ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA
CAMPUS DI CESENA

Scuola di Ingegneria e Architettura
Corso di Laurea Magistrale in Ingegneria e Scienze Informatiche

UTILIZZO DI DATI SOCIAL PER LA DEANONIMIZZAZIONE DI TRACCE GPS

Tesi in

DATA MINING

Relatore

Prof. MATTEO GOLFARELLI

Presentata da

NICOLA SANTOLINI

Co-relatore

Dott. ENRICO GALLINUCCI

Dott. MATTEO FRANCIA

Terza Sessione di Laurea
Anno Accademico 2018 – 2019

PAROLE CHIAVE

Trajectory Mining

De-anonimizzazione

Privacy

Social Network

Ai miei genitori, Mauro e Milena, che con i loro sacrifici hanno reso possibile il mio percorso. A Beatrice, per tutto ciò che significa per me.

Indice

Introduzione

Negli ultimi anni la diffusione massiva e pervasiva dei dispositivi di localizzazione ha portato a un aumento esponenziale della produzione di dati di traiettoria. Anche l'ascesa dell'*Internet of Things* ha contribuito a questo processo, promuovendo la diffusione di tecnologie come GPS, RFID e beacon. I dispositivi di localizzazione possono essere utilizzati per tracciare il movimento di animali, oggetti e perfino fenomeni naturali, accrescendo ulteriormente lo spettro delle sorgenti di dati di traiettoria. Un numero crescente di applicazioni, in maniera più o meno diretta, raccoglie e memorizza dati relativi al posizionamento degli utenti che le utilizzano. Questi dati racchiudono un potere informativo enorme, che da un lato ha stimolato un forte sviluppo delle tecniche di analisi (in particolare si parla di trajectory data mining) ma dall'altro ha portato a una maggiore esposizione della privacy dei soggetti che producono i dati stessi. Attraverso un'analisi approfondita degli spostamenti di un soggetto è infatti possibile dedurre informazioni relative a comportamenti, abitudini e preferenze. Queste informazioni possono essere utilizzate per molteplici scopi, ad esempio fornire servizi personalizzati agli utenti di specifiche applicazioni, sistemi di *recommendation*, o per finalità commerciali. I dati di traiettoria hanno quindi anche un rilevante valore economico, ed è ormai noto che molte compagnie monetizzino dalla loro cessione: ciò ha attribuito ulteriore rilevanza al dibattito sulla privacy degli utenti.

In letteratura sono presenti studi relativi alle possibili tecniche di protezione della privacy degli utenti, applicate anche ai dati di traiettoria, che comportano però nella maggior parte dei casi un forte deterioramento del potere informativo degli stessi. Spesso si tende ad anonimizzare i dati, sostituendo l'identità

e i riferimenti degli utenti con identificatori randomici. Molte ricerche dimostrano però come la privacy degli utenti, anche in questi casi, non sia del tutto inattaccabile. Sono state presentate numerose tecniche di de-anonimizzazione, volte non solo a distinguere gli utenti all'interno di gruppi anonimi, ma ad attribuire vere e proprie identità ai soggetti trovando collegamenti con dati presenti in altre sorgenti. Gli studi evidenziano quindi la possibilità di violare la privacy degli utenti che producono i dati di traiettoria, trovando ad esempio collegamenti con profili personali di social network come Twitter o Facebook. Questo lavoro di tesi in particolare è dedicato allo studio approfondito della letteratura riguardante le tecniche di de-anonimizzazione e all'applicazione concreta di algoritmi su un ampio dataset di traiettorie raccolte utilizzando dispositivi GPS.

La tesi è strutturata come segue. Nel Capitolo 1 vengono introdotte le tematiche relative ai dati di traiettoria e alle tecniche di mining che possono essere applicate su di essi; viene inoltre fatto riferimento alla problematica della tutela della privacy degli utenti, alle possibili tecniche di protezione e all'attacco di de-anonimizzazione. Il Capitolo 2 racchiude un'analisi estensiva dello stato dell'arte per quanto concerne le tecniche di de-anonimizzazione, in particolare su quelle progettate per i dati di traiettoria. Il Capitolo 3 contiene una presentazione delle principali tecnologie utilizzate durante lo svolgimento della tesi. Nel Capitolo 4 viene esposto nel dettaglio il contributo apportato con questo lavoro di tesi, ponendo l'accento sugli algoritmi implementati e sulla formalizzazione di nuova tecnica di de-anonimizzazione. Infine, nel Capitolo 5 vengono presentati i dati oggetto di studio e discussi i test effettuati e i risultati ottenuti, fornendo un confronto tra le diverse tecniche implementate.

Capitolo 1

Descrizione del problema

In questo capitolo viene presentato il problema oggetto di studio, partendo da una esposizione generale delle tecniche di *mining* applicabili ai dati di traiettoria per concludere con una trattazione del caso specifico della de-anonimizzazione.

