Università degli Studi di Padova
Dipartimento di Scienze Statistiche

Corso di Laurea Triennale in
Statistica per le Tecnologie e le Scienze

Tecniche per la riduzione della dimensionalità nei big data: un confronto tra diversi approcci

Relatore: Prof. Antonio Canale
Dipartimento di Scienze Statistiche

Laureando: Augusto Celon
Matricola: 1197827

Anno Accademico 2021/2022

Indice

Introduzione

L'argomento di questa relazione verte sulle tecniche di riduzione della dimensionalità che, dato l'esponenziale aumento della quantità di dati disponibili negli ultimi decenni, sono sempre più utilizzate. Vengono discussi diversi approcci e valutati i pregi e i difetti di ognuno di essi.
Nel primo capitolo viene data una descrizione generale di queste tecniche mentre nel secondo ci si focalizza su due metodi in particolare, t-SNE e UMAP; nel terzo capitolo viene proposto un approccio che risulta essere migliore rispetto ai due precedenti e nel quarto sono presenti delle illustrazioni grafiche.

Capitolo 1

Riduzione della dimensionalità

1.1 Di cosa si tratta

Negli ultimi decenni la disponibilità di dati provenienti da esperimenti è aumentata drasticamente; tra gli ambiti in cui questo fenomeno si è sviluppato maggiormente ci sono le scienze sperimentali come biologia e chimica.
Poiché gli strumenti di laboratorio sono diventati sempre più complessi e sofisticati, ad oggi si possono raccogliere centinaia di migliaia di rilevazioni per un singolo esperimento e dunque ci si trova spesso ad avere degli insiemi di dati in cui le variabili sono in numero di molto maggiore rispetto alle osservazioni.
Ad esempio, negli studi di genetica, si hanno spesso migliaia di variabili (geni) e campioni di numerosità nell'ordine delle decine.
I metodi statistici utilizzati per l'analisi sono dunque messi alla prova con dataset 'larghi'.
Tuttavia, avendo grandi masse di dati, si verifica spesso che non tutte le variabili rilevate siano informative e che possono per questo essere eliminate, sostituite o trasformate. Posso ottenere così una versione più 'pulita' dei dati che si presenta in dimensione minore rispetto a quella originale, pur mantenendo una buona parte del contenuto informativo presente nei dati di partenza.
I dati trasformati inoltre possono dare una visione di insieme che non era possibile avere prima data l'elevata dimensionalità.
Le procedure che rendono ciò possibile sono dette **tecniche di riduzione della dimen-**

sionalità il cui obiettivo è di trovare una rappresentazione dei dati con dimensione inferiore ma che trattenga il massimo dell'informazione dei dati originali; ad oggi sono un argomento di studio e sviluppo nell'ambito statistico.
Sono inoltre utilizzate in molte aree scientifiche, dalla ricerca medica all'analisi di testi (text mining) e in informatica.
Le tecniche di riduzione della dimensionalità possono essere affrontate principalmente in due modi: mantenendo le variabili rilevanti del dataset originale ed eliminando le altre o trovando un sottoinsieme di nuove variabili, ottenute da una combinazione di quelle originali, che contengano quanta più informazione delle variabili di partenza.
Questo ultimo metodo è quello su cui ci concentreremo.
Esistono diversi approcci per costruire un procedimento che riduca la dimensionalità di un dataset ma ne distinguiamo tre in particolare.

1.2 Metodi basati sulla statistica e sulla teoria dell'informazione

Questi metodi riducono la dimensione dei dati originali basandosi su criteri provenienti dalla statistica o dalla teoria dell'informazione. I metodi basati sulla teoria dell'informazione possono essere visti in un certo senso come una generalizzazione di quelli basati su criteri statistici poiché riescono a riconoscere relazioni non lineari tra variabili, e molti di questi sono invarianti a trasformazioni monotone delle variabili di partenza.
Fanno parte di questi metodi:

- **l'analisi delle componeneti principali o PCA**, uno dei metodi più utilizzati; dato un insieme di dati di partenza $X \in \mathbb{R}^p$, la PCA si basa su una trasformazione lineare delle colonne di X che crea un nuovo insieme di dati $Y \in \mathbb{R}^p$ con p nuove variabili (colonne) dette *componenti principali*. Queste nuove variabili hanno la peculiarità di avere varianza massima ed essere tra di loro incorrelate.
Dato che la prima componente principale è quella che contiene la maggior porzione di varianza dei dati e, a mano a mano, le successive ne contengono sempre meno, prendendo solamente le prime k delle p componenti totali si può ottenere una riduzione della dimen-

sionalità senza perdere troppa informazione poiché le restanti $p - k$ componenti scartate spiegano una porzione minima della struttura di varianza-covarianza dei dati iniziali.
Il valore di k viene scelto in base alla porzione di varianza totale spiegata dalle componenti (tengo tante componenti finché non raggiungo una porzione di varianza totale spiegata scelta a priori) e alla loro interpretabilità.
Un vantaggio della PCA è che, per poterla applicare, non si fa nessuna assunzione distributiva sui dati;

- **l'analisi di curve e superfici principali**, che è una generalizzazione dell'analisi delle componenti principali; ciò che viene generalizzato sono gli assi individuati dalla PCA, fornendo un'approssimazione dei dati in $\mathbb{R}^p$ come curva liscia (univariata).
Le superfici principali sono invece un concetto ancora più generale e forniscono un'approssimazione attraverso una superficie di dimensione 2 o maggiore.

- **l'analisi fattoriale o FA**, che cerca di ridurre la dimensionalità facendo uso dei fattori, ossia quantità inosservabili e responsabili della alta correlazione tra coppie di variabili. Il fattore è una dimensione latente, non osservabile e tutti i fattori sono incorrelati tra di loro.

- **l'analisi delle componenti indipendenti o ICA**, che si basa sull'individuazione di dimensioni latenti in maniera simile all'analisi fattoriale; in questo approccio però le dimensioni latenti, oltre ad essere incorrelate tra loro (come nella FA), sono anche indipendenti.

1.3 Metodi basati sui dizionari

Data una matrice X di dati, le cui righe rappresentano le unità statistiche e le cui colonne le variabili, un'altra famiglia di metodi di riduzione della dimensionalità si basa sulla decomposizione di X.
X viene trasformata in modo da ottenere una nuova matrice le cui variabili sono diverse.
La trasformazione non è altro che un cambiamento lineare della base tra i due gruppi di

variabili e la matrice che esprime il cambiamento di base è nota come dizionario.
Questi metodi non verranno ulteriormente approfonditi.

1.4 Metodi basati su proiezioni

Un'altra classe di metodi di riduzione è quella in cui si proiettano i dati originali in un sottospazio matematico con particolari caratteristiche vantaggiose per applicare la diminuzione della dimensionalità. I metodi maggiormente rilevanti che appartengono a questa classe sono le **proiezioni in varietà topologiche (manifolds)**.
Sono di particolare interesse poiché negli ultimi anni sono risultati sempre più utili per la loro capacità di identificare relazioni non lineari tra le variabili e di adattarsi a strutture locali dei dati.
Algoritmi distinti differiscono sicuramente a livello computazionale e di fondamenti teorici secondo i quali sono stati sviluppati ma hanno tutti lo stesso obiettivo: ridurre la complessità della struttura iniziale dei dati e fornire una rappresentazione interpretabile dell'informazione. Tra questi vediamo in particolare **t-SNE** e **UMAP**.

Capitolo 2

UMAP e t-SNE

Ci concentriamo adesso su due approcci in particolare, entrambi facenti parte della categoria dei metodi appartenenti alle proiezioni.
Sebbene siano due approcci pensati per svolgere lo stesso compito, operano in maniera differente.
D'ora in poi con 'mappa' intendiamo lo spazio in dimensione ridotta e con 'punti della mappa' i punti che giacciono su di esso.

