Università di Pisa

Facoltà di Matematica

Stimatori della dimensione intrinseca a confronto

Tesi di Laurea Triennale
in Matematica

Anno Accademico 2021/2022

Candidata
Eleonora Basilico

Relatore
Dario Trevisan
Università di Pisa

Anno Accademico 2021 - 2022

Indice

Abstract

Il manifold learning è una branca del machine learning ormai ben nota e in rapida espansione. Esso consiste in un insieme di tecniche che consentono a chi le usa di trovare informazioni interessanti ed utili sul dataset che si sta studiando. Più precisamente, le tecniche e gli algoritmi di manifold learning si basano sull'assunzione che i dati osservati siano campionati da una varietà di dimensione bassa immersa tramite embedding in uno spazio euclideo di dimensione più elevata. L'obiettivo del manifold learning è quindi quello di capire di quale varietà si tratta, o almeno di determinarne la dimensione, al fine di trovare una rappresentazione più semplice dei dati. La dimensione di tale varietà è detta *dimensione intrinseca del dataset* e il problema di determinarne il valore è noto come problema di *stima della dimensione intrinseca di un dataset*.

In questa tesi trattiamo il problema di stima della dimensione intrinseca di un dataset e analizziamo in dettaglio tre diversi stimatori, con l'obiettivo di studiarne le proprietà matematiche e di confrontarne la performance tramite sperimentazioni su datasets sia reali che generati.

Il primo capitolo contiene una breve introduzione al manifold learning e alle sue principali applicazioni.

Nel secondo capitolo viene trattato più in dettaglio il problema di stima della dimensione intrinseca di un dataset e vengono descritti due stimatori paradigmatici: CorrDim (Correlation Dimension) e PCA (Principal Component Analysis). Essi sono due buoni rappresentanti delle due principali classi esistenti di stimatori: *stimatori geometrici e stimatori proiettivi*. Vedremo, tuttavia, che CorrDim e PCA soffrono di due difetti tra loro complementari: il primo soffre del cosiddetto *curse of dimensionality*, ovvero per funzionare correttamente necessita di un numero esponenziale di punti (nella dimensione intrinseca d), cosa che si traduce in una sistematica sottostima del valore della dimensione intrinseca quando questa diventa elevata; il secondo, invece, tende a sovrastimare la dimensione intrinseca delle varietà curve.

Infine, il terzo capitolo è interamente dedicato allo studio di uno stimatore introdotto di recente, il *Full correlation integral*, il quale estende lo stimatore CorrDim con l'obiettivo di alleviare gli effetti del curse of dimensionality. In particolare, vedremo che il full correlation integral sembra essere uno stimatore robusto ed affidabile anche su datasets che non rispettano appieno le condizioni richieste, oppure in casi di forte undersampling. Vedremo, infine, che anche questo stimatore ha alcune limitazioni ma risulta spesso più preciso degli stimatori CorrDim e PCA.

Capitolo 1

Manifold learning e sue applicazioni

Il manifold learning è una branca del machine learning ormai ben nota e in rapida espansione. Questo capitolo contiene una breve introduzione al manifold learning. In particolare, vedremo qual è l'idea principale sulla quale esso si basa, diremo perchè viene utilizzato e quali sono le sue applicazioni più importanti. Definiremo poi un modello matematico di dataset, il cosiddetto *manifold model*, che sarà utilizzato nei capitoli successivi. Concluderemo, infine, con una lista di datasets artificiali che saranno poi impiegati più avanti nelle sperimentazioni.

1.1 Perchè il manifold learning

Per studiare un dataset la visualizzazione dei dati è di cruciale importanza: essa permette ai ricercatori di determinare l'eventuale presenza all'interno dei dati di trends generali, e di giungere così a conclusioni oppure elaborare nuove teorie.
Supponiamo ad esempio di avere un dataset costituito da 100 punti in $\mathbb{R}^3$, sul quale non abbiamo alcuna conoscenza a priori. Per ottenere informazioni utili a riguardo, la prima cosa da fare è visualizzarlo graficamente: possiamo così renderci conto che i punti del dataset sono collocati (ad esempio) sulla superficie di una sfera, e realizzare che possiamo ottenere informazioni interessanti sul dataset stimando il raggio della sfera o grandezze simili.
Cosa succede però se il nostro dataset è costituito da 60000 punti in 784 dimensioni? Al fine di studiare datasets in dimensioni elevate, abbiamo bisogno di strumenti analitici e computazionali che effettuino al nostro posto un'analisi qualitativa dei dati. Infatti, non potendo visualizzare il dataset graficamente, trovare informazioni utili risulta molto più difficile. Non solo, molti algoritmi soffrono anche del cosiddetto "curse of dimensionality", ossia la loro performance peggiora in dimensioni elevate.

Da quanto detto segue la necessità del *manifold learning*, con cui si intende un insieme di tecniche che consentono a chi le usa di trovare informazioni interessanti sul dataset che si sta studiando. Più precisamente, le tecniche e gli algoritmi di

manifold learning si basano, come vedremo in dettaglio più avanti, sull'assunzione che i dati osservati siano campionati da una varietà di dimensione bassa immersa tramite embedding in uno spazio euclideo di dimensione più elevata. L'obiettivo del manifold learning è quindi quello di capire di quale varietà si tratta, o almeno di determinarne la dimensione, al fine di trovare una rappresentazione più semplice dei dati.

Osservazione 1. Non bisogna fare l'errore di pensare che il manifold learning risolva del tutto il problema di visualizzazione dei datasets in alte dimensioni, e che ci permetta di ottenere una visione completa delle proprietà geometriche dei datasets. Infatti, spesso le tecniche di manifold learning risultano efficaci solo sotto determinate assunzioni sul dataset, che nella pratica non possono essere verificate o sono addirittura violate.

1.2 Alcune applicazioni

Riportiamo di seguito le principali applicazioni del manifold learning:

- **Stima della dimensione intrinseca:** Il problema di *stima della dimensione intrinseca* di un dataset consiste nell'identificare il minimo numero di parametri necessari a descriverlo interamente. Vediamo alcuni esempi. Supponiamo di avere dei dati tridimensionali che giacciono su una curva. In questo caso, due delle tre dimensioni sono effettivamente non necessarie a descrivere il dataset. Oppure, consideriamo un dataset costituito da immagini di aquile in cielo. Molti dei pixels che rappresentano il cielo saranno colorati di azzurro e saranno del tutto non informativi. Stimare la dimensione intrinseca di questi due datasets consiste nello stimare il minimo numero di dimensioni/pixels di cui abbiamo bisogno per catturare completamente la variabilità del dataset. Tratteremo in dettaglio il problema di stima della dimensione intrinseca nel prossimo capitolo.

- **Riduzione dimensionale:** Una volta stimato il numero di gradi di libertà necessari a descrivere il dataset, potremmo voler stimare quali sono le dimensioni rilevanti del dataset, oppure quali pixels contengono più informazione. Tornando all'esempio del dataset tridimensionale introdotto nel punto precedente, potremmo ad esempio voler trovare un punto della curva da prendere come origine e introdurre sulla curva una nozione di distanza in modo che ciascun punto del dataset possa effettivamente essere rappresentato tramite un solo valore, e cioè la sua distanza dall'origine lungo la curva. È in questi casi che si parla di *riduzione dimensionale*: essa consiste in una trasformazione dei dati da uno spazio ad alta dimensione in uno spazio di dimensione minore, in modo che quest'ultima mantenga alcune proprietà significative dei dati originali.

- **Clustering:** Obiettivo ben diverso dai precedenti è quello di capire se il proprio dataset è costituito da più nuvole distinte di punti, dette *clusters*. Il *clustering* ha l'obiettivo di riconoscere se il proprio dataset è costituito da clusters, e di fornire la regola che stabilisce se un certo dato è da collocare in un determinato cluster oppure in un altro. Esistono varie tecniche di

manifold learning impiegate per effettuare il clustering, che può essere quindi inserito tra le sue svariate applicazioni.

1.3 Un modello matematico di dataset

Per iniziare a sviluppare algoritmi di manifold learning abbiamo bisogno di un modello mattematico che rappresenti cosa sia un dataset. Introduciamo quindi il cosiddetto *manifold model* (da cui il nome di manifold learning). Il manifold model è un modello matematico costruttivo, ovvero un modello che caratterizza i datasets dicendo in che modo questi vengono generati in primo luogo.

Facciamo le seguenti assunzioni:

- Assumiamo che esista una varietà liscia d-dimensionale M, detta *varietà intrinseca*, sulla quale è definita una misura di probabilità μ. Tale varietà descrive l'informazione geometrica contenuta all'interno del dataset, ovvero l'informazione minima necessaria per riprodurlo;
- Assumiamo, inoltre, che esista una mappa sufficientemente regolare (diciamo liscia e iniettiva) dalla varietà M a valori in uno spazio euclideo D-dimensionale, con $D \geq d$. Chiameremo tale mappa *embedding*[1].

Diciamo allora che il dataset viene generato campionando dei punti i.i.d. sulla varietà intrinseca M secondo la distribuzione di probabilità μ, e mappando ciascuno di questi punti nello spazio euclideo tramite l'embedding. Inoltre, diremo che d è la *dimensione intrinseca* del dataset e che D è la *dimensione di embedding*.

In questo modello di dataset sono coinvolti tre fattori: la varietà intrinseca, che può risultare non banale; la distribuzione di probabilità ad essa associata, che può essere altamente non uniforme; ed infine l'embedding. Noi ci restringeremo a considerare distribuzioni di probabilità semplici e vedremo, invece, esempi non banali di varietà intrinseche ed embedding.

Nota 1. Notiamo che il manifold model può essere facilmente esteso a *datasets multidimensionali*, dati dall'unione di più datasets con dimensioni intrinseche diverse. In questo caso ogni componente del dataset è descritta dal modello separatamente, ma lo spazio di arrivo degli embedding è comune a ciascuna componente.

1.3.1 Perchè il manifold model

Cerchiamo di vedere con un esempio perchè ha senso usare il manifold model. Consideriamo il dataset MNIST: esso è costituito da 60000 immagini di 28×28 pixels a scala di grigi, ciascuna contenente una cifra tra 0 e 9 scritta a mano. Consideriamo, in particolare, il sottoinsieme costituito dalle sole cifre "uno". L'idea che tale sottoinsieme sia generato a partire da una varietà intrinseca proviene dall'intuizione che esiste un'immagine prototipo di "uno", e che ogni altra immagine

[1] In questa sezione compaiono diversi concetti tipici della geometria differenziale. Non entreremo nel dettaglio, in quanto ciò esula dal nostro obiettivo. Ad esempio, il termine embedding ha un significato ben preciso in geometria differenziale ma noi lo utilizzeremo in modo informale. Si guardi [PM13] per una trattazione più precisa.

può essere generata a partire da questa tramite trasformazioni lisce, quali rotazioni, traslazioni, riscalature, ecc. La varietà intrinseca è quindi generata da questo insieme di trasformazioni.
A questo punto è chiaro che la varietà intrinseca abbia una dimensione più piccola del numero di pixels utilizzati per rappresentare le immagini, quindi deve esistere un embedding che colleghi la "rappresentazione intrinseca" di ciascuna immagine al corrispondente dato osservato. Tale embedding può dipendere da come decidiamo di rappresentare i dati (per esempio dalla risoluzione delle immagini), e può contenere del rumore che va a corrompere i nostri dati.

1.4 Esempi di datasets artificiali

Riportiamo di seguito una lista di datasets creati artificialmente. La Figura 1.1 fornisce una rappresentazione grafica di alcuni di questi.

- **Datasets lineari:** In un dataset lineare la varietà intrinseca è data da un sottoinsieme di $\mathbb{R}^d$ di dimensione massima con distribuzione di probabilità uniforme. L'esempio più comune è l'ipercubo d-dimensionale $\mathcal{H}_d = [-1, 1]^d$. L'embedding è una mappa lineare da $\mathbb{R}^d$ in $\mathbb{R}^D$, ad esempio l'inclusione naturale oppure la mappa di inclusione seguita da una rotazione.

- **Datasets digitali:** In un dataset digitale la varietà intrinseca è l'insieme dei vertici di un ipercubo d-dimensionale dotato di distribuzione di probabilità uniforme. L'embedding è una mappa lineare da $\mathbb{R}^d$ in $\mathbb{R}^D$.

- **Datasets sferici:** In un dataset sferico la varietà intrinseca è la sfera d-dimensionale $S^d = \{x \in \mathbb{R}^{d+1} | \, ||x|| = 1\}$ dotato di distribuzione di probabilità uniforme. L'embedding è dato dalla composizione dell'inclusione naturale di S^d in $\mathbb{R}^{d+1}$ e di una mappa lineare da $\mathbb{R}^{d+1}$ in $\mathbb{R}^D$.

- **Datasets gaussiani:** In un dataset gaussiano la varietà intrinseca è $\mathbb{R}^d$ dotato di distribuzione gaussiana multivariata standard, avente densità

$$p_{Gauss}(x) = \frac{e^{-\frac{1}{2}||x||^2}}{\sqrt{(2\pi)^d}}. \tag{1.1}$$

  L'embedding è dato da una mappa lineare da $\mathbb{R}^d$ in $\mathbb{R}^D$.

- **Swiss roll dataset:** La varietà intrinseca è $[0, 1]^2$ dotato di distribuzione di probabilità uniforme. L'embedding è il seguente:

$$\phi(x, y) = (x \cos(2\pi x), y, x \sin(2\pi x)). \tag{1.2}$$

- **Hein dataset:** La varietà intrinseca del dataset è l'ipercubo d-dimensionale $[0, 2\pi]^d$ dotato della distribuzione uniforme. L'embedding è il seguente:

$$\phi(x_1, ..., x_d) = (x_2 \cos(x_1), x_2 \sin(x_1), ..., x_1 \cos(x_d), x_1 \sin(x_d)), \tag{1.3}$$

  e la minima dimensione di embedding è $D = 2d$. Dimensioni di embedding più alte possono essere considerate componendo ϕ con una funzione lineare da $\mathbb{R}^{2d}$ in $\mathbb{R}^D$. Il dataset è caratterizzato dalla presenza di una curvatura non banale.

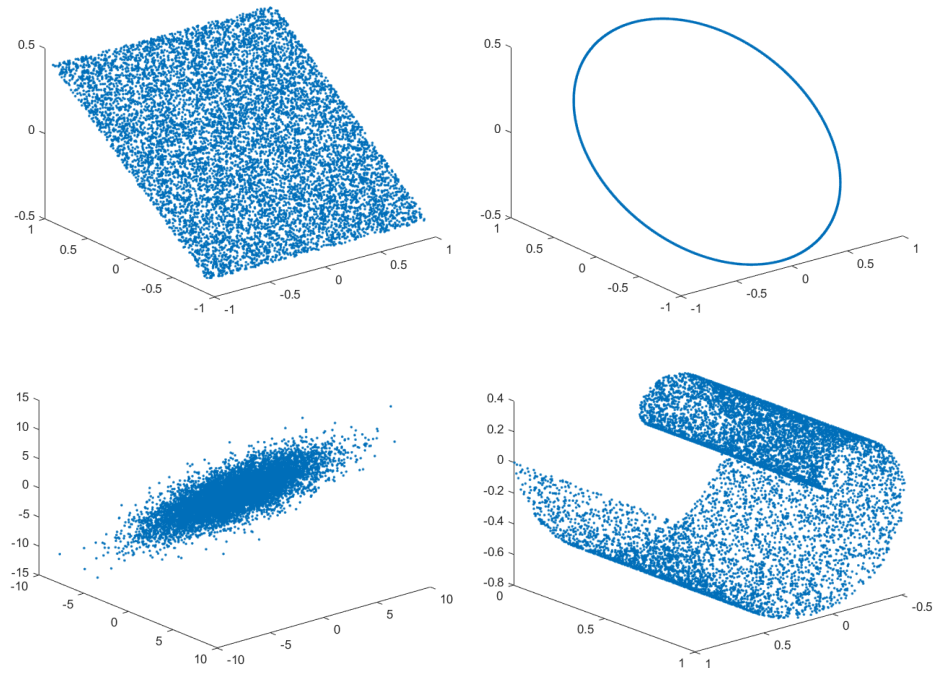

Figura 1.1: Esempi di datasets artificiali. (In alto a sinistra) Dataset lineare con dimensione intrinseca $d = 2$ e dimensione di embedding $D = 3$. (In alto a destra) Dataset sferico con dimensione intrinseca $d = 1$ e dimensione di embedding $D = 3$. (In basso a sinistra) Dataset gaussiano con dimensione intrinseca $d = 2$ e dimensione di embedding $D = 3$. (In basso a destra) Swiss roll dataset con dimensione intrinseca $d = 2$ e dimensione di embedding $D = 3$. Tutte le figure sono costituite da $N = 10000$ punti.

