Diario scientifico

04/09/25

E’ il momento di compilare una lista con i risultati migliori trovati questi giorni, per ogni dataset.

csv5

[OPTICS #0] Parametri: {'min_samples': 5, 'xi': 0.035, 'min_cluster_size': 0.1}

Distribuzione cluster per OPTICS #0:
Noise (outliers): 51 elementi
Cluster 0: 97 elementi
Cluster 1: 21 elementi
Totale elementi: 169

OPTICS #0: 2 cluster, DBCV score: 0.4507

[MeanShift #8] Parametri: {'bandwidth': 3.35}

Distribuzione cluster per MeanShift #8:
Cluster 0: 115 elementi
Cluster 1: 31 elementi
Cluster 2: 12 elementi
Cluster 3: 2 elementi
Cluster 4: 2 elementi
Cluster 5: 1 elementi
Cluster 6: 1 elementi
Cluster 7: 3 elementi
Cluster 8: 2 elementi
Totale elementi: 169

MeanShift #8: 9 cluster, DBCV score: 0.2281

[BIRCH #8] Parametri: {'threshold': 3.1, 'branching_factor': 100, 'n_clusters': 2}

Distribuzione cluster per BIRCH #8:
Cluster 0: 140 elementi
Cluster 1: 29 elementi
Totale elementi: 169

BIRCH #8: 2 cluster, DBCV score: 0.1322

csv6

[OPTICS #16] Parametri: {'min_samples': 7, 'xi': 0.03, 'min_cluster_size': 0.2}

Distribuzione cluster per OPTICS #16:
Noise (outliers): 20 elementi
Cluster 0: 32 elementi
Cluster 1: 15 elementi
Totale elementi: 67

OPTICS #16: 2 cluster, DBCV score: 0.4009

[OPTICS #17] Parametri: {'min_samples': 7, 'xi': 0.04, 'min_cluster_size': 0.25}

Distribuzione cluster per OPTICS #17:
Noise (outliers): 21 elementi
Cluster 0: 46 elementi
Totale elementi: 67

OPTICS #17: 1 cluster, DBCV score: 0.6866

[MeanShift #0] Parametri: {'bandwidth': 4.2}

Distribuzione cluster per MeanShift #0:
Cluster 0: 47 elementi
Cluster 1: 15 elementi
Cluster 2: 1 elementi
Cluster 3: 1 elementi
Cluster 4: 1 elementi
Cluster 5: 2 elementi
Totale elementi: 67

MeanShift #0: 6 cluster, DBCV score: 0.2339

[BIRCH #1] Parametri: {'threshold': 1.85, 'branching_factor': 20, 'n_clusters': 2}

Distribuzione cluster per BIRCH #1:
Cluster 0: 50 elementi
Cluster 1: 17 elementi
Totale elementi: 67

BIRCH #1: 2 cluster, DBCV score: 0.2461

csv10

[OPTICS #29] Parametri: {'min_samples': 4, 'xi': 0.04, 'min_cluster_size': 0.3}

Distribuzione cluster per OPTICS #29:
Noise (outliers): 20 elementi
Cluster 0: 396 elementi
Totale elementi: 416

OPTICS #29: 1 cluster, DBCV score: 0.9519

[MeanShift #11] Parametri: {'bandwidth': 3.55}

Distribuzione cluster per MeanShift #11:
Cluster 0: 398 elementi
Cluster 1: 12 elementi
Cluster 2: 1 elementi
Cluster 3: 5 elementi
Totale elementi: 416

MeanShift #11: 4 cluster, DBCV score: -0.0533

[BIRCH #34] Parametri: {'threshold': 3.0, 'branching_factor': 250, 'n_clusters': 2}

Distribuzione cluster per BIRCH #34:
Cluster 0: 20 elementi
Cluster 1: 396 elementi
Totale elementi: 416

BIRCH #34: 2 cluster, DBCV score: -0.3020

csv23

[OPTICS #0] Parametri: {'min_samples': 4, 'xi': 0.04, 'min_cluster_size': 0.05}

Distribuzione cluster per OPTICS #0:
Noise (outliers): 8 elementi
Cluster 0: 417 elementi
Totale elementi: 425

OPTICS #0: 1 cluster, DBCV score: 0.9812

[MeanShift #3] Parametri: {'bandwidth': 4.36}

Distribuzione cluster per MeanShift #3:
Cluster 0: 412 elementi
Cluster 1: 5 elementi

Cluster 2: 2 elementi
Cluster 3: 1 elementi
Cluster 4: 5 elementi
Totale elementi: 425

MeanShift #3: 5 cluster, DBCV score: -0.2230

csv66

[OPTICS #38] Parametri: {'min_samples': 10, 'xi': 0.02, 'min_cluster_size': 0.05}

Distribuzione cluster per OPTICS #38:
Noise (outliers): 810 elementi
Cluster 0: 239 elementi
Cluster 1: 143 elementi
Cluster 2: 65 elementi
Totale elementi: 1257

OPTICS #38: 3 cluster, DBCV score: 0.2192

[OPTICS #464] Parametri: {'min_samples': 8, 'xi': 0.032, 'min_cluster_size': 0.166}

Distribuzione cluster per OPTICS #464:
Noise (outliers): 33 elementi
Cluster 0: 1224 elementi
Totale elementi: 1257

OPTICS #464: 1 cluster, DBCV score: 0.9737

[OPTICS #534] Parametri: {'min_samples': 30, 'xi': 0.002, 'min_cluster_size': 0.2}

Distribuzione cluster per OPTICS #534:
Noise (outliers): 903 elementi
Cluster 0: 354 elementi
Totale elementi: 1257

OPTICS #534: 1 cluster, DBCV score: 0.2816

03/09/25

Ho creato un file di parametri per ogni dataset in modo da affinarli per i risultati dei singoli gruppi di dati.

02/09/25

Ho cercato di verificare cosa succede se tolgo il rumore che avevo inserito per evitare l'errore nel calcolo dell'indice, questo ho potuto farlo solo con csv6, csv23 e csv66 dato che gli altri due presentavano questi presunti punti uguali.

Con tutti e tre i dataset i risultati mi sembrano identici, quindi confermato che il problema non è il rumore.

Adesso cerco di trovare dei nuovi parametri nei range migliore. Un pochino meglio i risultati, domani voglio cambiare una cosa: usare per ogni dataset un file diverso di parametri in modo da poterlo affinare per singolo dataset, non so perchè ancora non lo avevo fatto.

28/08/25

Vediamo un pò i risultati con parametri nuovi.

csv5 → con OPTICS le poche soluzioni non banali sono quelle con 1 cluster grande + una quota di outlier di circa 50 (#8, #10, #12, #14, #16, #18) con valori DBCV 0.5–0.7
Con MeanShift solo soluzioni non significative.
Con BIRCH anche, per qualche ragione trova sempre 3 cluster, credo si deva segnare il numero di cluster su none per averli liberi.

csv6 → con optics conviene provare con parametri tipo min_samples ~6–10, min_cluster_size ~0.1–0.3, xi medio (0.05–0.08).
con meanshift ci vuole un bandwidth almeno di 2.5/3.
con BIRCH il risultato migliore si ha con threshold=2.0, branching_factor=200 (distribuzione: 49 – 17 – 1 elementi e DBCV = 0.2418)

csv10 → con optics conviene restringere la ricerca dei parametri attorno a quelli di #2 e #5 (min_samples=4–6, xi tra 0.02–0.05, min_cluster_size 0.08–0.15).
con meanshift nulla di significativo e nemmeno con birch, si devono liberare il numero di cluster che resta 3.

csv23 → solo OPTICS #0 e #6 non sono banali.
con meanshift anche qua conviene usare bandwidth ≥ 3.
con birch niente di utile.

csv66 → OPTICS #2 con DBCV = 0.1439 ha utilizzato parametri specifici (min_samples=4, xi=0.03, min_cluster_size=0.1) da vedere in questo range.
con meanshift niente (range da guardare 3.5-5.5) e birch niente di utile.

27/08/25

Voglio segnarmi qualche risultato un filo più interessante trovato con i clustering fatti con i parametri modificati.

Questi sono dal dataset csv6:

OPTICS #1, #6, #7, #10, #13, #17, #24, #28 → DBCV ≈ 0.6866, con 1 cluster grande (46 elementi) + 21 noise.

OPTICS #2, #26 → DBCV ≈ 0.40, 2 cluster (32+15) + ~20 outlier.

OPTICS #14, #29 → DBCV ≈ 0.23, con 2 cluster (47 e 16) e pochi outlier (4).

