LOCALLY LINEAR EMBEDDING

PROGETTO MATLAB
FONDAMENTI DI CALCOLO NUMERICO

Carlotta Marialuisa Guidi, Khalaf Abd El Rahman
Prof. Carlo De Falco

Febbraio 2019

POLITECNICO
MILANO 1863

Indice

Sommario

Per affrontare problemi di apprendimento automatico, grandi set di dati spesso devono essere adattati allo scopo di semplificare la loro visualizzazione. Ad esempio, immagini di volti, documenti di testo o spettrogrammi di discorso sono dati complessi, non-lineari e multidimensionali che necessitano di elaborazione prima di poterne studiare gli schemi sottostanti. È questa necessità di analizzare grandi quantità di dati multidimensionali che solleva il problema della riduzione della dimensionalità. Per questo progetto del corso di Fondamenti di Calcolo Numerico presentiamo il *Locally Linear Embedding*, un algoritmo di riduzione dimensionale che determina embedding a bassa dimensionalità per input N-dimensionali, sfruttando proprietà lineari locali per apprendere la struttura globale delle varietà non-lineari sottostanti.

1 Cenni di varietà k-dimensionali

1.1 Nozioni introduttive

L'entità più importante nella geometria differenziale è la *varietà differenziabile*. Una varietà differenziabile, intuitivamente, è un sottospazio all'interno di uno spazio più grande e rappresenta, nello spazio $\mathbb{R}^n$ di dimensione qualunque, una generalizzazione delle curve e delle superfici regolari nello spazio tridimensionale. Si chiameranno varietà k-dimensionali di $\mathbb{R}^n$ quei sottoinsiemi che si possono vedere come oggetti geometrici curvi, regolari e di dimensione k (per qualsiasi valore di k compreso tra 1 e $n-1$). Così come una superficie in $\mathbb{R}^3$, ad esempio una sfera, può essere definita sia in forma parametrica

$$\begin{cases} x = R\sin\varphi\cos\theta \\ y = R\sin\varphi\sin\theta \\ z = R\cos\varphi \end{cases} \qquad \varphi \in [\,0,\pi\,],\ \ \theta \in [\,0,2\pi)$$

come funzione $\mathbf{r} : \mathbb{R}^2 \to \mathbb{R}^3$, sia in forma implicita

$$x^2 + y^2 + z^2 - R^2 = 0$$

come un'equazione in tre variabili, allo stesso modo una varietà k-dimensionale si potrà definire sia in forma parametrica, con un'opportuna funzione $\mathbf{r} : \mathbb{R}^k \to \mathbb{R}^n$, sia in forma implicita, mediante un sistema di $n-k$ equazioni in n variabili. La Terra stessa può essere descritta come una sfera nello spazio tridimensionale, sebbene per gli individui che vivono sulla sua superficie sembri essere piatta. In linguaggio geometrico, la sfera è un oggetto tridimensionale paragonabile ad un oggetto bidimensionale – un piano – per un osservatore situato sulla sua superficie. Richiedere che questa sfera sia anche differenziabile ci permette di estendere alla varietà gli strumenti noti del calcolo differenziale. Una formalizzazione di queste idee porta ad una definizione matematica, presa in prestito da Doolin [1], che qui viene data solo per ragioni di completezza:

Definizione 1.1. Un sottoinsieme M di $\mathbb{R}^D$ è una varietà k-dimensionale se per ogni $x \in M$ esistono: sottoinsiemi aperti U e V di $\mathbb{R}^D$ con $x \in U$, e un diffeomorfismo f da U a V, tale che:

$$f(\,U \cap M\,) = \left\{\, y \in V :\ y^{k+1} = \ldots = y^D = 0 \,\right\}$$

Pertanto un punto y dell'immagine di f avrà sempre forma:

$$y = (y^1(x), y^2(x), ..., y^k(x), 0, ..., 0).$$

È comune inoltre descrivere la Terra tramite *mappe* della sua superficie, dove ogni mappa è la rappresentazione di una sezione di territorio abbastanza contenuta da potersi considerare localmente piana. Ovviamente, un'unica mappa non è in genere sufficiente per descrivere completamente la superficie della Terra dal momento che non vi potremo rappresentare contemporaneamente, ad esempio, Polo Nord e Polo Sud. Si può però collezionare un numero elevato e finito di queste mappe in un *atlas*: raggiunto il bordo di una mappa, per transizione, si passa alla mappa adiacente. Infine, non è escluso che alcune di queste mappe si sovrappongano. In questi stessi termini, possiamo parlare di varietà come collezione di mappe, laddove ogni mappa non è che una sezione ridotta approssimabile localmente ad un (iper)piano. Generalizzando queste idee si arriva alla seguente [1]:

Definizione 1.2. Una varietà C^∞ è una coppia (M, A) dove M è uno spazio topologico di Hausdorff completamente separabile e A un atlas C^∞ massimale.

Questa definizione all'apparenza complicata risulta essere proprio ciò che è necessario a soddisfare la nostra intuizione: le condizioni sulla topologia garantiscono che l'insieme delle mappe richiesto per descrivere completamente M sia numerabile, e il termine "massimale" è un tecnicismo che rende l'atlas la classe di collezioni di quelle sole mappe necessarie a formare una base numerabile per l'atlas.

1.2 LLE e le varietà k-dimensionali

L'ipotesi che sottende un algoritmo di riduzione della dimensionalità come LLE è che tutti i dati in input giacciano sulla superficie di una varietà e che ne rappresentino una versione campionata. Nel tentativo di rappresentare i dati in input nella loro descrizione più compatta possibile, l'algoritmo deve trovare le relazioni immanenti tra questi dati sotto l'ipotesi che questi giacciano localmente su sezioni della varietà approssimabili ad iperpiani: in affinità ai termini usati nel paragrafo precedente, la finalità dell'algoritmo diventa *la ricerca di un atlas della varietà*. Geometricamente, questo traguardo si formula come ricerca di un embedding a bassa dimensionalità dei dati ad alta dimensionalità, assunti su una varietà non-lineare, che preservi la geometria locale dei dati originali. Vale a dire: punti vicini nello spazio $\mathbb{R}^n$ dovranno rimanere vicini nello spazio $\mathbb{R}^n$ di destinazione.

1.3 La distanza Riemanniana

Considerati due punti su una varietà, siamo in generale interessati al percorso più breve tra i due punti *sulla superficie curva* della varietà stessa, ovvero alla *distanza geodetica* o *Riemanniana*, e non alla loro distanza euclidea o in *linea d'aria*. Nei casi fortunati in cui è nota la funzione costitutiva della varietà, questa distanza può essere calcolata sommando le distanze euclidee tra i passi infinitesimali sulla via dal primo al secondo punto [2]. Formalmente:

Definizione 1.3. Sia M una varietà connessa. Per ogni coppia di punti $p, q \in M$ si definisce la distanza Riemanniana come

$$d(p,q) \stackrel{\text{def}}{=} \inf \left\{ L(\sigma) \mid \sigma : [a,b] \to M \right\}$$

dove σ è una curva regolare a tratti, $\sigma(a) = p$, $\sigma(b) = q$ e

$$L(\sigma) = \int_a^b |\sigma'(u)|_{\sigma(u)} \, du$$

è la *lunghezza* di σ.

Ovvero, si vuole scegliere al variare di tutte le curve differenziabili σ, contenute in M che partono in p e arrivano in q, quella a lunghezza minore. Tuttavia, per quanto riguarda i problemi di apprendimento a cui è solito applicare gli algoritmo di riduzione dimensionale, la funzione costitutiva è raramente nota e bisogna ricorrere a stime della distanza geodetica, introducendo così una fonte di errore non trascurabile.