1.1 Dati di traiettoria

I servizi di posizionamento possono essere forniti da diversi tipi di tecnologie, ad esempio:

- satellitari, come il sistema di posizionamento globale (GPS) statunitense, che negli anni ha visto l'affiancamento del russo *GLONASS* e dell'europeo Galileo;
- *Radio Frequency Identification (RFID)*, ossia dispositivi a corto raggio e basso consumo spesso utilizzati in condizioni *indoor*, ovvero al chiuso, utili in contesti in cui le tecnologie satellitari sono inadatte o inapplicabili;
- sensori su smartphone, spesso collegati a tecnologie satellitari;
- *beacon*, ossia dispositivi a basso consumo basati su tecnologia *Bluetooth Low Energy*;

- stima di posizione basata sul collegamento alla rete Internet.

Lo sviluppo e la diffusione di queste tecnologie di acquisizione della posizione hanno portato alla generazione di notevoli quantità di dati di traiettoria. La traccia di un oggetto in movimento nello spazio geografico è continua, mentre la traiettoria consiste in un campionamento temporale di locazioni che l'oggetto occupa. La più semplice rappresentazione per dati di traiettoria consiste in una sequenza di locazioni associate a una indicazione temporale (tipicamente un *timestamp*); la durata e la frequenza di campionamento dipendono dall'applicazione. Ad esempio un sistema di navigazione stradale richiede una frequenza di campionamento elevata, mentre uno di monitoraggio può tollerare rilevazioni meno frequenti ma eseguite costantemente per un periodo di tempo continuato. Un esempio grafico è visibile in figura 1.1, presente in [16]. Più precisamente una traiettoria può essere definita come segue:

Definizione 1. *Traccia discreta di un oggetto in movimento nello spazio geografico, costituita da una sequenza di geo-locazioni associate alla rispettiva indicazione temporale, ad esempio* $\tau = < p_1, t_1 >, < p_2, t_2 >, ..., < p_n, t_n >$*, dove ogni elemento* $< p_i, t_i >$ *indica il posizionamento di un oggetto nella locazione* p_i *all'istante* t_i*.*

L'oggetto in movimento di cui si osserva la traiettoria può consistere in un essere umano, un animale, un oggetto o un fenomeno naturale. Una locazione è solitamente espressa come una tupla $<$ *latitudine, longitudine* $>$ corrispondente a un punto specifico nello spazio geografico e registrata utilizzando, ad esempio, un dispositivo GPS. Una particolare tipologia di dati di traiettoria è quella legata alla tecnologia RFID: in questo caso gli oggetti mobili corrispondono a device dotati di tag che emettono segnali radio di identificazione. Questi segnali vengono rilevati da altri dispositivi, detti *reader*, posizionati in locazioni specifiche: un tipico caso d'uso di questa tecnologia è quello del tracciamento di oggetti all'interno di magazzini o fabbriche. In questo caso la localizzazione avviene tramite il rilevamento di un tag da parte di un reader, e la traiettoria di un oggetto consiste sostanzialmente in una serie di aree geografiche corrispondenti ai reader che ne hanno rilevato il segnale.

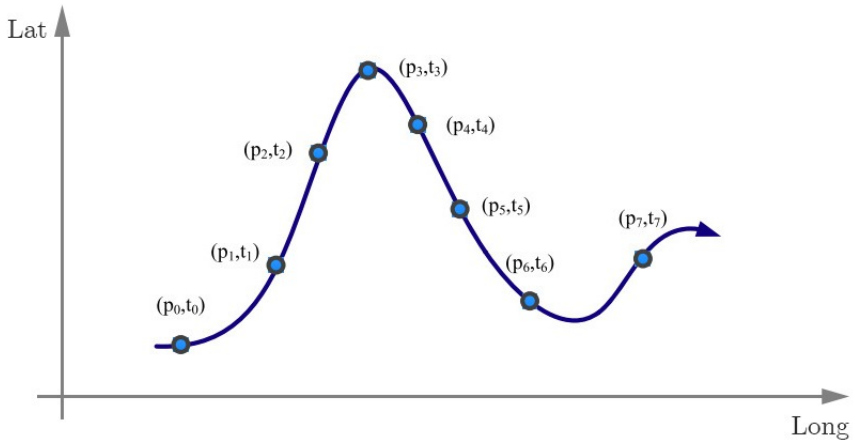


Figura 1.1: Rappresentazione di una traiettoria generata da una traccia continua, ogni punto campionato è caratterizzato dalla locazione p_i e dal riferimento temporale t_i.

