Di testi ed immagini

dott. Pollini Andrea

Università Cattolica del Sacro Cuore - Brescia

23/5/2005

Sommario della Parte I

Sommario della Parte I

1. Diagonalizzabilità di una matrice
   - Autovalori ed autovettori
2. Singular Value Decomposition di una matrice
   - Caratteristiche dell'SVD

Sommario della Parte II

Sommario della Parte II

3. Catalogazione automatica di documenti
   - Il clustering
   - Il problema del clustering
   - Un algoritmo di clustering
   - Il caso di collezioni di documenti

4. Compressione di immagini
   - Rappresentazione delle immagini
   - Compressione dell'immagine

Parte I

Richiami di algebra lineare

- Data una matrice $A \in \mathrm{Mat}_{n,n}(\mathbb{K})$ simmetrica esistono una matrice $V \in \mathrm{Mat}_{n,n}(\mathbb{K})$ ortogonale e $D \in \mathrm{Mat}_{n,n}(\mathbb{K})$ diagonale tali che vale la relazione

$$A = VDV^T$$

- Le colonne di V sono gli autovettori di A e formano una base ortonormale in $\mathbb{R}^n$
- La decomposizione VDV^T di chiama anche *Eigenvalue Decomposition* o, brevemente, EVD.

- Data una matrice $A \in \mathrm{Mat}_{n,n}(\mathbb{K})$ simmetrica esistono una matrice $V \in \mathrm{Mat}_{n,n}(\mathbb{K})$ ortogonale e $D \in \mathrm{Mat}_{n,n}(\mathbb{K})$ diagonale tali che vale la relazione

$$A = VDV^T$$

- Le colonne di V sono gli autovettori di A e formano una base ortonormale in $\mathbb{R}^n$
- La decomposizione VDV^T di chiama anche *Eigenvalue Decomposition* o, brevemente, EVD.

- Data una matrice $A \in \mathrm{Mat}_{n,n}(\mathbb{K})$ simmetrica esistono una matrice $V \in \mathrm{Mat}_{n,n}(\mathbb{K})$ ortogonale e $D \in \mathrm{Mat}_{n,n}(\mathbb{K})$ diagonale tali che vale la relazione

$$A = VDV^T$$

- Le colonne di V sono gli autovettori di A e formano una base ortonormale in $\mathbb{R}^n$
- La decomposizione VDV^T di chiama anche *Eigenvalue Decomposition* o, brevemente, EVD.

- Sia $A \in \mathrm{Mat}_{m,n}(\mathbb{K})$. Esistono $U \in \mathrm{Mat}_{m,m}(K)$ e $V \in \mathrm{Mat}_{n,n}(K)$ ortogonali e una matrice simmetrica $\Sigma \in \mathrm{Mat}_{m,n}(\mathbb{K})$ tali che

$$A = U\Sigma V^T$$

- Gli elementi di Σ sono detti *singular value* di A e le colonne di U e V sono i *singular vector* di A.
- I *singular value* e i *singular vector* soddisfano

$$\begin{aligned} Au &= \sigma v \\ Av &= \sigma u \end{aligned}$$

- Sia $A \in \mathrm{Mat}_{m,n}(\mathbb{K})$. Esistono $U \in \mathrm{Mat}_{m,m}(K)$ e $V \in \mathrm{Mat}_{n,n}(K)$ ortogonali e una matrice simmetrica $\Sigma \in \mathrm{Mat}_{m,n}(\mathbb{K})$ tali che

$$A = U\Sigma V^T$$

- Gli elementi di Σ sono detti *singular value* di A e le colonne di U e V sono i *singular vector* di A.
- I *singular value* e i *singular vector* soddisfano

$$\begin{aligned} Au &= \sigma v \\ Av &= \sigma u \end{aligned}$$

Analogie tra EVD ed SVD

Se si pensa alle matrici come trasformazioni lineari:

EVD Le colonne di V sono vettori (ortonormali) che formano una base rispetto ai quali la trasformazione identificata da A sono solo dilatazioni.

