

Desafio Computacional: Power Iteration vs. QR Iteration

1. Base Teórica

1.1 Autovalores e Autovetores (Revisão)

Seja $A \in \mathbb{R}^{n \times n}$. Um par (λ, \mathbf{v}) satisfaz

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

onde λ é um *autovalor* e $\mathbf{v} \neq \mathbf{0}$ é o *autovetor* associado. O polinômio característico de A é

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

cujas raízes são exatamente os autovalores de A . Em particular, se A é real e simétrica, todos os autovalores são reais e seus autovetores podem ser tomados como ortonormais.

1.2 Power Iteration

- **Objetivo:** encontrar o autovalor de maior módulo e seu autovetor associado.

- **Descrição do método:**

1. Escolher um vetor inicial não nulo $\mathbf{x}^{(0)}$.
2. Repetir para $k = 0, 1, 2, \dots$:
 - (a) Calcular $\mathbf{y}^{(k+1)} = A \mathbf{x}^{(k)}$.
 - (b) Normalizar $\mathbf{x}^{(k+1)} := \mathbf{y}^{(k+1)} / \|\mathbf{y}^{(k+1)}\|$.
 - (c) Estimar $\mu^{(k+1)} := (\mathbf{x}^{(k+1)})^T A \mathbf{x}^{(k+1)}$.
 - (d) Verificar critério de convergência, por exemplo:

$$\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| < \text{tol} \quad \text{ou} \quad |\mu^{(k+1)} - \mu^{(k)}| < \text{tol}.$$

3. Quando o critério for satisfeito, tomar $\lambda_{\max} \approx \mu^{(k+1)}$ e $\mathbf{v}_{\max} \approx \mathbf{x}^{(k+1)}$.

- **Convergência:** se os autovalores de A satisfazem $|\lambda_1| > |\lambda_2|$, então $\mathbf{x}^{(k)}$ converge (até normalização) ao autovetor associado ao maior valor λ_1 à taxa aproximada $|\lambda_2/\lambda_1|$.

- **Deflation (extração de múltiplos autovalores):**

1. Após obter $(\lambda_1, \mathbf{v}_1)$, formar

$$A^{(1)} = A - \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T.$$

2. Aplicar Power Iteration em $A^{(1)}$ para encontrar $(\lambda_2, \mathbf{v}_2)$, e assim por diante até extrair os k maiores autovalores.

- **Análise de Complexidade:**

- Cada iteração do Power Iteration requer uma multiplicação matriz-vetor $A \mathbf{x}$, que custa $O(n^2)$ para uma matriz densa de ordem n .
- A normalização do vetor e o cálculo do quociente de Rayleigh também custam $O(n)$.
- Portanto, cada iteração totaliza $O(n^2)$.
- Para extrair o autovalor dominante até convergência, o número de iterações pode variar de $O(\log(1/\text{tol})/\log(|\lambda_1/\lambda_2|))$.
- Para extrair os três maiores via Deflation, aplica-se Power Iteration em matrizes modificadas três vezes:

$$\text{Custo total aproximado} = 3 \times (\#\text{iters}) \times O(n^2).$$

- Em suma, a complexidade de extrair k autovalores dominantes via Power Iteration + Deflation é $O(k n^2 N)$, onde N é o número médio de iterações até convergência para cada autovalor.

1.3 Iteração QR

- **Objetivo:** aproximar simultaneamente todos (ou a maioria) dos autovalores de A .
- **Algoritmo sem deslocamento:**

1. Definir $A^{(0)} = A$.
2. Para $k = 0, 1, 2, \dots$:
 - (a) Calcular decomposição $A^{(k)} = Q^{(k)} R^{(k)}$, cujo custo é $O(n^3)$ para uma matriz densa $n \times n$.
 - (b) Atualizar $A^{(k+1)} = R^{(k)} Q^{(k)}$.
 - (c) Verificar critério de convergência nos subdiagonais de $A^{(k+1)}$, custos $O(n^2)$ para inspecionar os $n(n-1)/2$ elementos abaixo da diagonal principal.
3. Quando convergir, os autovalores aproximados aparecem na diagonal de $A^{(k+1)}$.

- **Algoritmo com deslocamento (single shift):**

1. Para cada passo k , escolher $\mu^{(k)} = (A^{(k)})_{n,n}$.
2. Definir $B = A^{(k)} - \mu^{(k)} I$.
3. Calcular $B = Q R$ (custo $O(n^3)$).
4. Atualizar $A^{(k+1)} = R Q + \mu^{(k)} I$.
5. Verificar convergência nos subdiagonais (custo adicional $O(n^2)$).

- **Análise de Complexidade:**

- Cada iteração do QR requer decomposição QR de uma matriz densa $n \times n$, com custo $O(n^3)$.
- A etapa de atualização RQ também custa $O(n^3)$, mas na prática, quando $A^{(k)}$ está em forma de Hessenberg (redução opcional de $O(n^3)$ inicial), cada iteração QR pode ser reduzida a $O(n^2)$.