A prescindere dal tipo di tecnologia utilizzata, i possibili scopi e applicazioni che possono beneficiare delle tecniche di mining applicate ai dati di traiettoria sono numerosi e possono rappresentare un valore sia per l'individuo che per organizzazioni pubbliche e private. La gestione ed elaborazione di dati di traiettoria non sono però prive di sfide da affrontare. Ci sono infatti alcuni importanti aspetti da gestire e tenere in considerazione:

- gestire una enorme quantità di dati di traiettoria può comportare difficoltà di memorizzazione, soprattutto quando il volume diventa inquadrabile in un contesto di *Big Data*;

- la velocità con cui i dati vengono generati può costituire un problema rilevante soprattutto se l'elaborazione deve essere eseguita in *real-time* o su uno *streaming*;

- è difficile stabilire appropriate metriche di similarità per confrontare tra loro traiettorie, dato che queste possono essere generate con strategie e frequenze di campionamento diverse;

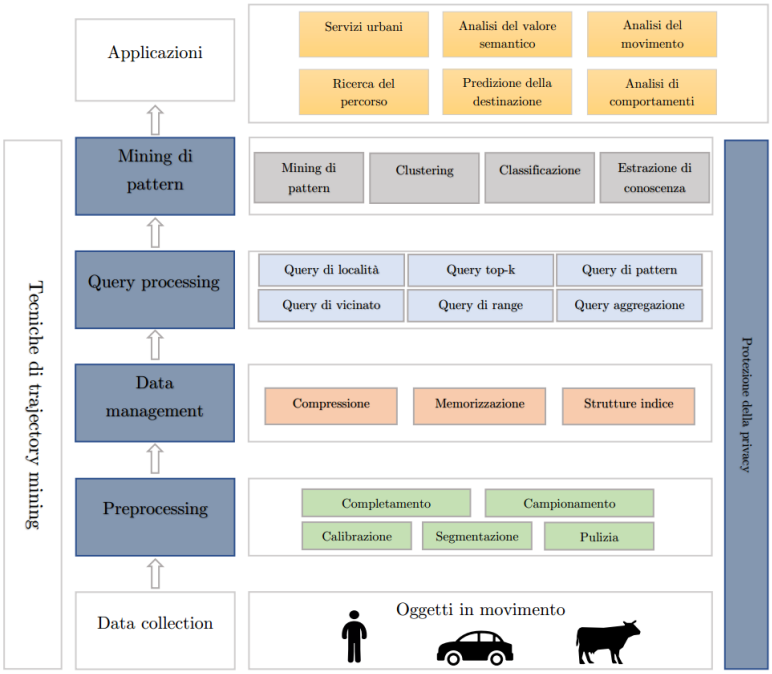


Figura 1.2: Processo complessivo di trajectory data mining.

- effettuare query su grandi quantità di dati di traiettoria comporta difficoltà in termini di complessità e tempo di esecuzione, soprattutto se non si sfruttano strutture idonee e strumenti scalabili.

Affrontare queste problematiche richiede spesso l'utilizzo di un processo ben strutturato che si compone di passaggi precisi, classificabili in tre macrolivelli [14]: raccolta dati, elaborazione con tecniche di *trajectory data mining*, applicativo. La composizione del processo è mostrata in 1.2. In particolare il livello di applicazione di tecniche di mining è a sua volta strutturato in diverse fasi:

- *preprocessing*, durante il quale le traiettorie vengono sottoposte ad operazioni di pulizia, segmentazione, calibrazione, campionamento;

- *data management*, in cui le traiettorie vengono eventualmente compresse o semplificate prima della memorizzazione; inoltre, per favorire una memorizzazione efficiente e la scalabilità delle traiettorie, vengono costruite

appropriate strutture di supporto (ad esempio indici), che diventeranno utili anche in fase di analisi;

- *query processing*, che comprende tutta la gamma di possibili interrogazioni effettuabili sulle traiettorie per estrarre dati ed informazioni: le query possono costituire i componenti base che i *task* di alto livello combinano e concatenano per eseguire elaborazioni complesse, ad esempio un task di *clustering* può richiedere l'esecuzione di query di vicinato;

- task di trajectory data mining, suddivisibili in diverse categorie come classificazione, clustering, mining di pattern, de-anonimizzazione; come nel data mining tradizionale, queste operazioni possono essere supervisionate o non supervisionate, con l'obiettivo di far emergere informazioni latenti nei dati o di rappresentarli in forme maggiormente significative;

- protezione della privacy, che costituisce un problema cruciale e trasversale a tutte le fasi citate: ogni operazione di data mining effettuata su traiettorie deve considerare adeguatamente la tutela dei dati sensibili degli utenti.

1.2 Trajectory mining

In questa sezione vengono brevemente trattati i principali ambiti di utilizzo delle tecniche di data mining applicate a dati di traiettoria.

Ricerca del percorso Si tratta di uno degli scenari applicativi più comuni, dove l'obiettivo è quello di trovare il percorso migliore tra due punti, dipendentemente dalla specifica applicazione; può ad esempio trattarsi di individuare il percorso più veloce, corto o frequente. Una frequente concretizzazione di questo scenario è il *route discovery*, in cui si cerca di trovare almeno un percorso tra un'origine e una destinazione che rispetti certi vincoli o caratteristiche. Solitamente strade e località vengono estratte da specifiche rappresentazioni della rete stradale (*road network*) per costruire i percorsi candidati.

Predizione di locazione o destinazione I cosiddetti servizi *location based (LBS)* o *location-aware* sono sempre più diffusi, e spesso si basano sulla predizione della prossima locazione o della destinazione finale degli utenti, ad esempio per fini commerciali, pubblicitari, di navigazione. Molti studi hanno evidenziato la forte regolarità e predicibilità della mobilità umana, rafforzando la popolarità di questo tipo di applicazioni. Si tratta di uno scenario strettamente legato a quello della ricerca del percorso. Se un percorso relativo a un viaggio in corso risulta molto simile o in match esatto con parte di uno presente in un dataset di percorsi frequenti, è possibile stimare con una certa confidenza che la destinazione finale sia la stessa.

