Calcolo Numerico

2019-2020

Esercitazione 3

Calcolo autovalori e valori singolari di una matrice - Minimi quadrati lineari

1. **Metodo delle potenze per il calcolo dell'autovettore relativo all'autovalore massimo in modulo.PageRank Algorithm: i motori di ricerca risolvono un problema agli autovalori.** Ogni volta che digitiamo alcune parole nella search bar, l'algoritmo di posizionamento di Google propone i siti web con un determinato ordine. Ci possono essere milioni di pagine web che contengono quelle precise parole e sono inerenti al tema cercato, eppure sono poche le pagine rilevanti che ci interessano davvero. Come riescono, allora, i motori di ricerca a fornirci cosí spesso le pagine giuste fra le prime 20-30 pagine suggerite? Elaborando le loro risposte sulla base di algoritmi di PageRank.

   Il primo algoritmo di page ranking é stato inventato da due studenti della Stanford University, Larry Page e Sergey Brin, e divenne un Google trademark nel 1998. Attualmente, il PageRank di Google é diventato parte di un sistema molto piú evoluto.

   L' idea alla base di questo algoritmo é semplice: ad ogni pagina web si associa un peso numerico (detto rango) che misura l'importanza relativa della pagina rispetto a tutte le altre possibili pagine e ne indica quindi il "grado di popolaritá". Il motore di ricerca propone le pagine in ordine di rango decrescente. Il rango di una pagina é detereminato dal numero di pagine web che contengono il link alla pagina stessa: maggiore é il numero, maggiore é il rango (la pagina in questione é evidentemente ritenuta importante da chi ha creato i contenuti delle altre pagine).

   Consideriamo un caso limitato a 4 pagine web, legate tra loro come mostrato nella figura 1.

   Costruiamo allora il corrispondente grafo orientato, riportato nella figura 2. Ogni pagina web rappresenta un nodo e ogni link da una pagina i alla pagina j equivale alla freccia dal nodo i al nodo j.

   Dal grafo, passiamo alla matrice A di adiacenza (detta anche, in questo contesto, matrice dei link). La si costruisce con questa regola:

$$A_{i,j} = \begin{cases} 1 & \text{se la pagina j contiene il link alla pagina i} \\ 0 & \text{altrimenti} \end{cases}$$

   Quindi risulta essere:

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

   L'autovettore associato al massimo autovalore di A corrisponde al vettore r di Page Rank, ossia al vettore (colonna) $r = [r_1, r_2, r_3, r_4]$ contenente i rank delle quattro pagine. Se consideriamo r normalizzato (tale che $r_1 + r_2 + r_3 + r_4 = 1$ con $0 \leq r_i \leq 1$), ogni suo elemento indica la probabilitá di una pagina di essere mostrata per prima.

   Richiami alla teoria e alla notazione.

   E' data una matrice $A \in \mathbb{R}^{n\times n}$. Un vettore $x \in \mathbb{R}^n$, $x \neq 0$, é definito *autovettore di A* se esiste un numero $\lambda \in \mathbb{C}$ tale che $Ax = \lambda x \in \mathbb{R}^n$.
   L'autovalore massimo in modulo $\bar{\lambda}$ é il raggio spettrale di A e lo si indica con $\rho(A) = |\bar{\lambda}|$.
   Vogliamo trovare $\bar{\lambda}$ e il corrispondente autovettore $\bar{x}$. Normalizzando $\bar{x}$ determino il vettore di ranking r.