Capitolo 2

Stima della dimensione intrinseca

Abbiamo visto che il manifold learning è uno strumento molto utile, ricco di possibili applicazioni, e che il manifold model ci fornisce una vasta gamma di datasets artificiali con cui lavorare. In questo capitolo descriveremo più in dettaglio il problema della stima della dimensione intrinseca di un dataset e studieremo due stimatori paradigmatici: CorrDim (Correlation Dimension) e PCA (Principal Component Analysis). Questi due stimatori non solo sono semplici da capire e da implementare ma sono anche due buoni rappresentanti delle due principali classi esistenti di stimatori: *stimatori geometrici* e *stimatori proiettivi* (discuteremo brevemente di questa classificazione al termine del capitolo). Vedrermo, infine, che CorrDim e PCA soffrono di due difetti tra loro complementari.

2.1 Descrizione del problema

Consideriamo per semplicità il caso di datasets artificiali: supponiamo di avere un dataset costituito da un insieme di N punti in $\mathbb{R}^D$ generati tramite il manifold model, con varietà intrinseca ed embedding sconosciuti (si noti che ciò è ovviamente equivalente a supporre che gli N punti del dataset siano collocati su una sottovarietà $M \subseteq \mathbb{R}^D$ a noi non nota). L'obiettivo è quello di trovare una stima del valore della dimensione intrinseca d che sia la più accurata possibile. L'idea è che quando uno stimatore funziona bene sui datasets artificiali, si può essere abbastanza fiduciosi del fatto che questo fornisca una stima ragionevole anche nel caso di datasets reali.

Il problema di trovare stime accurate di questo valore è stato affrontato più volte e in contesti diversi, quali la psicometria, la fisica e l'informatica. È stato riconsiderato più di recente con l'avvento dell'analisi dei big data e dell'intelligenza artificiale e sono stati proposti diversi stimatori. Iniziamo con l'analizzare lo stimatore CorrDim.

2.2 CorrDim

Lo stimatore CorrDim è stato introdotto per la prima volta nel 1983 da Grassberger e Procaccia [GP83]. In questa sezione ne descriveremo il funzionamento e ne dimostreremo la consistenza. Vedremo che CorrDim soffre di un importante difetto, noto in letteratura come *curse of dimensionality*. Infine, riporteremo alcune sperimentazioni che ne mettono in luce pregi e difetti.

2.2.1 Descrizione e consistenza

Indichiamo con $X = \{x_i\}_{i=1}^N$ il nostro dataset, e supponiamo che i punti del dataset siano collocati su una sottovarietà $d-$dimensionale $M \subseteq \mathbb{R}^D$, a noi non nota. Dato $r > 0$, consideriamo per ciascun punto $x_i \in X$ la quantità

$$\rho_i(r) = \frac{1}{N-1} \sum_{\substack{j=1 \\ j \neq i}}^{N} \theta(r - ||x_i - x_j||), \tag{2.1}$$

dove $\theta(x)$ è la funzione gradino di Heaviside, che vale 1 se x è positivo e 0 altrimenti. La quantità $\rho_i(r)$ misura la densità normalizzata di punti del dataset x_j, con $j \neq i$, distanti da x_i meno di r. Diamo ora la seguente definizione:

Definizione 2.2.1. Chiamiamo *correlation integral* la quantità

$$\rho(r) = \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq i}}^{N} \theta(r - ||x_i - x_j||). \tag{2.2}$$

$\rho(r)$ non è altro che la media sugli N punti del dataset delle quantità $\rho_i(r)$.

Nota 2. Più avanti, quando vorremo evidenziare la dipendenza del correlation integral dal numero N di punti del datset, scriveremo $\rho_N(r)$ anzichè $\rho(r)$.

Per stimare la dimensione intrinseca d del dataset lo stimatore CorrDim utilizza il correlation integral sfruttando la seguente relazione, di cui vedremo a breve la dimostrazione formale: per r piccolo, vale

$$\rho(r) \propto r^d, \tag{2.3}$$

(dove $\propto$ indica informalmente una proporzionalità asintotica). A questo punto risulta immediato stimare la dimensione intrinseca d: basta calcolare per r piccolo il valore del correlation integral $\rho(r)$ sul nostro dataset, ed eseguire un fit sul logaritmo dei punti $(r, \rho(r))$, in quanto in scala logaritmica il fit diventa lineare e la dimensione intrinseca va a coincidere con il coefficiente angolare della retta di interpolazione. Infatti, posto $r' = \log r$, sfruttando la relazione (2.3) si ottiene per r piccolo

$$\rho(e^{r'}) \propto (e^{r'})^d \Rightarrow \log \rho(e^{r'}) = \log(e^{r'})^d + c \Rightarrow \log \rho(r) = d \log r + c.$$

Passiamo ora a dimostrare formalmente la consistenza del nostro stimatore. A tal fine riscriveremo il problema in forma probabilistica. Inoltre, saranno utili i seguenti due risultati, dei quali riportiamo il solo enunciato:

Proposizione 2.2.1. *Se μ è una misura su $\mathbb{R}^D$ concentrata su una sottovarietà $d-$dimensionale $M \subseteq \mathbb{R}^D$ e avente densità liscia e strettamente positiva (rispetto alla misura superficie di M), allora per $\mu-$ q.o x vale:*

$$\lim_{r\to 0} \frac{\mu(B(x,r))}{r^d} = c(x) > 0. \tag{2.4}$$

Teorema 2.2.2. *(continuità dell'integrale) Siano (Λ, d) uno spazio metrico e (X, M, μ) uno spazio misurabile. Sia $E \in M$ e sia $f : \Lambda \times E \to \bar{\mathbb{R}}$ tale che:*

- *$f(\lambda, \cdot) : E \to \bar{\mathbb{R}}$ è misurabile per ogni $\lambda \in \Lambda$;*
- *$|f(\lambda, \cdot)| \leq g$ per ogni $\lambda \in \Lambda$, dove $g : E \to [0, +\infty]$ è integrabile;*
- *$\exists \lambda_0 \in \Lambda$ ed $\exists h : E \to \bar{\mathbb{R}}$ tali che per ogni $p \in E$ si ha $\lim_{\lambda\to\lambda_0} f(\lambda, p) = h(p)$.*

Allora h è integrabile e vale:

$$\lim_{\lambda\to\lambda_0} \int_E f(\lambda, p) d\mu(p) = \int_E h(p) d\mu(p). \tag{2.5}$$

Passiamo ora alla versione probabilistica del problema. Siano $\{X_i\}_{i=1}^N$ le variabili aleatorie che rappresentano gli N punti del nostro dataset: queste sono variabili aleatorie i.i.d. definite su $\mathbb{R}^D$ e aventi distribuzione μ su $\mathcal{B}(\mathbb{R}^D)$ con densità liscia e positiva sul supporto. Poiché assumiamo che i punti del nostro dataset siano collocati su una sottovarietà $d-$dimensionale $M \subseteq \mathbb{R}^D$, per la Proposizione 2.2.1 vale la proprietà (2.4).

Siamo pronti ora per dimostrare la consistenza dello stimatore CorrDim. Vale il seguente teorema:

Teorema 2.2.3. *Nelle ipotesi di cui sopra, quasi certamente*

$$\lim_{N\to\infty} \rho_N(r) = \int_{\mathbb{R}^D} \mu(B(x,r)) d\mu(x). \tag{2.6}$$

Quindi, se vale la proprietà (2.4) si ha che

$$\lim_{r\to 0} \frac{\log\left(\lim_{N\to\infty} \rho_N(r)\right)}{\log r} = d. \tag{2.7}$$

Dimostrazione. Per prima cosa riscriviamo il correlation integral $\rho_N(r)$ come variabile aleatoria:

$$\rho_N(r) = \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j\neq i}}^{N} \mathbb{1}_{\{||X_i - X_j|| < r\}}. \tag{2.8}$$

L'uguaglianza (2.6) è un'immediata conseguenza del seguente teorema più generale, per la cui dimostrazione si rimanda all'articolo [Hoe61]:

Teorema 2.2.4. *Sia $\{Y_i\}_{i\geq 1}$ una sequenza di variabili aleatorie i.i.d. a valori in uno spazio Y. Sia k un intero positivo e sia f una funzione reale misurabile definita su Y^k. Per $n \geq k$ definiamo $\bar{f}_n$ come la media aritmetica*

$$\bar{f}_n = \frac{1}{n(n-1)...(n-k+1)} \sum f(Y_{i_1}, ..., Y_{i_k}), \tag{2.9}$$

dove la somma è su tutte le $k-$uple $i_1, ..., i_k$ di interi positivi distinti minori di n. Se $\mathbb{E}[|f(Y_1, ..., Y_k)|] < \infty$, allora quasi certamente

$$\bar{f}_n \longrightarrow \mathbb{E}[f(Y_1, ..., Y_k)]. \tag{2.10}$$

Ponendo $Y_i = X_i$, $Y = \mathbb{R}^D$, $k = 2$, $n = N$ e $f(Y_{i_1}, Y_{i_2}) = \mathbb{1}_{\{||X_{i_1} - X_{i_2}||<r\}}$, si ottiene $\bar{f}_n = \rho_N(r)$; ed essendo

$$\mathbb{E}[|\,\mathbb{1}_{\{||X_1 - X_2||<r\}}|] = \int_{\mathbb{R}^D}\int_{\mathbb{R}^D} \mathbb{1}_{\{||x-y||<r\}} d\mu(y)d\mu(x) = \int_{\mathbb{R}^D} \mu(B(x,r))d\mu(x) < \infty,$$

possiamo applicare il Teorema 2.2.4 e la convergenza quasi certa (2.6) è immediata. Dimostriamo ora il limite (2.7). Poichè per ipotesi vale la proprietà (2.4), ovvero per $r \to 0$ vale

$$\mu(B(x,r)) = r^d(c(x) + o(1)), \tag{2.11}$$

per una certa $c(x) > 0$, allora per $r \to 0$

$$\begin{aligned}\frac{\log\left(\lim_{N\to\infty} \rho_N(r)\right)}{\log r} &= \frac{\log(\int_{\mathbb{R}^D} \mu(B(x,r))d\mu(x))}{\log r} = \\ &= \frac{\log(\int_{\mathbb{R}^D} r^d(c + o(1))d\mu(x))}{\log r} = \frac{\log(r^d \int_{\mathbb{R}^D}(c + o(1))d\mu(x))}{\log r} \\ &= \frac{\log r^d}{\log r} + \frac{\log(\int_{\mathbb{R}^D}(c + o(1))d\mu(x))}{\log r} = d,\end{aligned}$$

dove per la seconda uguaglianza utilizziamo la continuità del logaritmo e il Teorema 2.2.2. □

Osservazione 2. Se nell'espressione (2.6) si richiede la convergenza in probabilità anziché la convergenza quasi certa, per dimostrarla non è necessario passare per il Teorema 2.2.4, ma basta mostrare la convergenza in L^2, da cui segue per Chebyshev la convergenza in probabilità. Dimostriamo dunque la convergenza in L^2. Sia $A_{i,j} = \mathbb{1}_{\{||X_i - X_j||<r\}}$, con $i, j \in \{1, ..., N\}$. Allora possiamo scrivere:

$$\rho_N(r) = \frac{1}{N(N-1)} \sum_{i\neq j} A_{i,j}. \tag{2.12}$$

A questo punto ci basta notare che:

$$Var\Big(\frac{1}{N(N-1)} \sum_{i\neq j} A_{i,j}\Big) = \frac{1}{(N(N-1))^2} Var\Big(\sum_{i\neq j} A_{i,j}\Big) =$$

$$= \frac{1}{(N(N-1))^2}\Big(\sum_{i\neq j}\sum_{k\neq l} Cov(A_{i,j}, A_{k,l})\Big) \leq \frac{N^3 C}{N^4} \longrightarrow 0$$

per $N \to \infty$. La disuguaglianza finale nella precedente espressione discende dal fatto che $Cov(A_{i,j}, A_{k,l}) < C < \infty$ per ogni i, j, k, l e, in particolare, $Cov(A_{i,j}, A_{k,l}) = 0$ se $\{i, j\} \neq \{k, l\}$. Infatti, se $\{i, j\} \neq \{k, l\}$, allora $A_{i,j}$ e $A_{k,l}$ sono indipendenti, altrimenti vale:

$$Cov(A_{i,j}, A_{k,l}) = \mathbb{E}[A_{i,j}A_{k,l}] - \mathbb{E}[A_{i,j}]\mathbb{E}[A_{k,l}] \neq 0. \tag{2.13}$$

Ora:

$$\mathbb{E}[A_{i,j}] = \mathbb{E}[A_{k,l}] = \int_{\mathbb{R}^D} \mu(B(x,r))d\mu(x). \tag{2.14}$$

Inoltre, dati q e q' tali che $\frac{1}{q} + \frac{1}{q'} = 1$, per la disuguaglianza di Hölder vale:

$$\mathbb{E}[A_{i,j}A_{k,l}] \leq \mathbb{E}[A_{i,j}^{q'}]^{1/q'}\mathbb{E}[A_{k,l}^{q}]^{1/q} = \mathbb{E}[A_{i,j}]^{1/q'}\mathbb{E}[A_{k,l}]^{1/q} = \mathbb{E}[A_{i,j}]. \tag{2.15}$$

Segue che:

$$Cov(A_{i,j}, A_{k,l}) \leq \mathbb{E}[A_{i,j}] - \mathbb{E}[A_{i,j}]\mathbb{E}[A_{k,l}] = \mathbb{E}[A_{i,j}](1 - \mathbb{E}[A_{i,j}]) =$$

$$= \Big(\int_{\mathbb{R}^D} \mu(B(x,r))d\mu(x)\Big)\Big(1 - \int_{\mathbb{R}^D} \mu(B(x,r))d\mu(x)\Big) < C < \infty.$$

Poichè dunque

$$Var\Big(\frac{1}{N(N-1)}\sum_{i\neq j} A_{i,j}\Big) \longrightarrow 0, \tag{2.16}$$

si ha la convergenza in L^2, e quindi la convergenza in probabilità.

Osservazione 3. È possibile che si riesca a semplificare la dimostrazione del Teorema 2.2.3 studiando i momenti di ordine superiore al primo delle funzioni $f(X_i, X_j) = \mathbb{1}_{\{||X_i - X_j|| < r\}}$ che, in quanto funzioni indicatrici, ammettono momento di ogni ordine.

Osservazione 4. Il Teorema 2.2.3 copre anche il caso frattale, ovvero il caso in cui la dimensione d della varietà intrinseca non è intera, si pensi ad esempio alla curva di Koch. Nelle sperimentazioni andremo a considerare solo varietà intrinseche aventi dimensione intrinseca intera, pertanto se il valore restituito da CorrDim non è un intero, prenderemo come stima il valore intero a lui più vicino.