SVD In questo caso A identifica una trasformazione

$$\phi : \mathbb{R}^n \longrightarrow \mathbb{R}^m$$

e le colonne di U e V forniscono gli elementi di basi nel dominio e codominio rispetto alle quali la trasformazione è una dilatazione cui si aggiungono gli zeri per il cambio di dimensione.

Analogie tra EVD ed SVD

Se si pensa alle matrici come trasformazioni lineari:

EVD Le colonne di V sono vettori (ortonormali) che formano una base rispetto ai quali la trasformazione identificata da A sono solo dilatazioni.

SVD In questo caso A identifica una trasformazione

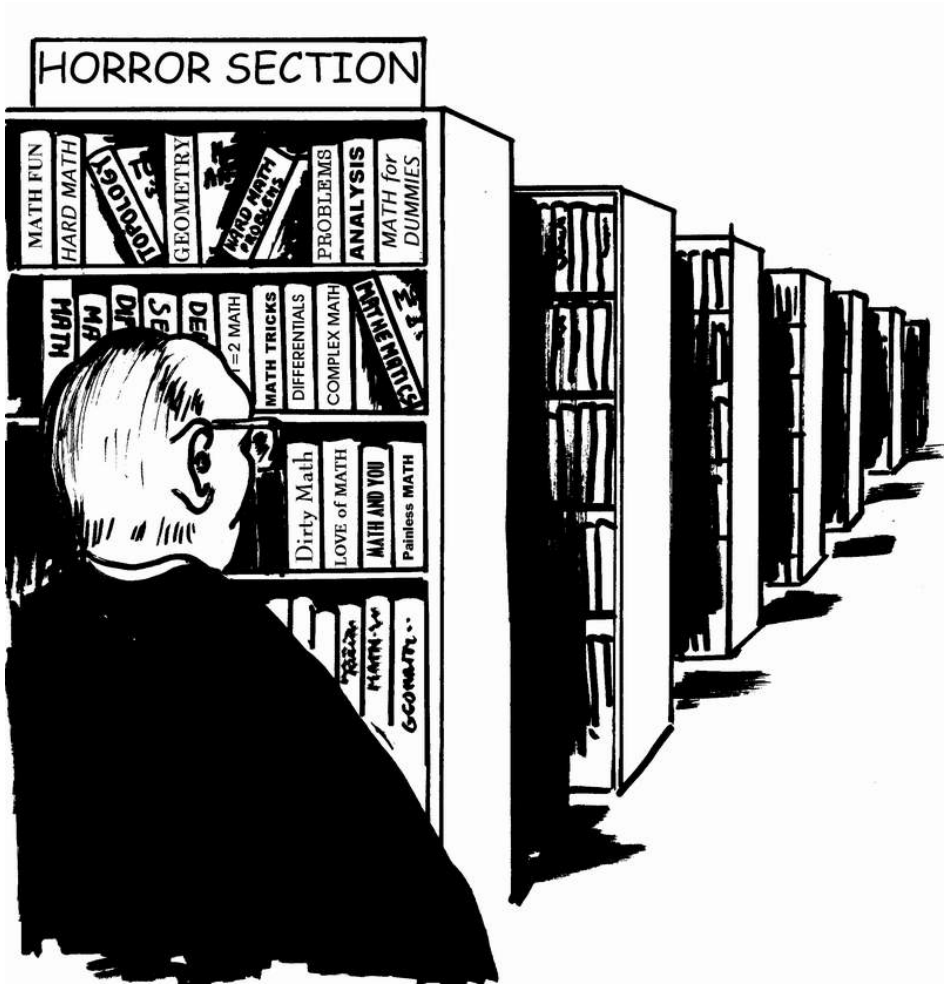
$$\phi : \mathbb{R}^n \longrightarrow \mathbb{R}^m$$

e le colonne di U e V forniscono gli elementi di basi nel dominio e codominio rispetto alle quali la trasformazione è una dilatazione cui si aggiungono gli zeri per il cambio di dimensione.