2.1 UMAP

La Uniform Manifold Approximation and Projection for Dimension Reduction, o UMAP, è una tecnica di riduzione della dimensionalità che pone le sue basi sulla geometria riemanniana [5]; il risultato è un algoritmo pratico e scalabile che si può applicare a varie tipologie di dati.
UMAP non ha restrizioni per quanto riguarda la dimensione della mappa che si vuole ottenere e ciò lo rende utile per un utilizzo generale nel contesto del machine learning.
L'obiettivo principale di UMAP è quello di preservare nei punti della mappa le distanze locali presenti nei dati originali a discapito di quelle globali.

2.1.1 Definizioni

Definiamo ora alcuni concetti matematici che ci servono per capire le basi di questo algoritmo.

Definizione 1 (Insieme fuzzy) *Un insieme fuzzy è un insieme che rientra in un'estensione della teoria classica degli insiemi, caratterizzato da una funzione detta grado di appartenenza, che mappa gli elementi in un intervallo reale continuo.*
Il valore 0 indica che un elemento non è per niente incluso nell'insieme fuzzy, il valore 1 indica che è certamente incluso, mentre valori tra zero e uno indicano il grado di appartenenza dell'elemento all'insieme fuzzy in questione.
L'appartenenza di un elemento a un insieme perciò non è più bivalente (appartiene o non appartiene), proprietà della teoria degli insiemi classica, ma è fuzzy e perciò assume valori nell'intervallo $[0, 1]$.
Un insieme fuzzy è una coppia (U, m) *dove* U *è un insieme e* $m\colon U \to [0, 1]$ *una funzione di appartenenza. L'insieme* U *è anche chiamato universo e, per ogni* $x \in U$, *il valore* $m(x)$ *è chiamato grado di appartenenza di* x *in* (U, m).
La funzione $m = \mu_A$ *è chiamata funzione di appartenenza dell'insieme fuzzy* $A = (U, m)$.
Per un insieme $U = \{x_1, \ldots, x_n\}$, *l'insieme fuzzy è il seguente:*

$$\{m(x_1)/x_1, \ldots, m(x_n)/x_n\}$$

.

Definizione 2 (Funtore) *In matematica, è spesso utile tradurre problemi geometrici o topologici in fatti algebrici o insiemistici, che spesso risultano di più facile risoluzione. Questo passaggio viene fatto normalmente tramite un funtore.*
Un ***funtore*** *è una mappa fra categorie che ne conserva le strutture.*

Definizione 3 (Varietà Riemanniana) *In geometria, una varietà è uno spazio topologico che è localmente simile a uno spazio topologico ben conosciuto (ad esempio lo spazio euclideo n-dimensionale), ma che globalmente può avere proprietà geometriche differenti*

(ad esempio può essere 'curvo' contrariamente allo spazio euclideo).
In particolare, una varietà riemanniana è una varietà differenziabile su cui sono definite le nozioni di distanza, lunghezza, geodetica (la curva più breve che congiunge due punti di uno spazio), area (o volume) e curvatura. È importante in quanto permette di modellizzare spazi 'curvi' di dimensione arbitraria.

2.1.2 Fondamenti teorici

In generale, UMAP si basa su un'ipotesi detta **manifold hypothesis** (ipotesi della varietà), che sostiene che i dati ad alta dimensionalità in realtà giacciono su delle varietà di dimensione minore. Questo significa che, nonostante abbiamo dati di dimensionalità elevata, esiste qualche rappresentazione di essi in dimensione ridotta.
UMAP ottiene un'approssimazione dei dati di partenza nei termini di una rappresentazione topologica.
Ottenere questa topologia è un processo che si divide in due fasi: trovare una varietà riemanniana su cui si assume che giacciano i dati e costruire un insieme fuzzy che approssimi adeguatamente la varietà.
Data una rappresentazione in dimensione ridotta dei dati iniziali, si può effettuare un processo analogo e ottenere una rappresentazione topologica anche per essi.
UMAP infine ottimizza la disposizione dei punti della mappa minimizzando una misura di similiarità tra le due topologie.

Varietà su cui si assume giacciano i dati

Avendo i dati di partenza $X = [X_1, ..., X_N]$ di dimensione N, risulta utile assumere che la varietà M su cui assumiamo giacciano sia distribuita uniformemente. Sotto questa assunzione possiamo inoltre approssimare la distanza geodetica tra ogni x_i e i suoi punti vicini, normalizzando la distanza che ha x_i dal suo k-esimo 'nearest neighbor'.
Otteniamo in questa maniera, per ogni punto del dataset, una metrica indipendente dalle altre.
Al fine di ottenere una struttura generale che riassuma le caratteristiche locali di ogni metrica, convertiamo le singole metriche in insiemi fuzzy e applichiamo ad essi l'operatore unione.

Trasformazione in insiemi fuzzy

Ogni spazio metrico (insieme di elementi in cui è definita una distanza, il più usato è quello euclideo), può essere tradotto in un insieme fuzzy grazie ad un funtore, che conserva le caratteristiche topologiche dello spazio.
Nel nostro caso gli spazi metrici sono quelli definiti sui punti del dataset, e una volta trasformati in insiemi fuzzy tramite funtore, si procede a raggrupparli (con l'operatore unione) in un singolo insieme fuzzy che cattura gli aspetti rilevanti della struttura topologica dei dati.
Una volta ottenuta questa approssimazione di X, possiamo implementare una riduzione trovando una rappresentazione in dimensione ridotta che abbia una struttura topologica simile a quella dei dati iniziali.

Rappresentazione in dimensione ridotta

Assumiamo che $Y = Y_1, \ldots, Y_N \subseteq \mathbb{R}^d$ con $(d \ll n)$ siano i punti di una mappa ottenuta a partire dai dati originali X, in modo che Y_i rappresenti il punto X_i.
Diversamente da quello che abbiamo fatto con X, dove è stata trovata una varietà in cui i dati sono distribuiti uniformemente, con Y scegliamo a priori una varietà (di solito è $\mathbb{R}^d$) nota, di cui conosciamo la metrica e che quindi ci permette di passare direttamente alla rappresentazione in insieme fuzzy.
Una volta ottenute le rappresentazioni di X e Y sotto forma di insiemi fuzzy, dobbiamo misurare la somiglianza tra questi; utilizziamo una funzione di perdita chiamata **entropia incrociata**.

Definizione 4 (Entropia incrociata) *L'entropia incrociata C tra due insiemi fuzzy (A, μ) e (A, ν) è definita come:*

$$C((A,\mu),(A,\nu)) \equiv \sum_{a \in A} \left(\mu(a) log \left(\frac{\mu(a)}{\nu(a)} \right) + (1 - \mu(a)) log \left(\frac{1-\mu(a)}{1-\nu(a)} \right) \right)$$

L'obiettivo finale è di minimizzare C, ottenendo così la versione ottima (secondo i vincoli imposti da UMAP) dei dati in dimensione ridotta.

2.1.3 Aspetti computazionali

Dal punto di vista computazionale, UMAP utilizza e opera su grafi pesati per rappresentare gli insiemi fuzzy.
Per utilizzare UMAP si fanno tre assunzioni:
- esiste una varietà sulla quale i dati sono distribuiti uniformemente;
- questa varietà è connessa localmente;
- preservare la struttura topologica di questa varietà è l'obiettivo principale.

Con i grafi pesati si riesce a mantenere queste assunzioni e a rappresentare bene dal punto di vista computazionale l'approssimazione data dall'insieme fuzzy.
UMAP, a livello computazionale, può essere descritto in queste fasi: per prima cosa si costruisce un particolare grafo pesato ottenuto con l'algoritmo K-NN (k-nearest neighbor), poi viene definita una funzione di perdita che preservi le caratteristiche del grafo e infine viene computato un grafo di dimensione ridotta che minimizzi questa funzione.
Il k-nearest neighbors è un algoritmo utilizzato nel riconoscimento dei punti basandosi sulle caratteristiche dei k punti vicini a quello considerato. Il parametro k può variare; generalmente all'aumentare di k si riduce il rumore, ma il criterio di riconoscimento diventa meno attendibile.

