 0 & 1/2 \\ 0 & 1/6 & 0 & 1/2 & 1 & 0 \end{bmatrix}$$

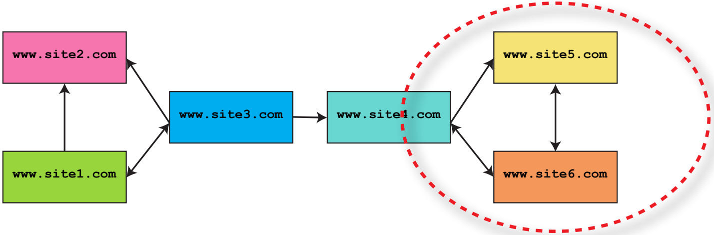
Aplicação: Pagerank
WWW



Agora precisamos evitar ciclos no grafo modificando **S** de modo a tornar o grafo irredutível:

$$\mathbf{G} = \alpha \begin{bmatrix} 0 & 1/6 & 1/3 & 0 & 0 & 0 \\ 1/2 & 1/6 & 1/3 & 0 & 0 & 0 \\ 1/2 & 1/6 & 0 & 0 & 0 & 0 \\ 0 & 1/6 & 1/3 & 0 & 0 & 1/2 \\ 0 & 1/6 & 0 & 1/2 & 0 & 1/2 \\ 0 & 1/6 & 0 & 1/2 & 1 & 0 \end{bmatrix} + (1-\alpha)\frac{\mathbb{1}\mathbb{1}^\top}{n}$$

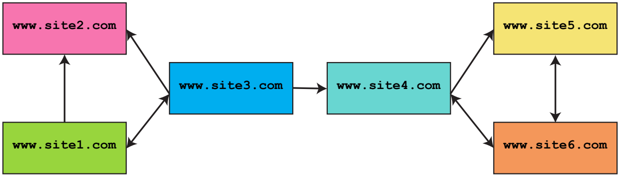
Aplicação: Pagerank
WWW



Agora precisamos evitar ciclos no grafo modificando **S** de modo a tornar o grafo irredutível:

$$\mathbf{G} = \alpha \begin{bmatrix} 0 & 1/2 & 1/3 & 0 & 0 & 0 \\ 1/2 & 1/2 & 1/3 & 0 & 0 & 0 \\ 1/2 & 1/6 & 0 & 0 & 0 & 0 \\ 0 & 1/6 & 1/3 & 0 & 0 & 1/2 \\ 0 & 1/6 & 0 & 1/2 & 0 & 1/2 \\ 0 & 1/6 & 0 & 1/2 & 1 & 0 \end{bmatrix} + (1-\alpha) \begin{bmatrix} 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \end{bmatrix}$$

Aplicação: Pagerank
WWW



Matriz Google:

$$\mathbf{G} = \alpha(\mathbf{A} + \mathbb{1}\mathbf{d}^\top) + (1-\alpha)\frac{\mathbb{1}\mathbb{1}^\top}{n}$$

Originalmente o valor escolhido para o peso foi $\alpha = 0.85$.

Aplicação: Pagerank

MATLAB

```
clear;
I = [2 3 2 4 5 6 6 4 5];
J = [1 1 3 3 4 4 5 6 6];
n = 6; A = zeros(n);

for idx=1:length(I)
    A(I(idx),J(idx)) = 1;
end

c = sum(A);
j = find(c == 0);
A(:,j) = ones(n,1); c(j) = n;
S = A./repmat(c,[n,1]);
alpha = 0.85;
G = alpha*S + (1-alpha)*ones(n)/n;
[~,v] = potencias_markov(G,1e-6);
[~,ranking] = sort(v,'descend')
```