Parte II

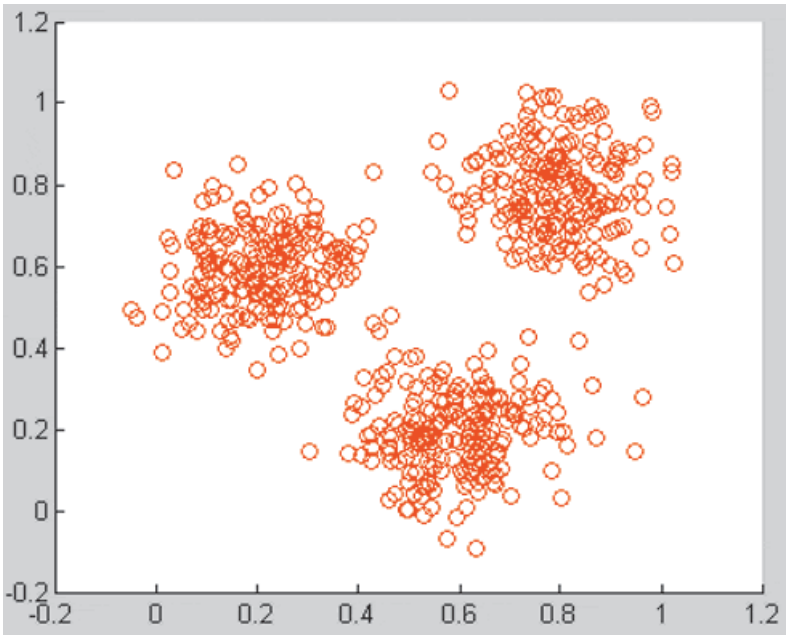
Applicazioni

Catalogazione automatica di documenti



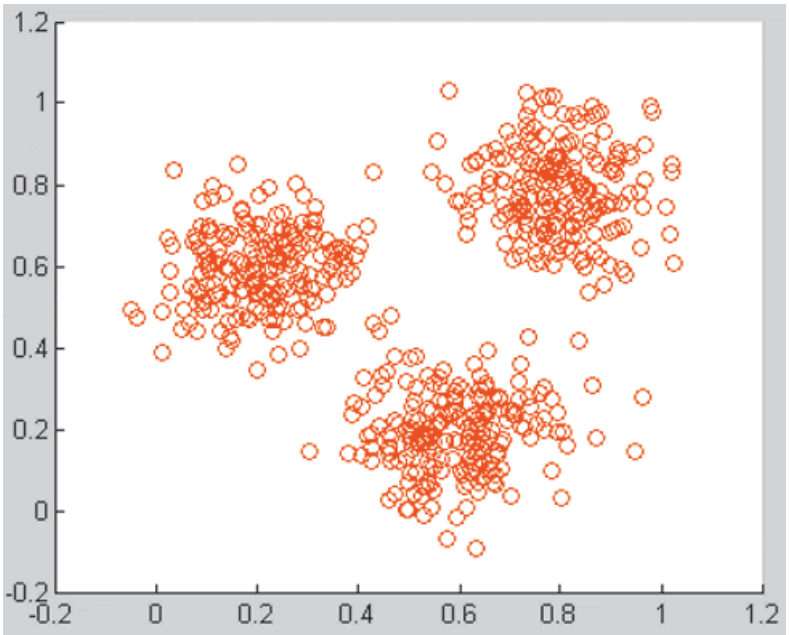
Cos'è il *clustering*

- **Processo di partionamento** di una collezione di oggetti in un insieme di sottoclassi significative, dette *cluster*.
- Ogni *cluster* è un insieme di oggetti che sono simili e che quindi formano un gruppo omogeneo.
- Il *clustering* aiuta a capire la **struttura** di una collezione di oggetti.



