Proposte progetti per l'esame di Ottimizzazione (mod. 2, a.a. 2021–22)

April 2022

Premessa

In questo PDF sono elencate alcune proposte per il progetto dell'esame di Ottimizzazione. Questa lista non è esaustiva per le studentesse e gli studenti del corso di Ottimizzazione (6 cfu), che possono concordare un progetto sugli argomenti del primo modulo con il prof. Pavarino. Il progetto va presentato con delle slide in un'esposizione di non più di 15 minuti a persona, e viene integrato con la consegna degli esercizi di Laboratorio e una prova orale (individuale) su tutto il programma del corso.

Di seguito alcune regole per la prenotazione e lo svolgimento del progetto:

- È possibile esprimere la propria preferenza - come singola/o o gruppo di due persone - per 2 o 3 delle tracce inviando una mail a `davide.duma@unipv.it`. Per l'assegnazione vale la policy del First Come First Served.
- È buona norma prenotare un progetto non più di 90 giorni prima dell'appello in cui si intende sostenere l'esame; le presenti tracce sono da intendersi valide fino al 31/03/2023.
- È possibile proporre entro un mese dall'esame un argomento alternativo a quelli proposti in questo PDF, che dovrà essere approvato dal docente.
- Le studentesse e gli studenti sono invitati a tenersi in contatto con il docente, soprattutto per quanto riguarda variazioni rispetto alla traccia, eventuali scelte modellistiche e l'organizzazione dell'analisi computazionale.

1 Collaborative filtering per stimare il flusso dei pazienti nelle nuove unità degli ospedali regionali

Informazioni principali

- Team di 2 persone

- Sviluppo in Python o MATLAB
- Dataset reale fornito dal docente
- Approfondimento su Collaborative Filtering e Similarity Matrix: *Mariette Awad, Rahul Khanna (2015). Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers. Sez. 2.3–2.4.* [PDF]

Descrizione

Siano C l'insieme di tutti i comuni di una regione italiana, H l'insieme di tutte le strutture ospedaliere e S l'insieme di tutte le specialità mediche. A partire dai dati storici degli accessi alle strutture ospedaliere regionali (SDO), è possibile determinare il flusso di pazienti di un certo comune $c \in C$ che hanno avuto un periodo di degenza nell'unità (s, h) della specialità $s \in S$ all'interno dell'ospedale $h \in H$, che si trova nel comune $c_h \in C$. La capacità giornaliera $\gamma_{(s,h)} \in \mathbb{N}$ dell'unità può essere stimata come il massimo numero di pazienti ricoverati in un certo arco di tempo.

Data una specialità s, è dunque possibile creare una matrice $M_s \in \mathbb{R}^{|C|,|H|}$ che indica le frequenze di accesso di pazienti residenti nel comune c all'unità (h, s).

Creare i seguenti script:

1. `ReadData`: legge il dataset e, data una data `DD/MM/YYYY`, genera le matrici $M_s, s \in S$, e le esporta in dei file `csv`;
2. `CityBasedFlow`: legge i file generati da `ReadData` e, dati in input il comune di una nuova unità e la sua capacità, stima il numero degli accessi dai comuni in C basandosi sull'affluenza delle altre specialità dello stesso ospedale attraverso lo user-based collative filtering;
3. `HospitalBasedFlow`: come sopra ma con l'item-based collaborative filtering;
4. `SimilarityFlow`: come sopra ma con la similarity matrix, dove il coefficiente `coeff` $\in$ {`Pearson, Spearman, Euclidean, Jaccard`} is given as input.

Hint: L'approccio usato può essere validato considerando unità che sono state aperte durante la raccolta dei dati.

2 Predizione dei pazienti che abbandonano il pronto soccorso senza essere visitati

Informazioni principali

- Team di 2 persone

- Sviluppo in Python o MATLAB
- Dataset reale fornito dal docente
- Approfondimento su Decision Tree: *(i) Steven L. Brunton, J. Nathan Kutz (2019). Data Driven Science & Engineering: Machine Learning, Dynamical Systems, and Control. Sez. 5.8.* [PDF] *(ii) T. Daniya, M. Geeta, Suresh Kumar (2020). Classification and regression trees with Gini index. Advances in Mathematics Scientific Journal.* [PDF]

Descrizione

A causa dell'impossibilità di pianificare le visite a priori, un problema che riguarda tutti i Pronto Soccorso (PS) è quello del sovraffollamento che si presenta in corrispondenza di un picco della domanda, causando alti tempi d'attesa prima dell'ammissione (DTDT - Door-To-Doctor Time) e di permanenza nella struttura (EDLOS - Emergency Department Length-Of-Stay). Uno degli effetti indesiderati è quello dei pazienti che abbandonano il PS prima di essere visitati (LWBS - Leaving Without Being Seen), che rappresenta un rischio per i pazienti e il cui tasso è uno dei più importanti indici di valutazione della qualità del servizio offerto dal PS.