- * *Redução a Hessenberg*: custo único $O(n^3)$ por transformações de Householder.
 - * *Iterações subsequentes*: $O(n^2)$ por passo, pois a matriz Hessenberg possui apenas $O(n)$ subdiagonais não nulos.
 - Sem redução a Hessenberg, cada iteração permanece $O(n^3)$.
 - Assim, para uma matriz já em forma de Hessenberg, a complexidade total de m iterações QR é $O(n^3 + m n^2)$.
 - No caso simétrico, costuma-se reduzir a uma forma tridiagonal, e cada iteração custa aproximadamente $O(n^2)$, resultando em custo total $O(n^3 + m n^2)$.
 - Em resumo:
- Complexidade sem redução: $O(m n^3)$, Com redução a Hessenberg: $O(n^3 + m n^2)$,
- onde m é o número de iterações até convergência.

- **Convergência:**

- Em matrizes simétricas, QR com deslocamento converge quadrática ou super-quadraticamente para autovalores isolados.
- Em matrizes não simétricas, podem aparecer blocos 2×2 relacionados a autovalores complexos conjugados.

2. Desafio: Power Iteration vs. QR Iteration

2.1 Objetivos

1. Implementar:

- Power Iteration (com Deflation para extrair os três maiores autovalores).
- QR Iteration sem deslocamento.
- QR Iteration com deslocamento (single shift).

2. Comparar os métodos em cada matriz de teste fornecida pelo instrutor, registrando:

- Número de iterações até adequar-se ao critério de convergência.
- Tempo de execução total (cronômetro antes e depois de cada método).
- Precisão numérica: erro relativo nos três maiores autovalores, calculado em relação aos valores de referência que o instrutor fornecerá.
- Para Power Iteration, efetuar verificação da ortonormalidade aproximada dos três autovetores obtidos, medindo $\|V^T V - I_{3 \times 3}\|$, onde V contém as colunas $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$.

3. Analisar:

- Casos em que $|\lambda_2/\lambda_1|$ esteja próximo de 1, tornando o Power Iteration lento ou instável.

- Desempenho comparativo de QR sem deslocamento vs. QR com deslocamento, especialmente em matrizes simétricas e não simétricas.
- Quando cada método é preferível (por exemplo, matrizes de pequeno ou grande tamanho, simétricas ou não, espaçamento do espectro).

3. Instruções de Implementação

3.1 Power Iteration

1. Receber uma matriz $A \in \mathbb{R}^{n \times n}$ e um vetor inicial $\mathbf{x}^{(0)} \in \mathbb{R}^n$ não nulo.
2. Normalizar $\mathbf{x}^{(0)} \leftarrow \mathbf{x}^{(0)} / \|\mathbf{x}^{(0)}\|$.
3. Iterar até convergência ou até atingir o número máximo de iterações (`max_iter`):
 - (a) Calcular $\mathbf{y} = A \mathbf{x}$.
 - (b) Normalizar $\mathbf{x}_{\text{novo}} = \mathbf{y} / \|\mathbf{y}\|$.
 - (c) Computar o quociente de Rayleigh $\mu = \mathbf{x}_{\text{novo}}^T A \mathbf{x}_{\text{novo}}$.
 - (d) Testar se $\|\mathbf{x}_{\text{novo}} - \mathbf{x}\| < \text{tol}$ ou $|\mu - \mu_{\text{antigo}}| < \text{tol}$. Se sim, parar e retornar μ e \mathbf{x}_{novo} . Caso contrário, atualizar $\mathbf{x} \leftarrow \mathbf{x}_{\text{novo}}$ e $\mu_{\text{antigo}} \leftarrow \mu$.
4. Caso não tenha convergido em `max_iter`, retornar a última aproximação obtida.
5. Para extrair os três maiores autovalores, repetir o processo (Deflation):
 - (a) Após obter $(\lambda_1, \mathbf{v}_1)$, formar

$$A^{(1)} = A - \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T.$$

- (b) Aplicar Power Iteration em $A^{(1)}$ para achar $(\lambda_2, \mathbf{v}_2)$.
- (c) Formar

$$A^{(2)} = A^{(1)} - \lambda_2 \mathbf{v}_2 \mathbf{v}_2^T = A - \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T - \lambda_2 \mathbf{v}_2 \mathbf{v}_2^T.$$

- (d) Aplicar Power Iteration em $A^{(2)}$ para obter $(\lambda_3, \mathbf{v}_3)$.