2.2.2 Curse of dimensionality

Lo stimatore CorrDim ha purtroppo un importante difetto: esso si basa fortemente sulla presenza di punti del dataset che siano poco distanti tra loro, ovvero affinchè CorrDim funzioni è necessario che ciascun punto del dataset abbia intorni densamente popolati. Mentre questa può sembrare una richiesta ragionevole in basse dimensioni, diventa un problema quando la dimensione intrinseca del dataset cresce. Tale fenomeno è noto come *curse of dimensionality* e fa sì che CorrDim sistematicamente sottostimi il valore della dimensione intrinseca d: vedremo che se d è grande, ma non necessariamente troppo grande (ad esempio già per $d = 6$),

per ottenere una stima di d sufficientemente accurata, CorrDim ha bisogno di un numero N di punti che sia esponenziale in d. Pertanto, se la dimensione intrinseca è troppo elevata, il numero di punti del dataset che si ha a disposizione è spesso insuficiente per calcolare una stima affidabile di tale valore. Per un esempio di curse of dimensionality si guardi la Figura 2.1.

Non dimostreremo formalmente quanto appena detto, ma vediamo di seguito un'euristica. Diamo prima la seguente definizione:

Definizione 2.2.2. Sia $M \subseteq \mathbb{R}^D$ una sottovarietà. Chiamiamo *diametro* di M la quantità

$$\delta = \sup_{x,y \in M} d(x,y), \tag{2.17}$$

dove d è la distanza euclidea.

Nota 3. Di seguito assumeremo δ finito: lo è ad esempio nel caso di sottovarietà compatte, ma in generale può non esserlo.

Affinchè lo stimatore CorrDim funzioni devono essere soddisfatte le seguenti due condizioni:

- Le distanze r utilizzate per effettuare la stima devono essere molto più piccole del diametro δ della varietà intrinseca in $\mathbb{R}^D$, ovvero deve valere
$$\frac{r}{\delta} \ll 1; \tag{2.18}$$
- Le distanze r utilizzate per effettuare la stima devono essere abbastanza grandi da far sì che il numero di coppie di punti a distanza minore r, sia esso $P(r)$, sia elevato:
$$P(r) \gg 1. \tag{2.19}$$
Tale condizione serve ad evitare fluttuazioni statistiche.

Nota 4. Di seguito, per evidenziare la dipendenza della quantità $P(r)$ dal numero N di punti del datset, scriveremo $P_N(r)$ anzichè $P(r)$.

L'euristica è la seguente: Siano N, $P_N(r)$ e δ come sopra, e sia d la dimensione intrinseca del dataset. Allora per $r \to 0$ vale:

$$P_N(r) = \frac{N^2}{2}(\frac{r}{\delta})^d + o(N^2). \tag{2.20}$$

Inoltre, se valgono le condizioni (2.18) e (2.19), si ha

$$d \lesssim \frac{2 \log N}{\log(\frac{\delta}{r})}. \tag{2.21}$$

Infatti, dal Teorema 2.2.3 segue che per r piccolo vale la seguente approssimazione:

$$\lim_{N \to \infty} \rho_N(r) = c r^d, \tag{2.22}$$

per una certa costante c. Infatti, da (2.7) si ottiene che per $r \to 0$

$$\log(\lim_{N \to \infty} \rho_N(r)) \propto d \log r \Rightarrow \lim_{N \to \infty} \rho_N(r) \propto r^d.$$

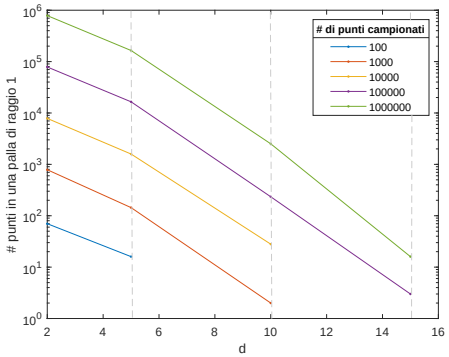

Figura 2.1: Il grafico (in scala semilogaritmica) mostra il numero di punti i.i.d. campionati su un ipercubo e distanti meno di 1 dal centro al variare della dimensione d dell'ipercubo, per diversi valori del numero totale di punti i.i.d. campionati. Si noti che il numero di punti che cadono in una palla di raggio 1 centrata nell'origine decrese esponenzialmente al crescere della dimensione d, a riprova del fatto che per riprodurre accuratamente una varietà intrinseca avente dimensione intrinseca elevata è necessario un numero esponenziale (nella dimensione d) di punti.

Vogliamo ora stimare la costante c. Notiamo che, se δ è il diametro della varietà intrinseca, allora $\rho_N(\delta) = 1$, e di conseguenza vale euristicamente che

$$\lim_{N\to\infty} \rho_N(\delta) = 1 = c\delta^d,$$

dove la seconda uguaglianza segue da (2.22). Ma allora $c = \delta^{-d}$. Dunque, per $r \to 0$

$$\lim_{N\to\infty} \rho_N(r) \propto (\frac{r}{\delta})^d \Rightarrow P_N(r) = \frac{N^2}{2}(\frac{r}{\delta})^d + o(N^2).$$

Passiamo ora alla disuguaglianza (2.21). Segue immediatamente da (2.19) e da (2.20) che

$$1 \lesssim \frac{N^2}{2}(\frac{r}{\delta})^d \Rightarrow (\frac{\delta}{r})^d \lesssim \frac{N^2}{2} \Rightarrow d\log(\frac{\delta}{r}) \lesssim 2\log N - \log 2$$

$$\Rightarrow d \lesssim \frac{2\log N}{\log(\frac{\delta}{r})} - \frac{\log 2}{\log(\frac{\delta}{r})},$$

da cui segue (2.21) in quanto, essendo per (2.18) $\frac{\delta}{r} \gg 1$, il secondo termine è trascurabile.

Nota 5. Il ragionamento appena fatto non è una dimostrazione formale di (2.20) e di (2.21). Infatti, le stime fatte valgono per $r \to 0$, mentre per stimare il valore della costante c abbiamo posto $r = \delta$, che non è in generale un valore piccolo.

Segue immediatamente da quanto detto che se la dimensione intrinseca d è elevata, per ottenere una stima di d sufficientemente accurata, lo stimatore CorrDim ha bisogno di un numero N di punti che sia esponenziale in d.

Osservazione 5. Usando $\frac{\delta}{r} = 10$ e $N = 10^3$, da (2.21) si ottiene $d \lesssim 6$. Questo bound, seppur approssimativo, rende ben chiaro il fatto che per stimare dimensioni intrinseche elevate è necessario un numero di punti N che sia esponenziale in d.

Osservazione 6. Posto $\frac{\delta}{r} = 10$, per ottenere una stima di d sufficientemente accurata CorrDim ha bisogno di almeno $10^{\frac{d}{2}}$ punti.

2.2.3 Sperimentazioni

In questa sezione riportiamo alcune sperimentazioni. Vedremo dapprima come lo stimatore CorrDim lavora su datasets artificiali, e lo faremo sia in situazioni ottimali che in situazioni di undersampling. Successivamente, proveremo a variare la matrice di covarianza di un dataset gaussiano, con l'obiettivo di vedere se in questi casi CorrDim restituisce stime ragionevoli. Infine, applicheremo lo stimatore su un dataset reale. Iniziamo con la prima sperimentazione.

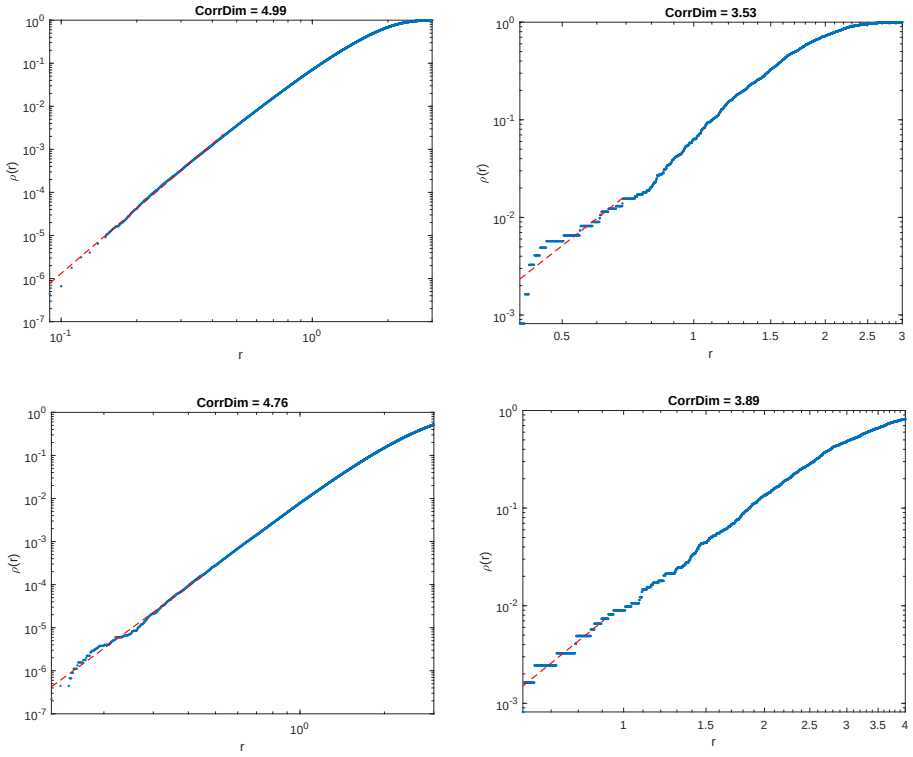

Figura 2.2: Tutti i grafici sono in scala logaritmica. I due grafici in alto mostrano la stima della dimensione intrinseca di due datasets lineari aventi dimensione intrinseca $d = 5$ in $\mathbb{R}^{20}$ e costituiti da $N = 3000$ punti (sinistra) e $N = 50$ punti (destra). La curva in blu mostra il valore del correlation integral al variare della distanza r, mentre la linea rossa tratteggiata rappresenta il fit lineare dei primi 300 punti. I due grafici in basso rispettano esattamente le stesse condizioni dei due grafici in alto, ma sono relativi a due datasets gaussiani costituiti rispettivamente da $N = 3000$ punti (sinistra) e $N = 50$ punti (destra).

Notiamo che nella sperimentazione sopra CorrDim stima correttamente sia la dimensione intrinseca del dataset lineare di $N = 3000$ punti ($d = 4.84$), sia quella del dataset gaussiano di $N = 3000$ punti ($d = 4.89$). Infatti, in entrambi i casi il numero di punti è maggiore di $10^{\frac{5}{2}}$, ed è quindi sufficiente per ottenere una stima

di d accurata.
Viceversa, CorrDim sottostima sia la dimensione intrinseca del dataset lineare di $N = 50$ punti ($d = 3.55$), sia quella del dataset gaussiano di $N = 50$ punti ($d = 3.88$). Ciò è dovuto alla mancanza di un numero di punti abbastanza alto da rappresentare la varietà intrinseca in modo accurato.

Vediamo ora che CorrDim restituisce stime ragionevoli anche se si varia la matrice di covarianza di un dataset gaussiano, ovvero se si considera come varietà intrinseca $\mathbb{R}^d$ dotato di disribuzione gaussiana multivariata avente densità

$$p(x) = \frac{1}{\sqrt{(2\pi)^d det(\Sigma)}} e^{-\frac{1}{2}x^T \Sigma^{-1} x}, \tag{2.23}$$

dove Σ è la matrice di covarianza.

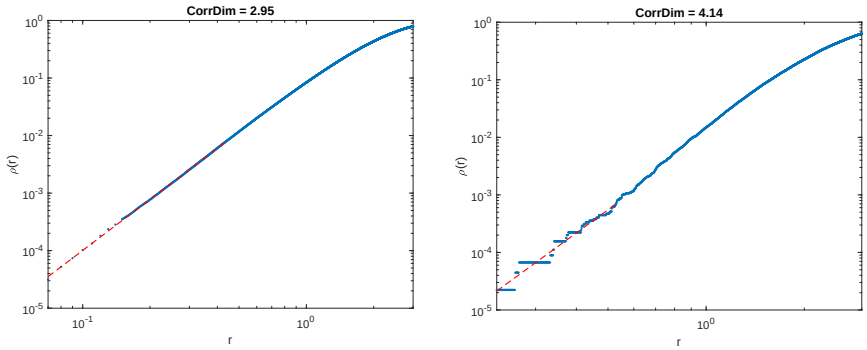

Figura 2.3: I due grafici (in scala logaritmica) mostrano la stima della dimensione intrinseca di due datasets gaussiani su $\mathbb{R}^5$ aventi matrice di covarianza $\Sigma_1 = diag(1, 1, 1, 0, 0)$ (sinistra) e $\Sigma_2 = diag(1, 1, 1, 1, 0.2)$ (destra). Entrambi i datasets sono costituiti da $N = 3000$ punti. Come nei grafici precedenti la curva in blu mostra il valore del correlation integral al variare della distanza r, mentre la linea rossa tratteggiata rappresenta il fit lineare dei primi 300 punti.

La stima della dimensione intrinseca restituita da CorrDim è di $d \simeq 3$ nel primo caso e di $d \simeq 4$ nel secondo. Noi non conosciamo con esattezza il valore della dimensione intrinseca dei due datasets, ma possiamo essere abbastanza fiduciosi del fatto che entrambe le stime siano buone.
Nella prossima sezione, quando parleremo della PCA, vedremo che la base di autovettori della matrice di covarianza empirica è quella che meglio cattura la varianza del dataset e che, lungo ciascun asse di tale base, la varianza del dataset coincide con il corrispondente autovalore della matrice. Nel nostro caso le matrici di covarianza empiriche saranno circa Σ_1 e Σ_2: hanno entrambe la base canonica di $\mathbb{R}^5$ come base di autovettori; inoltre, il primo dataset ha varianza nulla lungo due direzioni, mentre il secondo ha la stessa varianza lungo le prime 4 direzioni e varianza più bassa lungo la quinta. Pertanto, se pensiamo alla dimensione intrinseca come al minimo numero di parametri necessari a descrivere un dataset, sembra ragionevole che nel primo caso si abbia $d \simeq 3$ e nel secondo $d \simeq 4$.

Concludiamo ora questa sezione con una sperimentazione su un dataset reale.

Il dataset Digits è costituito da 10000 immagini di 28×28 pixels a scala di grigi, ciascuna contenente una cifra tra 0 e 9 scritta a mano. Noi utilizziamo come dataset il sottoinsieme degli "zeri" del dataset Digits:

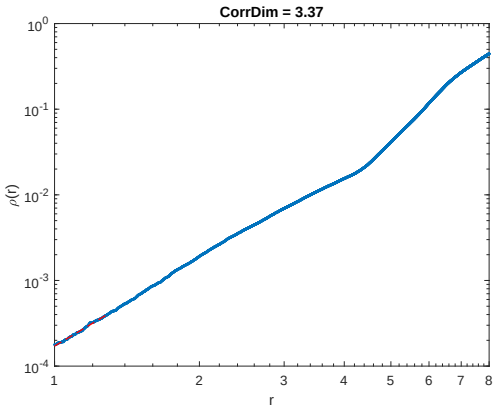

Figura 2.4: Il grafico (in scala logaritmica) mostra la stima della dimensione intrinseca del dataset costituito dagli "zeri" di Digits. Di nuovo, la curva in blu mostra il valore del correlation integral al variare della distanza r, mentre la linea rossa tratteggiata rappresenta il fit lineare dei primi 300 punti. La dimensione stimata è $d = 3.37$.

Anche in questo caso non conosciamo l'esatto valore della dimensione intrinseca del dataset. Tuttavia, il buon funzionamento dello stimatore CorrDim sui datasets artificiali supporta l'ipotesi di correttezza delle stime ottenute sui datasets reali.