Meanshift in generale molto male, frammentato e con DBCV bassi.

BIRCH: per molte combinazioni di parametri (threshold piccolo o medio), si ottiene sempre circa lo stesso partizionamento: un cluster da 50 e uno da 17 (o simili 49–18). Qui il DBCV è sempre ≈0.24, segno che i cluster non sono molto ben definiti. Per threshold alti (3.5–5.0), invece, BIRCH collassa tutto in un cluster grande e un outlier singolo (66–1), e in questo caso il DBCV è altissimo (≈0.985), ma chiaramente poco significativo.

Dataset csv10:

OPTICS male, o trova un singolo cluster o trova tantissimi outlier e DBCV basso. Da provare nuovi iperparametri.

MeanShift stessi problemi più o meno, o collassa in un singolo cluster oppure produce un cluster molto grande e tanti micro cluster.

BIRCH pure non sta dando risultati positivi dopo che ho forzato sempre di unire quello che trova naturalmente in due cluster. Anche questa cosa è da cambiare.

Dataset csv23:

OPTICS trova praticamente sempre un solo cluster, al massimo qualche outlier, questo dataset sembra sia davvero un unico gruppo coeso, vediamo gli altri modelli cosa dicono.

MeanShift va a forzare una divisione in tanti cluster che però non funziona, pochissimi elementi e DBCV negativi.

BIRCH di nuovo sta forzando due cluster, che comunque non funzionano, con threshold > 4 ne trova comunque solo 1, quando è minore ne trova due con bilanciamento variabile ma comunque DBCV pessimo.

Dataset csv66:

OPTICS anche in questo caso sembrerebbe indicare un unico cluster (risultato #2) ma i tanti risultati con tantissimi outlier fanno anche pensare che gli

iperparamentri non vanno bene per questa distribuzione.

MeanShift malissimo con DBCV sempre fortemente negativi.

Birch anche non sta rispondendo come vorrei, o trova due cluster con DBCV molto negativo, oppure uno enorme e un altro con qualche outlier, DBCV ovviamente molto alto ma che significa poco o nulla, devo provare a concentrarmi su threshold specifici e liberare il numero di cluster dato che mi sembrava meglio prima.

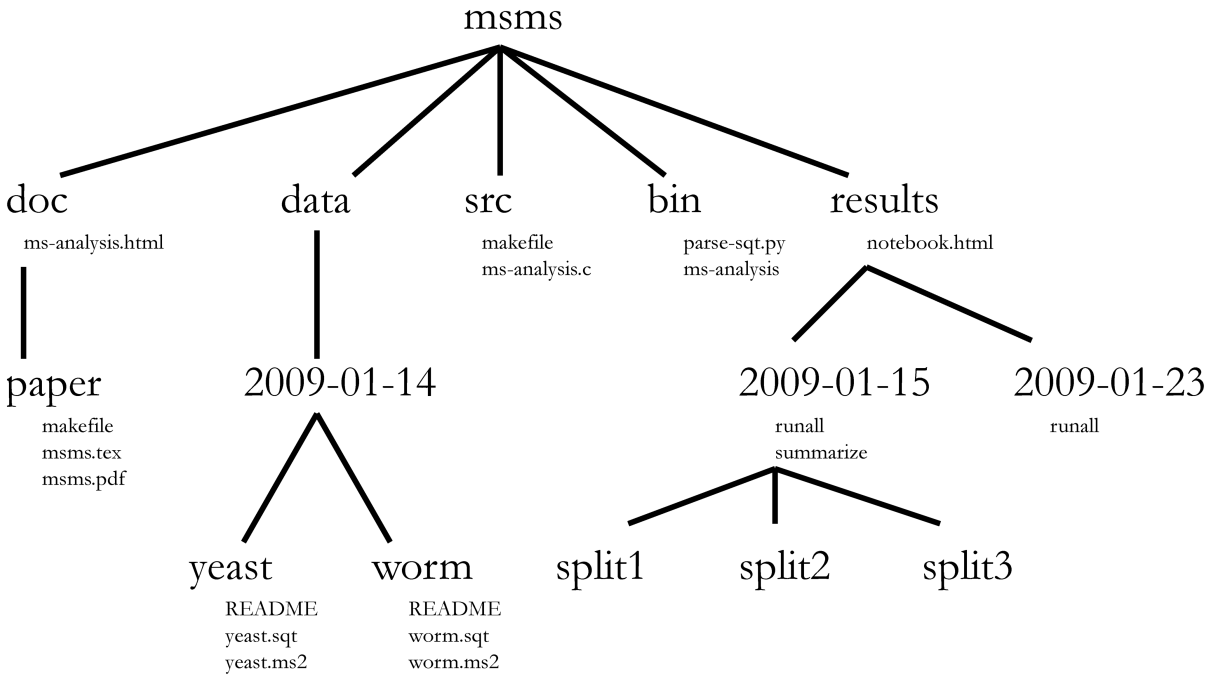
18/07/25

Dato che voglio cambiare un pò come lavorano gli algoritmi che ho scritto in modo da provare autonomamente tante combinazioni di parametri. Sto creando un file py tipo dizionario che contiene delle combinazioni più o meno sensate che poi posso importare nei vari algoritmi di applicazione.

16/07/25

Quello che voglio fare questi giorni è sistemare secondo le linee guida presentate in questo articolo → https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1000424 la cartella del progetto su github.

La struttura generale sarebbe la seguente:



18/06/25

5. Neuroblastomas in Eastern China: a retrospective series study of 275 cases in a regional center (10_7717_peerj_5665_dataYM2018_neuroblastoma)

Il neuroblastoma è il tumore solido extracranico maligno più comune nei bambini, l'articolo fa un analisi di un pò di casi. I dati che prende sono:

- **age**: età del paziente (0 categoria <18 mesi, 1 cat 18-60 mesi, 2 maggiore di 60 mesi)
- **sex**: Indica il sesso del paziente (0 femmina 1 maschio)
- **site**: indica la sede primaria del tumore (0 ghiandola surrenale, 1 mediastino, 2 altri posti).
- **stage**: stadio della malattia secondo l'international neuroblastoma staging system (0 stadio 1, 1 stadio 2,..., fino a 4 che è stadio 4s)
- **risk**: gruppo di rischio del paziente secondo i criteri di stratificazione del rischio del children's oncology group (0 rischio basso, 1 rischio intermedio, 2 rischio alto).
- **time_months**: periodo di follow-up in mesi dall'iniziale diagnosi.
- **autologous_stem_cell_transplantation**: Indica se il paziente ha ricevuto trapianto autologo (?) di cellule staminali (0 no 1 si).
- **radiation**: Indica se il paziente ha ricevuto radioterapia (0 no 1 si)
- **degree_of_differentiation**: grado di differenziazione patologica (?) del tumore (0 indifferenziato, 1 poco diff, 2 differenziante)
- **UH_or_FH**: classificazione istologica del tumore, se istologia favorevole (FH) o sfavorevole (UH) → non so cosa significhi
- **MYCN_status**: Indica lo stato di amplificazione del gene MYCN (?) → 0 amplificato, 1 non amplificato.
- **surgical_methods**: metodo chirurgico utilizzato (subtotal resection con 0, gross total resection con 1)
- **outcome**: esito del paziente in termini di sopravvivenza (0 deceduto all'ultimo follow up, 1 vivo all'ultimo follow up)

Lo studio retrospettivo su 275 casi di neuroblastoma nella cina orientale ha rivelato che l'amplificazione del gene MYCN è un fattore prognostico avverso indipendente nei pazienti cinesi con NB di stadio 3 e 4. la gross total resection del

tumore è associata a un miglioramento della sopravvivenza globale rispetto alla subtotal resection in questi pazienti.

La sopravvivenza dei pazienti con NB di stadio 4 e amplificazione MYCN è estremamente scarsa, suggerendo la necessità di una stratificazione del rischio più dettagliata e strategie di trattamento più efficaci per questa sottocategoria di pazienti.

16/06/25

3. Mortality after out-of-hospital cardiac arrest in a Spanish Region (journal.pone.0175818_S1Dataset_Spain_cardiac_arrest_EDITED)

Le colonne rappresentano le variabili raccolte per ogni paziente che ha subito un arresto cardiaco (fuori da ospedale) e ha ricevuto rianimazione cardiopolmonare dai servizi medici di emergenza nella provincia di Alicante nel 2013.