Analisi del movimento e del comportamento L'analisi dei comportamenti in termini di movimento e l'individuazione di eventuali pattern in questo senso può rivelarsi fondamentale per la comprensione del comportamento umano. Un aspetto particolarmente impegnativo è quello legato all'estrazione di indicazioni semantiche di alto livello, che permettano ad esempio di inferire scopi e ruoli degli oggetti in movimento. Un altro campo di ricerca molto importante è quello relativo alla predizione del comportamento umano in situazioni di emergenza, molto utile per scopi come la gestione di situazioni critiche e disastri naturali e le successive fasi di *recovery* e ricostruzione.

Analisi del comportamento di gruppo Gli oggetti in movimento, soprattutto se si tratta di persone o animali, tendono a formare gruppi o a organizzarsi in cluster a seconda dei propri comportamenti sociali. In questo senso i pattern di incontro (*gathering pattern*) modellano in particolare flussi e movimenti collettivi legati a gruppi di oggetti, ad esempio celebrazioni, parate, proteste, ingorghi stradali.

Servizi urbani La conoscenza che può essere estratta con le tecniche di data mining applicate ai dati di traiettoria può aiutare a migliorare la qualità della vita nelle aree urbane sotto diversi punti di vista. Le possibili applicazioni pratiche sono svariate, ad esempio:

- elaborazione della corretta distribuzione di risorse specifiche sul territorio (stazioni di ricarica/rifornimento, punti di raccolta, ecc.);
- sviluppo e progettazione della rete stradale basato sui dati relativi al traffico effettivo;
- programmazione dello sviluppo urbanistico e delle politiche ambientali a vantaggio di cittadini ed enti amministrativi.

Traiettorie e valore semantico Spesso i dati di traiettoria in forma grezza, espressi come sequenze di locazioni geografiche corredate da timestamp, non riescono ad avere un significato chiaramente interpretabile senza una descrizione semantica. In questo senso molti studi si stanno dedicando ad approcci per la generazione automatica di brevi descrizioni testuali che risultino intellegibili e *human-readable* a partire dai dati di traiettoria. Inoltre nel contesto in forte crescita degli applicativi LBS sono spesso richieste informazioni semantiche legate alle varie location e punti di interesse, ad esempio indicazioni come "luogo di lavoro" o "residenza" per arricchire le semplici coordinate espresse in termini di latitudine e longitudine. Questo tipo di informazioni in particolare svolge un ruolo fondamentale nell'approccio alla de-anonimizzazione presentato con questo lavoro.

1.3 Privacy e de-anonimizzazione

Quando si opera con dati di traiettoria, non solo in ambito di data mining, un annoso e difficile problema è quello della tutela della privacy dei soggetti coinvolti. L'utilizzo di dati riguardanti gli spostamenti di soggetti, ad esempio applicando tecniche e algoritmi di data mining per estrarre informazioni e conoscenza, comporta il rischio di ledere la privacy di questi ultimi. Come esposto in precedenza, le tecnologie in grado di effettuare la localizzazione di persone e cose sono sempre più diffuse, così come gli applicativi che ne fanno uso e che concretamente raccolgono e memorizzano dati. Sono molteplici le modalità con cui una persona può quotidianamente essere localizzata, con

conseguente generazione e memorizzazione di dati che possono poi andare a costituire delle traiettorie. Per citare alcuni semplici esempi:

- posizionamenti registrati dagli *Internet Service Provider (ISP)* tramite le celle agganciate dai telefoni cellulari, ad esempio quando si utilizzano per effettuare chiamate, inviare messaggi o connettersi alla rete;
- posizionamenti raccolti tramite dispositivi di localizzazione (tipicamente di tipo GPS) montati su smartphone, a cui può essere permesso l'accesso anche ad applicazioni di terze parti;
- localizzazioni effettuate tramite check-in o tag condivisi utilizzando i social network.

In molti casi, una volta concessa l'autorizzazione da parte dell'utente a raccogliere i dati, questi possono essere memorizzati, ad esempio dal provider o da un'applicazione a cui ne è stato permesso l'utilizzo. Tali soggetti, a seconda delle casistiche e nei limiti normativi, possono anche monetizzare su questi dati, cedendoli a terzi. Quando si tratta di dati sensibili, per poter essere condivisi o ceduti, questi devono essere trattati affinché non possano più essere in alcun modo ricondotti in maniera diretta all'identità del soggetto che li ha generati, per preservarne la privacy.