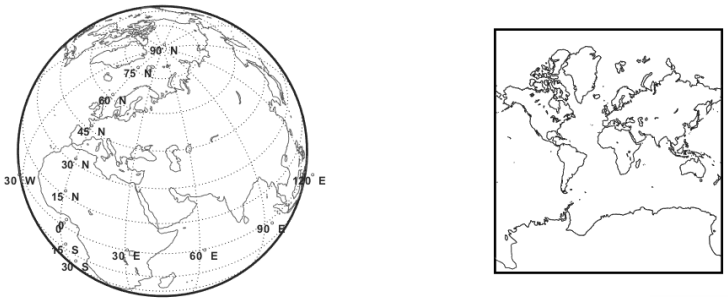


Figura 1: La proiezione di Mercatore è un noto esempio di riduzione dimensionale.

2 L'algoritmo

Come descritto dagli autori originali Roweis e Saul [3], l'algoritmo si basa sulle semplici intuizioni geometriche presentate finora, simulando la ricerca delle mappe e la costruzione dell'atlas attraverso i seguenti step:

1. Ricerca dei K punti più vicini per ogni campione.
2. Assegnamento dei pesi ricostruttivi ottimali per i K vicini.
3. Calcolo dell'embedding tramite conservazione dei pesi ricostruttivi.

2.1 Step I: Ricerca dei K vicini

Si supponga che i dati consistano in N vettori $\mathbf{x}_i$, ciascuno di dimensione D, campionati da una qualche varietà e collocati in una matrice $\mathbf{X} \in \mathbb{R}^{D \times N}$ dove ogni colonna è un punto[1] – o campione. Di nuovo, la premessa chiave in questa fase è che la varietà non-lineare campionata può essere ipotizzata localmente lineare se ciascun campione e i suoi vicini giacciono indicativamente sulla stessa sezione localmente lineare della varietà, ovvero, quando *l'area occupata dal campione e dai suoi K vicini è una mappa per la varietà*. Per un numero sufficiente di campioni questa premessa è generalmente soddisfatta e il calcolo delle distanze si riduce con buona approssimazione al calcolo delle distanze euclidee. Nel caso si adottassero altre metriche più raffinate, il risultato sarà comunque una matrice $\mathbf{N} \in \mathbb{R}^{K \times N}$, la cui j-esima colonna include gli indici dei K punti più vicini al j-esimo campione.

$\mathbf{x}_1$		$\mathbf{x}_3$		$\mathbf{x}_N$
x_{11}	x_{21}	x_{31}	$\cdots$	x_{N1}
x_{12}	x_{22}	x_{32}	$\cdots$	x_{N2}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
x_{1D}	x_{2D}	$\cdots$	$\cdots$	x_{ND}

1				*N*
3	$\cdots$	$\cdots$	$\cdots$	h_{N1}
$\vdots$	$\cdots$	$\cdots$	$\cdots$	$\vdots$
$\vdots$	$\ddots$	$\ddots$	$\ddots$	$\vdots$
N	$\cdots$	$\cdots$	$\cdots$	h_{NK}

Figura 2: In evidenza due dei K vicini di un campione $\mathbf{x}_1 \in \mathbf{X}$ con i relativi indici riportati nella 1-esima colonna di $\mathbf{N}$. Nell'esempio, si è assunto $\mathbf{x}_3$ come punto più vicino al campione $\mathbf{x}_1$ e $\mathbf{x}_N$ come punto più lontano, e per questo il numero 3 figura nella prima riga dell'1-esima colonna di $\mathbf{N}$, mentre il numero N nell'ultima.

[1] È importante notare che ad un vettore $\mathbf{r} = (x, y, z)$ è associato un punto nello spazio 3-dimensionale e che, generalizzando, si può interpretare un vettore $\mathbf{r} = (r_1, r_2, ..., r_D)$ come punto nello spazio D-dimensionale.

2.2 Step II: Assegnamento dei pesi ricostruttivi

Dal momento che si presuppone di lavorare all'interno di sottospazi lineari della varietà, ciascun punto può essere rappresentato da una combinazione lineare ponderata dei suoi K vicini complanari. In senso fisico, si vuole associare un *peso* ad ogni vicino in modo da far coincidere il *centro di massa* del sistema dato dai vicini con il campione. Matematicamente, si vuole minimizzazione la funzione

$$\varepsilon(\mathbf{w}_i) = \sum_{i=1}^{N} \left\| \mathbf{x}_i - \sum_{j=1}^{K} w_{ij}\mathbf{n}_j \right\|^2 \tag{1}$$

espressa come somma complessiva dei quadrati delle distanze tra tutti i punti e le loro ricostruzioni, dove $\mathbf{x}_i$ denota il campione corrente, $\mathbf{n}_j$ il suo j-esimo vicino e w_{ij} il peso ricostruttivo corrispondente. Introducendo l'ulteriore vincolo di normalizzazione

$$\sum_{j=1}^{K} w_{ij} = 1 \tag{2}$$

si rendono i pesi invarianti a trasformazioni lineari dei campioni come la traslazione, la rotazione ed il ridimensionamento. Questi potranno riflettere dunque proprietà della varietà indipendenti dal sistema di riferimento in cui questa viene fornita.

Dimostrazione. Siano $\mathbf{R}$ una matrice di rotazione e ridimensionamento, $\mathbf{v}$ il vettore dei pesi relativi ai vicini dei campioni a seguito della rotazione e del ridimensionamento e $\mathbf{w}$ il vettore dei pesi relativi ai vicini dei campioni originali. Se si suppone di aver ricostruito perfettamente ogni campione, per costruzione:

$$\begin{aligned}
\mathbf{R}\mathbf{x}_i &= \sum_{j=1}^{K} v_{ij}\mathbf{R}\mathbf{n}_j \\
\mathbf{R}^{-1}\mathbf{R}\mathbf{x}_i &= \sum_{j=1}^{K} v_{ij}\mathbf{R}^{-1}\mathbf{R}\mathbf{n}_j \\
\mathbf{x}_i &= \sum_{j=1}^{K} v_{ij}\mathbf{n}_j \\
\sum_{j=1}^{K} w_{ij}\mathbf{n}_j &= \sum_{j=1}^{K} v_{ij}\mathbf{n}_j
\end{aligned}$$

da cui segue $\mathbf{w}_i = \mathbf{v}_i$, ovvero: i vettori peso sono invarianti a rotazioni e ridimensionamenti. Si poteva giungere al medesimo risultato intuitivamente considerando che in un sistema di masse puntiformi, ruotando il centro di massa o moltiplicando tutte le masse del sistema per una costante, le distanze relative rimangono inalterate e così anche il contributo di ogni punto alla posizione del centro di massa.

Sommando inoltre un vettore costante qualsiasi ad ogni campione e servendosi del vincolo (2), è facile vedere che la funzione da minimizzare non cambia:

$$(\mathbf{x}_i + \mathbf{c}) - \sum_{j=1}^{K} w_{ij}(\mathbf{n}_j + \mathbf{c}) = (\mathbf{x}_i + \mathbf{c}) - \sum_{j=1}^{K} w_{ij}\mathbf{n}_j - \mathbf{c}\sum_{j=1}^{K} w_{ij} = \mathbf{x}_i - \sum_{j=1}^{K} w_{ij}\mathbf{n}_j$$