- **Exitus**: indica se il paziente è deceduto prima di arrivare in ospedale,1 indica il decesso, 0 indica la sopravvivenza.
- **sex_woman**: Il sesso del paziente (1 donna e 0 uomo)
- **Age_years**: L'età del paziente
- **Endotracheal_intubation**: Indica se al paziente è stata applicata la ventilazione assistita tramite intubazione endotracheale (tipo la maschera con il palloncino, 1 indica si 0 indica no)
- **Functional_status**: Lo stato funzionale del paziente prima dell'arresto cardiaco, misurato con il clinical performance score(?). La scala va da 0 (morte cerebrale) a 3 (cosciente normale).
- **Asystole**: Indica se il paziente presentava asistolia (no battito) al momento dell'intervento (1 si 0 no)
- **Cardiac_arrest_at_home**: luogo dell'arresto cardiaco, 1 significa a casa 0 non a casa.
- **Bystander**: Indica se è stata effettuata una bystander CPR (aiuto da parte di un passante) prima dell'arrivo dei soccorsi (1 si 0 no)
- **Time_min**: tempo in minuti tra l'arresto cardiaco e l'arrivo dell'ambulanza
- **Cardiogenic**: indica se la causa dell'arresto cardiaco era di origine cardiogena (cuore non pompa abbastanza sangue tipo, 1 si 0 no)

La mortalità per arresto cardiaco extra-ospedaliero nella regione studiata è risultata molto alta, circa quattro pazienti su cinque in cui è stata tentata la RCP sono deceduti prima dell'arrivo in ospedale. I fattori che sono risultati indipendentemente associati a una maggiore mortalità sono → sesso maschile, presenza di asistolia, un tempo maggiore tra l'arresto e l'arrivo dei soccorsi e il fatto che l'arresto cardiaco sia avvenuto in casa.

4. Comorbid Depression and Heart Failure: A Community Cohort Study (journal.pone.0158570_S2File_depression_heart_failure_v2)

Le colonne nel file di dati rappresentano le variabili cliniche, di laboratorio e di esito raccolte per i 425 pazienti con insufficienza cardiaca (heart failure) inclusi nello studio.

- **Age (years)**: L'età del paziente
- **Male (1=Yes, 0=No)**: sesso del paziente (1 uomo 0 donna)
- **PHQ-9**: Il punteggio di un questionario con 9 domande credo, utilizzato per valutare i sintomi depressivi, un punteggio ≥5 è stato usato per definire la presenza di depressione.
- **Systolic BP (mm Hg)**: pressione sanguigna sistolica
- **Estimated glomerular filtration rate**: La velocità di filtrazione glomerulare stimata, un marker della funzione renale.
- **Ejection fraction (%)**: frazione di eiezione ventricolare sinistra (?), una misura della funzione cardiaca
- **Serum sodium (mmol/l)**: concentrazione di sodio nel siero
- **Blood urea nitrogen (mg/dl)**: concentrazione di azoto ureico nel sangue.
- **Etiology HF(1=Yes, 0=No)**: Indica se l'insufficienza cardiaca aveva un'eziologia ischemica (1 si 0 no)
- **Prior diabetes mellitus**: Indica se il paziente aveva una storia pregressa di diabete mellito
- **Elevated level of BNP/NT-BNP (1=Yes, 0=No)**: Indica se i livelli del peptide natriuretico di tipo B o del frammento N-terminale del pro-BNP (?) erano elevati, secondo soglie specifiche per età (1 significa elevato, 0 non elevato)
- **Time from HF to Death (days)**: Il tempo in giorni trascorso dall'arruolamento nello studio fino al decesso del paziente, per coloro che sono morti entro il periodo di follow-up di 2 anni

- **Death (1=Yes, 0=No)**: esito di mortalità per qualsiasi causa, 1 indica che il paziente è deceduto entro 2 anni, 0 indica che era ancora in vita.
- **Time from HF to hospitalization (days)**: tempo in giorni trascorso dal recruiting fino alla prima ospedalizzazione del paziente.
- **Hospitalized (1=Yes, 0=No)**: L'esito di ospedalizzazione per tutte le cause → 1 indica che il paziente è stato ricoverato almeno una volta entro 2 anni, 0 indica che non è successo.

La depressione è un disturbo comune nei pazienti con insufficienza cardiaca nella comunità, con una prevalenza del 42,1% in questo gruppo di studio. La presenza di depressione (PHQ9≥5) è associata a un rischio più elevato di mortalità per tutte le cause (rischio raddoppiato) e di prima ospedalizzazione (rischio aumentato di circa il 42%) entro 2 anni.

15/06/25

2. Circulating osteocalcin as a bone-derived hormone is inversely correlated with body fat in patients with type 1 diabetes (**journal.pone.0216416_Takashi2019_diabetes_type1_dataset_preprocessed**)

Le colonne nel file di dati rappresentano le caratteristiche cliniche, di laboratorio e fisiche dei 67 (?) pazienti con diabete di tipo 1 partecipanti allo studio.

- **age**: L'età del paziente in anni
- **duration.of.diabetes**: La durata del diabete in anni.
- **body_mass_index**: L'indice di massa corporea del paziente, calcolato in kg/m^2
- **TDD**: Dose giornaliera totale di insulina in unità
- **basal**: La dose basale di insulina in unità
- **bolus**: La dose di insulina in bolo in unità.
- **HbA1c**: Il livello di emoglobina glicata, un indicatore del controllo glicemico a lungo termine.
- **eGFR**: Velocità di filtrazione glomerulare stimata, un indicatore della funzione renale.
- **perc.body.fat**: La percentuale di grasso corporeo.
- **adiponectin**: La concentrazione sierica di adiponectina (??).

- **free.testosterone**: La concentrazione sierica di testosterone libero (misurata solo nei pazienti di sesso maschile).
- **SMI**: Indice di massa muscolare scheletrica in kg/m^2
- **grip.strength**: forza della presa (grip strength) misurata in kg.
- **knee.extension.strength**: La forza di estensione del ginocchio (misurata in kg)
- **gait.speed**: La velocità di andatura (m/s) → una misura di quanto una persona cammina veloce in una certa distanza
- **ucOC**: La concentrazione sierica di osteocalcina non carbossilata (?)
- **OC**: La concentrazione sierica di osteocalcina totale (ng/ml)
- **weight_kg**: Il peso corporeo del paziente in chilogrammi.
- **insulin_regimen_binary**: Il regime insulinico seguito dal paziente, in formato binario, lo studio ha incluso pazienti in terapia con iniezioni multiple giornaliere (MDI) o con infusione sottocutanea continua di insulina (CSII).
- **sex_0man_1woman**: sesso del paziente in formato binario (0 per maschio, 1 per femmina).

È stata trovata una correlazione inversa significativa tra le concentrazioni sieriche di osteocalcina e la percentuale di grasso corporeo nei pazienti con diabete di tipo 1.

In pratica lo studio dimostra che uno degli effetti metabolici benefici dell'osteocalcina (l'associazione con una minore massa grassa) è osservabile anche nei pazienti con diabete di tipo 1, confermando in un contesto clinico umano quanto precedentemente riportato in modelli animali e in pazienti con diabete di tipo 2.

13/06/25

Voglio fare un breve studio sul significato dei dati che vado poi ad usare, per questo recupero i paper da cui deriva e cerco di capire che dati sono stati raccolti e le conclusioni che sono state tratte.

1. C-Reactive Protein and Hemogram Parameters for the Non-Sepsis Systemic Inflammatory Response Syndrome and Sepsis: What Do They Mean **(journal.pone.0148699_S1_Text_Sepsis_SIRS_EDITED)**

Le colonne nel file di dati rappresentano i parametri demografici, clinici e di laboratorio raccolti per ogni paziente nello studio:

- **Age**: L'età del paziente in anni.
- **sex_woman**: Il sesso del paziente in formato binario (dovrebbe essere 1 per donna, 0 per uomo).
- **diagnosis_0EC_1M_2_AC**: La diagnosi all'ammissione in terapia intensiva, i pazienti vengono divisi in tre categorie: "medical, elective and emergency surgery" (patologie mediche, chirurgia elettiva e chirurgia d'urgenza).
- **APACHE II**: È un punteggio utilizzato per classificare la gravità della malattia dei pazienti. Un punteggio più alto indica una condizione più grave.
- **SOFA**: È un punteggio che valuta il grado di disfunzione degli organi di un paziente. Un punteggio più alto è associato a una maggiore probabilità di sepsi.
- **CRP**: Il livello di proteina C-reattiva (C-Reactive Protein), un marker di infiammazione. Nello studio, un valore CRP≥4.0mg/dL è stato identificato come uno dei fattori che aumentano la probabilità di sepsi.
- **WBCC**: La conta dei globuli bianchi, lo studio conclude che questo parametro è poco utile per distinguere la sepsi dalla SIRS non settica (mi sembra).
- **NeuC**: La conta dei neutrofili, anche questo parametro è risultato poco indicativo per la diagnosi di sepsi.
- **LymC**: La conta dei linfociti, una bassa conta di linfociti è stata associata a una maggiore probabilità di sepsi.
- **EOC**: La conta degli eosinofili, lo studio ha rilevato che questo valore non è un marker importante per diagnosticare la sepsi.
- **NLCR**: Il rapporto tra neutrofili e linfociti (Neutrophil-Lymphocyte Count Ratio). Sebbene più alto nel gruppo con sepsi, l'analisi ha mostrato che è la diminuzione dei linfociti (Lymc), più che il rapporto in sé, ad essere significativa.
- **PLTC**: La conta piastrinica, una conta piastrinica bassa è stata associata a un rischio maggiore di sepsi
- **MPV**: Il volume piastrinico medio, non sono state trovate differenze significative di questo valore tra il gruppo con sepsi e quello con SIRS non settica.