A partire da un dataset reale, fissato un intervallo di tempo $I = [(n-1)\Delta t, n\Delta t], n \in \mathbb{N}$, è possibile etichettare i pazienti che hanno atteso per una durata $w \geq (n-1)\Delta t$ in due classi:

W_I: pazienti che hanno atteso per un tempo pari a o maggiore di $n\Delta t$ in attesa dell'ammissione (che include sia i pazienti che sono stati successivamente visitati, che quelli che hanno abbandonato più tardi);

L_I: pazienti che hanno abbandonato durante l'intervallo di tempo I.

In altre parole, ad essei può essere associata una variabile 0-1 a seconda che siano in attesa oppure .

Creare i seguenti script:

1. `LearnLWBS`: Dati il dataset i parametri T e Δt Per ogni $t \in \{\Delta t, 2\Delta t, \ldots, T\}$ fornire una classificazione dei pazienti LWBS sulla base delle loro caratteristiche tramite un albero decisionale basato sull'indice di Gini.
2. `PredictLWBS`: Dato un dataset con dei pazienti in attesa per l'ammissione, le loro caratteristiche e i rispettivi tempi di attesa (multipli di Δt), e un intero N, per ogni paziente stimare la probabilità che esso abbandoni prima dei prossimi $\Delta t, 2\Delta t, \ldots, N\Delta t$ minuti, con N dato in input.

Validare opportunamente la tecnica adottata e svolgere un'analisi per scegliere opportunamente il valore di Δt.

3 Predizione della presenza di patologie con il k-Nearest Neighbour classico e pesato

Informazioni principali

- Team di 2 persone
- Linguaggio di programmazione a piacere
- Open data: Diabete: Diabete e Parkinson
- Approfondimento su kNN: *Jianping Gou, Lan Du, Yuhong Zhang, Taisong Xiong (2012). A New Distance-weighted k-nearest Neighbor Classifier. Journal of Information & Computational Science.* [PDF]

Descrizione

Tra i più noti algoritmi di classificazione, vi è quello basato sul k-Nearest Neighbour: intuitivamente, dato un data point non ancora etichettato, si vanno a osservare i k data point più vicini ad esso nel dataset, e si effettua una predizione sulla base delle loro etichette.

Creare tre funzioni che, dati in input un set di dati etichettati e un set di dati da etichettare, fornisca un classificatore tramite le tre varianti del k-Nearest Neighbour, `kNN`, `WkNN` e `DWkNN`, riportate nell'articolo di Gou et al.

Creare uno script per la lettura dei dati da file e che effettui una predizione con tutti e 3 gli algoritmi per i dateset `Diabete` e `Parkinson`.

4 Un algoritmo genetico per il clustering

Informazioni generali

- Team di 2 persone
- Linguaggio di programmazione a piacere
- Approfondimento: *(i) Ujjwal Maulik, Sanghamitra Bandyopadhyay (1999). Genetic algorithm-based clustering technique. Pattern Recognition 33, 1455-1465* [PDF] *(ii) Vijini Mallawaarachchi. Introduction to Genetic Algorithms — Including Example Code. TowardsDataScience.com* [Link]
- Clustering data benchmark (S-sets and UCI datasets)

Descrizione

Gli algoritmi genetici sono una particolare classe di algoritmi euristici in grado di determinare una soluzione quasi-ottima di un problema di ottimizzazione in un ridotto tempo computazionale (si guardi l'articolo divulgativo su TowardsDataScience.com). Un algoritmo genetico è stato proposto per il clustering da

Maulik et al. L'obiettivo è quello di implementare tale algoritmo e testarlo su dei benchmark dataset. Si modifichi, inoltre, opportunamente la funzione di fitness in modo tale da imporre diversi criteri per la definizione del clustering ottimale (es. norma euclidea, norma infinito). Confrontare le performance dell'algoritmo (funzione obiettivo e tempo di calcolo) rispetto agli algoritmi di Lloyd e di Ward.