3.2 QR Iteration sem Deslocamento

1. Receber a matriz $A \in \mathbb{R}^{n \times n}$. (Opcional: antes, reduzir A à forma de Hessenberg se não for simétrica. Se souber fazer, ótimo; caso contrário, usar A diretamente.)
2. Para cada iteração $k = 0, 1, 2, \dots$ até convergir ou atingir `max_iter`:
 - (a) Calcular a decomposição $A^{(k)} = Q^{(k)} R^{(k)}$.
 - (b) Atualizar $A^{(k+1)} = R^{(k)} Q^{(k)}$.
 - (c) Verificar se $\max_{i>j} |(A^{(k+1)})_{ij}| < \text{tol}$.
3. Quando convergir, os autovalores aproximados são os n elementos da diagonal de $A^{(k+1)}$.
4. Se não convergir em `max_iter`, utilizar a diagonal de $A^{(\text{último})}$ como aproximação.

3.3 QR Iteration com Deslocamento (Single Shift)

1. Receber $A \in \mathbb{R}^{n \times n}$. (Opcional: reduzir a Hessenberg.)
2. Para cada iteração $k = 0, 1, 2, \dots$ até convergir ou `max_iter`:
 - (a) Escolher o *shift* $\mu^{(k)} = (A^{(k)})_{n,n}$.
 - (b) Definir $B = A^{(k)} - \mu^{(k)}I$.
 - (c) Calcular a decomposição $B = QR$.
 - (d) Atualizar $A^{(k+1)} = RQ + \mu^{(k)}I$.
 - (e) Verificar se $\max_{i>j} |(A^{(k+1)})_{i,j}| < \text{tol}$.
3. Quando convergir, aproximar os autovalores pela diagonal de $A^{(k+1)}$.
4. Se não convergir em `max_iter`, usar a diagonal de $A^{(\text{último})}$.

4. Matrizes de Teste

As matrizes de teste serão fornecidas pelo instrutor. Para cada matriz A recebida, procedam conforme descrito:

- Carregue A a partir de arquivo (por exemplo, formato `.npy` ou `.txt`).
- Juntamente com A , o instrutor fornecerá os três maiores autovalores de referência $\lambda_1^{\text{ref}}, \lambda_2^{\text{ref}}, \lambda_3^{\text{ref}}$.
- Escolha um vetor inicial $\mathbf{x}^{(0)}$ aleatório (componentes uniformes ou gaussiana) e normalize antes de iniciar Power Iteration.
- Aplicar Power Iteration + Deflation para extrair $\lambda_1, \lambda_2, \lambda_3$ e seus autovetores associados.
- Aplicar QR Iteration sem deslocamento para obter as aproximações de todos os autovalores, ordenar em ordem decrescente de módulo e extrair os três maiores.
- Aplicar QR Iteration com deslocamento e, novamente, extrair os três maiores.
- Para cada método, registrar:
 - *Tempo de execução total*, usando cronômetro de alta resolução.
 - *Número de iterações* até convergência.
 - *Erro relativo* nos três maiores autovalores:

$$\text{erro}_i = \frac{|\lambda_i^{(\text{método})} - \lambda_i^{\text{ref}}|}{|\lambda_i^{\text{ref}}|}, \quad i = 1, 2, 3.$$

- (Somente para Power Iteration) Verificar a *ortonormalidade aproximada* dos três autovetores extraídos, calculando

$$\|V^T V - I_{3 \times 3}\|,$$

onde $V = [\mathbf{v}_1 \ \mathbf{v}_2 \ \mathbf{v}_3]$.

- Repetir o processo em cada matriz de teste recebida.

5. Organização do Relatório

Siga este roteiro para montar o relatório final:

1. Introdução

- Contextualização da relevância do cálculo de autovalores/autovetores (ex.: compressão de dados, sistemas dinâmicos, PCA).
- Apresentação sucinta dos métodos a comparar: Power Iteration e QR Iteration.

2. Base Teórica (resumo)

- Definição breve de autovalor/autovetor e polinômio característico.
- Descrição conceitual do Power Iteration (com Deflation).
- Descrição conceitual do QR Iteration (sem deslocamento e com deslocamento).
- Análise de complexidade de ambos os métodos:
 - Power Iteration: $O(n^2)$ por iteração, total $O(k n^2 N)$ para extrair k autovalores com N iterações cada.
 - QR sem redução: $O(m n^3)$ para m iterações; com redução a Hessenberg, $O(n^3 + m n^2)$.
 - QR com deslocamento segue mesma ordem assintótica.