Creazione del grafo pesato

La prima fase di UMAP consiste nella costruzione di un grafo pesato ottenuto con il metodo k-neighbor.
Poniamo che $X = \{x_1, \ldots, x_N\}$ sia il dataset di input con una metrica d: $X \times X \to \mathbb{R}_{\geq 0}$. Dato un valore per l'iperparametro k, per ogni x_i costruiamo l'insieme $x_{i1}, \ldots, x_{ik}$ dei k nearest neighbors di x_i sotto la metrica d.
Per ogni x_i definiamo ρ_i e σ_i, dove

$$\rho_i = \min\{d(x_i, x_{i_j}) | 1 \leq j \leq k, d(x_i, x_{i_j}) \geq 0\},$$

e σ_i sia il valore tale che

$$\sum_{j=1}^{k} \exp\left(\frac{-\max(0, d(x_i, x_{i_j}) - \rho_i)}{\sigma_i}\right) = log_2(k).$$

La corretta selezione di ρ_i assicura che x_i si colleghi almeno con un altro punto con un arco di peso 1; questo è l'equivalente di un insieme fuzzy localmente connesso in x_i.
La scelta di σ_i corrisponde a una normalizzazione lisciata dei fattori, e definisce la metrica riemanniana locale relativa al punto x_i.
Possiamo adesso definire il grafico diretto pesato $\bar{G} = (V, E, \omega)$, dove V sono i nodi (vertices) e semplicemente rappresentano i punti di X. Possiamo quindi formare un insieme di archi diretti (edges) $E = \{(x_i, x_{i_j}) | 1 \leq j \leq k, 1 \leq i \leq N\}$, e definire la funzione peso ω nel seguente modo:

$$\omega(x_i, x_{i_j}) = \exp\left(\frac{-\max(0, d(x_i, x_{i_j}) - \rho_i)}{\sigma_i}\right).$$

Otteniamo così N diversi grafi locali e, per riassumerli in un'unica rappresentazione topologica, si procede nella seguente modalità.
Data la matrice di adiacenza A del grafo $\bar{G}$, definiamo la seguente matrice simmetrica:

$$B = A + A^\top - A \circ A^\top,$$

dove con $\circ$ si intende il prodotto di Hadamard (o pointwise product).
Questa formula deriva dall'operatore unione degli insiemi sfocati (t-conorm).
Se A_{ij} è interpretato come la probabilità che un arco diretto che va da x_i a x_j esista, allora B_{ij} si interpreta come la probabilità che almeno uno dei due archi diretti che vanno da x_i a x_j o viceversa esista.
Il grafo finale ottenuto, che riassume gli N grafi locali, e che chiamiamo G, è indiretto e ha come matrice di adiacenza B.

Costruzione del grafico

Nella pratica, UMAP costruisce il grafo dei punti della mappa utilizzando un algoritmo di force-directed placement, che sfrutta un insieme di forze attrattive applicate sugli archi in contrasto con delle forze repulsive applicate sui nodi. Questo algoritmo individua le posizioni spaziali dei nodi in modo da ottenere la configurazione che minimizzi queste forze.
Le forze descritte sopra sono derivate dall'ottimizzazione dell'entropia inctrociata tra il grafo G e un equivalente grafo pesato H costruito dai punti di dimensione ridotta $\{y_i\}_{i=1,\dots,N}$.
L'obiettivo finale è trovare una posizione per i punti della mappa tale che il relativo grafo approssimi al meglio G, e ciò si ottiene minimizzando l'entropia incrociata tra i due grafi

(dato che G cattura la struttura topologica dei dati di partenza, l'equivalente grafo in dimensione ridotta H ha la stessa proprietà, e quindi fornisce una buona rappresentazione in bassa dimensione della topologia generale dei dati).

2.2 t-SNE

Il t-SNE (t distributed Stochastic Neighbor Embedding) è un metodo di riduzione della dimensionalità il cui obiettivo principale è rappresentare graficamente dei dati di dimensione p proiettandoli in dimensione minore d (che di solito è 2 o 3 sempre definita a priori); dobbiamo quindi trovare una funzione $g : \mathbb{R}^p \to \mathbb{R}^d$ [1].

2.2.1 SNE

Questo metodo pone le sue basi su SNE (Stochastic Neighbor Embedding), un algoritmo per la riduzione della dimensionalità sviluppato all'inizio degli anni 2000, e si propone come una versione migliorata dello stesso.
SNE è definito in passi diversi.
Per prima cosa si definisce una distribuzione sulle coppie di unità p-dimensionali (una misura di somiglianza tra le coppie di unità) dove è assegnata probabilità alta alle unità simili e probabilità bassa a quelle meno simili. La somiglianza tra x_i e x_j è definita come la probabilità condizionata $p_{j|i}$ che x_i abbia x_j come punto più vicino dato che $x_j \sim \mathcal{N}_p(x_i, \sigma)$ e perciò centrata in x_i. Matematicamente abbiamo che:

$$p_{j|i} = \frac{\exp\{-\frac{1}{2}\frac{\|x_j - x_i\|^2}{\sigma_i^2}\}}{\sum_{k \neq i} \exp\{-\frac{1}{2}\frac{\|x_j - x_i\|^2}{\sigma_i^2}\}},$$

dove σ_i è un valore fissato.
Si definisce poi una posizione per le coppia di punti y_i e y_j nello spazio ridotto $\mathbb{R}^d$, che sono la controparte sottodimensionata di x_i e x_j, e una distribuzione per ogni coppie di punti in questo spazio $q_{j|i}$, che è l'equivalente di $p_{j|i}$ in dimensione d. Fissando la varianza σ_i di

$p_{j|i}$ uguale a $\frac{1}{\sqrt{2}}$, si ottiene:

$$q_{j|i} = \frac{\exp\{\|x_j - x_i\|^2\}}{\sum_{k \neq i} \exp\{-\frac{1}{2}\|x_j - x_i\|^2\}}.$$

I valori di y_i non sono noti, perciò dobbiamo sfruttare le informazioni che abbiamo per ottenerli.
Se i punti della mappa y_i e y_j riproducono la similarità tra i punti in dimensione originale x_i e x_j, allora questo varrà anche per le probabilità condizionate. Vogliamo quindi che le distribuzioni $p_{j|i}$ e $q_{j|i}$ siano il più possibile simili tra loro in modo che le $X \in \mathbb{R}^p$ siano rappresentate al meglio in dimensione ridotta.
La funzione di costo utilizzata per misurare la somiglianza tra le due distribuzioni è la **divergenza di Kullback-Leibler**, o $KL(p, q)$, che è la misura dell'informazione persa quando q è usata per approssimare p; è definita come:

$$KL(p, q) = \int p \log\left(\frac{p}{q}\right),$$

che nel nostro caso diventa:

$$KL(p_{j|i}, q_{j|i}) = \sum_{i=1}^{n} \sum_{j \neq i} p_{j|i} \log\left(\frac{p_{j|i}}{q_{j|i}}\right).$$

Data la funzione, SNE cerca i punti $Y \in \mathbb{R}^d$ le cui distribuizioni di probabilità condizionata $p_{j|i}$ minimizzino $KL(p_{j|i}, q_{j|i})$.

2.2.2 Asimmetria e problema del sovraffolamento

Nonostante SNE sia un buon punto di partenza, possiede alcuni difetti che gli implementatori di t-SNE hanno provato a risolvere.
Poiché la divergenza di Kullback-Leibler non è simmetrica, tipi diversi di errore nel rappresentare distanze coppia a coppia dei punti della mappa non vengono pesati in maniera equivalente.
In particolare, viene valutato maggiormente l'errore di distanziare due punti nella mappa che però sono vicini in dimensione originale piuttosto che commettere l'errore opposto.
In sintesi, la funzione di costo utilizzata da SNE massimizza la conservazione della struttura locale dei dati nella mappa e la sua minimizzazione è attuata utilizzando il metodo

della discesa del gradiente.
Poiché utilizziamo la distribuzione Normale per definire $p_{j|i}$ e $q_{j|i}$, e poiché la Normale è concentrata per valori attorno alla media, si verifica ciò che è chiamato **problema del sovraffolamento**.
Il sovraffolamento avviene quando ottengo una mappa in cui molte osservazioni sono sovrapposte, e questo rende l'output finale poco leggibile ed interpretabile.