□

Sapendo che $\|\mathbf{a}\|^2 = \mathbf{a}^\mathbf{T}\mathbf{a}$ si può rielaborare la (1) in una forma quadratica più utile ai nostri scopi:

$$\begin{aligned}
\sum_{i=1}^{N} \left\| \sum_{j=1}^{K} w_{ij}\mathbf{x}_i - \sum_{j=1}^{K} w_{ij}\mathbf{n}_j \right\|^2 &= \sum_{i=1}^{N} \left\| \sum_{j=1}^{K} w_{ij}(\mathbf{x}_i - \mathbf{n}_j) \right\|^2 = \\
&\sum_{i=1}^{N} \left[\sum_{j=1}^{K}\sum_{k=1}^{K} w_{ij}(\mathbf{x}_i - \mathbf{n}_j)^\mathbf{T} \cdot\ w_{ki}(\mathbf{x}_i - \mathbf{n}_k) \right] = \\
&\sum_{i=1}^{N} \left[\sum_{j=1}^{K}\sum_{k=1}^{K} w_{ij}w_{ki}(\mathbf{x}_i - \mathbf{n}_j)^\mathbf{T} \cdot\ (\mathbf{x}_i - \mathbf{n}_k) \right] = \\
&= \sum_{i=1}^{N} \left[\sum_{j=1}^{K}\sum_{k=1}^{K} w_{ij}w_{ki}G_{ijk} \right] = \sum_{i=1}^{N} \mathbf{w}_i^\mathbf{T}\mathbf{G}_i\mathbf{w}_i
\end{aligned}$$

per cui è stata definita la matrice $\mathbf{G}_i$ di elementi $G_{ijk} = (\mathbf{x}_i - \mathbf{n}_j)^\mathbf{T} \cdot (\mathbf{x}_i - \mathbf{n}_k)$. Risulta comodo in questa forma adoperare il Metodo dei moltiplicatori di Langrange:

$$\mathcal{L}(\lambda, \mathbf{w}_i) = \mathbf{w}_i^\mathbf{T}\mathbf{G}_i\mathbf{w}_i - \lambda(\mathbf{1}^\mathbf{T}\mathbf{w}_i - 1)$$

dove per convenienza si è impiegato il vettore $\mathbf{1}$ di tutti uni e di lunghezza K per rileggere il vincolo in forma matriciale: $\mathbf{1}^\mathbf{T}\mathbf{w}_i = 1$. Prendendo le derivate parziali e notando che $\mathbf{G}_i$ è simmetrica per definizione:

$$\frac{\partial\mathcal{L}}{\partial\mathbf{w}_i} = 2\,\mathbf{G}_i\mathbf{w}_i - \lambda\mathbf{1}^\mathbf{T} = 0$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = \mathbf{1^T} \mathbf{w}_i - 1 = 0$$

Nel caso $K < D$, in cui $\mathbf{G}_i$ risulta non-singolare, e quindi invertibile, i pesi si trovano nella forma:

$$\mathbf{w}_i = \frac{\lambda}{2} \mathbf{G}_i^{-1} \mathbf{1^T}$$

dove λ è scelta in modo da garantire la somma-a-uno dei pesi per ciascun campione. Nella pratica, risolvere il sistema di equazioni lineari $\mathbf{G}_i \mathbf{w}_i = \mathbf{1^T}$ (avendo scelto arbitrariamente $\lambda = 2$) e normalizzare successivamente i pesi ottenuti offre risultati numericamente più stabili [4].

Per valori di $K > D$, $\mathbf{G}_i$ può diventare singolare: avendo rango D, esisteranno infatti $K - D$ gradi di libertà e, di conseguenza, molteplici possibilità per incorporare un campione in modo ottimale. Applicare un termine di regolarizzazione alla diagonale di $\mathbf{G}_i$ rende la matrice non-singolare ed impone una particolare soluzione. Diversi di questi termini regolarizzatori sarebbero possibili, ma una scelta semplice è:

$$\mathbf{G}_i \leftarrow \mathbf{G}_i + \mathbf{I}_{K \times K} \cdot \Delta \cdot \mathrm{tr}(\mathbf{G}_i)$$

con Δ dell'ordine di 10^{-5} [5].
Al termine di questo step, il prodotto sarà una matrice $\tilde{\mathbf{W}} \in \mathbb{R}^{K \times N}$ di pesi che ancora richiede le informazioni date dagli indici della matrice $\mathbf{N}$.

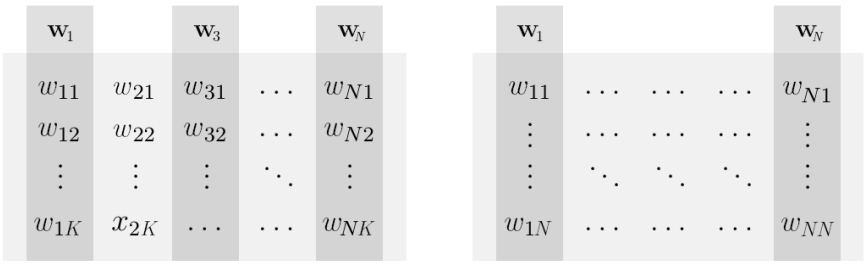


Figura 3: Se $w_{11} \in \tilde{\mathbf{W}}$ è il peso di $\mathbf{x}_3$ relativo ad $\mathbf{x}_1$, diventerà $w_{13} \in \mathbf{W}$.

Si supera questo inconveniente memorizzando i pesi in una nuova matrice $\mathbf{W} \in \mathbb{R}^{N \times N}$ costruita in modo tale da avere come ij-esimo elemento il peso ricostruttivo associato al j-esimo vicino del campione i. Dal momento che nei casi usuali vale

$N \gg K$, $\mathbf{W}$ sarà sparsa[2] e le sue colonne $\mathbf{w}_i = [\, w_{i1} \;\; w_{i2} \;\; ... \;\; w_{iN} \,]^{\mathbf{T}}$ saranno formate da quegli elementi $w_{ij} \neq 0$ solo per quelle j appartenenti all'i-esima colonna di $\mathbf{N}$.

2.3 Step III: Calcolo dell'incorporamento

Avendo legato ciascun campione ai suoi K adiacenti, si vuole trasportare al meglio l'informazione contenuta nei pesi a dimensioni inferiori stimando quei punti $\mathbf{y}_i$ che nello spazio d-dimensionale meglio rappresentano i corrispettivi campioni in input dati in D dimensioni. Similmente allo step precedente, è opportuno formulare un problema di minimizzazione in termini di una funzione costo, come la seguente:

$$\delta(\mathbf{Y}) = \sum_{i=1}^{N} \left\| \mathbf{y}_i - \sum_{j=1}^{N} w_{ij}\mathbf{y}_j \right\|^2 \tag{3}$$

mantenendo ora fissi i pesi. Con l'introduzione della matrice sparsa e simmetrica

$$\mathbf{M} = (\mathbf{I}_{N\times N} - \mathbf{W}) \cdot (\mathbf{I}_{N\times N} - \mathbf{W})^{\mathbf{T}}$$

diventa possibile anche in questo caso riformulare il problema in forma quadratica. Infatti, considerando $\mathbf{I}_i$ la i-esima colonna della matrice identità $\mathbf{I}_{N\times N}$

$$\delta(\mathbf{Y}) = \sum_{i=1}^{N} \| \mathbf{Y}\mathbf{I}_i - \mathbf{Y}\mathbf{w}_i \|^2 = \sum_{i=1}^{N} \| \mathbf{Y}(\mathbf{I}_i - \mathbf{w}_i) \|^2 = \| \mathbf{Y}(\mathbf{I}_{N\times N} - \mathbf{W}) \|^2$$

e ricordando che la norma $\|\mathbf{A}\|^2$ di ogni matrice quadrata è anche $\mathrm{tr}(\mathbf{A}^{\mathbf{T}}\mathbf{A})$, si può scrivere:

$$\delta(\mathbf{Y}) = \mathrm{tr}(\mathbf{Y}(\mathbf{I} - \mathbf{W}) \cdot (\mathbf{I} - \mathbf{W})^{\mathbf{T}}\mathbf{Y}^{\mathbf{T}}) = \mathrm{tr}(\mathbf{Y}\,\mathbf{M}\,\mathbf{Y}^{\mathbf{T}}) = \sum_{i=1}^{d} \mathbf{Y}_i \mathbf{M} \mathbf{Y}_i^{\mathbf{T}}$$

dove $\mathbf{Y}_i$ non è che l'i-esima riga di $\mathbf{Y}$ per $i = 1, 2, \ldots, d$.