Le tecniche esistenti per la protezione della privacy (applicabili, tra gli altri, anche ai dati di traiettoria) si possono classificare in due categorie: distorsione e pseudonimi. La prima tipologia è costituita da tecniche che, attraverso vari metodi, cercano di nascondere l'identità dei soggetti distorcendo i dati di traiettoria, aggiungendo rumore o riducendo la risoluzione delle traiettorie stesse, cercando di mascherare i precisi riferimenti spaziali e/o temporali. Di contro, questo tipo di tecniche basate sulla distorsione possono seriamente intaccare ed impattare l'integrità, l'utilità e la disponibilità dei dati, rischiando di comprometterne il potere informativo. Diversamente le tecniche basate su pseudonimi operano sostituendo all'effettiva identità dei soggetti o degli oggetti degli identificatori randomici univoci e consistenti. In questo caso risulta fondamentale che a partire dallo pseudonimo non sia possibile risalire tramite

nessun tipo di trasformazione o processo inverso all'identità originale del soggetto da cui ha avuto origine l'identificatore. Dato che sono facili da generare e che non richiedono di modificare i dati spaziotemporali originali, gli pseudonimi vengono largamente utilizzati quando si pubblicano dataset contenenti dati di traiettoria. Nonostante pseudonimi e vere identità dei soggetti non siano direttamente collegabili l'efficacia di queste tecniche di anonimizzazione è comunque questionabile. Molti studi hanno dimostrato come la regolarità e ripetitività, in particolare dei comportamenti umani, possano permettere di individuare pattern ricorrenti che possano fungere da semi-identificatori, o come anche all'interno di grandi dataset sia sufficiente un numero limitato di punti per identificare univocamente la maggior parte delle traiettorie. In molti casi si dimostra quindi insufficiente la sola rimozione di dati identificativi (nome, cognome, data di nascita, numero di telefono, ...) e la conseguente sostituzione con identificatori randomici. Esistono infatti tecniche in grado di re-identificare gli utenti facendo emergere le caratteristiche uniche dei soggetti anonimi o cercando corrispondenze con altri dati disponibili allo scopo di trovare un collegamento con identità note.

1.3.1 Tecniche di anonimizzazione

In letteratura sono presenti numerose pubblicazioni riguardanti tecniche e approcci utilizzabili per preservare l'anonimato degli individui legati a dati, non necessariamente di traiettoria, quando questi diventano accessibili ad altri ad esempio perché pubblicati a fini di ricerca. L'obiettivo principale, oltre a rendere impossibile l'identificazione dei soggetti osservati, è quello di riuscire a farlo senza alterare o deteriorare la qualità dei dati al punto da ridurne o inficiarne gravemente il potere informativo.

k-Anonimity Tra gli approcci più noti quello storicamente più datato è *k-Anonimity* [1]. Viene qui definita la proprietà di k-anonimia, che viene rispettata qualora ogni individuo di cui sono presenti dati non sia distinguibile da almeno altri k-1 soggetti all'interno della collezione di dati. Affinché i dati

ottengano questa proprietà vengono solitamente eseguite su di essi operazioni di due tipi:

- soppressione: i valori di certi attributi vengono sostituiti con altri predefiniti, ad esempio un '*', allo scopo di nascondere totalmente l'informazione sottostante con elementi non significativi;
- generalizzazione: i singoli valori di alcuni attributi vengo sostituiti con altri più generali, ad esempio indicando valori di intervallo al posto di singoli valori numerici o mantenendo solo prefissi comuni ed eliminando i caratteri distintivi all'interno di codici alfanumerici.

Una volta applicate queste trasformazioni i dati devono presentare gruppi di almeno k-1 record tra loro indistinguibili. Risulta evidente come questa tecnica possa portare a una notevole alterazione dei dati e ad una eventuale perdita di potere informativo.

l-Diversity Un altro approccio noto e proposto come estensione di *k-Anonimity* è *l-Diversity* [2]. In particolare questo metodo si propone di offrire una maggiore protezione per l'anonimato dei soggetti da attacchi che sfruttano dati esterni per la de-anonimizzazione. La proprietà di l-diversità risulta verificata su blocchi di record k-anonimi quando per gli attributi sensibili sono presenti almeno l valori definiti "ben rappresentati" all'interno del gruppo. Questa caratteristica viene introdotta per evitare l'eccessiva omogeneità dei valori di attributi sensibili all'interno della classe di equivalenza, che rende maggiormente vulnerabile la privacy degli utenti. La definizione di questi valori altamente rappresentativi può essere eseguita basandosi su tre criteri: l valori distinti, entropia, ricorsione.

t-Closeness Un ulteriore metodo largamente studiato nel campo dell'anonimizzazione dei dati è *t-Closeness* [3]. L'approccio si propone come ulteriore raffinamento, con un rafforzamento della protezione dell'anonimato, di *l-Diversity*. La proprietà di t-vicinanza richiede che la distribuzione dei valori di un attributo sensibile all'interno della singola classe di equivalenza o

gruppo sia vicina alla distribuzione dell'attributo su tutti i dati. Il concetto di vicinanza viene concretizzato tramite la definizione di una soglia che non deve essere superata dalla distanza tra le due distribuzioni.