- **Group**: Lo studio ha suddiviso i pazienti in due gruppi principali: "non-sepsis SIRS" (SIRS non settica) e "sepsis" (sepsi) → dovrebbe essere 0 non sepsi 1 sepsi.
- **LOS-ICU**: La durata della permanenza in terapia intensiva in giorni, risultata più lunga nei pazienti con sepsi.
- **Mortality**: La mortalità, risultata più alta nel gruppo con sepsi → dovrebbe essere 0 per sopravvisuto, 1 per deceduto.

La conclusione principale è che la combinazione di un livello di proteina C-reattiva CRP≥4.0, una conta linfocitaria Lymc<0.45 e una conta piastrinica PLTc<150 aumenta notevolmente la probabilità di diagnosticare la sepsi all'ammissione in terapia intensiva. La presenza di questi tre fattori contemporaneamente ha aumentato la probabilità di sepsi di 18.1 volte.

09/05/25

Ho trovato questi tre paper che sembrano interessanti sempre nell'ambito del DBCV e gli altri indici.

Automated segmentation of white matter fiber bundles using diffusion tensor imaging data and a new density based clustering algorithm

https://www.sciencedirect.com/science/article/abs/pii/S0933365716301117

Density-Based Clustering Validation

https://epubs.siam.org/doi/abs/10.1137/1.9781611973440.96

Cluster-based analysis of COVID-19 cases using self-organizing map neural network and K-means methods to improve medical decision-making.

https://www.sciencedirect.com/science/article/pii/S2352914822001484

07/05/25

Continuiamo il discorso sugli indicatori iniziato ieri.

DBCV Index → questo indice (Density-Based Clustering Validation) è progettato per validare clustering basati sulla densità, come DBSCAN.

A differenza degli altri indici visti non si basa solo su distanze geometriche, ma valuta direttamente la densità dei cluster e la separazione tra essi. Questo lo

rende ideale per cluster di forma arbitraria, dati con rumore (es. punti non assegnati), e densità non uniformi.

Funzionamento:

1. Mutual Reachability Distance (MRD) → per ogni coppia di punti a e b si calcola:

   $$\text{MRD}(a, b) = \max\left(\text{core-dist}(a), \text{core-dist}(b), \text{distance}(a, b)\right)$$

   dove $\text{core-dist}(x)$ è la distanza tra x e il suo k-esimo vicino in termini di distanza (sarebbe minPts in dbscan). Questo è una misura della densità locale.

2. Minimum Spanning Tree (MST) → per ogni cluster viene costruito un MST usando i valori MRD calcolati prima. Questo albero identifica le aree più dense tra i vari punti.

3. Si considerano:
   Sparseness interna → massimo valore di reachability all'interno dell'albero, più sono lunghi gli archi più il cluster è sparso.
   Separazione tra cluster → minimo valore di reachability tra punti di cluster diversi, due cluster sono ben separati se la MRD tra loro è alta.

4. **Punteggio DBCV** → per ogni cluster C_i si calcola:

$$\text{Validità}(C_i) = \frac{\text{Separazione}(C_i) - \text{Sparseness}(C_i)}{\max\left(\text{Separazione}(C_i), \text{Sparseness}(C_i)\right)}$$

Il punteggio finale è la media ponderata delle validità di tutti i cluster, pesata sulla dimensione dei cluster.

Il risultato è compreso tra -1 e +1. Un valore minore di 0 significa che il risultato del clustering non è buono (ad esempio nel caso in cui ci sono densità variabili improvvise) mentre un valore vicino a 1 (solitamente maggiore di 0.5) implica un buon risultato di clustering.

Un altra particolarità del DBCV è che considera esplicitamente i punti isolati nei dati, in genere i cluster con dimensione minore di 3 vengono ignorati, e un rumore eccessivo riduce il valore dell'indice.

Confronto tra DBCV e gli altri indici.

Tutti gli indici visti hanno l'obiettivo di quantificare coesione intra-cluster e separazione inter-cluster. Sono diversi però nel metodo di misura: silhouette, ch, dunn e dbi si basano su distanze fra punti e centroidi o fra punti stessi, senza considerare in modo esplicito la densità dei punti. Funzionano meglio quando i cluster sono globulari, assumendo implicitamente cluster convessi o comunque di forma regolare.

DBCV invece usa la densità locale per definire coesione e separazione, questo lo rende adatto a cluster di forma arbitraria separati da densità, scenario in cui gli altri indici tendono a fallire.

Anche la robustezza al rumore è una caratteristica distintiva di DBCV, che può gestire punti isolati (noise) come parte dell'algoritmo, mentre gli altri prevedono spesso di pre-processare o ignorare questi punti.

Il grosso problema di DBCV è che computazionalmente molto costoso, dato che deve calcolare la densità locale per ogni punto (usando ad esempio kNN). In generale è $O(n^2)$.

06/05/25

Dato che non ho ancora visto quasi nulla sull'argomento voglio approfondire un pò alcuni indicatori di valutazione del clustering. Questa valutazione si basa su indici che misurano la compattezza interna (cohesion) e la separazione fra cluster → in pratica punti dello stesso cluster dovrebbero essere vicini tra loro e cluster diversi dovrebbero essere ben separati.

Silhouette Index → valuta quanto ogni punto è simile al proprio cluster rispetto ai cluster vicini.

Indichiamo con i il punto, con $a(i)$ la distanza media tra i e gli altri punti del suo cluster, con $b(i)$ la minima distanza media di i dai punti di qualsiasi altro cluster. Si può trovare la silhouette del punto come:

$$s(i) = \frac{b(i) – a(i)}{max[a(i), b(i)]}$$

Quindi $s(i)$ può variare tra -1 e +1, valori vicini a +1 indicano un punto ben assegnato ($a(i)$ molto minore di $b(i)$), valori intorno a 0 un punto che si trova tra due cluster, valori negativi un punto che sarebbe più corretto metterlo in un altro cluster.

Il punteggio complessivo è la media della silhouette su tutti i punti considerati. Questo indice è intuitivo e fornisce anche una rappresentazione grafica (silhouette plot), ma assume implicitamente cluster di forma regolare o "globulare", se i cluster hanno geometrie irregolari e/o densità molto diverse può essere inadeguato.

Calinski-Harabasz Index → questo indice (anche detto *variance ratio criterion*) misura il rapporto tra la dispersione tra cluster diversi e la dispersione all'interno dei singoli cluster.

Considerando un clustering di n dati in k cluster si definiscono **BCSS** (Between-Cluster Sum of Squares) come la somma pesata (per numerosità) delle distanze quadratiche tra ciascun centroide dei cluster e il centroide (baricentro) globale, e **WCSS** (Within-Cluster Sum of Squares) come la somma delle distanze quadratiche di ogni punto dai rispettivi centroidi. L'indice è dato da:

$$CH = \frac{BCSS/(k–1)}{WCSS/(n–k)}$$

Non è limitato superiormente (dipende dai dati), un valore più alto indica cluster ben separati (BCSS alto) e compatti (WCSS basso). Spesso si confrontano diversi k, il massimo CH suggerisce il numero di cluster migliore.

Anche questo indice lavora bene con cluster di dimensioni simili e sferici, ma può non essere adeguato a cluster di forma arbitratia e densità variabile, proprio per il fatto che si basa sulla distanza dai centroidi.

Dunn Index → questo indice valuta la qualità di un clustering basandosi sulla distanza minima tra cluster e la distanza massima interna di un cluster.