2.2.3 Novità del t-SNE

Per risolvere i due problemi discussi sopra, t-SNE utilizza una versione differente della funzione di costo rispetto a SNE; le differenze principali sono due:
- si sostituisce la Normale con la distribuzione t di Student con 1 grado di libertà per definire $q_{j|i}$ in dimensione ridotta.
Utilizziamo la $q_{ij} \sim t_1$ poiché, avendo le code più pesanti della Normale, favorisce la repulsione di punti distanti tra loro attenuando così il sovraffolamento;
- viene utilizzata una versione simmetrica della funzione di costo, che sostituisce le probabilità condizionate $p_{j|i}$ e $q_{j|i}$ con delle probabilità congiunte

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$$

e

$$q_{ij} = \frac{(1 + \|y_i - y_j\|)^{-1}}{\sum_{k \neq i}(1 + \|y_i - y_j\|)^{-1}},$$

$p_{ij} \sim \mathcal{N}_p, q_{ij} \sim t_1$.
Minimizziamo dunque la seguente quantità:

$$\sum_{i=1}^{n} \sum_{j \neq i} p_{ij} \log \left(\frac{p_{ij}}{q_{ij}} \right).$$

Possiamo dire che questa funzione di costo, a differenza di quella utilizzata in SNE, è simmetrica poiché vale $p_{ij} = p_{ji}$ e $q_{ij} = q_{ji}, \forall i, j$.

Il problema di minimizzare la quantità $\sum_{i=1}^{n} \sum_{j \neq i} p_{ij} \log \left(\frac{p_{ij}}{q_{ij}} \right)$ ha però una complessità

temporale elevata ($O(n^2)$); per ridurla usiamo l'approssimazione del gradiente di Barnes-Hut, definita con una procedura iterativa basata su alberi [2]. Il gradiente della Kullback-Leibler viene scomposto in due parti, ottenendo:

$$\frac{\partial KL}{\partial y_i} = 4(F_{attr} + F_{rep}) = 4\left(\sum_{j\neq i} p_{ij}q_{ij}Z(y_i - y_j) - \sum_{j\neq i} q_{ij}^2 Z(y_i - y_j)\right),$$

con $Z = \sum_{i\neq j}(1 + \|y_i - y_j\|^2)^{-1}$.
Minimizzare questa approssimazione riduce il costo a $O(n\log(n))$.

Capitolo 3

Picasso Embedding e MCML

3.1 Perché t-SNE e UMAP potrebbero non fornire buoni risultati

Introduciamo due nuovi concetti: apprendimento supervisionato e apprendimento non supervisionato.
Per apprendimento supervisionato si intendono tutte quelle procedure statistiche per cui è nota la variabile risposta, sia essa un'etichetta nel caso di risposta qualitativa, sia essa un valore numerico in caso di risposta quantitativa; l'apprendimento non supervisionato invece comprende le tecniche statistiche che analizzano dati per cui non è nota la variabile risposta.
UMAP e t-SNE sono metodi non supervisionati, e la riduzione di dimensionalità viene applicata alla 'cieca'.
Prendendo i dati genomici, ad esempio, le informazioni spaziali e di ambiente non vengono considerate diversamente rispetto ai veri e propri dati.
Questo porta inevitabilmente ad avere delle riduzioni di dimensionalità a volte falsate poiché sono stati erroneamente considerati parte dell'informazione dati che in realtà non andavano considerati.
Sebbene molte ricerche considerino gli output grafici ottenuti con queste tecniche informativi, ossia che mantengono le proprietà principali dei dataset di partenza, c'è in realtà poca teoria a supporto di questa affermazione.

Per esempio, mentre i metodi t-SNE e UMAP sono intesi per rappresentare fedelmente la struttura locale e globale di dati ad alta dimensionalità in due o tre dimensioni, c'è evidenza che questi falliscano a tale riguardo, e i teoremi che garantiscono la qualità dei risultati si basano su numerose assunzioni che difficilmente vengono rispettate nella pratica.
A sostenere questa tesi c'è il **lemma di Johnson-Lindenstrauss** [8], che fornisce una condizione sufficiente per la presenza di distorsione in mappe ottenute tramite t-SNE e UMAP. Il lemma infatti sostiene che la conservazione delle distanze coppia a coppia di m punti con una precisione di fattore $1 \pm \epsilon$ può essere ottenuta con almeno $\log(m)/\epsilon^2$ dimensioni.
Questo ci dimostra quindi che la riduzione di dimensione di dati ad alta dimensionalità può avvenire, ma ci dice anche che il numero di dimensioni necessarie per ottenere dati di qualità è molto maggiore di due o tre.
Ad esempio, dato un insieme di dati avente $m = 10000$ punti, tenendo un termine d'errore pari al 20%, saranno necessarie almeno 1842 dimensioni per conservare le distanze coppia a coppia dei punti.
Al fine di comprendere meglio la distorsione dei dati ridotti a due dimensioni dalla dimensione originale, ci concentriamo inizialmente su un caso difficile: la mappatura di punti equidistanti.
È impossibile riprodurre più di $n + 1$ punti equidistanti in $\mathbb{R}^k$ con $k \leqslant n$, ma anche rilassando il vincolo di equidistanza e considerando la quasi-equidistanza, dove per tre punti qualsiasi in $\mathbb{R}^n$ la distanza tra due di questi è unitaria, si possono disporre solamente sette punti in $\mathbb{R}^2$ o dieci in $\mathbb{R}^3$.
Anche cercare di mantenere la quasi-equidistanza in dimensione ridotta è dunque inverosimile: la proporzione tra la massima distanza D e la minima d tra n punti in due dimensioni cresce con ritmo $O(\sqrt{n})$; inoltre la distorsione di punti equidistanti in dimensione ridotta può essere particolarmente acuta con la PCA, che è spesso utilizzata per pre-condizionare i dati.
Questa distorsione è tanto pronunciata, che cercare di mantenere i punti equidistanti in una mappa con la PCA ha la stessa efficacia di applicare una proiezione casuale; il risultato è che i punti proeittati non mostrano una struttura sensata o informativa dei dati iniziali.

