CASO DI STUDIO DEL CORSO DI "SISTEMI AD AGENTI" a.a. 2017/2018 – UNIVERSITA' DEGLI STUDI DI BARI "ALDO MORO"

DOCUMENTAZIONE DI PROGETTO

Colagrande Pierpasquale – matricola 651504

Monaco Daniele – matricola 654019

Debellis Rocco – matricola 655530

INDIVIDUAZIONE DELLE SIMILARITA' FRA UTENTI IN BASE ALLE LORO CRONOLOGIA DELLE POSIZIONI

Sommario

1 Introduzione

Con l'aumentare della sofisticatezza dei sistemi di tracciamento della posizione (si veda ad esempio la tecnologia GPS) e grazie all'inserimento di tali sistemi di geolocalizzazione all'interno di dispositivi di uso quotidiano e comune (smartphone, smartwatch, notebook ecc.), ogni giorno vengono prodotte enormi quantità di dati relativi ai movimenti effettuati da coloro che utilizzano tali strumenti.

Le informazioni contenute in questi dati (coordinate GPS, altitudine, timestamp ecc) risultano essere molto utili ai fini commerciali di un'impresa in quanto lo studio e l'analisi di tali dati consente l'implementazione di algoritmi atti a sfruttare tali dati per aumentare le capacità di monetizzazione dell'impresa stessa. In che modo? Facciamo qualche esempio.

Da molto tempo ormai, aziende come Google, Facebook, Twitter, ecc… suggeriscono, agli utilizzatori dei loro servizi, attività, locali, ristoranti e posti che sono simili a quelli già visitati dall'utente. In questo modo, oltre a

produrre un vantaggio per l'utente stesso che può vedersi suggerita un'attività che a lui può interessare, l'azienda può ricavare denaro da tale suggerimento grazie ad accordi commerciali fra l'impresa che gestisce il servizio stesso e il locale, ristorante o attività. Tale utilizzo delle posizioni è ormai attivo da molto tempo.

Un altro possibile utilizzo a fini commerciali di tali dati può essere quello di suggerire ad un utente posti, luoghi, ristoranti, eventi eccetera simili a quelli visitati da altri utenti che, al livello di posizioni e di movimenti effettuati, risultano essere simili all'utente stesso. Questo è proprio il problema che è stato affrontato e risolto con questo caso di studio.

2 Analisi

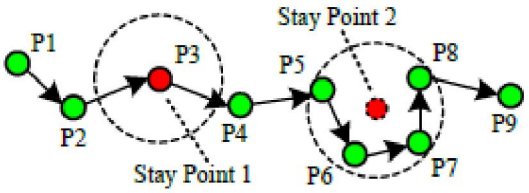
Il sistema utilizza sequenze di coordinate GPS, correlate ad utenti specifici, per calcolare le similarità fra i vari utenti.

Tale lavoro di calcolo delle similarità è stato già sviluppato da un gruppo di ricercatori cinesi per conto di Microsoft nel 2008[1]. Tale ricerca faceva parte di un progetto più grosso, denominato "GeoLife"[2], un social network dove le amicizie e le attività venivano suggerite in base alle posizioni.

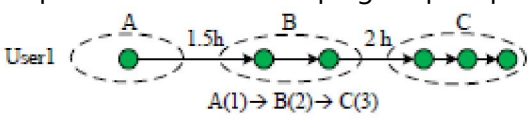
Per realizzare quindi tale strumento di estrazione delle similarità, ci si è basati sul lavoro di ricerca indicato in precedenza. Tuttavia, poiché tale paper risulta essere ormai datato, sono stati effettuati dei miglioramenti nei vari algoritmi utilizzati applicando anche algoritmi più recenti.

Il sistema si suddivide essenzialmente in cinque parti:

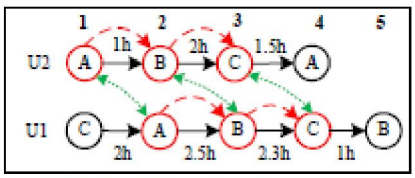
1. La prima parte estrae i dati memorizzati in un dataset ed effettua il parsing di tali dati affinché possano essere utilizzati a fini analitici
2. La seconda parte del sistema, invece, estrae i cosiddetti staypoints: uno staypoint non è nient'altro che una coppia di coordinate GPS (latitudine, longitudine) corrispondenti ad una posizione della superficie terrestre dove un utente ha sostato per molto tempo oppure, alternativamente, attorno al quale l'utente ha girato in un certo periodo di tempo. Gli staypoints vengono calcolati per tutti gli utenti ed inseriti in un unico dataset. All'interno del paper indicato[1] è spiegato un algoritmo di individuazione degli staypoints che è stato utilizzato nel progetto.



3. La terza parte del sistema utilizza un algoritmo per clusterizzare gli staypoints estratti in precedenza. L'algoritmo utilizzato nel paper di ricerca era OPTICS, un algoritmo di clustering gerarchico realizzato nel 1999 che migliora il precedente algoritmo DBSCAN. OPTICS utilizza due parametri, il numero minimo di punti che devono far parte di un cluster e il raggio massimo di grandezza di un cluster. Tale algoritmo è stato sostituito da HDBSCAN[3], un algoritmo di clustering gerarchico realizzato nel 2015 come miglioramento a OPTICS, che utilizza come parametro solo il numero minimo di punti necessari alla realizzazione del cluster. Vengono quindi creati dei cluster contenenti i vari staypoints.
4. La quarta parte estrae per ogni utente una sequenza di cluster. Tale sequenza non è nient'altro che un grafo dove ogni nodo è un cluster ed ogni arco indica il tempo che l'utente ha impiegato per spostarsi da un cluster all'altro. Ricordiamo che i cluster contengono staypoints, per cui il sistema analizza i vari staypoint per ogni utente e va a vedere a che cluster tale staypoint appartiene, costruendo nel mentre la sequenza.