2.3 PCA

La PCA (principal component analysis) è una tecnica ben nota utilizzata non solo per la stima della dimensione intrinseca di un dataset, ma anche come step iniziale in vari metodi di machine learning per la riduzione della dimensione. In questa sezione vedremo in dettaglio come la PCA stima la dimensione intrinseca dei datasets e ne studieremo pregi e difetti. Ne vedremo infine una prima versione multiscala, la PCA locale, ed una seconda versione probabilistica.

2.3.1 Descrizione del metodo

La PCA non è un semplice metodo di stima della dimensione intrinseca, ma ci permette di rispondere ad una domanda più generale: qual è la miglior base ortonormale attraverso cui studiare il nostro dataset? L'idea sulla quale essa si basa è la seguente: allineando alcuni degli assi di una base ortonormale alle direzioni in cui il dataset varia maggiormente, possiamo descrivere il nostro dataset utilizzando meno coordinate, oppure possiamo dare un'interpretazione a ciascun asse e vedere tale interpretazione come un aspetto nascosto del dataset responsabile della varianza del dataset lungo tale asse. Vediamo ora come quanto appena detto viene utilizzato per stimare la dimensione intrinseca.

Supponiamo di avere a disposizione un dataset $\{x_i\}_{i=1}^N$ costituito da N punti in $\mathbb{R}^D$, e supponiamo senza perdita di generalità che tale dataset sia centrato, ovvero che $\sum_{i=1}^N x_i$ sia il vettore nullo. Sia X la matrice $N \times D$ avente per $i-$esima riga il punto x_i e sia infine $C = \frac{1}{N} X^T X$ la matrice di covarianza empirica. Al fine di stimare la dimensione intrinseca del dataset la PCA utilizza il seguente risultato:

Teorema 2.3.1. *La base che meglio cattura la varianza del dataset coincide con la base di autovettori della matrice di covarianza empirica C. Inoltre, lungo ciascun asse di tale base, la varianza del dataset coincide con l'autovalore di C corrispondente all'asse.*

Dimostrazione. Determiniamo per prima cosa la direzione lungo la quale la proiezione del dataset lungo tale direzione ha la massima varianza, sia essa $w^{(1)}$:

$$w^{(1)} = \underset{\substack{w \in \mathbb{R}^D \\ ||w||=1}}{\arg\max} \Big[\frac{1}{N} \sum_{i=1}^N (w \cdot x_i)^2\Big] = \underset{\substack{w \in \mathbb{R}^D \\ ||w||=1}}{\arg\max} [w^T C w],$$

dove la seconda uguaglianza discende dalla seguente catena di uguaglianze:

$$\frac{1}{N} \sum_{i=1}^N (w \cdot x_i)^2 = \frac{1}{N} \sum_{i=1}^N (x_i^T w)^2 = \frac{1}{N} (Xw)^T Xw = \frac{1}{N} w^T X^T X w = w^T C w.$$

Essendo C simmetrica e semidefinita positiva, per il teorema spettrale esiste una matrice ortogonale U tale che $C = U^T \Lambda U$, con Λ matrice diagonale e U^T matrice ortogonale avente per colonne gli autovettori di C. Allora:

$$w^{(1)} = \underset{\substack{w \in \mathbb{R}^D \\ ||w||=1}}{\arg\max} [w^T U^T \Lambda U w] = \underset{\substack{w \in \mathbb{R}^D \\ ||w||=1}}{\arg\max} [(Uw)^T \Lambda (Uw)] =$$

$$= U^T \underset{\substack{y\in\mathbb{R}^D \\ ||y||=1}}{\arg\max}[y^T \Lambda y] = U^T \underset{\substack{y\in\mathbb{R}^D \\ ||y||=1}}{\arg\max}\Big[\sum_{i=1}^{D} \lambda_i y_i^2\Big],$$

dove per la terza uguaglianza poniamo $y = Uw$ e usiamo l'ortogonalità di U, mentre nell'ultima uguaglianza i $\lambda_i \geq 0$ sono gli autovalori di C (sono ≥ 0 perchè C è semidefinita positiva). A meno di riodinare possiamo supporre che i λ_i siano in ordine decrescente, ovvero $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_D$.
$w^{(1)}$ è l'autovettore di C corrispondente all'autovalore λ_1, infatti:

$$y^T \Lambda y = \sum_{i=1}^{D} \lambda_i y_i^2 \leq \lambda_1 \sum_{i=1}^{D} y_i^2 = \lambda_1 ||y||^2 = \lambda_1,$$

e $y^T \Lambda y = \lambda_1 \iff y = e_1$, dove indichiamo con e_1 il primo vettore della base canonica di $\mathbb{R}^D$. Segue che

$$\underset{\substack{y\in\mathbb{R}^D \\ ||y||=1}}{\arg\max}[y^T \Lambda y] = e_1 \quad \Longrightarrow \quad w^{(1)} = U^T e_1.$$

$w^{(1)}$ è quindi l'autovettore di C relativo all'autovalore λ_1, e la varianza del dataset lungo $w^{(1)}$ è

$$(w^{(1)})^T C w^{(1)} = \lambda_1 (w^{(1)})^T w^{(1)} = \lambda_1 \left|\left| w^{(1)} \right|\right|^2 = \lambda_1.$$

Cerchiamo ora una nuova direzione $w^{(2)}$, ortogonale a $w^{(1)}$, che catturi la maggior parte della varianza residua:

$$w^{(2)} = \underset{\substack{w\in\mathbb{R}^D \\ ||w||=1 \\ w\cdot w^{(1)}=0}}{\arg\max}[w^T C w] = \underset{\substack{w\in\mathbb{R}^D \\ ||w||=1 \\ w\cdot w^{(1)}=0}}{\arg\max}[w^T (C - \lambda_1 w^{(1)} (w^{(1)})^T) w],$$

dove, posto $C' = C - \lambda_1 w^{(1)} (w^{(1)})^T$, l'uguaglianza discende dal fatto che per ogni $w \in (w^{(1)})^\perp$ si ha

$$w^T C' w = w^T C w - \lambda_1 w^T w^{(1)} (w^{(1)})^T w = w^T C w.$$

In realtà possiamo rilassare la condizione $w \cdot w^{(1)} = 0$ e calcolare direttamente

$$\underset{\substack{w\in\mathbb{R}^D \\ ||w||=1}}{\arg\max}[w^T C' w].$$

Vedremo infatti che l'arg max si raggiunge comunque in corrispondenza di un vettore ortogonale a $w^{(1)}$.
Notiamo che C' non solo è simmetrica ma è anche semidefinita positiva, infatti ha lo stesso spettro di C con l'eccezione di λ_1 che è sostituito da 0:

$$C' w^{(1)} = \lambda_1 w^{(1)} - \lambda_1 w^{(1)} = 0;$$

$$C' U^T e_i = C U^T e_i - \lambda_1 w^{(1)} (w^{(1)})^T U^T e_i = C U^T e_i = \lambda_i U^T e_i$$

per ogni $i \neq 1$ e dove la penultima uguaglanza segue dal fatto che $U^T e_i \perp w^{(1)}$. Dunque, possiamo scrivere $C' = U^T \Lambda' U$, con $\Lambda' = diag(0, \lambda_2, ..., \lambda_D)$ e $\lambda_2 \geq ... \geq \lambda_D$. Allora, ragionando come prima otteniamo che

$$w^{(2)} = U^T \arg\max_{\substack{y \in \mathbb{R}^D \\ ||y||=1}} \Big[\sum_{i=1}^{D} \lambda_i y_i^2 \Big],$$

e che il massimo si ottiene in corrispondenza di e_2. Dunque $w^{(2)} = U^T e_2$ e la varianza del dataset lungo $w^{(2)}$ è

$$(w^{(2)})^T C w^{(2)} = \lambda_2 (w^{(2)})^T w^{(2)} = \lambda_2 \left|\left| w^{(2)} \right|\right|^2 = \lambda_2.$$

Iterando il ragionamento si ottiene che la base ortogonale che meglio cattura la varianza del dataset non è altro che la base di autovettori della matrice C, e la varianza del dataset lungo l'i-esimo asse di tale base coincide con l'autovalore λ_i ad esso corrispondente. □

Vediamo ora come utilizzare il Teorema 2.3.1 per stimare la dimensione intrinseca del dataset. L'idea è quella di ordinare le direzioni determinate dagli autovettori in ordine decrescente dei corrispondenti autovalori, e considerare le direzioni rilevanti fino a quando la varianza residua non diventa minore del 5% (ad esempio) della varianza totale. Le direzioni che risulteranno rilevanti saranno anche dette *componenti principali*. La dimensione intrinseca è quindi stimata come il numero di direzioni rilevanti (si guardi per un esempio la curva blu nel grafico di sinistra della Figura 2.5).

Osservazione 7. In generale lo spettro della matrice di covarianza empirica è caratterizzato dalla presenza di ampi salti nella grandezza degli autovalori. Ciò garantisce che la stima sia stabile contro eventuali modifiche del valore soglia della varianza residua, che è comunque scelto arbitrariamente.

2.3.2 Difetti ed esempi

La PCA ha purtroppo due difetti:

1. In generale, nel caso di varietà intrinseche non lineari la PCA sovrastima la dimensione intrinseca. Infatti, se due coordinate sono correlate, come ad esempio nel caso di un disco in $\mathbb{R}^2$, la PCA le considera entrambe rilevanti, in quanto contribuiscono entrambe in modo significativo alla varianza del dataset. Tuttavia, essendo le due coordinate correlate, possono essere descritte da un singolo parametro e ciò comporta una sovrastima del valore della dimensione intrinseca. Si vedano per un esempio le curve rossa e verde nei grafici della Figura 2.5;

2. La matrice di covarianza empirica può non avere ampi salti nella grandezza dei suoi autovalori (ciò non accade negli esempi mostrati in Figura 2.5). In tal caso la scelta del valore soglia della varianza residua diventa arbitraria ed eventuali modifiche possono restituire stime ben diverse.

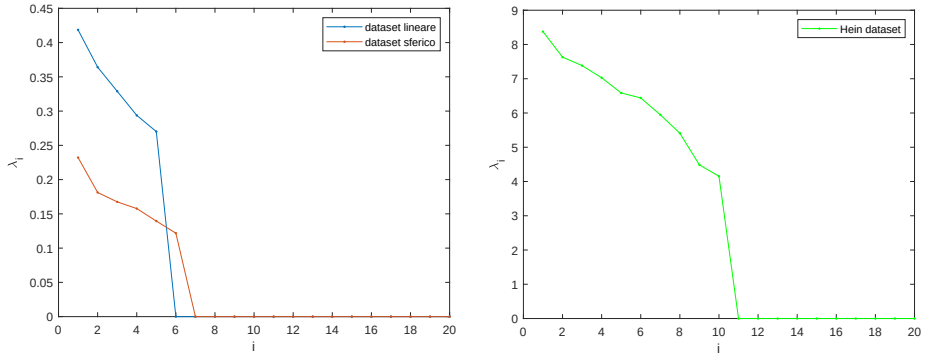

Figura 2.5: I due grafici mostrano gli autovalori della matrice di covarianza empirica di tre datasets: un dataset lineare (blu), un dataset sferico (rosso) e un Hein dataset (verde). In tutti i casi la dimensione intrinseca è $d = 5$, la dimensione di embedding è $D = 20$ e il numero di punti campionati è $P = 200$. Le dimensioni intrinseche stimate sono $d = 5, 6, 10$ per dataset lineare, dataset sferico e Hein dataset rispettivamente, e sono ottenute guardando agli ampi salti presenti nella grandezza degli autovalori (in questo caso si otterrebbero le stesse stime se si utilizzasse il criterio della varianza residua). Notiamo che la PCA stima correttamente la dimensione intrinseca del dataset lineare, invece sovrastima quella dei due datasets curvi.

A differenza di CorrDim, la PCA non ha problemi nel caso di dimensioni intrinseche elevate: è possibile mostrare che essa ha bisogno di $\sim d \log d$ punti per stimare in modo affidabile la matrice di covarianza, e tale valore è indipendente dalla dimensione di embedding D. Si guardi per ulteriori dettagli [LMR17].

2.3.3 PCA locale

Abbiamo visto che il principale difetto della PCA è che riesce a stimare correttamente solo la dimensione intrinseca di varietà lineari, mentre sovrastima la dimensione intrinseca dei datasets curvi. Per risolvere questo problema si è pensato ad una versione multiscala della PCA, che prende il nome di *PCA locale*.
A differenza della PCA globale, in cui la matrice di covarianza empirica è calcolata su tutto il dataset, sia esso X, nella PCA locale si effettua l'analisi spettrale su sottoinsiemi $X(x_0, r_c)$ di X, ottenuti selezionando un punto $x_0 \in X$ e includendo nella matrice di covarianza empirica locale solo quei punti distanti da x_0 meno di r_c. Le dimensioni intrinseche locali vengono così calcolate e combinate per determinare un dimensione intrinseca che caratterizzi l'intero dataset.

Così facendo, la PCA locale risolve il problema della curvatura. Tuttavia, sappiamo che la PCA funziona solo quando il numero N di punti campionati è $\gtrsim d \log d$. Questo è un problema per la versione multiscala: per restituire una stima corretta, il valore di r_c deve essere abbastanza piccolo, il che implica che il campione della varietà deve essere abbastanza denso da garantire la presenza di un numero sufficiente di punti nei sottoinsiemi $X(x_0, r_c)$.
Un altro problema che rende la PCA locale difficile da utilizzare per la stima della dimensione intrinseca è che l'ampiezza dei salti nella grandezza degli autovalori dipende dai dati, e la scelta delle regioni locali e dei valori soglia può essere difficile.

2.3.4 PCA probabilistica

La PCA è una tecnica di analisi dati che in principio non si basa su un modello probabilistico. In un articolo del 1999 Tipping e Bishop [BT99] hanno dimostrato come le componenti principali di un insieme di dati osservati possano essere determinate attraverso la stima di massima verosimiglianza dei parametri di un modello a variabile latente, fortemente collegato all'analisi fattoriale. Hanno così associato alla PCA un modello probabilistico: tale riformulazione della PCA prende il nome di *PCA probabilistica*.

Prima di addentrarci nella descrizione vera e propria della PCA probabilistica, ricordiamo brevemente cosa si intende per funzione di verosimiglianza e per metodo di massima verosimiglianza. Introduciamo poi i concetti di variabile latente e di modello a variabile latente, concentrandoci in particolare sull'ambito dell'analisi fattoriale. Vediamo infine come la PCA emerge da una particolare parametrizzazione di tale modello.

Definizione 2.3.1. Consideriamo un insieme di osservazioni $\{x_i\}_{i=1}^n$ e una famiglia di funzioni di densità, parametrizzate tramite il vettore θ:

$$x \to f(x|\theta).$$

La *funzione di verosimiglianza* associata è:

$$\mathcal{L}(\theta|x_1, ..., x_n) = f(x_1, ..., x_n|\theta). \tag{2.24}$$

Nel caso in cui, come normalmente si ipotizza, gli x_i siano i.i.d., allora:

$$\mathcal{L}(\theta|x_1, ..., x_n) = \prod_{i=1}^{n} f(x_i|\theta). \tag{2.25}$$

Poichè l'espressione (2.25) può risultare poco trattabile, soprattutto nei problemi di massimizzazione collegati al metodo di massima verosimiglianza, spesso risulta preferibile lavorare sul logaritmo della funzione di verosimiglianza, detto *log-verosimiglianza*:

$$L(\theta|x_1, ..., x_n) = \ln \mathcal{L}(\theta|x_1, ..., x_n) = \sum_{i=1}^{n} \ln f(x_i|\theta). \tag{2.26}$$

Il *metodo di massima verosimiglianza* ricerca il valore più verosimile di θ, ovvero ricerca all'interno dello spazio Θ di tutti i possibili valori di θ, i valori dei parametri che massimizzano la probabilità di aver ottenuto il campione dato. Lo stimatore di massima verosimiglianza è ottenuto come:

$$\hat{\theta} = \arg\max_{\theta \in \Theta} \mathcal{L}(\theta|x_1, ..., x_n), \tag{2.27}$$

da cui il nome di metodo di massima verosimiglianza, in quanto ricerca il valore $\theta \in \Theta$ che massimizza la funzione di verosimiglianza.