Per la corretta minimizzazione di questa funzione, si noti che un embedding che si attiene alla conservazione delle distanze tra vicini è di sua natura invariante a traslazioni e rotazioni, dato che non mutano le distanze tra i punti. Per rendere il problema ben posto bisogna quindi forzare l'unicità della soluzione eliminando

[2] Una matrice quadrata di dimensione N si dice *sparsa* se il numero dei suoi elementi non nulli è di ordine N (pertanto asintoticamente minore del numero totale N^2 di elementi).

questi due gradi di libertà. A questo scopo si sono scelti arbitrariamente i seguenti vincoli: il primo per centrare i punti in output, ad esempio, nell'origine

$$\sum_{i=1}^{N} \mathbf{y}_i = \mathbf{0} \tag{4}$$

ed il secondo per fissare rotazione e scala

$$\mathbf{Y}\mathbf{Y}^{\mathbf{T}} = \mathbf{I} \tag{5}$$

Infatti, nell'algebra multilineare la nozione di covarianza impiegata nel secondo vincolo si riferisce al modo in cui la descrizione di una data entità geometrica varia quando si effettua un cambiamento di coordinate, come una rotazione nello spazio.

Applichiamo nuovamente il Metodo di Langrange alla funzione $\Phi(\mathbf{Y}_i) = \mathbf{Y}_i\mathbf{M}\mathbf{Y}_i^{\mathbf{T}}$ sotto il vincolo (5):

$$\mathcal{L}(\mu, \mathbf{Y}_i) = \mathbf{Y}_i\mathbf{M}\mathbf{Y}_i^{\mathbf{T}} - \mu(\mathbf{Y}_i\mathbf{Y}_i^{\mathbf{T}} - 1)$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{Y}_i} = 2\,\mathbf{M}\mathbf{Y}_i - 2\,\mu\mathbf{Y}_i = 0$$

$$\mathbf{M}\mathbf{Y}_i = \mu\mathbf{Y}_i$$

Dovrebbe essere evidente ora dagli studi di algebra lineare che questo risultato si ottiene quando le righe $\mathbf{Y}_i$ di $\mathbf{Y} \in \mathbb{R}^{d\times N}$ sono gli autovettori associati agli autovalori di $\mathbf{M}$. In particolare, vale il seguente teorema[3]:

Teorema 2.1. *Siano* $\mathbf{A}$ *e* $\mathbf{V}$*, rispettivamente, una matrice simmetrica* $n \times n$ *ed una matrice ortogonale* $d \times n$*. Allora la traccia di* $\mathbf{V}\mathbf{A}\mathbf{V}^{\mathbf{T}}$ *è minimizzata quando* $\mathbf{V}$ *è una matrice di basi ortogonali degli autospazi associati agli autovalori più piccoli di* $\mathbf{A}$*.*

Si poteva raggiungere questo risultato ricordando che gli autovettori di una matrice simmetrica, come $\mathbf{M}$, sono ortogonali e che per queste ultime vale la proprietà $\mathbf{Y^{-1}} = \mathbf{Y^{T}}$. Per cui [4]:

$$\mathrm{tr}(\mathbf{Y}\mathbf{M}\mathbf{Y}^{\mathbf{T}}) = \mathrm{tr}(\mathbf{Y}\mathbf{M}\mathbf{Y}^{-1}) = \mathrm{tr}(\mathbf{M}) = \lambda_1 + \lambda_2 + \cdots + \lambda_d$$

e la funzione sarà minimizzata per i d autovalori più piccoli.

[3] Questo teorema è un'immediata conseguenza della caratterizzazione di Courant-Fisher.
[4] La traccia è invariante per similitudine, ovvero due matrici simili hanno la stessa traccia.

3 L'implementazione

Dal momento che l'algoritmo fa uso intensivo di operazioni tra matrici e di strumenti dell'algebra lineare, è opportuno implementarlo in un ambiente ottimizzato per eseguire questo tipo di calcoli. Per questo progetto si è utilizzato il software di calcolo matriciale più prominente – **MATLAB** – seguendo le linee guida tracciate nelle sezioni 2.1–2.3 e attenendoci allo pseudo-codice pubblicato dagli autori originali [3].

```
% Setting default values
if ~exist('d', 'var')
        d = 2;
end
if ~exist('K', 'var')
        K = 12;
end

% Extracting frequently used parameteres
[D, N] = size(X);

% STEP 1:
% COMPUTING EUCLIDEAN DISTANCES & FINDING K-NEAREST NEIGHBORS

SQ = sum(X.^2,1);
ED = repmat(SQ,N,1) + repmat(SQ',1,N) - 2*(X')*X;

[~, INDXS] = sort(ED);
NGHBS = INDXS(2:(1+K), :);
```

La funzione che si vuole realizzare prenderà in ingresso tre parametri: la matrice $\mathbf{X} \in \mathbb{R}^{D \times N}$ dei campioni supposti giacenti su una varietà non-lineare, il numero d di dimensioni a cui si auspica di ridurre la varietà campionata ed il numero K di vicini che si vogliono considerare per ciascun campione. Per buona pratica, nelle prime righe di codice, si sono assegnati valori di default a questi ultimi due parametri e si sono estratti il numero N di campioni e la loro dimensionalità D, poiché verranno utilizzati di frequente in seguito.

Notando che la differenza quadrata tra due vettori $(\mathbf{x} - \mathbf{y})^2$ è fattorizzabile come $\mathbf{x}\mathbf{x}^\mathbf{T} + \mathbf{y}\mathbf{y}^\mathbf{T} - 2\mathbf{x}\mathbf{y}^\mathbf{T} = \|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 - 2\mathbf{x}\mathbf{y}^\mathbf{T}$, diventa possibile costruire la matrice $\mathbf{ED}$ delle distanze euclidee impiegando esclusivamente il calcolo matriciale [5].

[5] L'informazione che si ricava dal quadrato della distanza è la stessa che si otterrebbe dal calcolo della sua radice, con il vantaggio di un'operazione in meno.

Per ricavare i K elementi più vicini ad ogni campione basta ordinare in senso crescente gli elementi di **ED** e conservare solo i primi K.

```
% STEP 2:
% COMPUTING OPTIMAL RECOSTRUCTION WEIGHTS

% Cheking regularization
if(K > D)
          DELTA = 10e-5;
else
         DELTA = 0;
end

W = spalloc(N, N, K * N);
for i = 1 : N
         XN = X(:, NGHBS(:, i)) - repmat(X(:, i), 1, K);
         G = XN' * XN;
         G = G + eye(K, K) * DELTA * trace(G);
         w = G\ones(K, 1);
         W(i, NGHBS(:, i)) = w/sum(w);
end
```

L'implementazione degli ultimi due passi è abbastanza diretta perchè segue la descrizione matematica fornita nelle sezioni 2.2-2.3 [6].

```
% STEP 3:
% COMPUTING OPTIMAL EMBEDDING
IW = spdiags(ones(N, 1), 0, N, N) - W;
M = (IW)' * (IW);

% Computing embedding
[YT, ~] = eigs(
                M,
                d+1,
                0,
                'Display', 0,
                'StartVector', ones(size(M, 1), 1)
          );
Y = YT(:, 2:d+1)' * sqrt(N);
```

[6] Un attento lettore potrebbe notare la discrepanza tra la discussione teorica e le ultime righe di codice, dove si sono calcolati, non i d, ma i $d+1$ autovalori più piccoli di $\mathbf{M}$. In realtà, vengono comunque considerati sempre solo d autovalori, ma si scarta il più piccolo perché sempre nullo. Infatti, per il vincolo (2): $\mathbf{M1} = (\mathbf{I}-\mathbf{W})\cdot(\mathbf{I}-\mathbf{W})^{\mathbf{T}}\cdot\mathbf{1} = 0$

3.1 Complessità

L'analisi della complessità temporale totale dell'algoritmo si riconduce all'analisi disgiunta dei singoli step, di cui la complessità finale è la somma. Si è scelto di impostare l'analisi attorno alla variabile N di campioni e di considerare fisse le altre quantità in gioco come K e D.