5. La quinta parte del sistema va a calcolare la similarità fra le varie sequenze estratte in precedenza, determinando così le similarità fra i vari utenti. Per calcolare le similarità, il sistema tiene conto dei vari



C) Finding *3-length* similar sequences

cluster in comune fra le sequenze che si stanno paragonando, delle tempistiche impiegate per lo spostamento da un cluster all'altro e del numero di staypoints dei vari utenti presenti nei vari cluster delle sequenze e della lunghezza massima delle sequenze simili da individuare.

Il risultato fornito dal sistema è una matrice di dimensione n*n (dove n è il numero di utenti), contenente le similarità fra gli utenti stessi.

Il sistema è inoltre stato dotato di un'interfaccia grafica che consente di settare i vari parametri delle operazioni di clustering, individuazione di staypoint, estrazione e confronto di sequenze e consente di visualizzare su una mappa gli utenti (e quindi i percorsi) degli utenti simili ad un utente specificato.

I miglioramenti effettuati quindi consistono nell'aver sostituito l'algoritmo di clustering con un algoritmo più nuovo, preciso, veloce ed efficace e con l'aver semplificato alcune operazioni, oltre ad aver aggiunto una interfaccia grafica per visualizzare i vari dati.

La distanza utilizzata per clusterizzare i punti e per individuare gli staypoint è la distanza di haversine.

3 L'applicazione

Per sviluppare il sistema, sono stati adottati i seguenti software/librerie:

- **Git/GitHub** per il controllo di versione
- **Python 3** come linguaggio di programmazione
- **PyCharm/Spyder** come IDE per sviluppare il sistema con Python
- **numpy** per la gestione dei dati (array n-dimensionali)
- **hdbscan**, facente parte di scikit-learn
- **kivy** per la realizzazione della grafica
- **kivymd**, estensione di kivy che implementa il Material Design di google
- **cefpython** per la visualizzazione di una pagina HTML contenente la mappa
- **folium** per la creazione di una pagina HTML contenente una mappa interattiva

Il dataset usato nel programma è quello di GeoLife offerto sul sito web della Microsoft.[4]

Il sistema è stato sviluppato creando delle classi per le entità fondamentali (punto, utente, traiettoria, staypoint, sequenza, nodo) e delle classi per la gestione di entità di contorno (coordinate).

É stata anche aggiunta la possibilità di cambiare il tema (chiaro/scuro) dell'applicazione ed il colore principale dell'applicazione.

4 Moduli

- Modulo **clustering.py** per la gestione delle operazioni di clustering
- Modulo **dataset_parser.py** per la gestione del parsing del dataset
- Modulo **point_utilities.py** per la gestione dei punti (calcolo delle distanze)
- Modulo **sequence_manager.py** per la gestione delle sequenze (creazione) e l'estrazione delle similarità fra le varie sequenze
- Modulo **staypoint_detector.py** per l'estrazione e la gestione degli staypoints
- Modulo **entities.py** per la gestione delle classi entity
- Modulo **sides.py** per la gestione delle classi di contorno
- Modulo **gui.py** per la gestione dell'UI
- Modulo **map_manager.py** per la gestione della mappa
- Modulo **main.py** che contiene il main del programma (l'avvio del software)

5 Conclusioni

Il sistema, anche se molto simile a quello presentato dai ricercatori cinesi, è stato molto migliorato con l'aggiunta dell'algoritmo di clustering HDBSCAN in sostituzione di OPTICS.

Sviluppi futuri possono essere l'aggiunta del multithreading, di cython e dell'esecuzione di istruzioni sulla scheda video al fine di velocizzare operazioni quali quelle di staypoint detection, sequence extraction e sequence matching. Infatti, tali operazioni sono le più lente e quelle che richiedono più tempo, mentre l'algoritmo di clustering hdbscan, rispetto a tali operazioni, risulta essere di gran lunga più veloce in quanto già implementato in scikit-learn e, per tanto, già sfruttante i miglioramenti descritti immediatamente qui sopra.

Questo sistema può essere il nucleo di applicazioni di social networking basate su posizione, che possono quindi suggerire possibili amicizie suggerendo utenti simili all'utilizzatore, luoghi, posti e qualsiasi posizione geografica simile a quelle già visitate dall'utente. Può inoltre essere inserito all'interno di un'applicazione che sfrutta l'utilizzo della posizione (ad esempio, banalmente, un semplice navigatore o social network) per estrarre informazioni preziose riguardo gli interessi della gente che possono poi essere sfruttate da altre aziende di advertising oppure da multinazionali a fini commerciali.

Lo svantaggio di un sistema basato su tale algoritmo è che, affinché le raccomandazioni e l'estrazione delle similarità siano efficienti al massimo, è necessario avere a disposizione una grande quantità di dati. Questo può essere un problema durante la fase di sviluppo in quanto è necessario l'utilizzo di un dataset per testare il sistema; tuttavia per un utilizzo reale del sistema non ci dovrebbero essere problemi di carenza di dati, considerando la velocità con cui le informazioni vengono prodotte nell'epoca moderna.

6 Riferimenti

[1] Mining user similarities based on location history: https://www.microsoft.com/en-us/research/publication/mining-user-similarity-based-on-location-history/

[2] GeoLife project: https://www.microsoft.com/en-us/research/project/geolife-building-social-networks-using-human-location-history

[3] HDBSCAN: https://github.com/scikit-learn-contrib/hdbscan

[4] GeoLife Dataset: https://www.microsoft.com/en-us/download/details.aspx?id=52367