Definiamo due valori:

- $\delta(C_i, C_j)$ → è la distanza inter-cluster minima tra due cluster C_i e C_j, calcolata come la minima distanza tra due punti appartenenti a cluster diversi.
- Δ_k → il massimo diametro del cluster C_k, la massima distanza tra due punti all'interno dello stesso cluster.

L'indice si definisce come:

$$\text{Dunn} = \frac{\min_{i<j} \delta(C_i, C_j)}{\max_k \Delta_k}$$

In pratica considera il worst-case scenario, il minimo delle distanze tra cluster e il massimo dei diametri intra cluster, quindi la vautazione dipende dal cluster meno compatto e la coppia di cluster più vicini. Valori elevati indicano cluster ben separati e compatti.

Davies-Bouldin Index → valuta la qualità del clustering misurando il rapporto tra la dispersione interna dei cluster e la loro distanza reciproca.

Calcolo del DBI:

- Per ogni cluster i si calcola la dispersione come distanza media dei punti dal relativo centroide.
- Per ogni coppia di cluster i e j si misura la distanza tra i loro centroidi.
- Si definisce il rapporto di similarità, che quantifica la somiglianza tra due cluster, un valore alto significa cluster vicini e/o dispersi.

$$R_{i,j} = \frac{S_i + S_j}{d(i,j)}$$

- Per ogni cluster i, si seleziona il valore massimo di R_{ij} tra tutti i $j \neq i$.
- L'indice è la media di questi massimi su tutti i cluster.

A differenza degli altri indici visti, un DBI basso indica cluster ben definiti, con alta seperazione e bassa dispersione. Il valore "ideale" è DBI=0, che significa cluster puntiformi infinitamente distanti tra loro, più è alto invece più mostra la presenza di cluster sovrapposti e/o molto dispersi.

28/04/25

https://link.springer.com/article/10.1007/s41109-021-00395-2

questo articolo presenta un metodo per analizzare e strutturare i dati testuali non strutturati delle cartelle cliniche elettroniche (EHR), per valutarne la qualità a livello di una città, prendendo come caso studio san pietroburgo.

Il problema di base è la variabilità dei dati provenienti da diversi ospedali e sistemi informativi sanitati; il metodo usa tecniche di natural language processing (NLP). In particolare viene applicato l'algoritmo di clustering OPTICS ai vettori TF-IDF (trasformazione in vettori numerici) degli header delle cartelle per identificare automaticamente i tipi di cartelle cliniche.

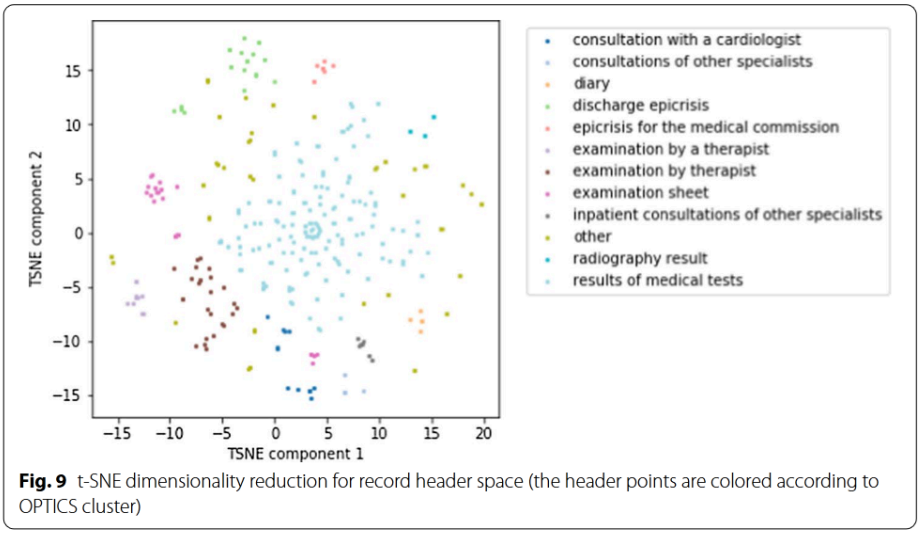


Fig. 9 t-SNE dimensionality reduction for record header space (the header points are colored according to OPTICS cluster)

L'approccio si è dimostrato più efficace dell'etichettatura manuale nel raggruppare le cartelle in tipi significativi e contribuisce a valutare la completezza delle informazioni presenti nelle EHR.

La trasformazione Term Frequency - Inverse Document Frequency funziona così:

- Term Frequency (TF) → si calcola la frequenza di ogni parola all'interno di un singolo documento (header della cartella clinica).
- Inverse Document Frequency (IDF) → viene calcolato quanto è rara una parola nell'intero insieme di documenti considerati, le parole frequenti hanno score IDF basso, le parole rare hanno IDF alto. In questo modo si da meno peso alle parole comuni (stop words) che non sono molto informative per distinguere i documenti.
- Calcolo di TF-IDF → il valore per ogni parola in un documento è il prodotto dei due valori trovati, in questo modo si crea un vettore numerico.

E' interessante anche il motivo per cui viene usato OPTICS, l'articolo spiega che i nomi di alcuni tipi di record sono molto uniformi tra i vari sistemi informativi sanitari, creando cluster molto densi e compatti. Altri tipi di record invece hanno nomi molto variabili (ad esempio con aggiunta di nomi di reparti, specialisti, medici, sinonimi), portando a cluster più sparsi e meno densi, e (potenzialmente) anche di dimensioni diverse.

27/04/25

Ho recuperato un articolo simile a quello di ieri, che ha l'obiettivo sempre di trovare un approccio migliore alla segmentazione automatica di aree di interesse nelle immagini mediche.

https://link.springer.com/article/10.1007/s11042-020-09640-9

Per ovviare sempre a problemi di gestione del rumore, di bordi deboli o variazioni di intensità luminosa, il lavoro propone la combinazione di due tecniche:

1. Clustering BIRCH → viene usato l'algoritmo di clustering gerarchico per creare un raggruppamento a più livelli che suddivide i pixel in base alla densità dei loro valori di grigio.
2. Active Contour Model → è un metodo che fa evolvere una curva iniziale fino ad aderire ai contorni dell'oggetto da segmentare, usando una 'energy function' che considera sia informazioni globali (legate al clustering) che informazioni locali (legate a intensità e caratteristiche statistiche).

Il punto di partenza del metodo è un passo di "pre-elaborazione spaziale", in cui l'immagine viene smussata. Per ogni pixel (i, j) dell'immagine originale R viene costruita una nuova immagine Q in cui il valore di ciascun pixel è la media aritmetica dei 9 valori (3*3) intorno al punto (mean filter).

L'articolo che ho letto ieri e questo hanno la stessa applicazione ma sono sostanzialmente diversi. Entrambi combinano due tipi di informazioni, il primo integra il clustering globale (BIRCH) e le statistiche locali all'interno della energy function, integrando l'algoritmo di clustering profondamente per modificare l'active contour model e la sua funzione di energia, il secondo usa clustering globale (mean shift) per la fase iniziale e il fitting locale (RSF) per l'evoluzione, ponendo maggiore enfasi sull'automazione dell'inizializzazione e sulla stima dei parametri derivata dal clustering.

26/04/25

Ho recuperato un paper che propone un approccio moderno alla segmentazione (riconoscimento dei vari elementi) di immagini del mondo medico:

https://doi.org/10.1016/j.compbiomed.2013.08.024

I metodi classici iniziano con un contorno casuale e poi cercano di muoverlo verso i bordi giusti, andando spesso in difficoltà per inizializzazione instabile, diventando computazionalmente complessi (a causa di continue re-inizializzazioni) e trovando problemi quando ci sono situazione di rumore o luminosità variabile.

Il paper propone un approccio in due fasi:

1. Con il clustering mean shift vengono "raggruppati" i pixel simili nell'immagine, costruendo un contorno iniziale già vicino a quello reale dell'oggetto che ci

interessa.

2. RSF (region scalable fitting) Level Set → in questa fase viene “aggiustato” il contorno iniziale fino a farlo combaciare con i bordi reali. E’ region based perchè guarda sia l’interno che l’esterno del contorno (non solo i bordi), diventando più robusto a situazioni di rumore o zone sfumate.
   Attorno ad ogni punto del contorno RSF calcola le medie di intensità dentro e fuori il bordo iniziale usando una finestra pesata (kernel) che da’ più importanza ai pixel vicini e meno a quelli lontani.