Prima di descrivere cosa si intende per modello a variabile latente diamo le seguenti definizioni:

Definizione 2.3.2. In statistica, chiamiamo *variabile latente* ogni variabile che non è osservata direttamente, ma è dedotta attraverso un modello matematico a partire da altre variabili che vengono invece osservate.

Definizione 2.3.3. Una *distribuzione di probabilità a priori* di una quantità incognita p è la distribuzione di probabilità che esprimerebbe l'incertezza di p prima che i dati osservati vengano presi in considerazione. La quantità incognita può essere un parametro oppure una variabile latente.

Allora:

Definizione 2.3.4. Un *modello a variabile latente* ha l'obiettivo di trovare una relazione tra un insieme di vettori $D-$dimensionali $\{x_i\}_{i=1}^N$, che rappresentano dei dati osservati, e un insieme $\{t_i\}_{i=1}^N$ di variabili latenti $d-$dimensionali, della forma:

$$x = y(t;\theta) + \epsilon, \tag{2.28}$$

dove $y(t;\theta)$ è una funzione del vettore di variabili latenti t avente parametri θ, mentre ϵ rappresenta l'errore ed è indipendente da t. In generale si ha $d < D$, così che le variabili latenti diano una descrizione dei dati più semplice.

Noi siamo interessati in particolare ai modelli a variabili latenti nell'ambito dell'analisi fattoriale, la quale è una tecnica che permette di evidenziare l'esistenza di una struttura di tratti latenti (o fattori o dimensioni), non misurabili direttamente, all'interno di un insieme di variabili direttamente osservabili.
Nell'analisi fattoriale standard, la funzione $y(t;\theta)$ che compare nell'equazione (2.28) è lineare, ovvero si ha:

$$x = Wt + m + \epsilon, \tag{2.29}$$

dove il vettore di variabili latenti t ha distribuzione a priori $N(0, I)$; l'errore ϵ ha distribuzione $N(0, \Psi)$, con Ψ diagonale; $W \in \mathbb{R}^{D\times d}$ è la cosiddetta *matrice dei loadings*, ovvero l'entrata $w_{i,j}$ descrive la forza della relazione tra la $j-$esima variabile latente e l'$i-$esimo dato osservato; infine, m è una costante il cui stimatore di massima verosimiglianza è dato dalla media dei dati osservati.
Data questa formulazione, segue che x ha distribuzione $N(m, S)$, con $S = \Psi + WW^T$. Ciò discende immediatamente dalle seguenti due proposizioni:

Proposizione 2.3.2. *Sia X un vettore gaussiano con media $m_X \in \mathbb{R}^n$ e matrice di covarianza $K_X \in \mathbb{R}^{n\times n}$. Sia $A : \mathbb{R}^n \to \mathbb{R}^m$ un'applicazione lineare e sia $b \in \mathbb{R}^m$. Sia $Z = AX + b$; allora Z è ancora un vettore gaussiano con media $m_Z = Am_X + b \in \mathbb{R}^m$ e matrice di covarianza $K_Z = AK_XA^T \in \mathbb{R}^{m\times m}$.*

Proposizione 2.3.3. *Se Z e Y sono vettori gaussiani indipendenti, con $Z \sim N(m_Z, \Sigma_Z)$ e $Y \sim N(m_Y, \Sigma_Y)$, allora $Z + Y \sim N(m_Z + m_Y, \Sigma_Z + \Sigma_Y)$.*

Vediamo ora che, sotto determinate ipotesi, lo stimatore di massima verosimiglianza di W va a coincidere con la matrice avente per colonne le componenti principali (ruotate e riscalate) dei dati osservati. Vale infatti il seguente teorema, della cui dimostrazione riportiamo solo le idee principali (per ulteriori dettagli si rimanda all'articolo [BT99]):

Teorema 2.3.4. *Consideriamo il modello* (2.29) *con* $\Psi = \sigma^2 I$, *con* $\sigma^2 \in \mathbb{R}$ *non noto. Lo stimatore di massima verosimiglianza di* W, *sia esso* W_{ML}, *coincide con la matrice le cui colonne sono gli autovettori principali (ruotati e riscalati) della matrice di covarianza empirica* C.
Inoltre, per $W = W_{ML}$, *lo stimatore di massima verosimiglianza di* σ^2 *è dato da:*

$$\sigma^2_{ML} = \frac{1}{D-d} \sum_{j=d+1}^{D} \lambda_j, \tag{2.30}$$

dove $\lambda_1, ..., \lambda_D$ *sono gli autovalori di* C *ordinati in ordine decrescente.*

Dimostrazione. Vediamo per semplicità solo la dimostrazione della prima parte del teorema. Consideriamo dunque il modello (2.29) con $\epsilon \sim N(0, \sigma^2 I)$, per un certo $\sigma^2 \in \mathbb{R}$. Segue da (2.29) che $x - Wt - m = \epsilon \sim N(0, \sigma^2 I)$. Pertanto, condizionatamente a t, si ha la seguente distribuzione di probabilità su x:

$$p(x|t) = (2\pi\sigma^2)^{-D/2} e^{-\frac{1}{2\sigma^2}||x-Wt-m||^2}. \tag{2.31}$$

Se poniamo una distribuzione a priori gaussiana standard sulle variabili latenti:

$$p(t) = (2\pi)^{-d/2} e^{-\frac{1}{2}t^T t}, \tag{2.32}$$

otteniamo la seguente distribuzione marginale di x:

$$p(x) = \int_{\mathbb{R}^d} p(x|t)p(t)dt = (2\pi)^{-D/2}|S|^{-1/2} e^{-\frac{1}{2}(x-m)^T S^{-1}(x-m)}, \tag{2.33}$$

con $S = \sigma^2 I + WW^T$.
La log-verosimiglianza dei dati osservati è dunque data da:

$$\mathcal{L} = \sum_{i=1}^{N} \ln p(x_i) = -\frac{ND}{2} \ln 2\pi - \frac{N}{2} \ln |S| - \frac{N}{2} tr[S^{-1}C], \tag{2.34}$$

dove

$$C = \frac{1}{N} \sum_{i=1}^{N} (x_i - m)(x_i - m)^T \tag{2.35}$$

è la matrice di covarianza empirica dei dati osservati $\{x_i\}_{i=1}^N$. I parametri per questo modello possono quindi essere stimati massimizzando la log-verosimiglianza $\mathcal{L}$. A tal fine, consideriamo la derivata di $\mathcal{L}$ rispetto a W. Utilizzando alcuni risultati standard di differenziazione matriciale, si ottiene

$$\frac{\partial \mathcal{L}}{\partial W} = N(S^{-1}CS^{-1}W - S^{-1}W). \tag{2.36}$$

È possibile dimostrare che, con $S = \sigma^2 I + WW^T$, gli unici punti stazionari non nulli di (2.36) sono della forma:

$$W = U_d(\Lambda_d - \sigma^2 I)^{1/2} R, \tag{2.37}$$

dove le d colonne di U_d sono autovettori di C, con i corrispondenti autovalori nella matrice diagonale Λ_d, mentre R è una qualche matrice di rotazione $d \times d$.
È possibile inoltre dimostrare che il massimo globale di $\mathcal{L}$ si ha quando le colonne di U_d coincidono con gli autovettori principali di C, e che ogni altra combinazione di autovettori rappresenta invece un punto di sella. Pertanto, le colonne dello stimatore di massima verosimiglianza W_{ML} coincidono con gli autovettori principali di C, con una riscalatura determinata dall'autovalore corrispondente e dal parametro σ^2, e con una certa rotazione. □

Conseguenza importante del precedente teorema è che la PCA può essere ottenuta a partire da un modello probabilistico.

2.4 Stimatori geometrici e proiettivi: una breve overview

Abbiamo visto che CorrDim e PCA sono stimatori facili da capire e da implementare, tuttavia sono alla base di algoritmi di stima ben più avanzati.
CorrDim può essere considerato il rappresentante dei cosiddetti *stimatori geometrici*, i quali si basano sul calcolo di determinate grandezze che, al limite, dipendono esplicitamente dal valore d della dimensione intrinseca del dataset. La PCA, invece, può essere considerata il rappresentante dei cosiddetti *stimatori proiettivi*, i quali in genere ricercano decomposizioni o rappresentazioni utili dello spazio di embedding, al fine di evidenziare quali sono le direzioni rilevanti e quali invece quelle irrilevanti.
In generale, tutti gli stimatori proiettivi sovrastimano la dimensione intrinseca delle varietà curve. Gli stimatori geometrici, invece, evitano il problema della curvatura, ma soffrono del cosiddetto curse of dimensionality, e per funzionare correttamente necessitano di un numero esponenziale di punti (in d), cosa che si traduce in una sistematica sottostima della dimensione intrinseca quando questa diventa elevata.

Il prossimo capitolo sarà interamente dedicato allo studio di un particolare stimatore introdotto di recente, il *Full correlation integral*, il quale estende lo stimatore CorrDim con l'obiettivo di alleviare gli effetti del curse of dimensionality.

Capitolo 3

Full Correlation Integral

Nel precedente capitolo abbiamo visto che la stima della dimensione intrinseca di un dataset ha due principali nemici: da un lato, gli stimatori proiettivi sovrastimano la dimensione intrinseca dei datasets curvi, dall'altro gli stimatori geometrici soffrono del curse of dimensionality. Purtroppo nessuno stimatore al momento è immune ad entrambi questi difetti. Come fare dunque a migliorare gli stimatori già esistenti? È stato proposto di recente un nuovo stimatore, il *full correlation integral*, che combina alcune delle caratteristiche degli stimatori geometrici con quelle degli stimatori proiettivi [EGR19]. In questo capitolo studieremo in dettaglio il full correlation integral: ne descriveremo il funzionamento, ne valuteremo la performance generale tramite sperimentazioni e la confronteremo con quella degli stimatori CorrDim e PCA. Infine, ne descriveremo una versione multiscala.

3.1 Descrizione del metodo

Il full correlation integral nasce a partire dalla seguente osservazione: nel caso di alcuni datasets semplici (ad esempio un dataset sferico $(d-1)-$dimensionale immerso in $\mathbb{R}^d$ tramite mappa di inclusione), il correlation integral calcolato sul dataset ha le seguenti proprietà, delle quali daremo la dimostrazione formale più avanti:

- Il valor medio del correlation integral ammette un'espressione in forma chiusa che dipende in modo parametrico dalla dimensione intrinseca d. Chiameremo tale espressione *correlation integral analitico*;
- La distribuzione di probabilità del correlation integral si concentra attorno al suo valor medio.

Quanto detto suggerisce il seguente algoritmo di stima della dimensione intrinseca:

1. Centrare il dataset, ovvero sottrarre a ciascun punto x_i del dataset la posizione del baricentro $\frac{1}{N}\sum_{i=1}^{N} x_i$;
2. Proiettare ciascun punto x_i sulla sfera unitaria normalizzandolo;
3. Calcolare il correlation integral sul dataset normalizzato;

4. Ricavare la dimensione intrinseca del dataset facendo un fit del correlation integral analitico calcolato per la sfera con il correlation integral calcolato sul dataset normalizzato. Infine, aggiungere uno al valore trovato, in quanto la procedura di normalizzazione fa diminuire di uno il numero totale di gradi di libertà.

Questo algoritmo è esatto se il dataset ha una varietà intrinseca lineare embedded in modo isometrico in $\mathbb{R}^D$ e dotata di una distribuzione di probabilità che sia invariante per rotazione. Infatti, in questo caso il dataset centrato e normalizzato non è altro che un campione uniforme da S^{d-1}, e per isometria tutte le distanze possono essere misurate equivalentemente su $\mathbb{R}^d$ o $\mathbb{R}^D$.

Nota 6. Le condizioni che si hanno sul dataset affinchè l'algoritmo sia esatto sono piuttosto forti, pertanto è estremamente importante studiarne la performance nei casi in cui il dataset non le soddisfi appieno. Nella sezione dedicata alle sperimentazioni vedremo che si ottengono risultati affidabili anche in casi di estremo undersampling oppure quando le condizioni di linearità, isotropia e isometria non sono del tutto rispettate. Questo ci permetterà di dire che il full correlation integral è uno stimatore robusto.

Dimostriamo ora formalmente le due proprietà sulle quali si basa il full correlation integral. Consideriamo il caso particolare di un dataset sferico avente dimensione intrinseca $d-1$ e immerso in $\mathbb{R}^d$ tramite mappa di inclusione. Consideriamo, inoltre, la versione probabilistica del problema: supponiamo di avere N variabili aleatorie $X_1, ..., X_N$ i.i.d. definite su $\mathbb{R}^d$ e aventi distribuzione uniforme concentrata sulla superficie della sfera unitaria $(d-1)-$dimensionale centrata nell'origine. Ciascuna variabile aleatoria rappresenta un diverso punto del nostro dataset. In questo caso il correlation integral può essere riscritto come variabile aleatoria nella seguente forma:

$$\rho(r) = \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq i}}^{N} \mathbb{1}_{\{||X_i - X_j|| < r\}}. \tag{3.1}$$

Diamo le seguenti definizioni:

Definizione 3.1.1. L'*angolo solido d-dimensionale* Ω_d misura la superficie della sfera unitaria $d-$dimensionale. Esso è dato da:

$$\Omega_d = \frac{2\pi^{\frac{d+1}{2}}}{\Gamma(\frac{d+1}{2})}, \tag{3.2}$$

dove Γ è la funzione Gamma di Eulero.

Definizione 3.1.2. Chiamiamo *funzione ipergeometrica gaussiana o standard* ${}_2F_1$ una funzione speciale rappresentata dalla serie ipergeometrica

$${}_2F_1(a, b; c; z) = \sum_{n=0}^{\infty} \frac{\beta_n z^n}{n!}, \tag{3.3}$$

dove $\beta_0 = 1$ e $\frac{\beta_{n+1}}{\beta_n} = \frac{(n+a)(n+b)}{(n+c)}$.

Valgono i seguenti teoremi:

Teorema 3.1.1. *Nelle ipotesi di cui sopra, fissato $0 < r \leq 2$, il valor medio del correlation integral $\rho(r)$ è dato da:*

$$\mathbb{E}[\rho(r)] = \frac{1}{2}\Big(1 + \frac{\Omega_{d-1}}{\Omega_d}(r^2-2)\,{}_2F_1\Big(1-\frac{d}{2},\frac{1}{2};\frac{3}{2};\frac{(r^2-2)^2}{4}\Big)\Big), \tag{3.4}$$

dove Ω_d è l'angolo solido d-dimensionale, mentre ${}_2F_1$ è la funzione ipergeometrica standard.