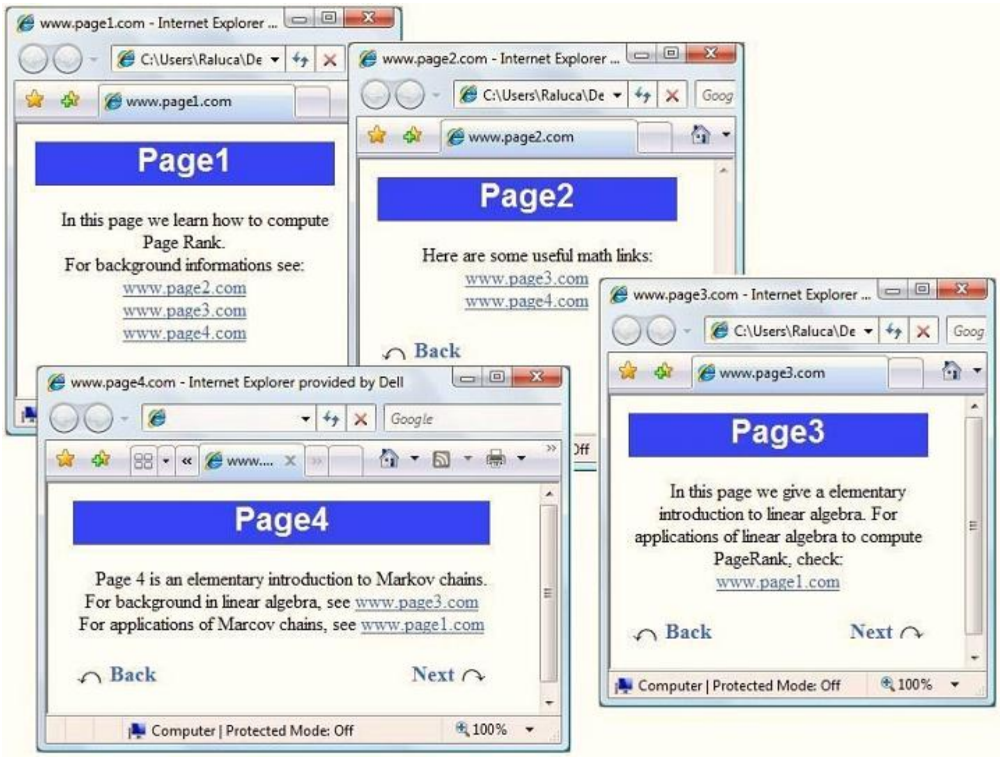

Figura 1:

Esercizio

Implementare il metodo delle potenze in una function:

- passare in input la matrice A, il vettore di partenza x_0 e il parametro per il test di arresto
- ottenere in output $\bar{x}$

Scrivere uno script Matlab in cui:

- si fissa la dimensione n del problema
- si costruisce la matrice dei link A
- si utilizza la vostra function per determinare $\bar{x}$
- si determina l'ordine con cui mostrare le pagine web, tramite il vettore r.
- si confronta il proprio risultato con quanto si ottiene usando la function Matlab `eig`, che implementa il calcolo di tutti gli autovalori e gli autovettori di A. Modificare il criterio di arresto del metodo delle potenze, se necessario.

Testare lo script sia sulla matrice A riportata nell'equazione (1), sia sulla matrice che si ottiene dal grafo in figura 3.

2. **Rappresentazione di un'immagine tramite valori singolari della matrice.**

Introduzione

Un'immagine è la riproduzione o imitazione di un oggetto. L'occhio umano è un sistema un grado di formare e registrare immagini date da radiazioni elettromagnetiche nello spettro visibile. Altri tipi di immagini, con diverse lunghezze d'onda, possono essere formate da sistemi diversi dall'occhio umano. La formazione dell'immagine è dunque un processo continuo, ma nel trattamento delle immagini al

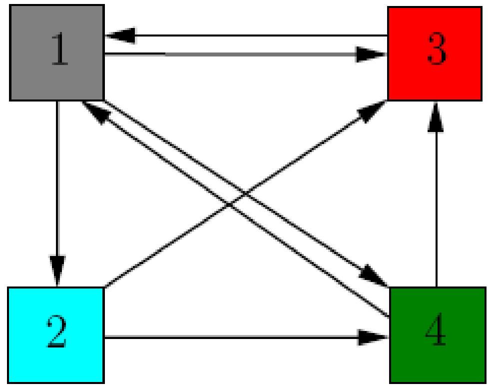

Figura 2:

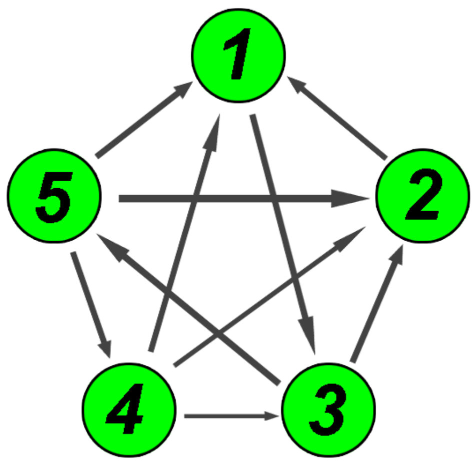

Figura 3:

computer si effettua una discretizzazione (la cosiddetta digitalizzazione dell'immagine). In particolare, i valori che possono essere assunti dall'immagine discretizzata G vengono suddivisi in regioni discrete e ciascuna è associata ad un dato numero. Mediante questo processo, detto quantizzazione, ogni elemento $G_{i,j}$, denominato *pixel* assume un valore intero compreso nell'intervallo $[0, 2^k - 1]$. Se si pone $k = 8$ (8 bit=1 byte) l'intervallo di rappresentazione è $[0, 255]$. Il valore 0 rappresenta il valore minimo di luminosità (nero) mentre il valore 255 rappresenta il massimo valore di luminosità ed è perciò associato al bianco.

Usando la decomposizione in valori singolari dell'immagine G si ha:

$$G = \sum_{i=1}^{r} \sigma_i u_i v_i^t$$

dove σ_i sono i valori singolari e r rappresenta il rango di G.