3. Instruções de Implementação (em linguagem natural)

- Descrever passo a passo o que cada rotina deve fazer, sem fornecer o código-fonte completo:
 - (a) Power Iteration:
 - Inicialização e normalização de $\mathbf{x}^{(0)}$.
 - Multiplicação $A \mathbf{x}$, normalização e cálculo do quociente de Rayleigh.
 - Critério de convergência.
 - Deflation para extrair múltiplos autovalores.
 - (b) QR Iteration sem deslocamento:
 - (Opcional) Redução de A à forma de Hessenberg ou usar A diretamente.
 - Decomposição QR e atualização RQ .
 - Critério de convergência nos subdiagonais.
 - (c) QR Iteration com deslocamento:
 - Escolha de μ a cada iteração (elemento (n, n)).
 - Decomposição QR de $A - \mu I$.
 - Atualização $RQ + \mu I$.
 - Critério de convergência.
- Especificar parâmetros padrão:

$$\text{tol}_{\text{PI}} = 10^{-8}, \quad \text{tol}_{\text{QR}} = 10^{-10}, \quad \text{max_iter} = 1000.$$

4. Descrição das Matrizes de Teste

- Listar as matrizes fornecidas pelo instrutor, identificando-as (por exemplo: Matriz A1: simétrica, $n = 100$; Matriz B2: não-simétrica, $n = 300$; etc.).
- Para cada matriz, incluir informações fornecidas (valores de referência dos três maiores autovalores, propriedades relevantes como proximidade dos dois maiores, simetria ou não).

5. Resultados Experimentais

- Para cada matriz, apresentar tabela contendo:
 - *Método* (PI+Deflation, QR sem deslocamento, QR com deslocamento).
 - *Tempo de execução* (em segundos).
 - *Número de iterações* até convergência.
 - *Erro relativo em λ_1* .
 - *Erro médio relativo em $\{\lambda_1, \lambda_2, \lambda_3\}$* .
- Se houver matrizes de várias dimensões, incluir gráficos comparativos:
 - (a) *Tempo vs. dimensão n* para cada método.
 - (b) *Número de iterações vs. n* .
 - (c) *Erro relativo em λ_1 vs. n* .

6. Discussão de Estabilidade e Desempenho

- Analisar casos em que $|\lambda_2/\lambda_1| \approx 1$:
 - Efeito no Power Iteration (convergência lenta ou falha).
 - Comportamento do QR sem deslocamento (normalmente lento quando autovalores próximos).
 - Como o QR com deslocamento contorna esse problema (convergência quadrática).
- Discutir quando cada método é mais adequado:
 - Matrizes *simétricas*: espaçamento do espectro e custo/estabilidade.
 - Matrizes *não-simétricas*: possíveis autovalores complexos e blocos 2×2 .
- Comentar sobre custo computacional e uso de memória:
 - Power Iteration: $O(n^2)$ por iteração, ideal para extrair poucos autovalores em matrizes densas grandes.
 - QR sem redução: $O(n^3)$ por iteração, viável apenas para n moderados.
 - QR com redução a Hessenberg: $O(n^3 + m n^2)$, adequado se $m \ll n$.

7. Conclusão

- Resumir as principais descobertas (por exemplo: Em matrizes simétricas até $n = 500$, QR com deslocamento converge em menos de 50 iterações, enquanto Power Iteration exigiu mais de 500 iterações quando $\lambda_2/\lambda_1 \approx 0,95$).
- Sugerir extensões, tais como:
 - Uso do Método de Lanczos para extrair vários autovalores de matrizes simétricas esparsas.
 - QR com múltiplos *shifts*.

- Deflation via transformações de Householder em vez de rank-1 updates simples.

8. Apêndice (opcional)

- Incluir pseudocódigo resumido ou fluxogramas de cada método, se necessário.
- Anexar dados brutos de medições, caso sejam volumosos.

6. Observações Finais

- **Focar nas matrizes fornecidas pelo instrutor:** não use outras instâncias para as medições finais, garantindo comparabilidade entre todos os alunos.
- **Parâmetros de convergência padrão** (caso não informado de outra forma):

$$\text{tol}_{\text{PI}} = 10^{-8}, \quad \text{tol}_{\text{QR}} = 10^{-10}, \quad \text{max_iter} = 1000.$$

- **Medição de tempo:** use cronômetros de alta resolução (em Python, `time.perf_counter()`; em C++, `std::chrono::high_resolution_clock`; em Java, `System.nanoTime()`). Meça apenas a execução principal de cada método.
- **Validação preliminar:**
 - Teste em matrizes pequenas (3E3, 5E5) cujos autovalores sejam conhecidos a priori.
 - Se Power Iteration não convergir em `max_iter`, registre não convergiu e discuta no relatório.
 - Verifique a ortonormalidade aproximada dos autovetores extraídos pelo Power Iteration, calculando $\|V^T V - I\|$ para $V \in \mathbb{R}^{n \times 3}$.
- **Formato de entrega:**
 - Um único arquivo PDF com o relatório final, estruturado conforme indicado.
 - Código-fonte organizado em diretório separado (não anexar código ao PDF, apenas instruções de como executar).
 - Instruções claras de compilação/execução, incluindo dependências necessárias.

May the force be with you!