3.2 Studio su dati genomici

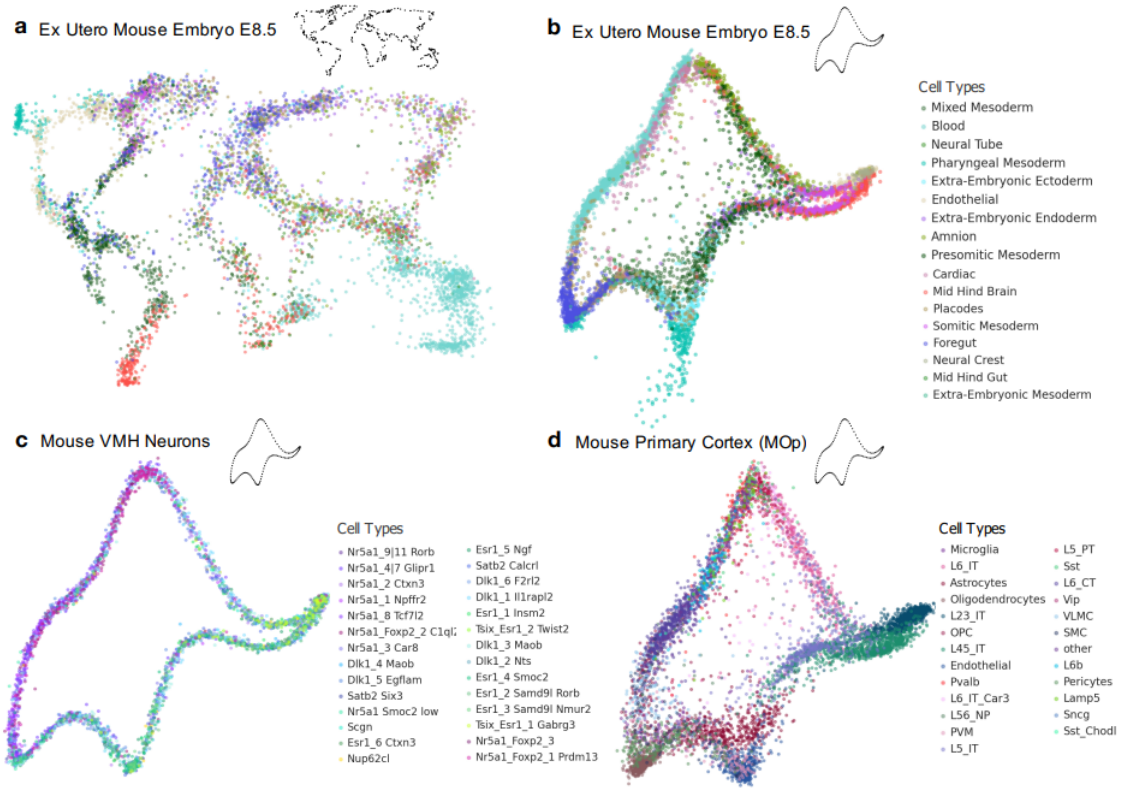
Uno studio condotto da Tara Chari, Joeyta Banerjee e Lior Pachter al California Institute of Technology [4], ha evidenziato come algoritmi non supervisionati siano particolarmente inefficaci su dati genomici, evidenziando la presenza di ostacoli nella riduzione della dimensionalità ed evidenti distorsioni. Per risolvere questi problemi, vengono proposti dei metodi supervisionati.
Lo studio ha valutato se fossero presenti punti equidistanti in dati biologici, e se i metodi di riduzione della dimensionalità come t-SNE e UMAP, applicati a dati precondizionati con la PCA, possano indurre distorsione.
Sono stati presi in considerazione dati relativi a cellule embrionali appartenenti a topi coltivate in due ambienti diversi: all'interno dell'utero dei roditori e in un ambiente artificiale (in-utero e ex-utero) [9].
Sono stati selezionati questi dati perché la struttura della loro riduzione in due dimensioni è stata utilizzata per validare dei risultati inerenti all'embriogenesi artificiale.
Sono risultate visibili distorsioni di punti equidistanti in 1,511,502 gruppi distinti di cellule definiti 'vicini ed equidistanti' e 1,020,120 gruppi distinti di cellule definiti 'lontani'.
Le distorsioni sono dovute al fatto che nella mappa creata con t-SNE e UMAP i gruppi 'vicini ed equidistanti' e quelli 'lontani' appaiono raggruppati similmente, nonostante le loro differenti proprietà in dimensione originale.
È stato misurato il cambiamento di varianza all'interno di questi gruppi e la distorsione delle distanze, ossia la proporzione tra la massima e la minima distanza tra coppie di cellule.
Per i gruppi 'vicini ed equidistanti', la varianza delle distanze coppia a coppia è aumentata nella mappa creata da UMAP, in media, da 135 a 1040 volte relativamente alla varianza dei dati originali. La distorsione delle distanze è aumentata da 25 a 95 volte. Anche se alta, la distorsione in una mappa di 15 dimensioni ottenuta con la PCA è risultata minore, con un aumento della varianza di 214 volte e una distorsione delle distanze di 4.5 volte.
Per i gruppi 'lontani' si è osservato un trend simile con varianza maggiorata da 321 a 1029 volte e la distorsione da 22 a 39.
Separatamente è stato analizzato un dataset contente informazioni relative ad un grande numero di cellule provenienti dall'ipotalamo di topi [10], dove sono stati trovati oltre 10

milioni di gruppi differenti di cellule equidistanti con fino a 14 cellule per gruppo.
La varianza dei punti della mappa rispetto ai dati originali è aumentata da 42 a 77 volte usando UMAP, e da 443 a 625 volte usando t-SNE. La distorsione è aumentata da 104 a 154 volte.
Guardando i dati da una prospettiva meno specifica e cioè osservando i tipi di cellula, la distorsione era comunque presente, da 3.1 a 4.3 volte nelle mappe create da UMAP e t-SNE.
Ciò ha dimostrato che questi algoritmi non riescono a mantenere nemmeno la struttura di larga scala dei tipi di cellula.
Per entrambi i dataset sono state identificate grandi distorsioni nelle mappe rispetto ai dati originali, evidenziando che le rappresentazioni grafiche non sono informative.
I problemi osservati relativi alle tecniche di riduzione della dimensionalità non supervisionate (come UMAP e t-SNE) evidenziano come queste siano inefficaci a produrre un output grafico fedele ai dati di partenza e di conseguenza la loro limitata potenzialità.

3.3 Picasso embedding

Per avere un'ulteriore prova che t-SNE e UMAP non creano risultati di qualità, è stata implementata una nuova tecnica di riduzione della dimensionalità, chiamata **Picasso embedding**.
La peculiarità di Picasso embedding è che, oltre a ridurre la dimensionalità dei dati di partenza, tramite alcuni vincoli, permette all'utente di scegliere una qualsiasi forma da far assumere ai punti della mappa di dimensione ridotta.
Ci si aspetterebbe che un tale algoritmo abbia delle prestazioni significativamente inferiori rispetto ai primi due; sorprendentemente si è però riscontrato che Picasso fornisce risultati che hanno un errore simile, se non inferiore, a t-SNE e UMAP, e tutto ciò con il vincolo di creare un pattern grafico prescelto dall'utente.
Picasso è stato applicato due volte sui dati relativi alle cellule embrionali in e ex utero, creando due pattern con i punti della mappa: l'elefante di Von Neumann nel primo caso e un planisfero nel secondo; è stato inoltre applicato a due dataset differenti creando come pattern grafico l'elefante di Von Neumann.

Tutte le rappresentazioni derivate dall'applicazione di Picasso hanno precisione simile ai risultati ottenuti applicando t-SNE e UMAP agli stessi insiemi di dati.



I grafici ***a*** *e* ***b*** *sono relativi al dataset delle cellule embrionali dei topi* [9]*, in* ***a*** *i dati assumono la forma di un planisfero mentre in* ***b*** *di un elefante.*
I grafici ***c*** *e* ***d*** *rappresentano entrambi un elefante e sono stati applicati a due dataset differenti* [10][11].

Poter rappresentare un dataset con una mappa dove i punti formano un elefante o un planisfero e poter rappresentare due diversi dataset dove la mappa ha la forma di elefante per entrambi, tutto ciò con precisione comparabile tra le varie applicazioni (e simile a quella di t-SNE e UMAP), è la dimostrazione del fatto che visualizzazioni in due dimensioni ottenute con metodi come Picasso, t-SNE e UMAP non sono in alcun senso canoniche e spesso forniscono risultati non buoni in termini assoluti.
Dati i limiti della riduzione della dimensionalità e gli errori che si commettono dando valore ai risultati ottenuti utilizzando tecniche non supervisionate, è chiaro che per ottenere

risultati significativi sia necessario cambiare approccio.