Si può interpretare (2) come una somma pesata di r matrici di rango 1 (diadi) dette anche autoimmagini $u_i v_i^t$. L'errore che si commette troncando i termini della somma ad un valore $k < r$ è :

$$\|G - G^{(k)}\|_2 = \sigma_{k+1}$$

dove

$$G^{(k)} = \sum_{i=1}^{k} \sigma_i u_i v_i^t$$

.

L'utilizzo della rappresentazione SVD può consentire anche una riduzione dell'occupazione di memoria passando da n^2 word intere a $k(2n+1)$ word floating point. Pertanto se i valori singolari decrescono molto rapidamente, si può ottenere una buona rappresentazione con una minore occupazione di memoria.

Se consideriamo un'immagine $m \times n$ a livelli di grigio (in cui c'è quindi un'unica matrice G) il cosiddetto *fattore di compressione* viene quindi definito come:

$$c = k(\frac{1}{m} + \frac{1}{n}).$$

La definizione si può facilemnte estendere alle immagini a colori, in cui sono necessarie tre matrici G_1, G_2, G_3, una per ogni canale di colori (rosso,verde,blu).

La differenza fra l'immagine G e quella ottenuta dalla rappresentazione SVD troncata al termine k–esimo viene valutata secondo i seguenti parametri di errore:

Errore Relativo:

$$E\_REL = \frac{\|G - G^{(k)}\|_2}{\|G\|_2}$$

Esercizio

Scrivere una function Matlab per rappresentare un'immagine a livelli di grigio mediante la fattorizzazione SVD (usare la funzione matlab `svd` con un numero k di diadi a piacere.

- Considerare 2-3 immagini scelte a piacere (usare immagini concaratteristiche differenti per numero di livelli di grigio, numero e dimensione dei dettagli). Per ognuna delle immagini considerate:
- Visualizzare l'autoimmagine relativa al valore singolare piú grande e quella relativa al valore singolare piú piccolo e alcune immagini $G^{(k)}$
- riportare nella seguente tabella il numero k di diadi richieste per avere un errore relativo $< 1\%$, $< 5\%$, e $< 10\%$:

nome immagine	errore$< 1\%$	errore$< 5\%$	errore$< 10\%$

- Plottare il grafico dei valori singolari; plottare il grafico dell'errore relativo al variare del numero delle diadi; plottare il grafico del fattore di compressione

$$c = k(\frac{1}{m} + \frac{1}{n})$$

(immagine $m \times n$) diadi sempre al variare del numero di diadi (in ascissa).

3. **Minimi quadrati per approssimazione dati.**

Assegnato un data set di m elementi $(x_i, y_i), i = 1, \ldots m$, determinare il polinomio di approssimazione ai minimi quadrati di grado $n = 1, \ldots 5$ dei dati, dove i coefficienti del polinimio

$$p(x) = \alpha_0 + \alpha_1 x + \ldots \alpha_n x^n$$

sono gli elementi del vettore $\alpha = (\alpha_0, \alpha_2, \ldots \alpha_n)$ soluzione del problema:

$$min\|A\alpha - \mathbf{y}|_2^2, \tag{2}$$

Scrivere una funzione Matlab che:

- calcoli la soluzione risolvendo le equazioni normali

$$A^T A\alpha = A^T \mathbf{y}$$

dove

$$A = \begin{pmatrix} 1 & x_1 & \dots & x_1^n \\ 1 & x_2 & \dots & x_2^n \\ \dots & \dots & \dots & \dots \\ 1 & x_m & \dots & x_m^n \end{pmatrix}$$

- calcoli la soluzione utilizzando la SVD di A

$$\alpha = \sum_{i=1}^{n} \frac{(\mathbf{u}_i^T \mathbf{y})\mathbf{v}_i}{\sigma_i}$$

- Plottare i dati e il grafico delle funzioni ottenute.
- Considerare i seguenti insiemi di dati:
  a) (1.0, 1.18), (1.2, 1.26), (1.4, 1.23), (1.6, 1.37), (1.8, 1.37), (2.0, 1.45), (2.2, 1.42), (2.4, 1.46), (2.6, 1.53), (2.8, 1.59), (3.0, 1.50).
  c) Un data set con difficoltà= lower dal sito web NIST: http://www.itl.nist.gov/div898/strd/lls/lls.shtml

Traccia per la relazione

1.]**Esercizio 1.** Mostrare i risultati ottenuti negli esemi proposti e verificarne la corretezza.
2. **Esercizio 2.** Commentare latabella e i grafici richiesti. Mostrare qualche immagine ottenuta con un numero di diadi variabile (considerare i valori singolari ordinati in senso crescente e decrescente). Spiegare, utilizzando le informazioni teoriche studiate, i risultati ottenuti.
3. **Esercizio 3.** Verificare innanzitutto se ci sono differenze significative sui coefficienti calcolati tramite equazioni normali e tramite SVD. Disvutere quindi le approssimaazioni ottenute al variare del grado del polinomio.