STEP I. Il costo della somma dei quadrati di ogni elemento, di ogni colonna, di una matrice $\mathbf{X} \in \mathbb{R}^{N \times D}$ è $O(ND) = O(N)$, mentre l'ordinamento di una matrice di N vettori di N elementi, sia in ordine crescente che decrescente, è di $O(N^2 \log(N))$. L'operazione asintoticamente più dispendiosa è quindi la seconda [7].

STEP II. La risoluzione di N sistemi lineari di K equazioni in K incognite tramite metodi diretti, come il Metodo di eliminazione di Gauss, richiede un tempo $O(NK^3) = O(N)$.

STEP III. La ricerca dei d autovalori più piccoli di una matrice $N \times N$ reale e simmetrica è messa in pratica da **MATLAB** per mezzo dell'algoritmo di Krylov-Arnoldi, la cui complessità media è $O(dN^2) = O(N^2)$.

[7] L'algoritmo adottato da **MATLAB** per l'ordinamento è il Quicksort, di cui è nota la complessità temporale: $O(N \log(N))$.

4 Valutazione dell'algoritmo

Nella letteratura pertinente agli algoritmi di riduzione dimensionale non è raro esaminare la validità di un nuovo metodo con set di dati artificiali di cui si conoscono la geometria e l'esito ideale. In questa sezione si vogliono presentare gli effetti del metodo finora sviluppato su alcune varietà bidimensionali *sviluppabili* in $\mathbb{R}^3$ dal momento che sono di semplice visualizzazione e permettono la conoscenza a priori del risultato desiderato[8]. Si analizzeranno inoltre le performance al variare di parametri come il numero di campioni N, il numero di vicini K ed il fattore di regolarizzazione Δ.

4.1 Swiss Roll

Si consideri la varietà mostrata in **Figura 4** e definita in forma parametrica dalle seguenti equazioni [6]:

$$\begin{cases} x = \left(\frac{3\pi}{2}(1+2t)\right)\cos\left(\frac{3\pi}{2}(1+2t)\right) \\ y = s \\ z = \left(\frac{3\pi}{2}(1+2t)\right)\sin\left(\frac{3\pi}{2}(1+2t)\right) \end{cases} \qquad 0 \leqslant s \leqslant L, \;\; 0 \leqslant t \leqslant 1.$$

e a cui è stato applicato un gradiente di colore che consente di verificare la conservazione delle relazioni di distanza tra i punti dalla dimensione di origine al piano.

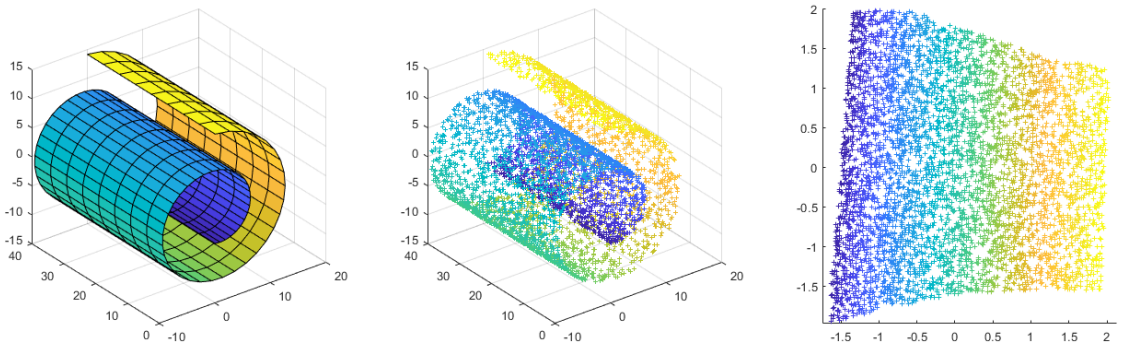


Figura 4: Risultato dell'algoritmo per $N = 5000$ e $K = 12$.

[8] Per quanto concerne la geometria differenziale, una superficie sviluppabile è una superficie liscia con curvatura Gaussiana zero, ovvero: può essere appiattita su un piano senza distorsioni. Il risultato ideale della riduzione dimensionale di queste superfici è dunque un piano.

La conservazione del gradiente e la forma quasi-rettangolare della varietà ridotta indicano proprio il successo dell'algoritmo. L'allontanamento dal risultato teorico di piano perfetto è da ricercarsi in non-idealità, quali il campionamento non uniforme o rumoroso, la scelta non ottimale del numero di vicini, la sensibilità al parametro Δ, ecc.

4.2 S-Curve

Ripetiamo ora l'applicazione su un secondo data-set molto noto, che prende forma in **Figura 5** e che può essere descritto dalle seguenti equazioni parametriche:

$$\begin{cases} x = -\cos(1.5\pi t) \\ y = s \\ z = \begin{cases} -\sin(1.5\pi t) & 0 \leqslant t \leqslant 1 \\ 2 + \sin(1.5\pi t) & 1 < t \leqslant 2 \end{cases} \end{cases} \qquad 0 \leqslant s \leqslant L, \;\; 0 \leqslant t \leqslant 2.$$

Sebbene l'insieme di dati usato per questo esempio sia molto differente da quello precedente, il metodo applicato conduce ad un risultato simile. Per una riduzione tendente a quella ideale, i risultati sarebbero indistinguibili. *Più insiemi possono quindi generare il medesimo embedding.* Se volessimo per qualche ragione ricostruire la varietà di partenza a partire da un embedding, inciamperemmo nell'incertezza della non-unicità della soluzione. Questa osservazione all'apparenza banale getta luce su un fatto importante: nel passaggio da dimensioni elevate a dimensioni inferiori, si perde il contenuto geometrico dell'insieme di dati di partenza a favore di conformazioni più comprensibili all'occhio umano.

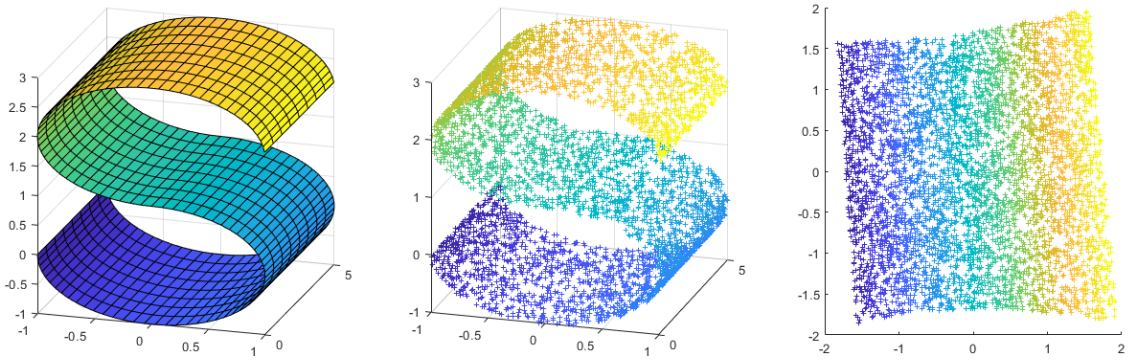


Figura 5: Risultato dell'algoritmo per $N = 5000$ e $K = 12$.

4.3 Superfici non sviluppabili

Riprendiamo le equazioni parametriche della sfera introdotte nel paragrafo 1.1. Facendo variare il parametro θ nel nuovo intervallo $[\phi, 2\pi)$, con $0 \leq \phi < 2\pi$, si possono ottenere figure interessanti, come la *sfera bucata* ed il *paraboloide ellittico*.