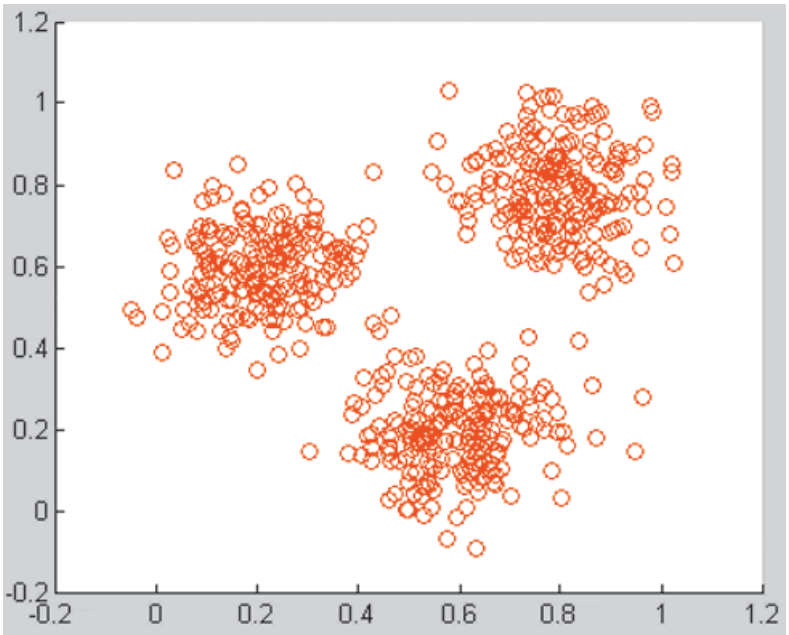
Cos'è il *clustering*

- **Processo di partionamento** di una collezione di oggetti in un insieme di sottoclassi significative, dette *cluster*.
- Ogni *cluster* è un insieme di oggetti che sono simili e che quindi formano un gruppo omogeneo.
- Il *clustering* aiuta a capire la **struttura** di una collezione di oggetti.



Cos'è il *clustering*

- **Processo di partionamento** di una collezione di oggetti in un insieme di sottoclassi significative, dette *cluster*.
- Ogni *cluster* è un insieme di oggetti che sono simili e che quindi formano un gruppo omogeneo.
- Il *clustering* aiuta a capire la **struttura** di una collezione di oggetti.



Tipi di classificazione

Data una collezione di oggetti, si potranno effettuare su di essi due tipi di classificazione

Classificazione supervisionata Si conoscono in anticipo sia il numero che le descrizioni delle classi distinte cui possono appartenere gli oggetti.

Classificazione non supervisionata non si conoscono né il numero di classi né le loro caratteristiche principali.

E il clustering?

Il clustering è una **classificazione non supervisionata**, in quanto le classi in cui si divide la collezione non sono note a priori.

Il problema del clustering

- Rappresentare una collezione di oggetti in uno spazio vetttoriale k-dimensionale.
  - Selezione di k features.
  - Sclelta di una distanza.
- Suddividerla in sottinsiemi (detti appunto **cluster**) tali che
  - Sia minima la distanza tra oggetti del medesimo cluster.
  - Sia massima la distanza da oggetti appartenenti ad altri cluster.

Il problema del clustering

- Rappresentare una collezione di oggetti in uno spazio vetttoriale k-dimensionale.
  - Selezione di k features.
  - Sclelta di una distanza.
- Suddividerla in sottinsiemi (detti appunto **cluster**) tali che
  - Sia minima la distanza tra oggetti del medesimo cluster.
  - Sia massima la distanza da oggetti appartenenti ad altri cluster.

Algoritmi di clustering

Algoritmi partizionanti Costruiscono una suddivisione della collezione di partenza e poi la ottimizzano procedendo in maniera ricorsiva.

Algoritmi Gerarchici Costruiscono un albero detto dendogramma le cui foglie sono i cluster ottenuti.

Agglomerativi Uniscono i cluster ricorsivamente.

Divisivi Suddividono i cluster ricorsivamente.