Dimostrazione. Per linearità del valor medio ed essendo le variabili aleatorie $X_1, ..., X_n$ identicamente distribuite, vale:

$$\mathbb{E}[\rho(r)] = \mathbb{E}\Big[\frac{1}{N(N-1)}\sum_{i=1}^{N}\sum_{\substack{j=1\\ j\neq i}}^{N}\mathbb{1}_{\{||X_i - X_j||<r\}}\Big] = \mathbb{E}[\mathbb{1}_{\{||X-Y||<r\}}], \tag{3.5}$$

dove X e Y sono due variabili aleatorie che rappresentano due punti i.i.d. campionati uniformemente da S^{d-1}. Pertanto:

$$\mathbb{E}[\rho(r)] = \int_{\mathbb{R}^d}\int_{\mathbb{R}^d}\mathbb{1}_{\{||x-y||^2<r^2\}}\frac{\delta(||x||-1)}{\Omega_d}\frac{\delta(||y||-1)}{\Omega_d}dydx, \tag{3.6}$$

dove abbiamo elevato al quadrato entrambi i membri della disequazione nella funzione indicatrice e possiamo farlo perchè entrambi i termini sono positivi. Inoltre, Ω_d è l'angolo solido d-dimensionale, mentre δ è la funzione delta di Dirac.
Passiamo ora in coordinate sferiche. Scegliendo l'asse azimutale di y in modo tale che sia allineato con x, possiamo riscrivere $||x-y||^2$ nel seguente modo:

$$||x-y||^2 = 2(1-\cos(\beta_d)), \tag{3.7}$$

dove β_d è l'angolo azimutale di y, o analogamente l'angolo tra x e y. Allora:

$$\begin{aligned}\mathbb{E}[\rho(r)] &= \frac{1}{\Omega_d^2}\int_0^{2\pi}d\alpha_1\int_0^{\pi}d\alpha_2\sin(\alpha_2)...\int_0^{\pi}d\alpha_d\sin^{d-1}(\alpha_d)\times\\ &\quad\times\int_0^{2\pi}d\beta_1\int_0^{\pi}d\beta_2\sin(\beta_2)...\int_0^{\pi}d\beta_d\sin^{d-1}(\beta_d)\mathbb{1}_{\{2(1-\cos(\beta_d))<r^2\}} \qquad (3.8)\\ &= \frac{\Omega_{d-1}}{\Omega_d}\int_0^{\arccos(1-\frac{r^2}{2})}\sin^{d-1}(\beta_d)d\beta_d,\end{aligned}$$

dove per l'ultima uguaglianza abbiamo risolto gli integrali in $\alpha_1, ..., \alpha_d, \beta_1, ..., \beta_{d-1}$ usando la definizione di Ω_d e Ω_{d-1}.
L'integrale (3.8) può essere risolto applicando il cambio di variabile $t = \cos(\beta_d)$ e

poi espandendo in serie di Taylor nell'origine la funzione integranda. Si ottiene:

$$\begin{aligned}\mathbb{E}[\rho(r)] &= \frac{\Omega_{d-1}}{\Omega_d}\int_0^{\arccos(1-\frac{r^2}{2})} \sin^{d-1}(\beta_d)d\beta_d \\ &= \frac{\Omega_{d-1}}{\Omega_d}\int_{1-\frac{r^2}{2}}^{1}(1-t^2)^{\frac{d}{2}-1}dt \\ &= \frac{\Omega_{d-1}}{\Omega_d}\sum_{n\geq 0}\frac{\Gamma(n+1-\frac{d}{2})}{n!\Gamma(1-\frac{d}{2})}\int_{1-\frac{r^2}{2}}^{1} t^{2n}dt \\ &= \frac{\Omega_{d-1}}{\Omega_d}\sum_{n\geq 0}\frac{\Gamma(n+1-\frac{d}{2})}{n!\Gamma(1-\frac{d}{2})}\frac{1}{2n+1}\Big(1-\Big(1-\frac{r^2}{2}\Big)^{2n+1}\Big).\end{aligned} \tag{3.9}$$

Infine, utilizzando il seguente fatto:

$$\sum_{n\geq 0}\frac{\Gamma(n+a)}{n!\Gamma(a)}\frac{1}{2n+1}b^{2n+1} = b\,{}_2F_1\Big(a,\frac{1}{2};\frac{3}{2};b\Big), \tag{3.10}$$

dove ${}_2F_1$ è la funzione ipergeometrica standard, si ottiene l'uguaglianza (3.4). □

Teorema 3.1.2. *Fissato $0 < r \leq 2$, al crescere del numero N di punti del dataset la distribuzione del correlation integral $\rho_N(r)$ si concentra attorno al suo valor medio. In particolare:*

$$Var(\rho_N(r)) = O_r(N^{-1}) \tag{3.11}$$

per $N \to \infty$.

Dimostrazione. Per prima cosa riscriviamo il correlation integral $\rho_N(r)$ nella seguente forma:

$$\rho_N(r) = \frac{2}{N(N-1)}\sum_{1\leq i<j\leq N}\mathbb{1}_{\{||X_i-X_j||<r\}}, \tag{3.12}$$

da cui immediatamente segue che

$$\rho_N(r)^2 = \Big(\frac{2}{N(N-1)}\Big)^2\sum_{1\leq i<j\leq N}\sum_{1\leq k<l\leq N}\mathbb{1}_{\{||X_i-X_j||<r\}}\mathbb{1}_{\{||X_k-X_l||<r\}}. \tag{3.13}$$

Per linearità del valor medio vale:

$$\mathbb{E}[\rho_N(r)^2] = \Big(\frac{2}{N(N-1)}\Big)^2\sum_{1\leq i<j\leq N}\sum_{1\leq k<l\leq N}\mathbb{E}[\mathbb{1}_{\{||X_i-X_j||<r\}}\mathbb{1}_{\{||X_k-X_l||<r\}}]. \tag{3.14}$$

Notiamo che $\mathbb{E}[\mathbb{1}_{\{||X_i-X_j||<r\}}\mathbb{1}_{\{||X_k-X_l||<r\}}]$ non dipende dagli indici i,j,k,l ma solo dalla relazione che c'è tra i due insiemi di indici $\{i,j\}$ e $\{k,l\}$: si ottiene lo stesso valore a seconda che si abbia $\{i,j\} = \{k,l\}$, oppure $|\{i,j\} \cap \{k,l\}| = 1$, oppure se gli indici i,j,k,l sono a due a due distinti. Possiamo quindi riordinare i termini della doppia sommatoria in (3.14) in base alle distribuzioni degli indici ed ottenere così uno sviluppo in N. Notiamo, inoltre, che per N grande la doppia sommatoria in (3.14) è dominata dai termini aventi indici tutti diversi, il cui numero è dell'ordine di N^4 (più precisamente $3\binom{N}{4}$).

Analogamente, possiamo sviluppare $\left(\frac{N(N-1)}{2}\right)^2 \mathbb{E}[\rho_N(r)]^2$. Essendo $\mathbb{1}_{\{||X_i - X_j|| < r\}}$ e $\mathbb{1}_{\{||X_k - X_l|| < r\}}$ indipendenti quando gli indici i, j, k, l sono a due a due distinti, l'ordine del leading term del suo sviluppo coincide con quello dello sviluppo della doppia sommatoria in (3.14), ovvero N^4.
Considerando ora la differenza tra il primo e il secondo sviluppo, i termini di ordine N^4 si elidono. Dunque, moltiplicando per $\left(\frac{2}{N(N-1)}\right)^2$, si ottiene che per N grande vale:

$$\mathbb{E}[\rho_N(r)^2] - \mathbb{E}[\rho_N(r)]^2 = Var(\rho_N(r)) = O(N^{-1}).$$

□

3.2 Sperimentazioni

In questa sezione riportiamo i risultati di alcune sperimentazioni volte a valutare la performance generale del full correlation integral e a confrontarla con quella degli stimatori CorrDim e PCA.

3.2.1 Sperimentazione 1

Con questa prima sperimentazione vediamo che il full correlation integral sembra essere affidabile anche su datasets che non soddisfano appieno le condizioni di isotropia, linearità e isometria, che nel paragrafo precedente abbiamo visto essere sufficienti per il suo corretto funzionamento:

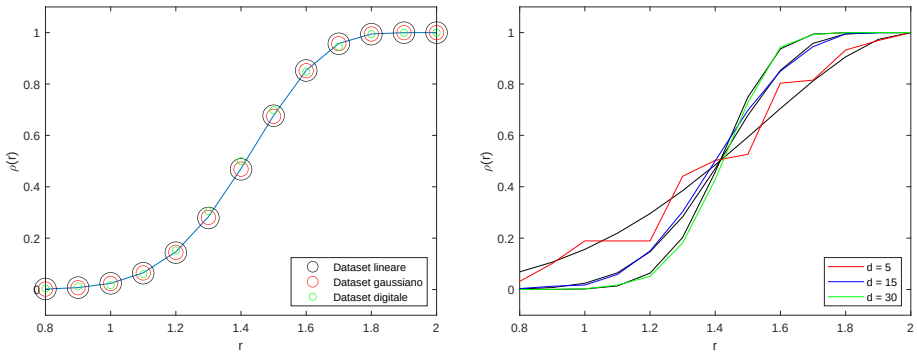

Figura 3.1: Il grafico a sinistra mostra il correlation integral di tre datasets (lineare, gaussiano e digitale) centrati e normalizzati. I tre datasets sono constituiti da $N = 500$ punti, hanno dimensione intrinseca $d = 15$ e dimensione di embedding $D = 60$. La linea in blu rappresenta il correlation integral analitico, che è dato dall'Equazione (3.4) con $d = 14$ (il valore di d è diminuito di uno rispetto all'effettivo valore della dimensione intrinseca dei datasets per via della procedura di normalizzazione). Il grafico a destra mostra il correlation integral di tre datasets digitali centrati e normalizzati e aventi dimensione intrinseca $d = 5, 15, 30$ e dimensione di embedding $D = 60$. Ciascun dataset è costituito da $N = 500$ punti. Ognuno dei tre grafici è sovrapposto al corrispondente correlation integral analitico (in nero).

Dal grafico a sinistra si evince che per datastes gaussiani, lineari e digitali (centrati e normalizzati), l'Equazione (3.4) con d diminuito di un'unità rispetto all'effettivo valore della dimensione intrinseca, interpola bene il correlation integral calcolato sul dataset. Notiamo, inoltre, che il fit risulta più preciso nei primi due casi e meno preciso sul dataset digitale: a differenza dei cerchi in rosso e in nero, il cui centro si colloca esattamente sulla linea blu, il centro dei cerchi in verde cade a volte al di fuori della linea, generando così delle imperfezioni che sono però del tutto trascurabili.
Dal grafico a destra, invece, risulta evidente che per stimare correttamente la dimensione intrinseca dei tre datasets digitali è necessario l'intero correlation integral analitico sull'intervallo $[0.8, 2]$. Infatti, le tre curve colorate spesso si allontanano dalle corrispondenti curve in nero, pertanto un fit locale (ad esempio per un valore di r piccolo) porterebbe quasi certamente ad un errore di stima. Questo risulta evidente sopratutto per valori della dimensione intrinseca non elevati. Si

noti, infatti, che al crescere della dimensione intrinseca, l'andamento quasi a scalini del correlation integral calcolato sul dataset scompare, andando a coincidere rapidamente con il corrispondente correlation integral analitico.

3.2.2 Sperimentazione 2

Con questa seconda sperimentazione vediamo che il full correlation integral risulta affidabile anche in casi di estremo undersampling, ad esempio quando il numero N di punti campionati è minore della dimensione intrinseca d del dataset:

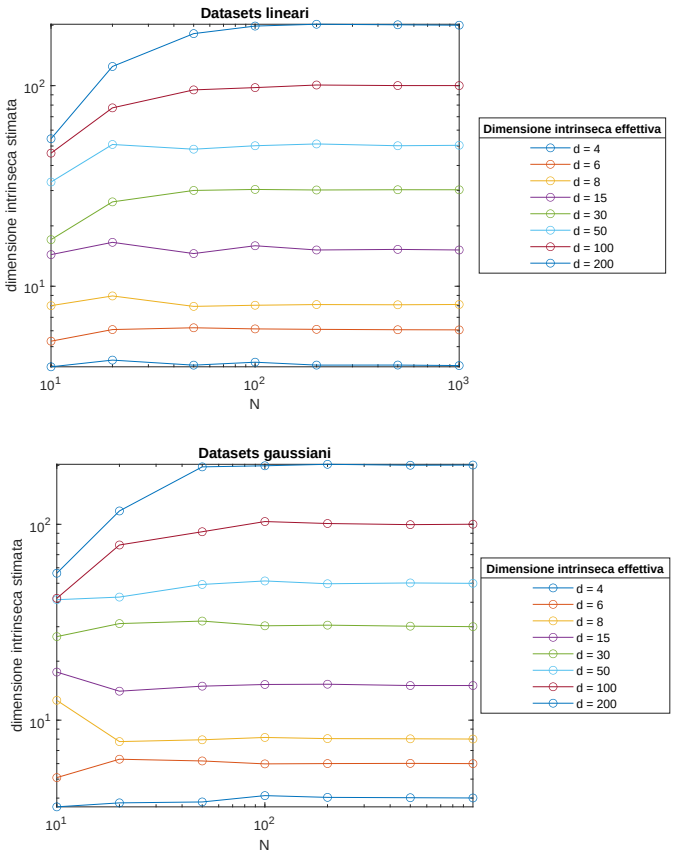

Figura 3.2: Il grafico in alto mostra la dimensione intrinseca stimata dal full correlation integral al variare del numero N di punti campionati, per svariati datasets lineari aventi dimensione intrinseca $d = 4, 6, 8, 15, 30, 50, 100, 200$ e dimensione di embedding $D = 500$. Il grafico in basso mostra la stessa cosa, con la differenza che i datasets considerati sono gaussiani anzichè lineari.

Dai due grafici in figura emerge che sia nel caso lineare che in quello gaussiano il full correlation integral riesce a stimare correttamente il valore della dimensione intrinseca, non solo quando quest'ultima è bassa ma anche quando invece è molto elevata. Si noti, in particolare, che lo stimatore funziona bene anche nel caso di estremo undersampling in cui il numero N di punti campionati è minore della

dimensione intrinseca d del dataset. Ad esempio, in entrambi i casi in dimensione $d = 200$ e con $N = 100$ punti, il full correlation integral riesce a stimare il valore della dimensione intrinseca con un errore di circa uno. Inoltre, a riprova del fatto che il full correlation integral funziona bene nei casi di undersampling, è possibile verificare che, per $N = 20$, se si considerano i datasets aventi dimensione intrinseca da $d = 4$ a $d = 30$, la radice dell'errore quadratico medio è 0.70 nel caso gaussiano e 1.83 nel caso lineare.
Quanto appena visto ci permette di dire che lo stimatore è più robusto rispetto agli stimatori CorrDim e PCA: ricordiamo, infatti, che la PCA, per stimare correttamente il valore della dimensione intrinseca, necessita di almeno $\sim d \log d$ punti, mentre CorrDim necessita di un numero di punti che sia esponenziale nella dimensione d.

3.2.3 Sperimentazione 3

In questa terza sperimentazione vediamo come il full correlation integral si comporta se si varia la matrice di covarianza di un dataset gaussiano. Nei grafici che seguono abbiamo considerato solo matrici di covarianza diagonali aventi autovalori che tendono lentamente a zero: in questi casi noi non sappiamo qual è l'effettivo valore della dimensione intrinseca dei datasets, ma risulta interessante cercare di capire se le stime che si ottengono sono plausibili oppure no. Iniziamo con i primi due casi:

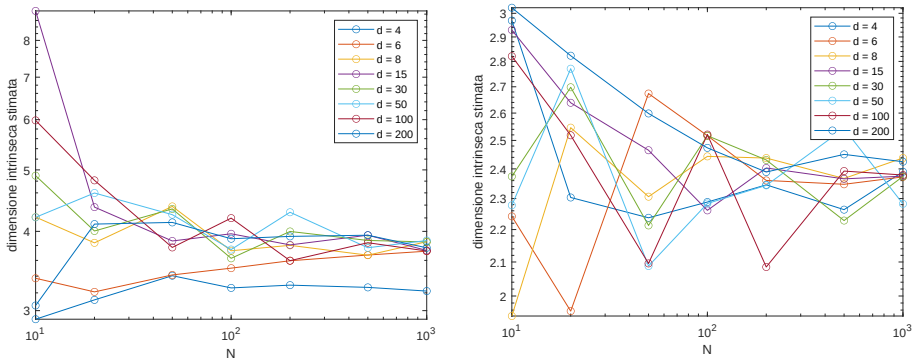

Figura 3.3: Il grafico a sinistra mostra la dimensione intrinseca stimata dal full correlation integral al variare del numero N di punti campionati, per svariati datasets gaussiani aventi dimensione intrinseca $d = 4, 6, 8, 15, 30, 50, 100, 200$, vettore delle medie zero e matrice di covarianza $\Sigma_1 = diag(\frac{1}{2^k})$, per $k = 0, 1, ..., d-1$. Il grafico a destra mostra la stessa cosa, con la differenza che i datasets gaussiani considerati hanno matrice di covarianza $\Sigma_2 = diag(\frac{1}{4^k})$.