Differential privacy Per completezza si cita anche la *differential privacy* [14], approccio di tipo statistico che alcuni studi hanno applicato a dati di traiettoria per valutarne le performance [15]. L'intuizione alla base di questo metodo è quella di aggiungere rumore in modo controllato ai dati, ad esempio a partire da una distribuzione di Laplace. Negli ultimi anni questo tipo di approccio è stato oggetto di studio e ricerca anche da parte di aziende leader come Microsoft, Apple e Google.

1.3.2 Attacco di de-anonimizzazione

Le osservazioni precedenti evidenziano come l'utilizzo di tecniche di anonimizzazione, ad esempio tramite l'applicazione di pseudonimi, non renda completamente inattaccabile la privacy dei soggetti che hanno generato dati di traiettoria. Nello specifico, in letteratura sono numerosi gli studi riguardanti attacchi di de-anonimizzazione effettuati contro dataset di traiettorie anonime. Questo tipo di analisi non è mirata solamente ad evidenziare quanto una traccia, seppur anonimizzata, possa risultare unica e distinguibile dalle altre, ma a cercare una vera e propria corrispondenza tra le tracce e identità di soggetti. Molto spesso quello che si cerca di ottenere è il collegamento tra traiettorie anonime contenute in un dataset (che costituisce il bersaglio dell'attacco) e traiettorie/profili non anonimi contenuti in un secondo dataset, spesso definito come *external dataset* o *side information*. Queste informazioni esterne possono essere già disponibili in forma di traiettorie oppure consistere in dati di altra natura a disposizione dell'attaccante che cerca di de-anonimizzare le tracce anonime. In questo caso sono necessarie operazioni aggiuntive per estrarre e costruire le traiettorie non anonime a partire dai dati disponibili. Nel capitolo seguente vengono esposti nello specifico tecniche e approcci noti in letteratura per quanto riguarda la de-anonimizzazione di tracce anonime.

Capitolo 2

Stato dell'arte

In letteratura sono numerosi gli studi volti a valutare l'efficacia delle tecniche di anonimizzazione applicabili ai dati di traiettoria e le proposte di approcci e metodi per la deanonimizzazione di tali dati. Nella maggior parte dei casi lo scenario è quello di un attaccante che intende de-anonimizzare il maggior numero possibile di traiettorie rese anonime (ad esempio utilizzando pseudonimi) presenti in un dataset sfruttando informazioni esterne. Tipicamente queste informazioni consistono in altre traiettorie di soggetti noti o dati in qualche modo riconducibili a profili di utenti o identità specifiche. Volendo classificare i vari approcci alla de-anonimizzazione, un possibile criterio di suddivisione è quello del tipo di informazioni esterne che vengono sfruttate:

- contenuto: informazioni riguardanti attività dell'utente (ad esempio posizioni spaziali corredate da indicazioni temporali);
- profilo: dati relativi ad attributi degli utenti (username, genere, età, ...);
- network: informazioni riguardanti relazioni e connessioni tra utenti.

La maggior parte degli approcci presentati in letteratura utilizza dati di contenuto, poiché i dati di traiettoria appartengono a questa categoria. Molti metodi si basano infatti sull'utilizzo di dataset di traiettorie non anonime come informazioni esterne; queste possono essere già elaborate oppure inferite

o estratte da dataset contenenti dati di vario genere che possono essere sfruttati per costruire traiettorie. Ogni tecnica può poi cercare di giungere alla de-anonimizzazione in modi diversi, basandosi ad esempio su incontri o co-occorrenze spaziotemporali, costruzione e confronto di rappresentazioni del comportamento in termini di movimento, individuazione di pattern ricorrenti, approcci probabilistici.

Diversamente, le tecniche che sfruttano dati di profilo (età, genere, lingua, ecc.) sono difficilmente adattabili allo specifico caso della de-anonimizzazione di traiettorie, e per questo poco rilevanti in questo ambito. Gli algoritmi dell'ultima categoria possono invece essere adattati alla de-anonimizzazione di tracce spaziotemporali, tipicamente andando a costruire strutture come grafi sociali o delle relazioni/incontri che intercorrono tra gli utenti, ma richiedono dati di partenza spesso non disponibili o comunque non applicabili a questo specifico caso.

Una schematizzazione globale delle tecniche analizzate è visibile in tabella 2.1. A queste viene aggiunto il nuovo approccio alla de-anonimizzazione proposto in questo lavoro di tesi, che verrà esposto in una sezione successiva 4.3.2. Alla luce di quanto detto in precedenza, tutti gli algoritmi descritti utilizzano informazioni di tipo "contenuto". Per ogni approccio viene indicata la capacità o meno di tollerare *mismatch* ed errori presenti nei dati di traiettoria. È possibile notare come alcuni algoritmi operino costruendo modelli del comportamento degli utenti in termini di movimento, ad esempio tramite Catene di Markov [5] o vettori di *feature* [9], utilizzati poi come termine di paragone in fase di matching. L'approccio presentato in [13] è il più recente in termini di pubblicazione e si caratterizza soprattutto per essere stato progettato per tollerare mismatch spaziotemporali e distorsioni che potrebbero essere presenti nei dati; viene inoltre utilizzato un modello markoviano per la modellazione del comportamento dell'utente. Esistono anche approcci che operano in maniera più diretta e senza la costruzione di rappresentazioni particolari [7], ma le comparazioni presenti in letteratura [13] dimostrano come le performance in termini di accuratezza non siano inferiori rispetto a tecniche più sofisticate. Se ne deduce quindi che approcci più semplici non portino necessariamente a

Tabella 2.1: Schematizzazione degli approcci alla de-anonimizzazione presenti in letteratura, nell'intestazione della tabella M. = Mismatch.