Gli esperimenti riportati nell’articolo mostrano come sia su ecografie che su TAC, il modello MS-RSF ha prodotto segmentazioni più precise e veloci rispetto ad altri metodi classici.

14/04/25

Ricerca di articoli scientifici. Direi di fare per ogni algoritmo 2 più teorici e 2 applicativi idealmente in ambito bio-medico.

https://www.sciencedirect.com/science/article/abs/pii/S174680942100793X

In questo paper applicativo viene presentato un metodo semi supervisionato per trovare outlier nel processo di monitoring continuo del glucosio (aka CGM). Usa una variante di OPTICS (ICP-OPTICS), assegnando a ogni campione un peso ottimizzato tramite una funzione di ottimizzazione che bilancia reachability distance e information entropy (una misura di dispersione/incertezza).

L’algoritmo viene inizialmente addestrato su dati puliti (già noti) e poi validato su un test set. L’idea è di rilevare malfunzionamenti dei sensori CGM, che vengono divisi in 4 categorie.

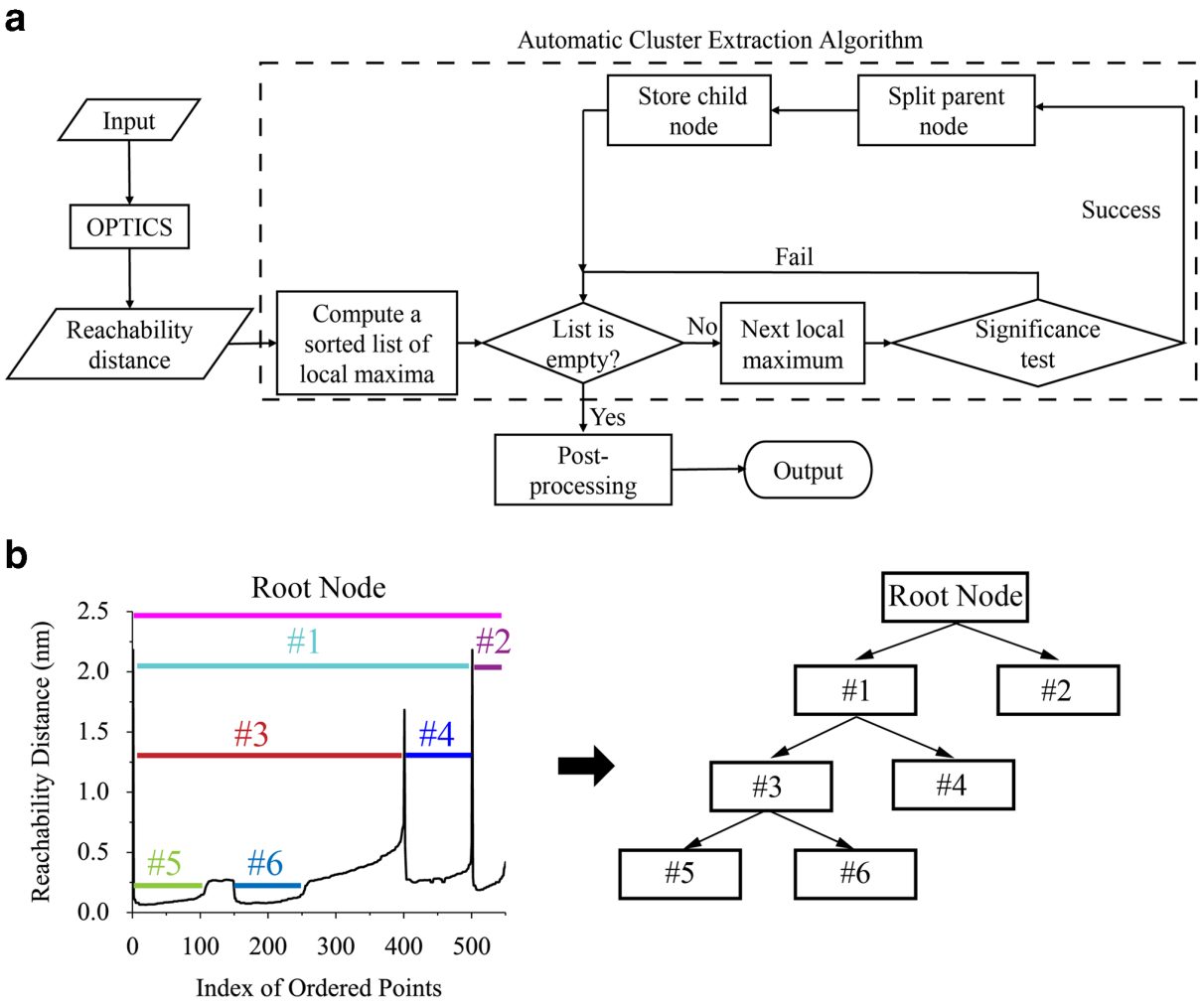
31/03/25

https://academic.oup.com/mam/article-abstract/25/2/338/6887445

L’obiettivo di questo articolo è migliorare l’analisi di cluster in una tecnica avanzata (atom probe tomography) per visualizzare strutture nanometriche 3d in alcuni materiali. In particolare vuole mostrare come uno degli algoritmi più usati per questo tipo di clustering è il DBSCAN, che ha limitazioni nel gestire varazioni di densità atomica e richiede vari iperparametri, proponendo una specifica applicazione di OPTICS come soluzione.

Il reachability plot rende più facile l'interpretazione e la regolazione dei parametri, offrendo un approccio più flessibile e accurato; anche in casi di necking (cluster collegati) e per separare cluster con gradienti di densità molto bassi (che causano facilmente problemi con DBSCAN).

Il paper presenta anche un algoritmo (Sander, 2003) per la rilevazione dei cluster a partire dal RP, diverso dal semplice cut per una certa epsilon arbitraria. L'algoritmo rileva i picchi locali sul RP (massima densità), li ordina dal più alto al più basso e per ognuno verifica se il picco è abbastanza alto rispetto alla densità media circostante (usando una certa soglia), se si divide in due il gruppo. Ripete la procedura su tutti i gruppi, costruendo un albero gerarchico dove le foglie sono i cluster finali.



28/03/25

https://arxiv.org/abs/1503.00687

Questo è un articolo teorico che fa una review degli algoritmi mean shift, metodi non parametrici per il clustering basati sulla stima della densità del kernel (KDE). L'obiettivo è identificare cluster come regioni ad alta densità, sfruttando i massimi locali (o "modi") della KDE. A differenza di metodi parametri (come k-means) non richedono di specificare a priori il numero di cluster e gestiscono bene forme complesse.

I due algoritmi più classici sono:

1. Mean-Shift → inizializza tutti i punti e itera spostandoli verso la media locale pesata dei vicini, convergendo ai massimi locali della finestra kernel. Come svantaggi ha un costo computazionale elevato $(O(N^2))$ ed è sensibile al parametro di bandwidth (σ).
2. Blurring Mean-Shift → aggiorna iterativamente l'intero dataset, sostituendo ogni punto con la media locale. Il dataset progressivamente collassa in cluster, senza un criterio di arresto l'intero dataset avrà punti tutti coincidenti.
   Quindi il BMS fa una sorta di "smoothing" (o filtering) dei dati.

Il bandwidth determina la scala di analisi, un sigma piccolo produce più cluster, uno più grande ne trova meno (li "unifica"). Anche il kernel è variabile, c'è il kernel gaussiano ma anche il kernel Epanechnikov.

20/03/25

BIRCH → threshold (raggio massimo), branching factor

MeanShift → bandwidth

OPTICS → minPts (per essere core), max_eps (ϵ_{max}, distanza max considerata per definire un vicinato (core distance))

19/03/25

Per tenere i test che sto facendo anche su github sto usando pycharm. Ho creato una repository:

https://github.com/spinalscratch/clustering

---

visto una demo con meanshift (meanShift.py):

https://scikit-learn.org/stable/auto_examples/cluster/plot_mean_shift.html#sphx-glr-auto-examples-cluster-plot-mean-shift-py

che mostra come si può fare una rilevazione automatica del bandwidth usato dall'algoritmo, e qualche altro modo per migliorare la visualizzazione dei cluster nel plot.

https://github.com/scikit-learn/scikit-learn/blob/98ed9dc73/sklearn/cluster/_mean_shift.py#L300

Questa è l'implementazione di sklearn dell'algoritmo, molto complessa. Utilizza un approccio basato su kernel flat quindi una finestra fissa definita dal bandwidth.

18/03/25

Continuo a leggere la documentazione di scikit learn sul clustering.