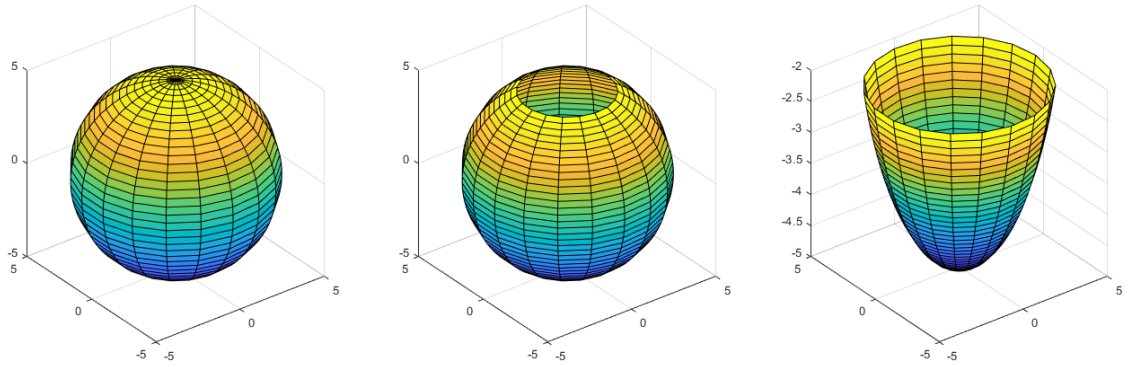
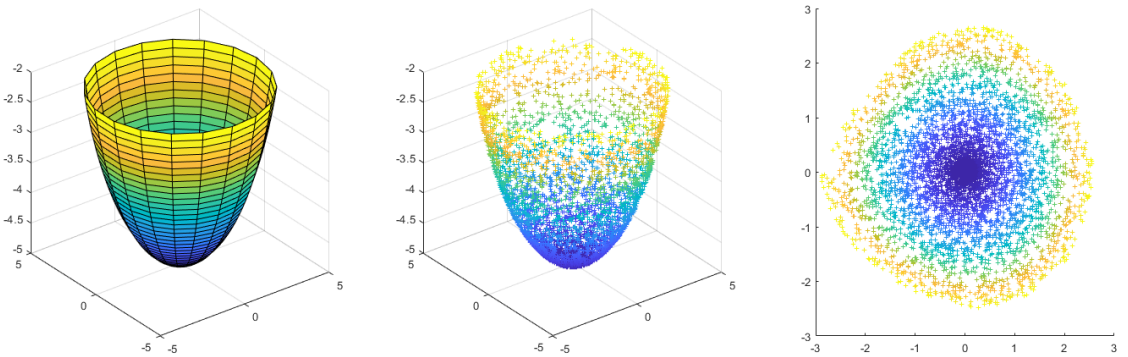


Figura 6: Dalla sfera al paraboloide ellittico: $\phi = 0$, $\phi = 0.5$, $\phi = 2$.

Queste superfici, pur non essendo sviluppabili nel senso dei due esempi precedenti, sono comunque abbastanza semplici da consentire la conoscenza a priori di una loro riduzione dimensionale. Si è visto infatti che l'effetto della riduzione è uno di *srotolamento* ed *appiattimento*. Per un paraboloide ellittico non è difficile immaginare che l'embedding desiderabile sia una circonferenza, come mostrato nella figura seguente



dove è ora evidente un fenomeno di *clustering* nel campionamento vicino al vertice, dovuto al nostro tentativo di campionare uniformemente la varietà lungo direzioni in cui questa varia non-linearmente: ad intervalli costanti, per esempio, lungo la direzione dell'asse **x**, corrisponderanno intervalli sempre più grandi lungo l'asse **z**.

4.4 Sensibilità al parametro N

Si vuole ora studiare la performance dell'algoritmo al variare della quantità di campioni forniti. Non è difficile sostenere che un buon risultato dipenderà sia dalla densità di campionamento che dal rapporto tra l'area campionata e l'area effettiva della varietà: un numero troppo basso di campioni risulterà in una perdita significativa di dettagli geometrici, un numero troppo elevato richiederà invece una potenza computazionale eccessiva. Viceversa, si può trattare la variazione dei campioni considerati ad ogni esecuzione dell'algoritmo, e quindi quella dell'area campionata, come conseguenza di un rumore casuale di campionamento attorno a dei campioni stabiliti. Infatti, fissato un insieme di dati, l'embedding prodotto sarà sempre identico perché sono deterministici i calcoli che lo generano. Variare la posizione dei campioni di poco in un intorno dei punti fissati può produrre effetti significativi, come mostrano le prove eseguite di seguito sulla curva ad S, per cui si è operata una campionatura casuale su un numero costante di campioni $N = 5000$.

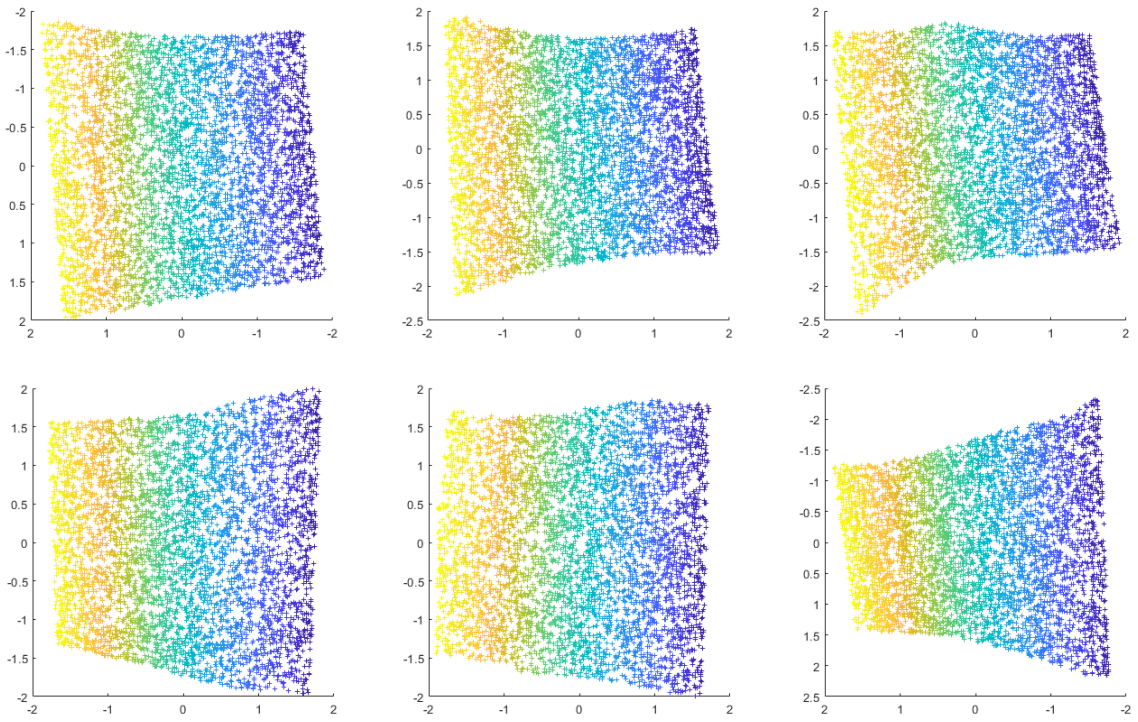
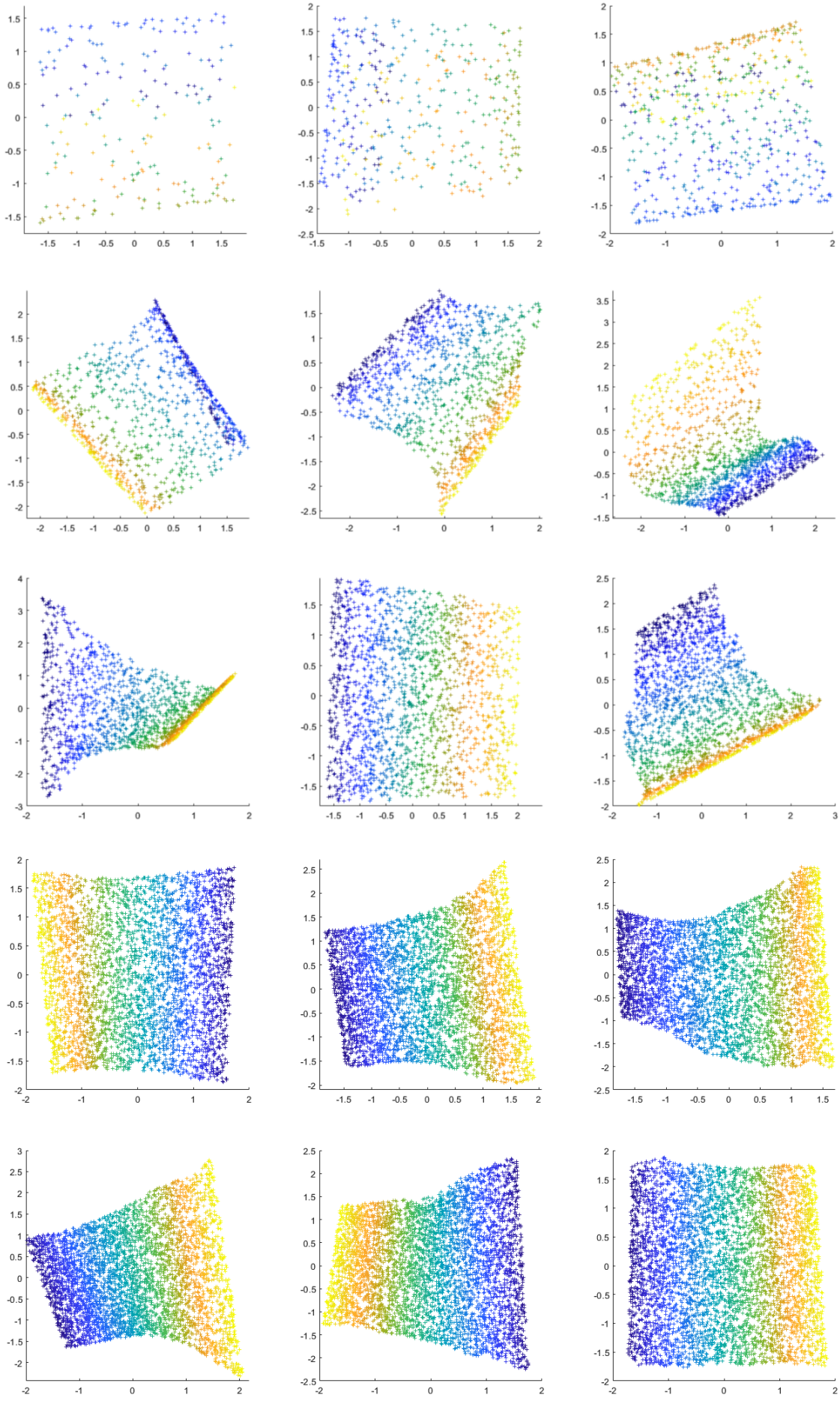