3.4 MCML

Molte pubblicazioni hanno dimostrato i vantaggi offerti da tecniche di riduzione supervisionate nel costruire spazi di dimensione ridotta interpretabili. Per i dati genomici è stata proposta una tecnica che consiste nell'avere i dati disposti in diverse classi, dove ogni classe (es: tipo di cellula, condizioni sperimentali) contiene un'etichetta per ogni cellula; la tecnica è chiamata **MCML** (multi-classe, multi-etichetta) e cerca di ottenere risultati migliori rispetto a quelli ottenuti con metodi non supervisionati.
MCML utilizza una funzione di costo basata sulle etichette servendosi dell'algoritmo NCA (Neighborhood Component Analysis), che ottimizza la probabilità che cellule aventi la stessa etichetta siano tradotte come punti vicini nella mappa senza avere overfitting.
Combina inoltre l'errore di ricostruzione dei dati in dimensione ridotta con la funzione di costo basata sulle etichette ottimizzando così la struttura dei punti vicini della mappa e mantenendo allo stesso tempo la struttura dei dati originali.
Questo algortimo è stato applicato ai dati relativi alle cellule embrionali dei topi [9] ed è stata ottenuta una mappa di 15 dimensioni che si è rivelata migliore rispetto ai metodi non supervisionati nel mantenere le caratteristiche locali e globali dei dati originali.
MCML è stato inoltre utilizzato per fare previsione su cellule a cui non era stata assegnata una etichetta ed è stato confrontato (sempre sui dati dei topi) con SCANVI e LDVAE, due algoritmi per la previsione di etichette su cellule. Si è osservato un miglioramento della precisione nel classificare le cellule del 2.0% rispetto a SCANVI e del 12.9% rispetto a LDVAE. La precisione massima ottenuta con MCML non è comunque elevata (0.56).
Questi risultati mostrano comunque che la tecnica proposta è buona per la previsione.
Un altro lato positivo dell'algoritmo MCML, differentemente alla PCA, ad esempio, è l'abilità di ricostruire la struttura di correlazione tra dati rari e ortogonali (come espressioni genetiche non condivise da diversi tipi di cellule). Le componenti principali individuano solo le direzioni di massima varianza sopprimendo dunque le espressioni rare ed ortogonali.
È stato inoltre definito un metodo che si concentra nel mantenere le informazioni di una particolare metrica di interesse (ad esempio la varianza intra-etichetta) in dimensione ri-

dotta. Ciò è possibile poiché la sua funzione di costo tiene in considerazione la distorsione risultante dal processo di ridurre la dimensionalità e, per questo, il metodo è chiamato **bMCML** (biased-MCML).
bMCML si è rivelato utile nel mantenere la struttura di una particolare metrica di interesse anche se ciò avviene a discapito delle altre metriche; questa proprietà non è presente in nessuno dei metodi non supervisionati.

3.4.1 Conclusione

Fino ad ora si è creduto che le tecniche di riduzione della dimensionalità non supervisionate utilizzate su dati genomici preservino le relazioni locali e globali tra le cellule, ma le analisi effettuate dimostrano che queste applicazioni possono causare distorsioni significative.
Metodi supervisionati come MCML e bMCML sono un'innovazione nel campo della riduzione della dimensionalità.

Capitolo 4

Applicazioni e illustrazioni empiriche

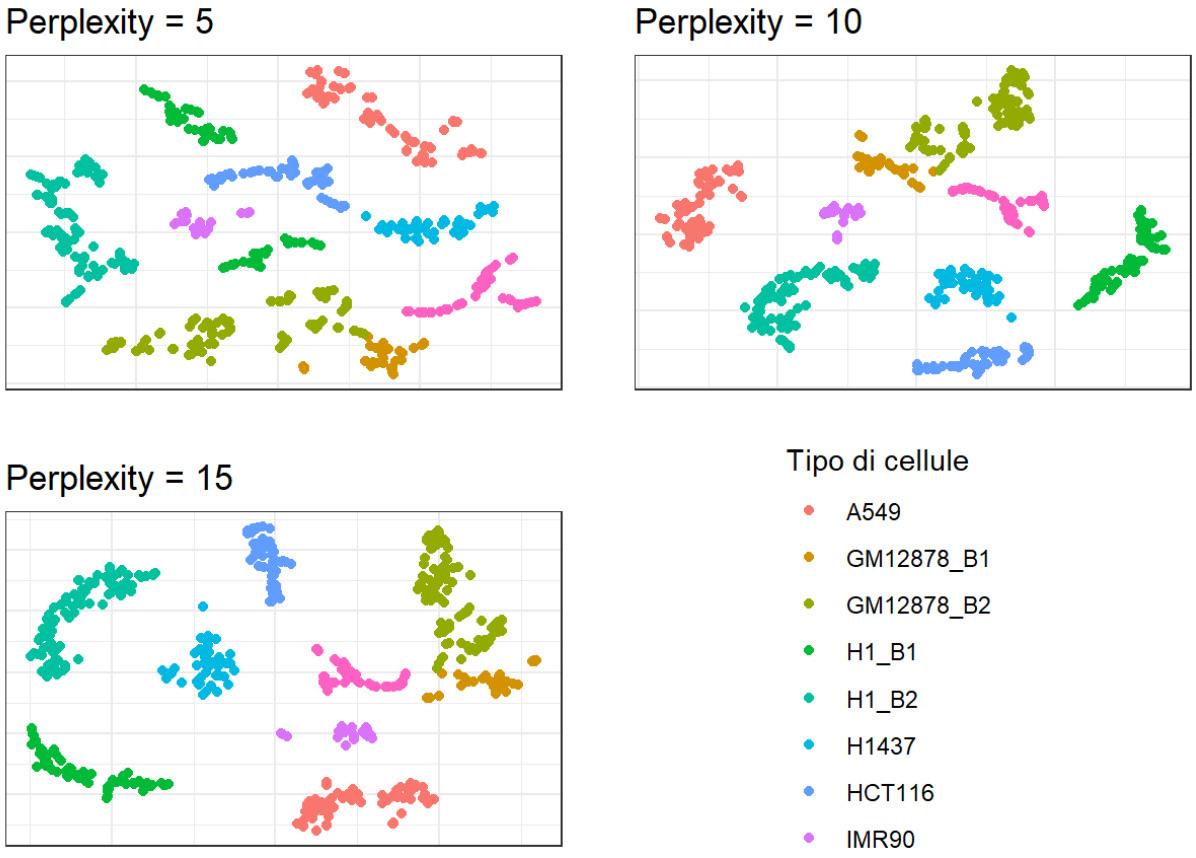
4.1 Applicazione su dati

Applichiamo ora t-SNE, UMAP e Picasso embedding ad un dataset proveniente da uno studio riguardante cellule tumorali di tumori del colon-retto [6].
L'insieme di dati è composto da 531 unità provenienti da 9 diverse linee cellulari (sono presenti nove gruppi) e su ogni unità sono state rilevate 14766 variabili. Vediamo in che modo cambia l'output grafico modificando i parametri di questi algoritmi, mantenendo sempre la dimensione della mappa pari a 2.