5 Clustering per i gironi delle competizioni sportive

Informazioni generali

- Team di 2 persone
- Linguaggio di programmazione a piacere
- Approfondimento su Constrained k-Means Clustering: *P.S. Bradley, K.P. Bennett, A. Demiriz (2000). Constrained k-Means Clustering. Microsoft MSR-TR-2000-65* [PDF]
- Dati disponibili al link FIT Lombardia

Descrizione

Diversi tipi di competizioni, sportive e non, richiedono il partizionamento dei concorrenti in sottoinsiemi della stessa dimensione.

Ad esempio nella Coppa Italia del Tennis, le squadre di serie D svolgono il primo turno di gioco (Prima Fase Provinciale) in un girone di M squadre in base alla distanza (intesa come tempo di percorrenza in auto) dei circoli di appartenenza. L'obiettivo è quello di minimizzare la distanza (es. massima o somma) tra due circoli all'interno dei gironi creati.

La soluzione al problema presentato può essere calcolata come soluzione di un modello di Programmazione Lineare Intera oppure con un'euristica (es. una variante degli algoritmi di clustering presentati durante il corso). Si prenda spunto dall'articolo di Bradley et al. sul Constrained K-Means Clustering.

- Creare uno script `CreaGironi`, che legge un dataset con le squadre e le coordinate geografiche del proprio circolo e il criterio di scheduling `criterion` $\in$ {`MINIMAX, MINSQSUM`}, producendo un file `csv` con i gironi.
- Confrontare le soluzioni ottenute con quelle della FederTennis riportando le distanze massime e la somma delle distanze al quadrato per i due criteri e il caso reale.

6 Scoperta di community per individuare comportamenti fraudolenti delle giurie all'Eurovision Song Contest

Informazioni generali

- Team di 2 persone
- Linguaggio di programmazione a piacere
- Approfondimento su partizionamento di grafi e ricerca di community: *Avrim Blum, John Hopcroft, and Ravindran (2020). Foundations of Data Science. Cambridge University Press. Sez. 7.10 e 7.11* [PDF]
- Dati disponibili al link GitHub

Descrizione

L'Eurovision Song Contest è l'evento non sportivo più seguito al mondo. Durante l'ultima edizione, che si è tenuta a Torino, l'European Broadcasting Union (EBU) ha annullato i voti di 6 Paesi (Azerbaigian, Georgia, Montenegro, Polonia, Romania e San Marino) dopo aver scoperto un pattern nei voti delle giurie della Semifinal 2, dove ognuno dei 6 paesi ha posto gli altri 5 nelle primissime posizioni della propria classifica.

Implementare un algoritmo di clustering per la ricerca di community per determinare comportamenti anomali nei voti delle giurie nelle semifinali e le finali delle edizioni tra il 2016 e il 2022 (N.B. la competizione non si è tenuta nel 2020; inoltre si considerino i voti reali per il 2022). Si considero solo cluster di dimensione maggiore o uguale a 3.

Ulteriori dettagli sul meccanismo di voto delle giurie a questo link.

7 Scheduling online basato sull'apprendimento dei pesi

Informazioni generali

- Team di 2 persone
- Python
- Approfondimento: *Silvio Lattanzi, Thomas Lavastiday, Benjamin Moseleyy, Sergei Vassilvitskii (2020). Online Scheduling via Learned Weights. Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms (SODA)* [PDF]
- Dati da generare artificialmente

Descrizione

Gli algoritmi di ottimizzazione online sono dei particolari algoritmi in cui si devono prendere delle decisioni (es. sullo scheduling di alcuni lavori) con una conoscenza dell'istanza che avviene solo nel tempo. Spesso essi vengono valutati sulla base della loro performance nel caso peggiore, ma nel mondo reale le istanze sono spesso lontane dal peggiore dei casi e alcune caratteristiche possono essere apprese con delle tecniche di machine learning. Lattanzi et al. integrano le informazioni fornite da modelli predittivi all'interno di un algoritmo di scheduling online.

Presentare i risultati teorici dell'articolo di Lattanzi et al., insieme ai due algoritmi implementati in Python e testati su diverse istanze generate in modo casuale.