Visto questa lecture https://www.youtube.com/watch?v=xqWsAkIby6c in cui viene mostrato iterazione per iterazione come opera OPTICS su un gruppo di dati. Una cosa che non avevo ancora visto è come si deve considerare la possibilità che un border point prima di una certa valle si deva includere nello specifico cluster.

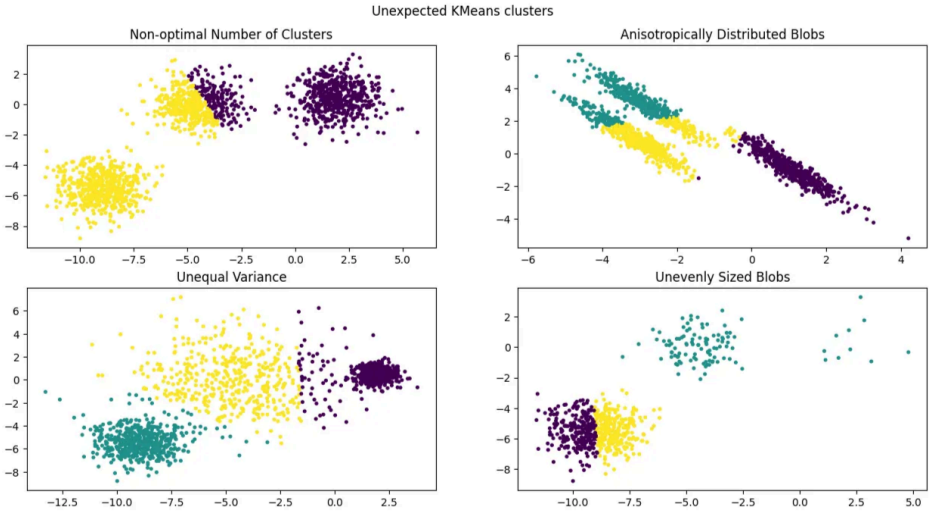
Il video successivo https://www.youtube.com/watch?v=VziKnZSffRc continua mostrando come si interpreta un reachability plot. Non avevo pensato all'influenza che ha il punto di partenza e la struttura del cluster sull'ordine con cui vengono visitati i dati (e quindi costruito il reachability plot).

Mostra come fare un taglio orizzontale ad una certa altezza del RP costruisce gli stessi cluster che produrrebbe DBSCAN con quella specifica epsilon. Come alternativa più elegante fa vedere anche lo xi method (ξ). Il parametro ξ definisce quanto deve essere "ripida" una variazione nel reachability plot per considerarla un confine tra cluster. Quando viene superata questa soglia viene riconosciuto un punto di separazione che delimita l'inizio o la fine di un cluster.

Con questo altro esempio https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_assumptions.html mostra alcuni possibili problemi dell'algoritmo k means.

Ma più che quello sto capendo come funziona matplotlib per produrre i grafici con le varie opzioni.

Vengono mostrati due limiti dell'algoritmo k means: il primo si ha in presenza di varianze anisotropiche, cioè quando la varianza dei dati cambia a seconda della direzione, questo significa che i cluster non sono sferici ma possono essere ellittici o con forme più complesse. Il k means si basa sull'assunzione che i cluster siano sferici e di dimensioni simili.



Il secondo si vede nel risultato ottenuto nel grafico in basso a sx. Il k means in questa situazione di varianze variabile, dove il cluster giallo ha alta varianza, quelli ai lati più bassa, tende ad assegnare più punti ai cluster con varianza bassa (punti più compatti).

Si vede come molti punti del cluster più compatto (viola) vengono assegnati per errore a lui invece che correttamente a quello centrale.

17/03/25

Dato che vorrei iniziare a fare qualche prova di applicazione di questi algoritmi, ho iniziato a guardare la pagina della documentazione di scikit learn relativa al clustering.

https://scikit-learn.org/stable/modules/clustering.html

Questo progetto ha l'obiettivo di fornire tool semplici ed efficienti per l'utilizzo di tecniche di machine learning con python.

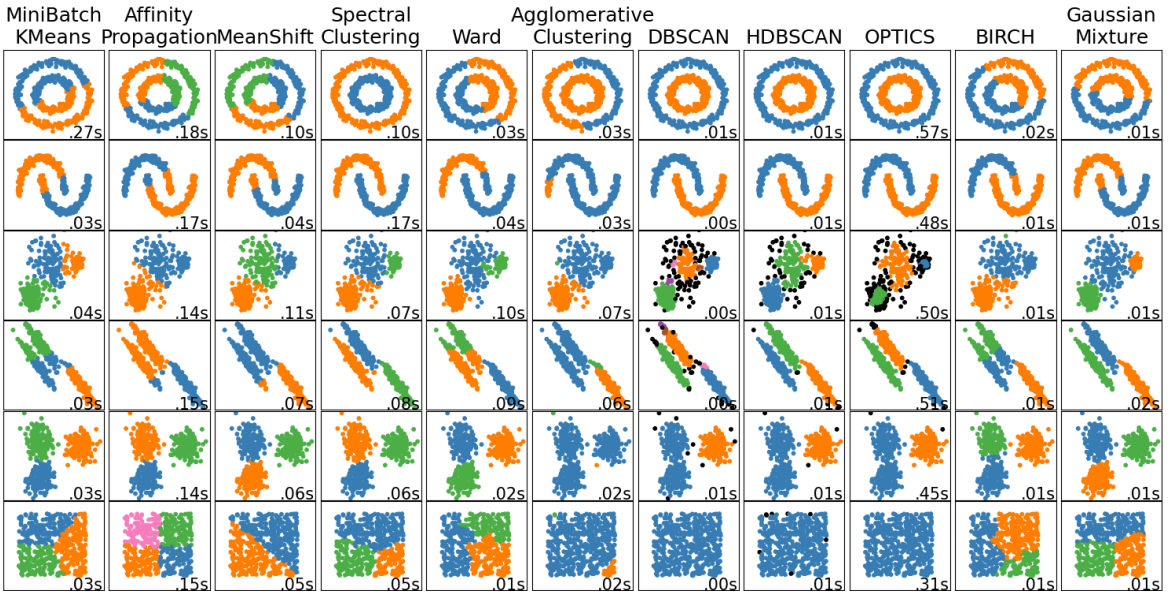
Dato che l'intera documentazione è legata a python e sarebbe non banale portare tutto in R per ora non lo userò.

Quindi inizio a preparare l'ambiente, ho aggiunto sklearn alla install di python da cmd → `pip install scikit-learn`.

Allo stesso modo ho installato anche matplotlib, una libreria per la visualizzazione di dati in python.

Per controllare che tutto funzionasse ho provato il primo esempio mostrato sulla documentazione, in cui vengono provati alcuni algoritmi di clustering su un 6 dataset di esempio, si cui l'ultimo è un esempio di dataset omogeneo in cui non c'è una soluzione corretta.

Con il codice dato viene fuori una rappresentazione molto interessante tramite matplotlib.



La prima cosa che ho notato è la differenza tra i tempi di esecuzione, è evidente come OPTICS sia decisamente più lento delle alternative, allo stesso tempo offre uno dei risultati migliori tra le varie possibilità.

In particolare nella prima riga si vede un dataset che presenta due corone concentriche di punti, molti algoritmi fanno fatica a distinguere correttamente i due, invece quelli “dbscan based” riconoscono correttamente la struttura dei punti.

Interessante anche il 5° dataset dove ci sono 3 gruppi di punti, due dei quali sono più vicini tra loro rispetto all’altro. Alcuni algoritmi identificano 3 cluster, altri ne identificano 2.

14/03/25

Oggi voglio fare una sintesi di ognuno dei principali algoritmi di clustering che sono venuti fuori in questi giorni.

K-means

E’ un algoritmo basato sui centroidi che divide i dati in k cluster (il numero di cluster deve essere definito all’inizio), dove ogni cluster è rappresentato della media (cioè il centroide) dei suoi punti.

Funzionamento dell’algoritmo:

- I k centroidi vengono inizializzati casualmente.
- Ogni punto viene assegnato al centroide più vicino.
- I centroidi vengono aggiornati calcolando la media dei punti assegnati ad ognuno dei cluster.
- Ripete questi due passaggi finchè i centroidi non cambiano più la loro posizione (convergono).

Hierarchical clustering

Il clustering gerarchico raggruppa data point simili in cluster, organizzati secondo una struttura ad albero che si chiama dendrogramma. Si usa per trovare relazioni gerarchiche nei dati.