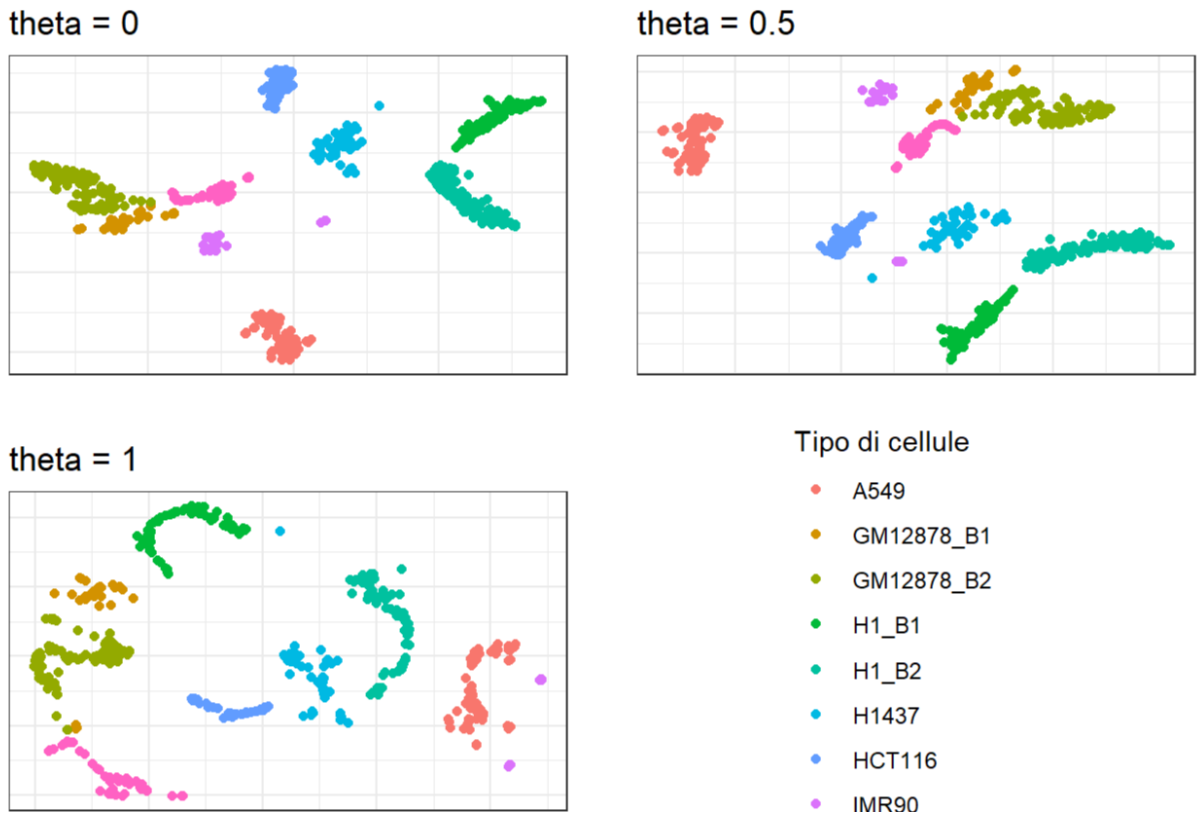
4.2 t-SNE

La funzione Rtsne (libreria Rtsne) implementa t-SNE in R; i parametri che andiamo a modificare sono:
- **perplexity**, per cui vale $3 \cdot perplexity = n$, dove n è il numero di nearest neighbors che vengono considerati nella creazione delle distribuzioni sulle coppie di unità;
- **theta**, che è il compromesso tra velocità e accuratezza dell'algorimo; con il valore zero si ottiene il t-SNE esatto e all'aumentare di theta l'accuratezza diminuisce (il valore di default è 0.5).

I seguenti grafici rappresentano le mappe fornite dalla funzione Rtsne per tre diversi valori del parametro perplexity e del parametro theta.



Si nota che nel primo grafico, con *perplexity* $= 5$, i vari gruppi di cellule sono formati, ma comunque leggermente dispersi, mentre per valori maggiori di perplexity i gruppi sono rappresentati in maniera più compatta.



L'output grafico varia leggermente in queste tre mappe ma ciò che è di interesse è il tempo di esecuzione dell'algoritmo perché theta è un parametro che regola questo. Utilizzando la funzione system.time() di R ottengo che, per $theta = 0$, il tempo di esecuzione è 15.36 secondi, per $theta = 0.5$ è 13.52 secondi e per $theta = 1$ risulta 13.21 secondi. Questo risultato è in accordo con ciò che ci si aspettava (valori grandi di theta $\rightarrow$ minor tempo di esecuzione).

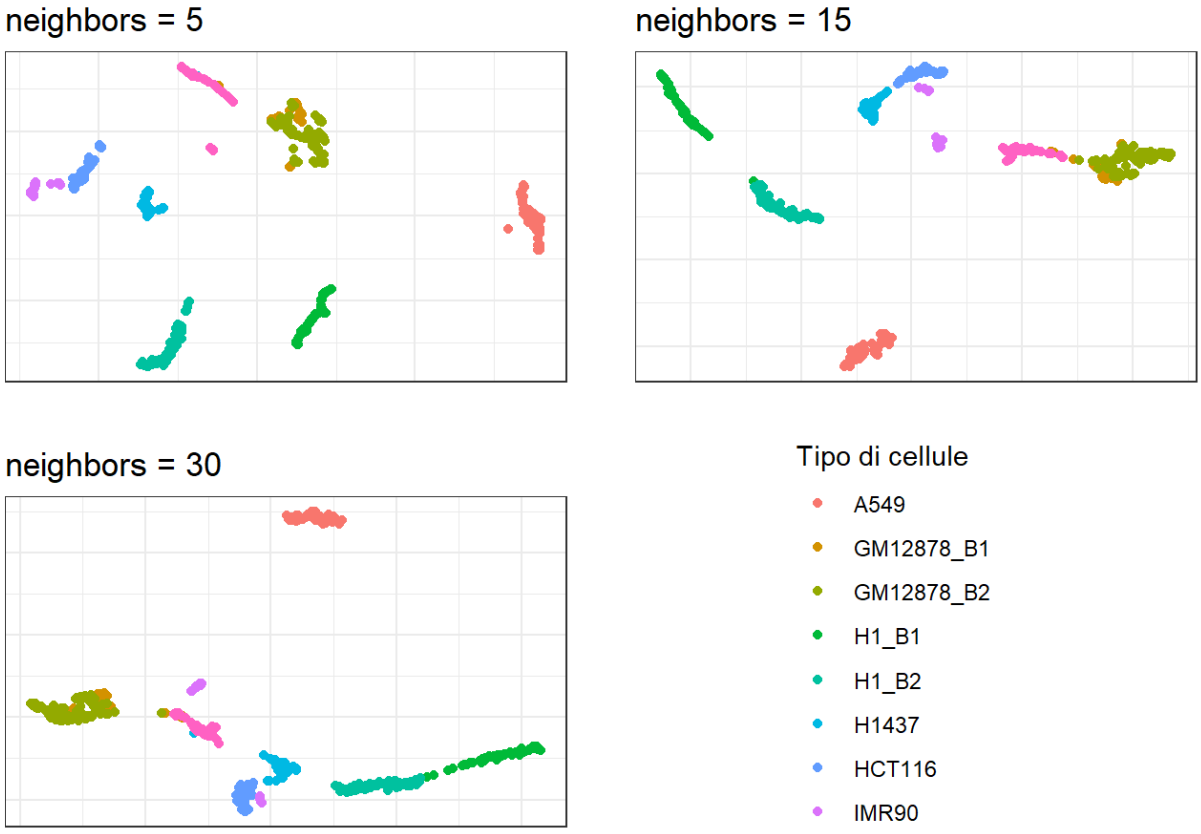
4.3 UMAP

La funzione umap (libreria umap) implementa UMAP in R; i parametri che andiamo a modificare sono:

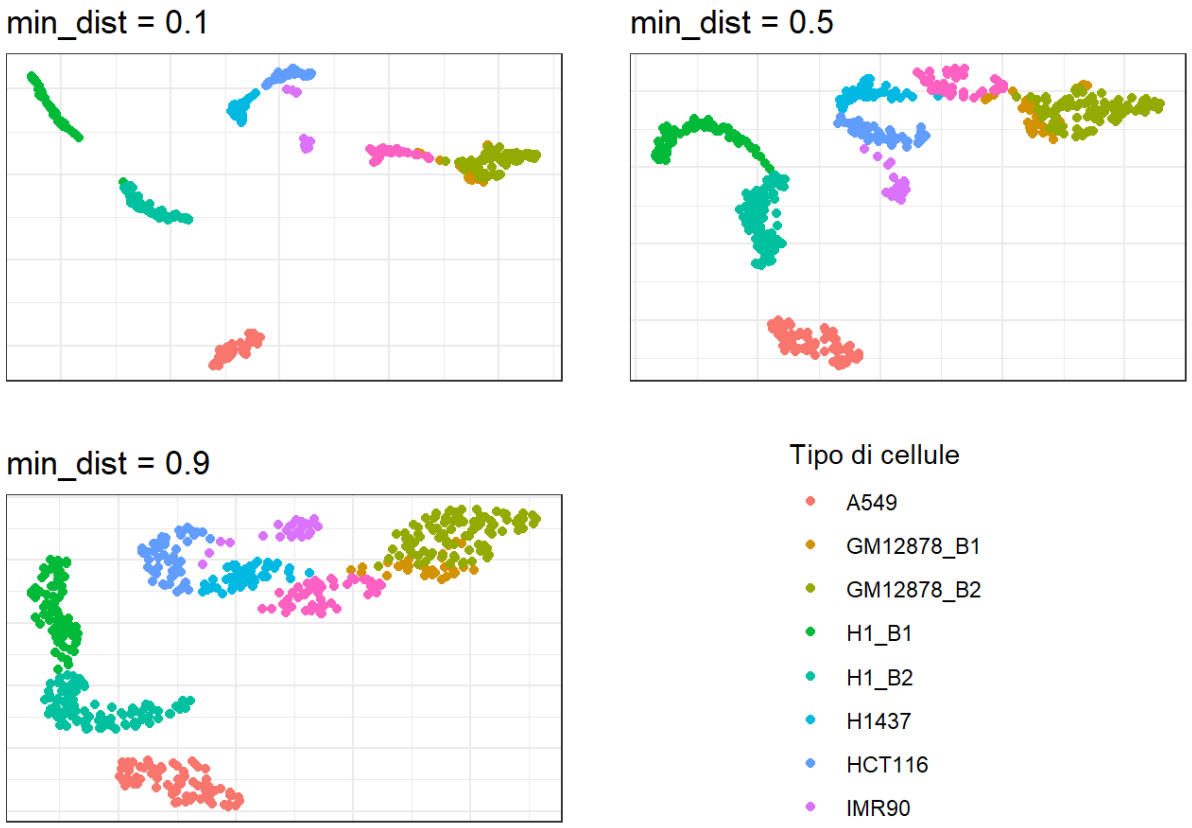
- **n**, il numero di neighbors da considerare quando avviene l'approssimazione della metrica locale;
- **min_dist**, che rappresenta la separazione desiderata tra punti vicini nella mappa.