Osserviamo che nel primo caso (sinistra) la dimensione stimata dal full correlation integral tende a 4 all'aumentare del numero N di punti campionati (fa eccezione il caso $d = 4$ in cui la dimensione stimata è di circa 3), viceversa nel secondo caso (destra) la dimensione stimata tende ad un valore che si trova all'incirca a metà tra 2 e 3. Ci chiediamo se le stime ottenute siano plausibili oppure no. Per rispondere a questa domanda utilizziamo un ragionamento analogo a quello che si fa per la PCA.

Studiando la PCA, abbiamo visto che la base di autovettori della matrice di covarianza empirica è quella che meglio cattura la varianza di un dataset e che, lungo ciascun asse di tale base, la varianza del dataset coincide con il corrispondente autovalore della matrice. Nel nostro caso le matrici di covarianza empiriche saranno all'incirca Σ_1 e Σ_2, le quali, in quanto matrici diagonali $d \times d$, hanno la base canonica di $\mathbb{R}^d$ come base di autovettori. La PCA stima la dimensione intrinseca guardando alla presenza di ampi salti tra gli autovalori, tuttavia nel nostro caso gli autovalori di Σ_1 e Σ_2 tendono lentamente a zero, pertanto non ci sono ampi salti tra un autovalore e l'altro. Andiamo allora a calcolare la varianza residua: nel primo caso la dimensione stimata tende a 4; notiamo che se consideriamo i primi quattro autovalori della matrice di covarianza Σ_1, si ottiene una varianza residua v minore del 6% della varianza totale:

$$v = \frac{1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8}}{\sum_{k=0}^{d} \frac{1}{2^k}} = \frac{\frac{15}{8}}{\frac{1-(\frac{1}{2})^{d+1}}{1-\frac{1}{2}}} = \frac{\frac{15}{8}}{2 - (\frac{1}{2})^d} \simeq \frac{15}{16} \simeq 0.94,$$

dove le approssimazioni finali valgono per d grande. Dunque, se pensiamo alla dimensione intrinseca come al minimo numero di parametri necessari a descrivere un dataset, sembra ragionevole che la dimensione stimata sia $d = 4$.
Per quanto riguarda il grafico a destra, con un ragionamento del tutto analogo si ottiene che se si considerano i primi tre autovalori della matrice di covarianza Σ_2, allora la varianza residua è minore del 2% della varianza totale; se invece si considerano i primi due autovalori, allora la varianza residua è circa il 7% della varianza totale. Pertanto, anche in questo caso il risultato ottenuto è ragionevole.

Proviamo ora a variare ulteriormente le matrici di covarianza. Consideriamo i grafici in Figura 3.4. Anche in questo caso, come nel precedente, non conosciamo l'esatto valore della dimensione intrinseca dei datasets considerati, ma con un ragionamento analogo a quello fatto prima possiamo dire che i valori stimati dal full correlation integral sono del tutto plausibili. Notiamo in particolare come le dimensioni stimate quando la matrice di covarianza del dataset gaussiano è $\Sigma_2 = diag(\frac{1}{\log(k+1)})$ (in alto a destra) sono molto più alte rispetto agli altri tre casi: ad esempio, in dimensione $d = 200$, le dimensioni intrinseche stimate dal full correlation integral per $N = 10^3$ sono rispettivamente $d = 26.53$ e $d = 163.81$ nei primi due casi (grafici in alto), $d = 8.46$ e $d = 3.71$ negli altri due casi (grafici in basso). Ciò è dovuto al fatto che le funzioni $\frac{1}{k}$, $\frac{1}{k\log(k+1)}$ e $\frac{1}{k\log^2(k+1)}$ vanno a zero molto più rapidamente rispetto alla funzione $\frac{1}{\log(k+1)}$.

3.2.4 Stimatori a confronto

Con le precedenti sperimentazioni abbiamo visto che il full correlation integral è uno stimatore robusto. Vogliamo ora confrontare la sua performance con quella degli stimatori CorrDim e PCA. I grafici in Figura 3.5 mettono a confronto le stime di questi tre stimatori per datasets gaussiani e lineari sia in dimensione bassa che in dimensione alta.
Dai grafici emerge che il full correlation integral risulta più preciso rispetto agli altri due stimatori: a differenza di PCA e CorrDim, esso riesce a stimare corret-

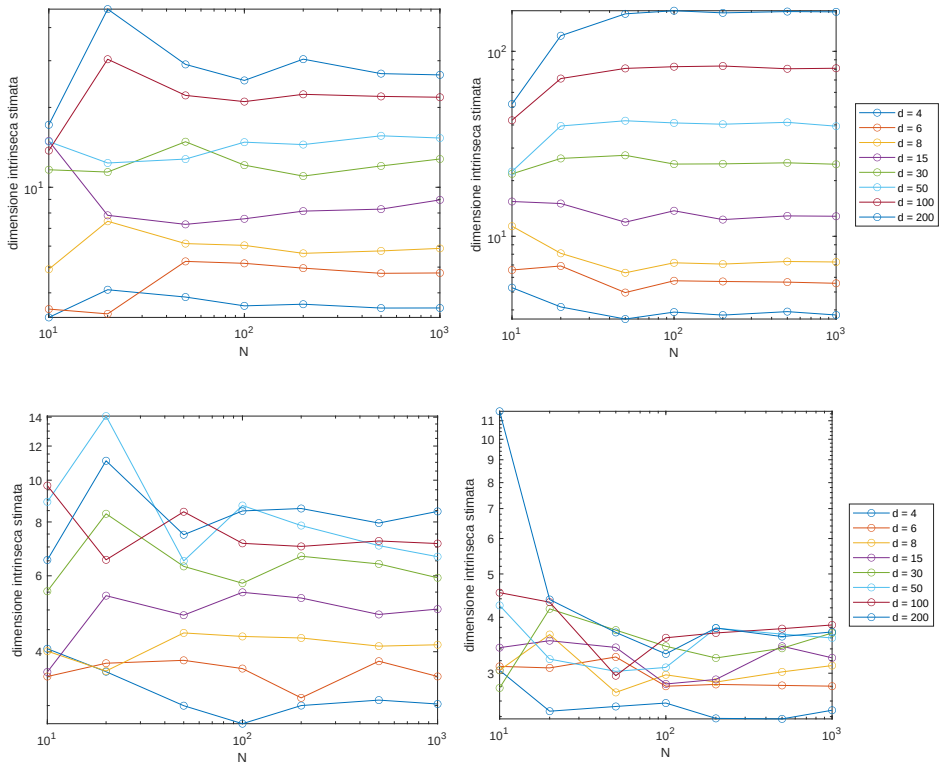

Figura 3.4: I quattro grafici mostrano la dimensione intrinseca stimata dal full correlation integral al variare del numero N di punti campionati, per svariati datasets gaussiani aventi dimensione intrinseca $d = 4, 6, 8, 15, 30, 50, 100, 200$, vettore delle medie zero e matrice di covarianza rispettivamente $\Sigma_1 = diag(\frac{1}{k})$ (in alto a sinistra), $\Sigma_2 = diag(\frac{1}{\log(k+1)})$ (in alto a destra), $\Sigma_3 = diag(\frac{1}{k\log(k+1)})$ (in basso a sinistra) e $\Sigma_4 = diag(\frac{1}{k\log^2(k+1)})$ (in basso a destra) per $k = 1, ..., d$.

tamente la dimensione intrinseca (o comunque l'errore è molto piccolo) anche nei casi in cui il numero N di punti campionati è molto basso. Ad esempio, con soli $N = 20$ punti il full correlation integral restituisce una stima di 20.48 sul dataset lineare avente dimensione intrinseca $d = 20$, e di 9.97 sul dataset gaussiano avente dimensione intrinseca $d = 10$.
Anche la PCA fornisce sempre stime corrette, ad eccezione del caso lineare con dimensione intrinseca $d = 20$, in cui essa sottostima il valore della dimensione intrinseca di uno. CorrDim, d'altro canto, risulta molto meno preciso: in dimensione bassa riesce a fornire una stima corretta sia nel caso gaussiano che in quello lineare soltanto con $N = 3000$ punti. Viceversa, in dimensione elevata fornisce stime non corrette, cosa che sappiamo essere dovuta agli effetti del curse of dimensionality. A causa del curse of dimensionality, per stimare correttamente il valore della dimensione intrinseca d, CorrDim necessita di un numero di punti che sia esponenziale in d, pertanto anche $N = 3000$ punti sono troppo pochi.

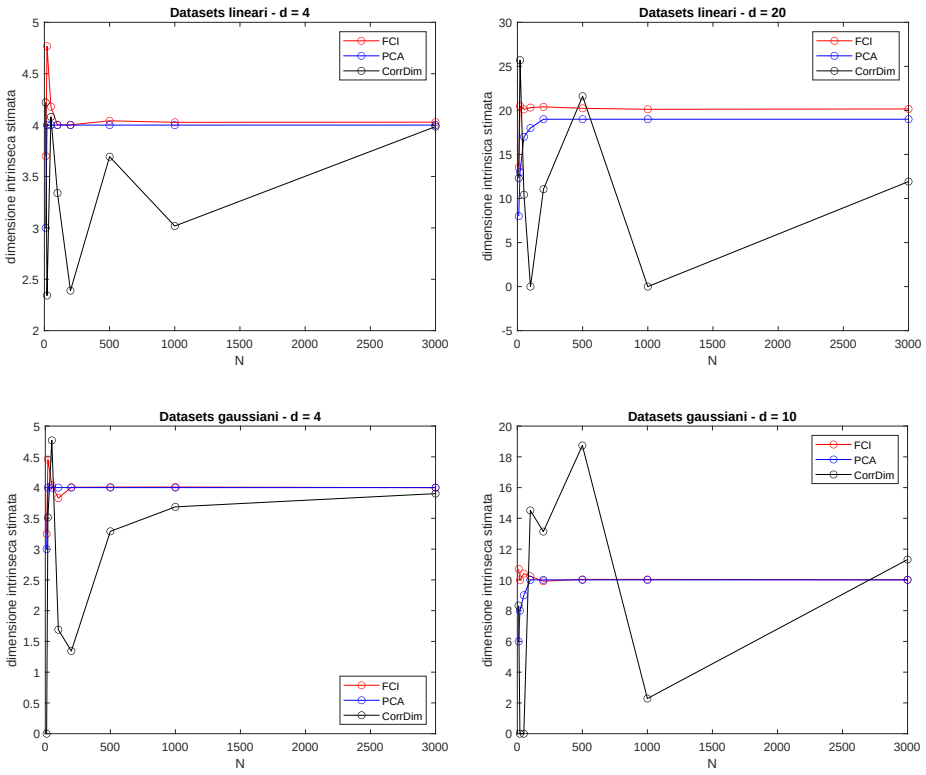

Figura 3.5: I due grafici in alto mostrano la dimensione intrinseca stimata dagli stimatori CorrDim, PCA e full correlation integral al variare del numero N di punti campionati per datasets lineari aventi dimensione intrinseca $d = 4$ (sinistra) e $d = 20$ (destra). I due grafici in basso mostrano la stessa cosa ma per datasets gaussiani aventi dimensione intrinseca $d = 4$ (sinistra) e $d = 10$ (destra).

3.3 Alcuni difetti

Abbiamo visto che il full correlation integral sembra essere uno stimatore robusto ed affidabile anche su datasets che non rispettano appieno le condizioni di linearità, isotropia e isometria, oppure in casi di forte undersampling. È chiaro che anche questo stimatore ha alcuni difetti: da un lato, similmente a quanto accade per la PCA, se il dataset è molto curvo anche il full correlation integral tenderà a sovrastimare il valore della dimensione intrinseca; dall'altro, quando la geometria del dataset risulta molto complessa, l'Equazione (3.4) non riuscirà ad interpolare correttamente il correlation integral calcolato sul dataset, cosa che si traduce in un risultato privo di significato. Vediamo un esempio di queste due limitazioni in Figura 3.6.
Il grafico a sinistra è un chiaro esempio di sovrastima del valore della dimensione intrinseca su datasets curvi. La dimensione intrinseca dell'Hein dataset stimata dal full correlation integral è pari a $d = 10$ anzichè $d = 5$: notiamo infatti che i cerchi in blu si collocano quasi perfettamente al di sopra della linea nera, la quale rappresenta l'Equazione (3.4) con $d = 9$.

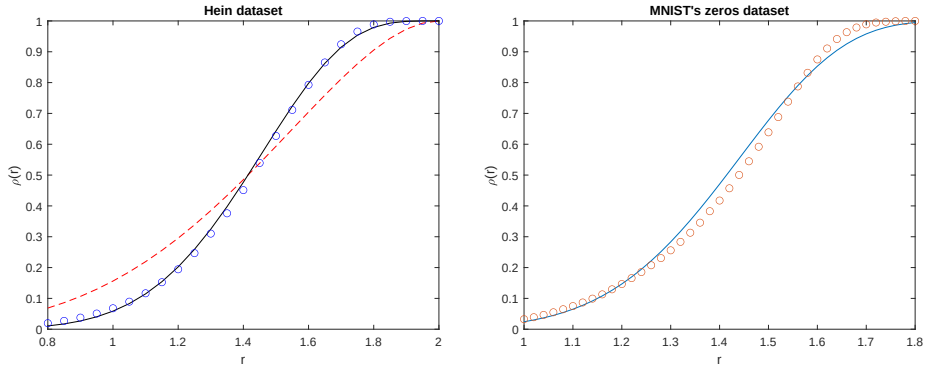

Figura 3.6: Il grafico a sinistra mostra il correlation integral di un Hein dataset (centrato e normalizzato) avente dimensione intrinseca $d = 5$, dimensione di embedding $D = 10$ e costituito da $N = 200$ punti. La linea rossa tratteggiata e la linea in nero rappresentano l'Equazione (3.4) con $d = 4$ e $d = 9$ rispettivamente. Il grafico a destra mostra il correlation integral del dataset costituito da $N = 200$ zeri di MNIST. La linea in blu rappresenta l'Equazione (3.4) con $d = 14$.

Il grafico a destra, invece, mostra che l'Equazione (3.4) con $d = 14$ [1] non interpola bene il correlation integral del dataset: questo risulta particolarmente evidente per $1.3 < r < 1.5$, dove i cerchi in arancione sono visibilmente al di sotto della linea blu, e per $r > 1.6$, dove invece si collocano al di sopra.

[1] Utilizziamo l'Equazione (3.4) con $d = 14$ in quanto la dimensione stimata dal full correlation integral è di $d = 15$. È interessante vedere come la stima cambi al variare della norma utilizzata nel calcolo del correlation integral: la stima di $d = 15$ si ottiene considerando la norma di Frobenius; si ottengono, invece, stime più basse se si considerano le norme uno, due e infinito (si ottengono rispettivamente $d = 11$, $d = 10$ e $d = 10$).