Algoritmi di clustering

Algoritmi partizionanti Costruiscono una suddivisione della collezione di partenza e poi la ottimizzano procedendo in maniera ricorsiva.

Algoritmi Gerarchici Costruiscono un albero detto dendogramma le cui foglie sono i cluster ottenuti.

- **Agglomerativi** Uniscono i cluster ricorsivamente.
- **Divisivi** Suddividono i cluster ricorsivamente.

L'algoritmo PDDP

- Proposto del prof. Daniel Boley (University of Minnesota).
- Algoritmo di tipo gerarchico divisivo.
- Bisecante: ad ogni passo suddivide un cluster in due sottocluster.
- Ogni oggetto è rappresentato da un vettore colonna $d_i \in \mathbb{R}^n$, dove n è il numero di feature selezionate per il clustering.
- Una collezione di k oggetti è rappresentata da una matrice

$$M = (d_1, \ldots, d_k)$$

- Definizione dello scatter, ovvero della misura della coesione dei cluster (scatter maggiori corrispondono a cluster meno coesi).

Processo di suddivisione

- Calcolo del centroide w di M e della direzione principale u di $\tilde{M} = M - w \cdot e$ con $e = (1, \ldots, 1)$
- Divisione degli oggetti che formano M in due sottocluster, M_L ed M_R, secondo il valore della proiezione di $d_i \in M$ su u.

Processo di suddivisione

- Calcolo del centroide w di M e della direzione principale u di $\tilde{M} = M - w \cdot e$ con $e = (1, \ldots, 1)$
- Divisione degli oggetti che formano M in due sottocluster, M_L ed M_R, secondo il valore della proiezione di $d_i \in M$ su u.

Processo di suddivisione

- Calcolo dello scatter.
- Selezione del cluster con lo scatter maggiore, che verrà diviso al passo successsivo.
- Il processo continua fino al raggiungimento di un numero massimo di cluster o finché tutti i cluster non hanno scatter minore di una certa soglia.

Caratteristiche del problema

- Elevata dimensionalità sia in termini di numero di oggetti che in termini di numero di feature (In questo dominio le feature sono i termini distinti)
- Possibilità tramite il clutering di suddividere una grande collezione e di ottenere anche un riassunto del contenuto di ogni cluster.
  - Necessità di prettrattare i dati per migliorare le caratteristiche dei cluster e delle loro sintesi.
- Valutazione della qualità della divisione prodotta tramite il concetto di entropia.

Caratteristiche del problema

- Elevata dimensionalità sia in termini di numero di oggetti che in termini di numero di feature (In questo dominio le feature sono i termini distinti)
- Possibilità tramite il clutering di suddividere una grande collezione e di ottenere anche un riassunto del contenuto di ogni cluster.
  - Necessità di prettrattare i dati per migliorare le caratteristiche dei cluster e delle loro sintesi.
- Valutazione della qualità della divisione prodotta tramite il concetto di entropia.

Problemi aperti

- Algoritmi efficienti per SVD (parallelizzazione e distribuzione calcolo)
- Hardware e software adeguato.
- Utilizzo di più autovalori.

Compressione delle immagini tramite SVD

Come sono visualizzate le immagini?

- Le immagini vengono rappresentate tramite una griglia di punti colorati, detto *pixel*. Se i *pixel* sono abbastanza vicini approssimano un'immagine continua.
- Un'immagine scannerizzata viene realizzata
  - prendendo un'immagine e, suddivisa in una griglia fitta quanto sarà fitta la griglia di pixel su cui viene visualizzata.
  - Si opera un'interpolazione furba dei dati sulle celle.
  - Quindi una visualizzazine dell'immagine è sempre un'approssimazione.

Rappresenatazione dei pixel

bianco e nero sono elementi di $\mathbb{Z}_2$.

grayscale teoricamente valori nel range $[0, \infty)$, ma è chiaramente comodo $[0, 1]$. Problema della quantizzazione

coloscale Diversi *colour space*

- **RGB** monitor
- **CMY** stampanti
- **YIQ** Tv standard NSTC.