	Anno	M. Spazio	M. Tempo	Modello Utente	Contesto
NFLX [4]	2002	×	✓	×	×
MSQ [6]	2008	✓	×	×	×
HMM [5]	2011	✓	×	✓	×
WYCI [11]	2013	×	✓	✓	×
ITF-IDF [9]	2014	×	×	✓	×
ME [7]	2015	×	×	×	×
HIST [8]	2016	×	✓	✓	×
POIS [10]	2016	×	×	×	✓
CO-LOC [12]	2016	×	×	✓	×
GM [13]	2018	✓	✓	✓	×
DDT	2019	✓	✓	✓	✓

risultati peggiori, dipendentemente dai casi e dai dati oggetto di analisi.

Un'ultima caratteristica raffigurata in tabella è quella della considerazione del contesto locale (in termini geografici), ad esempio la densità di persone o incontri in una certa location, per migliorare le performance. Questo tipo di meccanismo, poco considerato dagli approcci studiati, verrà ulteriormente approfondito in seguito in 4.3.2.

Come prima fase operativa, si è deciso di implementare e testare due degli algoritmi analizzati. Il primo [7] è un approccio semplice e diretto, che è stato giudicato adatto al caso di studio e come termine di paragone per altre tecniche. Di contro, il secondo algoritmo scelto [13] risulta essere uno dei più sofisticati; si è deciso di implementarlo in quanto rappresenta la soluzione più recente tra tutte quelle analizzate, presentata dagli stessi autori come tecnica in grado superare le performance di quelle precedenti. Per contrapposizione, si sono quindi confrontati due algoritmi profondamente diversi per valutarne

nel dettaglio le caratteristiche prima di elaborare una nuova proposta per la de-anonimizzazione.

Di seguito si espongono, in ordine cronologico, i principali approcci noti in letteratura, considerabili come lo stato dell'arte per quanto riguarda la de-anonimizzazione di traiettorie. L'insieme di queste opere ha costituito il punto di partenza per gli studi e gli esperimenti successivi.

2.1 Tecniche di de-anonimizzazione

NFLX Questo lavoro [4] presenta un approccio originariamente proposto per essere applicato al dataset del *Netflix Prize* ma adattabile al contesto della de-anonimizzazione di traiettorie. L'algoritmo nello specifico si focalizza sullo sfruttare la caratteristica della sparsità dei dati, molto frequente quando si opera con traiettorie, andando a pesare maggiormente le occorrenze di eventi rari comuni tra i due dataset per trovare il miglior link possibile tra utenti. In particolare vengono favorite dall'algoritmo con un contributo maggiore:

- locazioni visitate meno di frequente perché considerate maggiormente discriminanti;
- visite frequenti alla stessa location;
- visite che occorrono in una finestra temporale vicina agli eventi della traccia obiettivo.

MSQ In questo articolo [6] vengono esposte alcune possibili tecniche per calcolare la similarità tra tracce e individuare quelle con i match migliori, variando anche la misura in cui viene valutato il possibile rumore presente nelle informazioni esterne che vengono sfruttate per la de-anonimizzazione. Si propone infine come soluzione migliore quella di collegare tra loro le tracce secondo il criterio di *Minimum Square Approach*, andando quindi a ricercare le tracce che minimizzano i quadrati delle distanze:

$$D_{MSQ} = -\sum_{t \epsilon \tau} |L(t) - S(t)|^2,$$

dove $L(t)$ è una traccia anonima e $S(t)$ una collegata a un'identità.

HMM La soluzione proposta in questo articolo [5] si basa sull'assunzione di poter modellare la mobilità degli utenti attraverso delle Catene di Markov. Il comportamento P^u di un utente è quindi rappresentabile come la matrice di transizione corrispondente alla Catena di Markov; ogni entry P^u_{ij} di P^u rappresenta la probabilità che l'utente u si muova nella regione r_j nello slot temporale successivo, posto che si trovi al momento nella regione r_i. Questa rappresentazione può essere ulteriormente raffinata costruendo più Catene di Markov per ciascun utente, ognuna riferita a diversi momenti della giornata (ad esempio mattina, pomeriggio e sera), assumendo che il modello di comportamento dell'utente possa variare in funzione della fascia oraria. Per poter calcolare le matrici, è necessario colmare i punti in cui le traiettorie di partenza non sono complete, ovvero gli istanti temporali per i quali non è campionata nessuna posizione dell'utente. Una volta fatte queste stime, è possibile calcolare il punteggio di similarità come:

$$D_{HMM} = P(S_u|T^v) = \sum_Z \prod_{t \epsilon T} f(Z(t), S(t)) T^v_{Z(t-1),Z(t)},$$

dove la funzione $f(\cdot)$ descrive il mismatch tra la conoscenza dell'attaccante, indicata con S, e le vere posizioni dell'utente, rappresentate da Z, variabile nascosta del modello markoviano.