3.4 Una versione multiscala

Al fine di stimare la dimensione intrinseca di datasets più complessi, quali datasets sferici[2], Hein datasets o MNIST, utilizziamo una versione multiscala del full correlation integral. L'idea sulla quale questa si basa è la seguente: localemente i datasets artificiali sono ben descritti da datasets lineari campionati uniformemente (tenendo sempre in considerazione i limiti dovuti al curse of dimensionality); ha senso dunque utilizzare il full correlation integral localmente, ovvero applicarlo a sottoinsiemi del dataset X della forma

$$X(c, r) = \{x_i \in X \mid ||c - x_i|| \leq r\}, \tag{3.15}$$

dove r è tale per cui il sottoinsieme $X(c, r)$ è ben approssimato da un dataset lineare, mentre c è un qualsiasi punto in X. Ripetendo questa procedura per vari punti $c \in X$ e per diversi $r > 0$ (piccoli), si ottiene una serie di stime locali della dimensione intrinseca che devono poi essere combinate per ricavare la dimensione intrinseca del dataset di partenza. Purtroppo al momento non esiste alcun risultato teorico che ci dica come scegliere il valore di r in modo tale che valga l'approssimazione lineare oppure come combinare le stime locali per ricavare la dimensione intrinseca del dataset originale. Quello che si fa in questi casi è considerare una serie di esempi per cercare di trovare una procedura di stima euristica.

Guardiamo l'esempio in Figura 3.7. Abbiamo applicato il full correlation integral a due sottoinsiemi di uno swiss roll dataset (i rispettivi centri sono mostrati in figura), e abbiamo fatto il plot delle stime ottenute viste come funzioni della distanza r dal centro. È possibile osservare i seguenti fatti:

- Le stime legate alla parte più interna del dataset sono sistematicamente più alte di quelle invece legate alla parte più esterna. Questa è una chiara conseguenza della curvatura: laddove il dataset è più curvo, il full correlation integral tenderà a sovrastimare il valore della dimensione intrinseca;
- Le stime sono per lo più costanti, dunque le proprietà geometriche del dataset sono per lo più costanti per $0 < r \leq 1$;
- Per r vicino a 0 o a 1 le stime risultano più elevate. Ciò è dovuto nel primo caso all'assenza di un numero di punti adeguato (il full correlation integral è affidabile anche in casi di undersampling, ma comunque sbaglia se il numero di punti è troppo piccolo); nel secondo caso al fatto che il full correlation integral riesce a captare la struttura globale del dataset, sovrastimando così la dimensione intrinseca a causa della curvatura;
- L'altezza della parte piana della curva più in basso è circa 2, che è la dimensione intrinseca del dataset.

Le precedenti osservazioni suggeriscono la seguente procedura euristica, che prende il nome di *multi-scale full correlation integral*:

[2]Nonostante il correlation integral analitico venga calcolato sulla sfera, i datasets sferici sono considerati datasets complessi. Questo accade perchè noi non sappiamo a priori se un certo dataset è sferico, pertanto l'ultimo passo del full correlation integral (aggiunguere uno alla dimensione stimata) porterebbe a sovrastimare di uno la dimensione intrinseca di un dataset sferico.

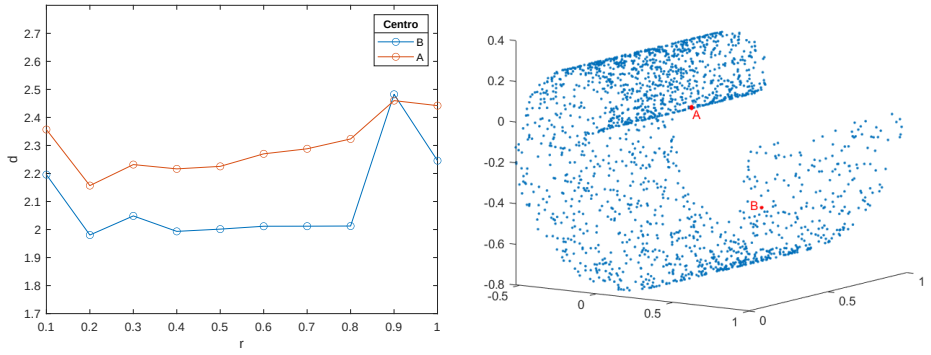

Figura 3.7: Il grafico a sinistra mostra la dimensione intrinseca stimata dal full correlation integral su due sottoinsiemi di uno swiss roll dataset al variare della distanza r dal centro. Il grafico a destra è una rappresentazione dello swiss roll dataset utilizzato: i punti A e B sono i centri dei due sottoinsiemi considerati per il grafico a sinistra. Il dataset è costituito da $N = 2000$ punti.

1. Si applica il full correlation integral su sottoinsiemi del dataset X della forma descritta in (3.15), facendo variare sia il punto c che la distanza r; si ottenengono così varie stime locali $d(c, r)$.
   Una possibile variante consiste nel sostituire la distanza r con il numero k di punti più vicini, e cioè si applica il full correlation integral su sottoinsiemi di X della forma:
   $$X(c, k) = \{x_i \in X \mid x_i \text{ è uno dei primi k vicini di c in } X\}. \tag{3.16}$$
2. Si fa il plot delle stime $d(c, r)$ viste come funzioni di r, per tutti i centri c;
3. Si considerano affidabili le curve che mostrano una parte piana ben pronunciata, che chiameremo *plateau*: plateau più lunghi corrispondono a regioni del dataset in cui le proprietà geometriche del dataset variano meno;
4. Si seleziona l'altezza del plateau più basso come dimensione intrinseca del dataset.

A riprova della ragionevolezza di tale procedura guardiamo la Figura 3.8: abbiamo applicato il multi-scale full correlation integral ad un Hein dataset (sinistra) e ad un dataset multidimensionale costituito dall'unione di due datasets lineari aventi dimensione intrinseca diversa (destra).
Notiamo che in entrambi i casi il plateau più basso identifica la dimensione intrinseca corretta. Inoltre, mentre nel primo caso, in cui il dataset è molto curvo, la maggior parte delle curve identificano dimensioni intrinseche locali più elevate, nel caso multidimensionale le stime si concentrano maggiormente attorno ai valori $d_1 = 20$ e $d_2 = 30$. Notiamo, infine, che per ragioni computazionali abbiamo utilizzato in entrambi i casi la versione dell'algoritmo con i primi k vicini.

In conclusione, il multi-scale full correlation integral è una procedura utile per trattare datasets più complessi. In generale, ogni stimatore della dimensione intrinseca ammette una versione multiscala (ad esempio nel capitolo precedente abbiamo visto la PCA locale). In questo caso, però, il multi-scale full correlation

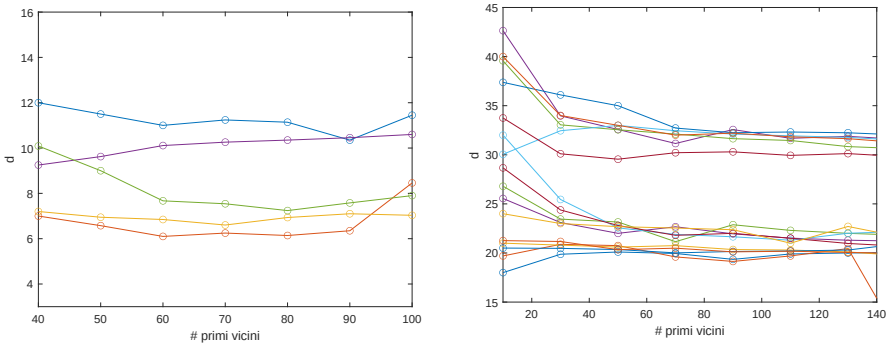

Figura 3.8: (Sinistra) Multi-scale full correlation integral applicato ad un Hein dataset avente dimensione intrinseca $d = 6$, dimensione di embedding $D = 12$ e costituito da $N = 10^4$ punti. (Destra) Multi-scale full correlation integral applicato ad un dataset multidimensionale costituito dall'unione di due datasets lineari aventi dimensione intrinseca $d_1 = 20$ e $d_2 = 30$ rispettivamente, dimensione di embedding $D = 50$ e costituiti da $N = 1000$ punti.

integral gode di alcune proprietà che mancano agli altri stimatori: la robustezza del full correlation integral nei casi in cui le condizioni di linearità, isotropia e isometria del dataset sono in parte violate oppure nei casi di undersampling, fa sì che nella sua versione multiscala risultino più contenuti sia gli effetti della curvatura che quelli del curse of dimensionality, che in genere è inevitabile nelle versioni multiscala.

Capitolo 4

Conclusioni e prospettive future

In questo lavoro di tesi abbiamo introdotto il manifold learning e abbiamo trattato in particolare il problema di stima della dimensione intrinseca di un dataset. Abbiamo poi analizzato in dettaglio gli stimatori CorrDim (Correlation Dimension), PCA (Principal Component Analysis) e Full correlation integral, con l'obiettivo di studiarne le proprietà matematiche e di confrontarne la performance tramite sperimentazioni su datasets sia reali che generati. L'obiettivo di questo paragrafo è quello di riassumere i principali risultati ottenuti e di fornire qualche spunto per possibili sviluppi futuri.

Nel primo capitolo abbiamo dato una breve introduzione del manifold learning e delle sue principali applicazioni.

Nel secondo capitolo abbiamo analizzato in dettaglio gli stimatori CorrDim e PCA: ne abbiamo dimostrato formalmente la consistenza tramite i Teoremi 2.2.3 e 2.3.1 rispettivamente; inoltre, abbiamo visto che CorrDim e PCA soffrono di due difetti tra loro complementari: il primo soffre del cosiddetto *curse of dimensionality*, ovvero per funzionare correttamente necessita di un numero esponenziale di punti (nella dimensione intrinseca d), cosa che si traduce in una sistematica sottostima del valore della dimensione intrinseca quando questa diventa elevata; il secondo, invece, tende a sovrastimare la dimensione intrinseca delle varietà curve. Non siamo però riusciti a dare una dimostrazione formale del curse of dimensionality ma solo un'euristica; potrebbe dunque essere interessante cercare di formalizzare quanto detto a riguardo. Notiamo inoltre che per dimostrare l'Equazione (2.6) nel Teorema 2.2.3 abbiamo utilizzato un teorema molto più generale (Teorema 2.2.4), ma potrebbe risultare interessante cercare di trovare una dimostrazione alternativa che non ne faccia uso.

Il terzo capitolo, infine, è interamente dedicato al Full correlation integral. I Teoremi 3.1.1 e 3.1.2 ne dimostrano la consistenza nel caso in cui si abbia un dataset sferico avente dimensione intrinseca $d-1$ e immerso in $\mathbb{R}^d$ tramite mappa di inclusione. Una possibile alternativa sarebbe quella di provare a ripercorrere le due dimostrazioni nel caso in cui si abbia invece un dataset gaussiano.
Abbiamo visto tramite opportune sperimentazioni che il Full correlation integral

sembra essere uno stimatore robusto ed affidabile anche su datasets che non rispettano appieno le condizioni di isotropia, isometria e linearità, che sappiamo essere sufficienti per il suo corretto funzionamento. Inoltre, esso si comporta bene anche in casi di forte undersampling. Sembrerebbe dunque che il Full correlation integral non soffra del curse of dimensionality; bisogna però tenere presente che la maggior parte delle sperimentazioni sono state fatte su datasets generati: cosa succederebbe se invece il dataset fosse molto più complesso? In questi casi il Full correlation integral soffrirebbe del curse of dimensionality?
Chiaramente anche il Full correlation integral non è privo di difetti: ad esempio, similmente a quanto accade per la PCA, anche il Full correlation integral tende a sovrastimare la dimensione intrinseca di datasets curvi.
Purtroppo al momento non esistono stimatori che siano totalmente privi di limitazioni, e questo fa sì che il problema di trovare stimatori che siano il più possibile robusti ed affidabili sia ancora ampiamente trattato.

Appendice A

Alcuni codici

Riportiamo di seguito le principali funzioni in MATLAB utilizzate nelle sperimentazioni fatte.

Funzione A.1: La seguente funzione calcola sul dataset X il valore del correlation integral relativo alla distanza r. I dati corrispondono alle righe della matrice X.

```
function rho = calcolo_di_rho(r,X)
    [N,D] = size(X);
    n = 0;
    for i = 1:N-1
        for j = i+1:N
            if r - norm(X(i,:)-X(j,:)) >= 0
                n = n+1;
            end
        end
    end
    rho = (2/(N*(N-1)))*n;
end
```

Funzione A.2: La seguente funzione restituisce la dimensione intrinseca d stimata dallo stimetore CorrDim applicato al dataset X.

```
function d = CorrDim(X)
    [N,D] = size(X);
    R = [0.07:0.01:0.15,0.151:0.001:3];
    n = length(R);
    rho = zeros(1,n);
% Calcolo il correlation integral al variare di r in R
    for i = 1:n
        rho(i) = calcolo_di_rho(R(i),X);
    end
% Mi restringo a considerare i valori non nulli
    ids = rho ~= 0;
    rho1 = rho(ids);
    R1 = R(ids);
```

```
14 % Eseguo il fit lineare sui primi 300 punti
15    p = polyfit(log10(R1(1:300)),log10(rho1(1:300)),1);
16 % La dimensione intrinseca coincide con il coefficiente
17 % angolare della retta
18    d = p(1);
19 end
```

Funzione A.3: La seguente funzione restituisce la dimensione intrinseca stimata dal full correlation integral. Dataset è la matrice $N \times D$ che rappresenta il dataset. $x0$ è l'initial guess richiesta dalla funzione lsqcurvefit che esegue il fit non lineare.

```
1  function dim = FCI_estimator(dataset,x0)
2    R = [0.1:0.005:1.9];
3    [N,D] = size(dataset);
4    X = zeros(N,D);
5    barycenter = zeros(1,D);
6    rho = zeros(1,length(R));
7  % Calcoliamo il baricentro del dataset
8    for i = 1:D
9        barycenter(i) = mean(dataset(:,i));
10   end
11 % Centriamo e normalizziamo il dataset
11   for i = 1:N
12       X(i,:) = dataset(i,:) - barycenter;
13       X(i,:) = X(i,:)/norm(X(i,:));
14   end
15 % Calcoliamo il correlation integral al variare di r in R
16   for i = 1:length(R)
17       rho(i) = calcolo_di_rho(R(i),X);
18   end
19 % La funzione analytical_FCI rappresenta il correlation
20 % integral analitico calcolato sulla sfera
21   fun = @analytical_FCI;
22 % Eseguiamo un fit del correlation integral analitico
23 % calcolato sulla sfera con il correlation integral
24 % calcolato sul dataset normalizzato
25   dim = lsqcurvefit(fun,x0,R,rho,0);
26 % Aggiungiamo uno alla dimensione stimata
27   dim = dim+1;
28 end
```

Bibliografia

[BT99] Christopher M. Bishop e Michael E. Tipping. "Probabilistic principal component analysis." In: *J.R Statist. Soc. B, 61 (Part 3)* (1999), pp. 611–622.

[EGR19] Vittorio Erba, Marco Gherardi e Pietro Rotondo. "Intrinsic dimension estimation for locally undersampled data". In: *Scientific Reports, 9(1):17133* (nov. 2019).

[Erb21] Vittorio Erba. "Aspects of data structure in machine learning [tesi di dottorato]". Milano: Università degli studi di Milano (Dipartimento di fisica), 2021.

[GP83] Peter Grassberger e Itamar Procaccia. "Measuring the strangeness of strange attractors". In: *Physica 9D* (1983), pp. 189–208.

[Hoe61] Wassily Hoeffding. "The strong law of large numbers for U-statistics". In: *North Carolina State University. Dept. of Statistics* (1961).

[LMR17] Anna V. Little, Mauro Maggioni e Lorenzo Rosasco. "Multiscale geometric methods for data sets I: Multiscale SVD, noise and curvature." In: *Applied and Computational Harmonic Analysis, 43* (2017), pp. 504–567.

[PM13] Dominique Perraul-Joncas e Marina Meila. "Non-linear dimensionality reduction: riemannian metric estimation and the problem of geometric discovery". In: *arXiv: 1305.7255 [stat.ML]* (2013).