Rappresenatazione dei pixel

bianco e nero sono elementi di $\mathbb{Z}_2$.

grayscale teoricamente valori nel range $[0, \infty)$, ma è chiaramente comodo $[0, 1]$. Problema della quantizzazione

coloscale Diversi *colour space*

RGB monitor

CMY stampanti

YIQ Tv standard NSTC.

Rappresenatazione dei pixel

bianco e nero sono elementi di $\mathbb{Z}_2$.

grayscale teoricamente valori nel range $[0, \infty)$, ma è chiaramente comodo $[0, 1]$. Problema della quantizzazione

coloscale Diversi *colour space*

- **RGB** monitor
- **CMY** stampanti
- **YIQ** Tv standard NSTC.

Struttura dell'immagine

- Un pixel è un vettore nello spazio dei colori.
- Se l'immagine è $m \times n$ in uno spazio RGB, allora occupa $3mn$ elementi. Se si considerano quantizzazioni a 256 colori, un'immagine $480x640$ occupa $0.9Mb$.
- Poco spazio ma se le immagini devono essere trasmesse spreco di risorse.

Idea

Un'immagine non è una raccolta casuale di punti, ma ha una qualche struttura. Quindi sfrutto la struttura per rappresentare l'immagine in maniera furba.

Tipi di compressione

Si hanno due tipi di compressioni possibilià

loseless compression Niente perdita di informazioni. Salva i pattern, non i pixel.

lossy compression Sfrutta le limitazioni dell'occhio umano per approssimare l'immagine lasciandola uguale all'originale, almeno alla vista.

SVD per la compressione

- La SVD di $A \in \mathrm{Mat}_{m,n}(\mathbb{R})$ matrice reale è

$$A = USV^T$$

con le colonne di V autovettori di $A^T A$ e quelle di U autovettori di AA^T. A reale, anche U e V lo sono.

- La precedente relazione si può scrivere

$$A = \sum_{i=1}^{\min\{m,n\}} u_i \sigma_i v_i^T$$

- Posso realizzare l'approssimazione di rango k

$$A \approx \widehat{A} = \sum_{i=1}^{k} u_i \sigma_i v_i^T$$

- La norma 2 di $(A - \widehat{A})$ è uguale al primo autovalore non usato.

SVD per la compressione

- La SVD di $A \in \mathrm{Mat}_{m,n}(\mathbb{R})$ matrice reale è

$$A = USV^T$$

con le colonne di V autovettori di A^TA e quelle di U autovettori di AA^T. A reale, anche U e V lo sono.

- La precedente relazione si può scrivere

$$A = \sum_{i=1}^{\min\{m,n\}} u_i \sigma_i v_i^T$$

- Posso realizzare l'approssimazione di rango k

$$A \approx \widehat{A} = \sum_{i=1}^{k} u_i \sigma_i v_i^T$$

- La norma 2 di $(A - \widehat{A})$ è uguale al primo autovalore non usato.

Questioni di spazio...

- La matrice dei colori, nel caso di immagine grayscale di dimension $m \times m$, è di m^2 valori.
- L'occupazione richiesta dall'SVD è di $2m^2 + m$. Usando un'approssimazione di rango k l'occupazione è $2mk + k$.
- Devo usare un'approssimazione di rango al più

$$\frac{m^2}{1 + 2m}$$

Questioni di spazio...

- La matrice dei colori, nel caso di immagine grayscale di dimension $m \times m$, è di m^2 valori.
- L'occupazione richiesta dall'SVD è di $2m^2 + m$. Usando un'approssimazione di rango k l'occupazione è $2mk + k$.
- Devo usare un'approssimazione di rango al più

$$\frac{m^2}{1 + 2m}$$

Miglioramenti

- Codifica e decodifica in poarallelo tramite tecniche di *block splitting*.
- Buoni algoritmi di calcolo della SVD.
- Basse richieste in termini di storage.


That's All
Folks