Il parametro n rappresenta il compromesso tra le caratteristiche di piccola e grande sca-

la della varietà. Piccoli valori di n assicurano che i dettagli di piccola scala della varietà vengano fedelmente riprodotti nella mappa (a discapito di quelli di grande scala), mentre valori più grandi catturano i dettagli di grande scala della varietà a discapito però della struttura dettagliata.
Per piccoli valori di n la varietà tende ad essere separata in tante piccole componenti connesse nella mappa.
In contrasto min_dist è un parametro che controlla la costruzione dell'insieme sfocato; può essere visto come un parametro estetico.
Valori piccoli di min_dist creano regioni della mappa molto dense di punti rendendo l'output grafico non sempre di facile interpretazione con il vantaggio però di rappresentare la struttura della varietà in maniera più fedele.
Valori maggiori forzano i punti della mappa ad espandersi, aiutando così l'interpretazione visuale evitando l'overplotting.
I prossimi grafici rappresentano l'applicazione di umap per diversi valori di n e min_dist.



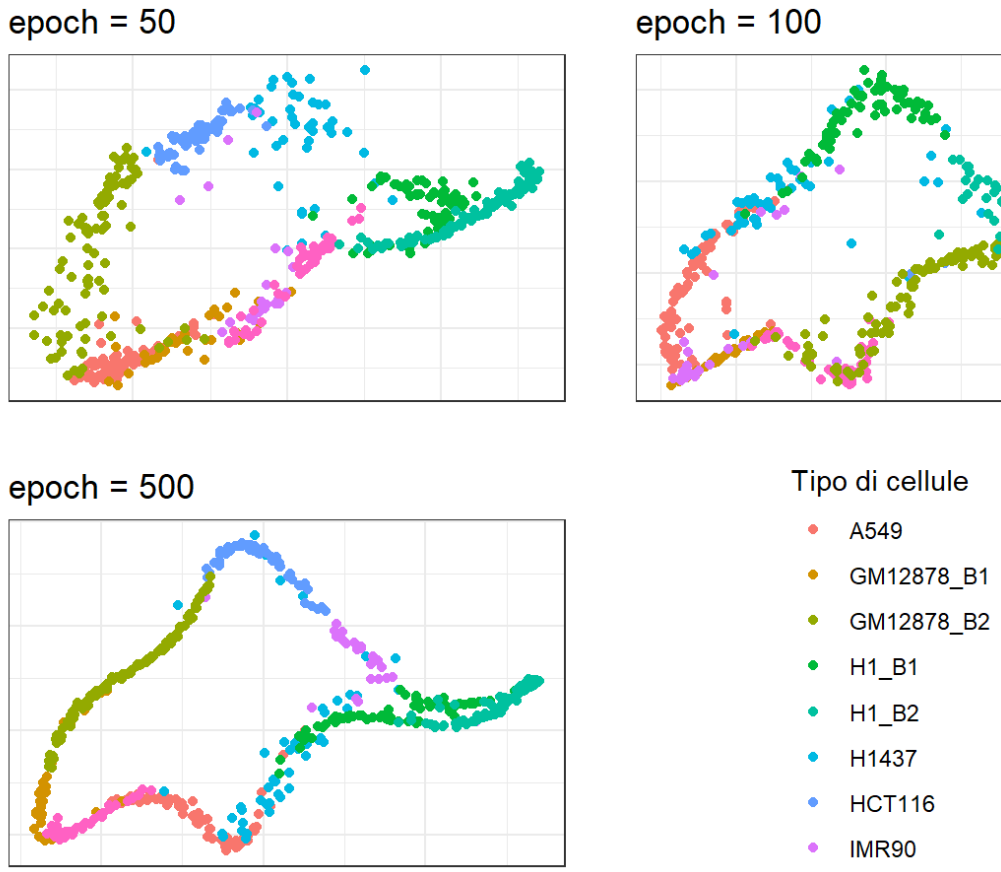
Le differenze tra valori di n piccoli (5) e grandi (30) non sono molto marcate.



Come ci si aspettava, valori piccoli di min_dist hanno creato zone della mappa dense di punti, mentre per valori maggiori i punti tendono ad espandersi.

4.4 Picasso embedding

Picasso embedding è stato implementato solamente in Python; la forma scelta da far assumere ai punti del grafico è un elefante e il parametro che è stato modificato è epoch.



Aumentando il valore di epoch, i punti della mappa assumono una forma sempre più simile a quella di un elefante.

Bibliografia

[1] van der Maaten, Hinton, (2008). *Visualizing High-Dimensional Data Using t-SNE*, Journal of Machine Learning Research

[2] van der Maaten (2014), *Accelerating t-SNE using Tree-Based Algorithms*, Journal of Machine Learning Research.

[3] C.O.S. Sorzano, J. Vargas, A. Pascual-Montano (2014), *A survey of dimensionality reduction techniques*, Natl. Centre for Biotechnology (CSIC), Madrid, Spain.

[4] Tara Chari, Joeyta Banerjee, Lior Pachter (2021), *The Specious Art of Single-Cell Genomics*, Caltech, CA, USA.

[5] Leland McInnes, John Healy, James Melville (2018), *UMAP: Uniform Manifold Approximation and Pojection for Dimension Reduction*, Tutte Institute for Mathematics and Computing.

[6] Huipeng Li, Elise T Courtois, Debarka Sengupta, Yuliana Tan, Kok Hao Chen, Jolene Jie Lin Goh, Say Li Kong, Clarinda Chua, Lim Kiat Hon, Wah Siew Tan, et al. *Reference Component Analysis of Single-Cell Transcriptomes Elucidates Cellular Heterogeneity in Human Colorectal Tumors.* Nature Genetics, 49:708, 2017.

[7] Richard Johnson, Dean Wichern, *Applied Multivariate Statistical Analysis: Pearson New International Edition*, 2014

[8] Johnson, W. B. and Lindenstrauss, J. Extensions of Lipschitz mappings into a Hilbert space 26. *Contemp. Math.* 26 (1984).

[9] Aguilera-Castrejon, A. *et al.* Ex utero mouse embryogenesis from pre-gastrulation to late organogenesis. *Nature* 593, 119–124 (May 2021).

[10] Kim, D.-W. *et al.* Multimodal Analysis of Cell Types in a Hypothalamic Node Controlling Social Behavior. *Cell* 179, 713–728.e17 (Oct. 2019).

[11] Zhang, M. et al. Molecular, *spatial and projection diversity of neurons in primary motor cortex revealed by in situ single-cell transcriptomics*, June 2020.