Figura 7: Risultato dell'algoritmo per $N = 5000$ al variare della posizione dei campioni selezionati.

È chiaro perciò che il nostro metodo è altamente sensibile al rumore di campionamento. Nella pagina seguente è anche possibile studiare il comportamento dell'algoritmo al variare del numero di campioni da $N = 200$ ad $N = 5000$.

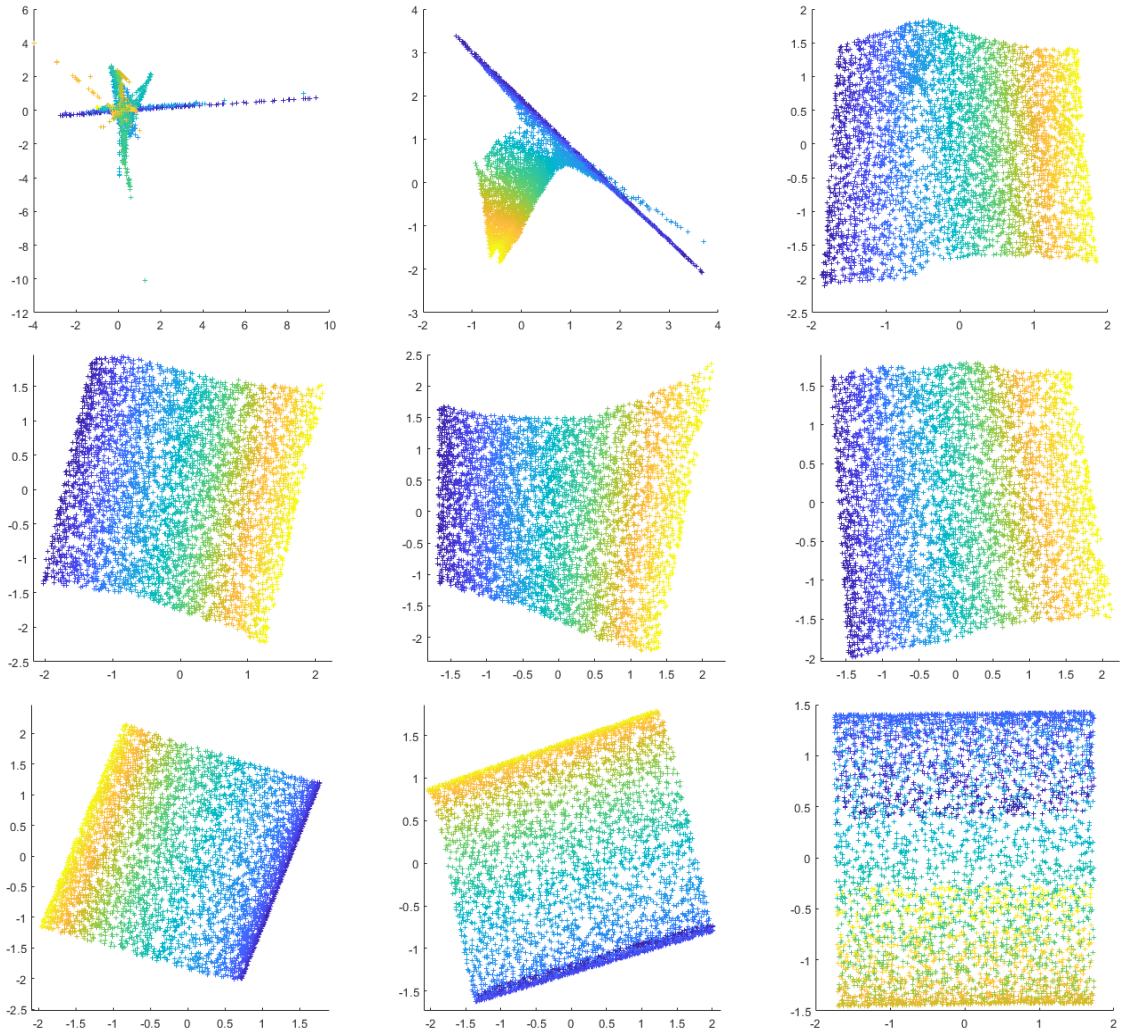


4.5 Sensibilità al parametro K

Si considerino i due casi estremi:

1. $K = 1$.
2. $K \to \infty$

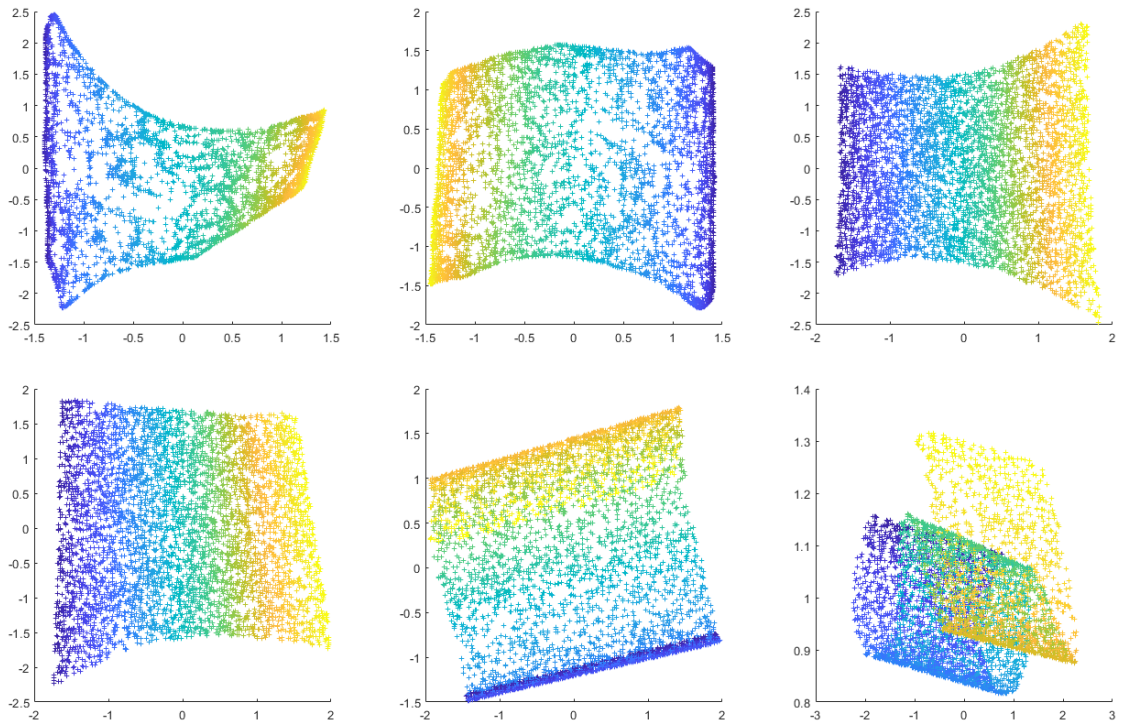
Nella prima evenienza si conserva solamente la distanza tra ogni punto e il suo campione più vicino, e l'algoritmo perde di senso ed efficacia. Nel secondo caso la varietà viene totalmente linearizzata e l'LLE degenera ad un algoritmo di riduzione lineare. Ovviamente i valori ragionevoli di K si troveranno nel mezzo. Per dimostrare l'importanza della corretta scelta di questo parametro, si propongono nella seguente figura le prove eseguite per $K \in \{4, 6, 8, 12, 20, 40, 80, 100, 200\}$ ed $N = 5000$.



Si può vedere come valori di K troppo piccoli non consentano all'algoritmo di apprendere la reale forma della varietà, tanto meno operarne una riduzione dimensionale. Viceversa, valori troppo elevati danno origine, nell'embedding, a fenomeni di clustering. In particolare, il gradiente di colore non è più correttamente mantenuto.

4.6 Sensibilità al parametro Δ

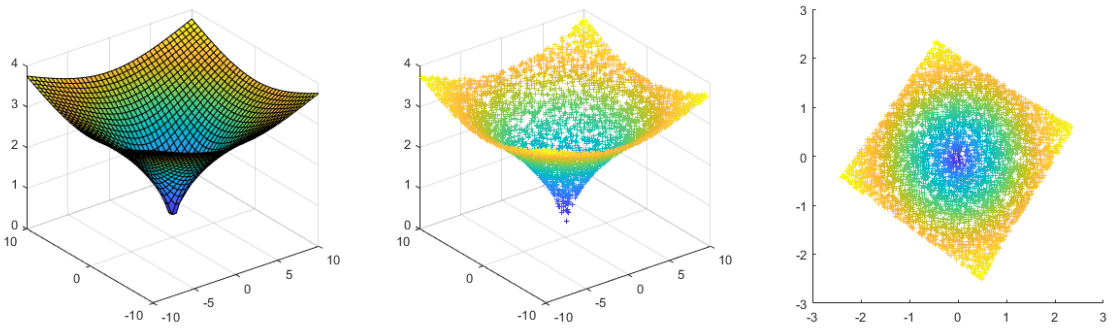
La determinazione del fattore di regolarizzazione può essere guidata solamente da risultati empirici e sperimentali dal momento che il suo impatto sul successo dell'algoritmo non è dettato da relazioni dirette con la varietà che si vuole ridurre. Per questo non è comune trovare trattazioni che discutano l'effetto della sua variazione, ma solo stime che variano tra 10^{-3} e 10^{-6}. È opportuno rivalutare il valore di questo fattore di volta in volta, sulla base di elementi come la dimensione del data-set e l'uniformità del campionamento. Di seguito sono mostrati i risultati delle prove svolte per $\Delta \in \{10^{-1}, 10^{-2}, 10^{-4}, 10^{-5}, 10^{-7}, 10^{-9}\}$, $N = 5000$ e $K = 12$.



Questo dimostra come si possano ottenere risultati meno che ideali anche con una selezione accurata degli altri due parametri N e K.

5 Osservazioni finali

In questo progetto ci si è limitati a data-set artificiali, dati da campioni prelevati da superfici abbastanza regolari – ad esempio differenziabili, come le varietà discusse nei paragrafi introduttivi. Si può però constatare che il metodo presentato è applicabile anche a varietà non differenziabili, come quella proposta nella figura seguente



data dalla rotazione attorno all'asse **z** della funzione $f(x) = \sqrt[2]{|x|}$. Questo accade perché l'algoritmo non pone condizioni sulla varietà in ingresso: sono sufficienti un numero elevato di campioni e la corretta scelta del parametro K. Il limite è dato da varietà intricate, che si intrecciano o si interrompono a tratti.

6 Conclusioni

L'algoritmo di riduzione non-lineare presentato in questo progetto ha il vantaggio di non comportare lo studio di minimi locali, di essere non-iterativo e di richiedere pochi parametri in ingresso: di fatto, l'unica euristica richiesta da un eventuale utente è la dimensione del vicinato K, gli altri parametri dipendono esplicitamente dal data-set. Questi fattori, insieme all'utilizzo di matrici sparse, rendono l'esecuzione dell'algoritmo più veloce di quello di sue controparti, come **ISOMAP**. Dai test eseguiti sulla curva ad S in questi ultimi paragrafi si possono tirare le seguenti conclusioni riguardo ai valori più idonei per i tre parametri discussi:

1. $2000 \leq N \leq 5000$,
2. $8 \leq K \leq 40$,
3. $10^{-4} \leq \Delta \leq 10^{-5}$.

Valori di N e K prossimi agli estremi inferiori offrono il beneficio di una minore potenza computazionale, mentre valori mediani presentano risultati più consistenti tra prove successive. Bisogna notare che queste linee guida dipendono completamente dall'insieme di dati di partenza. Come per ogni problema di apprendimento, il numero di campioni è *piccolo* o *grande* solo in relazione alla dimensione del problema che si cerca di risolvere.

Si è visto inoltre che i limiti esibiti dall'algoritmo sono principalmente:

1. l' estrema sensibilità alle fonti di rumore o di disturbo,
2. la dipendenza dei parametri dalla qualità del data-set in ingresso.

Riferimenti bibliografici

[1] Clyde Martin Brian F. Doolin. *Introduction to differential geometry for engineers*. Dover Civil and Mechanical Engineering. Dover Publications, 1990.

[2] F. Tovena Marco Abate. *Geometria Differenziale*. La Matematica per il $3 + 2$. Springer-Verlag Mailand, 2011.

[3] Lawrence K. Saul Roweis T. Sam. LLE algorithm pseudocode. `https://cs.nyu.edu/~roweis/lle/algorithm.html`.

[4] Lawrence K. Saul Roweis T. Sam. An introduction to locally linear embedding. 2001.

[5] Lawrence K. Saul Roweis T. Sam. Think globally, fit locally: Unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research*, 4:119–155, 2003.

[6] Jianzhong Wang. *Geometric Structure of High-Dimensional Data and Dimensionality Reduction*. Springer-Verlag Berlin Heidelberg, 2011.