Ci sono due tipi: agglomerative (o bottom-up), che inizia considerando ogni punto come un cluster e va iterativamente ad unire le coppie più vicine finchè tutti i punti formano un singolo cluster, è il più usato.

Il secondo tipo è il divisive (o top-down), che fa il percorso inverso, iniziando da un singolo cluster per tutti i punti, dividendolo ricorsivamente fino a quando i punti sono tutti divisi singolarmente. Viene usato meno perchè ha complessità computazionale elevata $\rightarrow O(n^3)$.

Un parametro variabile è il linkage, cioè come viene calcolata la distanza tra i cluster, vengono usate varie possibilità (single, complete, avarage...). Non richiede di specificare in anticipo il numero di cluster.

BIRCH (balanced iterative reducing and clustering using hierarchies)

Nasce per essere usato con grandi dataset, costruisce una struttura ad albero detta CF-tree, che sintetizza i dati in clustering feature (CF), ognuna della quali contiene informazioni come il numero di punti (del cluster), la somma dei vettori e dei quadrati dei vettori.

- Viene costruito il CF-Tree in memoria.
- Opzionalmente l'albero viene ulteriormente ridotto, per rimuovere outlier e ridurre la granularità.
- Viene quindi usato un altro algoritmo (come k means o clustering gerarchico) sulle foglie dell'albero.

Questo sistema è molto efficiente e fa una gestione incrementale dei dati (senza dover salvare in memoria grandi quantità di dati contemporaneamente). Un parametro importante è la soglia per il CF-tree, cioè quanto vicini devono essere due data point per rientrare nella stessa foglia (micro-cluster).

DBSCAN (density-based spatial clustering of applications with noise)

Questo è un algoritmo basato sulla densità, raggruppa i punti che sono densamente connessi secondo un raggio definito all'inizio (ϵ) e il numero di punti minimo (minPts) che devono essere contenuti nella "circonferenza" definita da questo raggio.

I punti isolati o in zone a bassà densità vengono etichettati come outlier.

1. L'algoritmo inizia selezionando un data point qualsiasi che non è stato ancora visitato.
2. Secondo epsilon viene controllato il "vicinato" del punto scelto.
3. Se il numero di punti nel raggio è maggiore o uguale a minPts il punto viene etichettato come core point e viene creato un nuovo cluster.
4. Se il punto è centrale (core), l'algoritmo raccoglie tutti i punti raggiungibili (sia core che border) e li aggiunge al cluster. Il processo continua ricorsivamente finchè non ci sono più punti raggiungibili per densità.
5. I data point che non sono ne centrali ne di confine vengono etichettati come punti di rumore (noise o outlier).
6. I punti 1-5 vengono ripetuti fino a quando tutti i punti non sono stati etichettati come parte di un cluster o noise.

L'utilizzo di parametri fissati può essere un limite, soprattuto in dataset con densità molto variabili.

OPTICS (ordering points to identify the clustering structure)

Questo algoritmo si può vedere come un estensione di DBSCAN, ma necessita di due concetti ulteriori.

- Core distance → per un punto p, è la minima distanza epsilon tale che entro questo raggio ci siano almeno minPts punti (compreso p). Se p non ha abbastanza vicini, la sua core distance è indefinita (non è core point).
- Reachability distance → per un punto p rispetto ad un altro punto o (core point già processato) è il valore massimo tra la core distance di o e la distanza (euclidea) tra o e p. Il suo scopo è quello di garantire che p sia raggiungibile da o mantendo la densità locale del punto o.

I parametri che vengono usati sono minPts, che rappresenta il numero minimo di punti per definire un core point, e un raggio massimo di ricerca ϵ_{max}. Il raggio massimo è un parametro facoltativo e si usa per limitare i calcoli necessari.

1. Inizializzazione: per ogni punto p viene calcolata la core distance. Viene mantenuta una lista ordinata dei punti, che è la base per la costruzione del reachability plot, e una coda con priorità per gestire i punti da esplorare. Questa coda è una struttura dati che mantiene - dopo che un core point viene identificato - l'elenco dei vicini da esplorare, la priorità sta nel fatto che vengono esplorati prima i vicini con reachability distance più bassa.
2. Processamento dei punti: si seleziona un punto non processato p, se questo è un core point (ha una core distance definita), viene calcolata la reachability distance per tutti i punti nel suo intorno, che vengono inseriti nella coda prioritaria.
   Viene estratto dalla coda il punto con RD minima e aggiunto alla lista ordinata, questo passaggio si ripete finchè tutti i punti sono processati.
3. Generazione del reachability plot: l'output prodotto è una lista ordinata di punti con la loro RD. Il reachability plot è un grafico dove l'asse x rappresenta l'ordine dei punti e l'asse y la loro RD.

Usando questa struttura un cluster più denso presentare delle "valli" con RD bassa, uno meno denso avrà RD più alta, mentre cluster con densità variabile avranno valli a diverse altezze. I picchi saranno le transizioni tra cluster o noise points.

L'estrazione dei cluster viene fatta scegliento un valore epsilon a posteriori, facendo un taglio orizzontale in cui tutti i punti con RD minore o uguale ad epsilon appartengono allo stesso cluster, con i picchi sopra epsilon che separano i vari cluster.

MeanShift

E' un algoritmo che utilizza la stima della densità, ogni punto viene spostato iterativamente verso la zona a densità massima (una sorta di centro di massa locale) secondo una finestra di dimensione arbitraria (kernel).

1. Per ogni data point, viene definito un kernel con una larghezza specifica.
2. Per ogni punto viene calcolato il centro di massa dei punti all'interno della sua finestra, ponderando i punti in base alla loro distanza.
3. Il punto viene spostato verso il centro di massa trovato.
4. I passaggi 2 e 3 vengono ripetuti finche i punti non convergono, cioè fino a quando lo spostamento diventa trascurabile.
5. I punti che convergono verso lo stesso centro di massa vengono raggruppati nello stesso cluster.

Non richiede di specificare il numero di cluster, può avere costo computazionale alto.

13/03/25

https://www.frontiersin.org/journals/artificial-intelligence/articles/10.3389/frai.2022.842306/full

In questo altro lavoro che sembra essere una guida per i clinici alle possibilità offerte dal clustering di vari tipi, viene mostrato come il clustering possa rivelare associazioni non facilmente rilevabili nella patofisiologia delle malattie.

Si focalizza anche sull'importanza di un approccio iterativo e validato per evitare errori di interpretazione e massimizzare il valore clinico delle scoperte che vengono fatte.

12/03/25

Questo articolo che definirei applicativo approfondisce l'applicazione di algoritmi di clustering a dati clinici (EHR).

https://pubmed.ncbi.nlm.nih.gov/22275205/

Il paper esplora l'uso di tecniche avanzate di clustering per migliorare la classificazione dei pazienti mediante dati EHR. Vengono comparati otto algoritmi di clustering (inclusi K-Means, DBSCAN, Clustering Gerarchico, Mean Shift).

Fa anche una valutazione basata su criteri come qualità dei cluster, scalabilità, robustezza al rumore, forma e densità dei cluster, interpretabilità, numero di cluster.

11/03/25

L'obiettivo del lavoro di questa settimana è comprendere meglio gli algoritmi di density based clustering e iniziare a leggere qualche paper che li applica (idealmente) all'ambito biomedico.

Un problema che sto riscontrando è la mancanza di accesso ad alcuni articoli linkati da google scholar.

Ho letto questa analisi per approfondire DBSCAN e OPTICS.

https://www.atlantbh.com/clustering-algorithms-dbscan-vs-optics/

Il testo spiega il concetto di clustering, distinguendo tra apprendimento supervisionato e non supervisionato per raggruppare dati simili. Viene mostrato il funzionamento di DBSCAN, che usa i parametri ε e minPts per identificare cluster basati sulla densità dei dati. Parla del limite di DBSCAN nel rilevare cluster con densità variabili, dato che presume una densità costante in tutto il dataset. Si parla di OPTICS come estensione di DBSCAN, capace di generare una struttura gerarchica e di gestire meglio cluster a densità variabili attraverso il reachability plot.

Un esempio in python fa vedere come OPTICS (rispetto a DBSCAN) identifica correttamente tutti i cluster, evidenziando l'importanza del giusto approccio nel data clustering.

Usando **https://sci-hub.se/ ho trovato il paper completo IEEE che mi sembrava interessante:**

https://ieeexplore.ieee.org/abstract/document/7479923

Che usa una piattaforma di visualizzazione dei cluster che si chiama weka, prossimamente lo voglio provare. Ho trovato anche un altro tool (ggplot) ma farei una cosa alla volta.