WYCI L'approccio qui proposto è probabilistico [11]. L'utente viene caratterizzato dalla frequenza con cui visita certe location. La probabilità di aver visitato le location viene espressa come:

$$P(r|L_v) = \frac{n^v_r + \alpha}{\sum_{r \in R} n^v_r + \alpha |R|},$$

dove n^v_r è il numero di volte in cui l'utente v ha visitato la location r, $|R|$ è il numero di location nel dataset e $\alpha > 0$ consiste in un parametro di *smoothing* per eliminare le probabilità zero. In conclusione il punteggio totale di similarità

può essere calcolato come:

$$D_{WYCI} = \prod_{t \epsilon \tau, S(t) \neq \emptyset} P(S(t)|L_v).$$

ITF-IDF Questo approccio [9] si basa sull'assunzione che i comportamenti in termini di movimento dei soggetti presentino pattern piuttosto stabili nel tempo e dal forte potere discriminante. Estraendo questi pattern partendo da un numero limitato di segmenti appartenenti a traiettorie di soggetti noti, questi possono essere utilizzati come semi-identificatori e sfruttati per ricercare in dataset di traiettorie storicizzate anonime le migliori corrispondenze. Gli autori sostengono che per estrarre questi pattern significativi siano sufficienti porzioni limitate di informazioni sull'utente, ottenibili ad esempio tramite osservazione diretta della vittima, monitoraggio, social engineering, attività condivise tramite social network. I pattern vengono modellati estraendo per ogni traccia segmenti stradali preferiti e punti di stop rilevanti, che vengono poi utilizzati per costruire dei vettori di feature che modellano i pattern di movimento applicando l'*Improved Term Frequency-Inverse Document Frequency (ITF-IDF)* ai valori. I segmenti stradali preferiti vengono estratti in base al numero di volte in cui il soggetto vi transita. Per quanto riguarda i cosiddetti "*Stops of Interest*", vengono selezionati tra i punti in cui le traiettorie mostrano delle soste dell'oggetto in movimento applicando alcuni filtri specifici per il dominio di origine dei dati utilizzati nello studio, campionati da dispositivi GPS montati su taxi. Questi punti vengono poi mappati sui punti di intersezione stradale più vicini per poter essere poi elaborati con l'ITF-IDF e ottenere i vettori di feature. Una volta estratte le rappresentazioni dei pattern sotto forma di vettori si procede alla deanonimizzazione calcolando il punteggio di matching tra le coppie di vettori relativi alle diverse tracce fino a individuare quella con il risultato maggiore. Il punteggio può essere calcolato come:

$$M_{Sc}^{i,\sigma} = \frac{f_i \cdot \widetilde{f}_\sigma}{||f_i||||\widetilde{f}_\sigma||},$$

ponendo al numeratore il prodotto scalare di f_i e $\widetilde{f}_\sigma$, ovvero i vettori relativi alle tracce da confrontare, e al denominatore il prodotto delle rispettive norme.

ME L'approccio presentato in questo articolo [7] stima la probabilità di ogni coppia di tracce di appartenere allo stesso soggetto in base al loro numero di elementi che producono un match spaziotemporale. L'algoritmo viene testato utilizzando come raccolta di dati da de-anonimizzare un dataset di record di cartellini di traffico (*CDR*) pubblicato da una compagnia telefonica dopo aver anonimizzato con degli pseudonimi gli utenti. Da questi record è possibile estrarre delle traiettorie andando a selezionare le coordinate spaziali (latitudine e longitudine) delle celle telefoniche indicate. Le informazioni esterne consistono invece in dati relativi a post e check-in geo-localizzati effettuati sui social network Twitter e Flickr, estratti tramite le API disponibili. Formalmente, il punteggio di similarità che viene proposto si può definire come:

$$D_{ME} = \sum_{t \epsilon \tau} I(S(t) = L(t)),$$

dove $L(t)$ è una traccia anonima, $S(t)$ una collegata a un'identità e $I(\cdot)$ è una funzione che ritorna uno in caso di match spaziotemporale, oppure zero. L'algoritmo prevede inoltre una cosiddetta condizione di esclusione: qualora una coppia di utenti generi due eventi vicini nel tempo ma lontani nello spazio, si valuta come impossibile la corrispondenza tra i due soggetti in quanto fisicamente lontani in un certo momento e la coppia viene esclusa dall'analisi. Questo specifico approccio è stato scelto come caso di studio ed è quindi stato implementato e testato; nei prossimi capitoli verrà presentato ed analizzato in dettaglio.

HIST L'approccio proposto in questo lavoro [8] si discosta dagli altri, basando la de-anonimizzazione sul match tra gli istogrammi ricavabili dalle traiettorie, utilizzati come validi identificatori degli utenti. Il metodo assume che le abitudini umane in termini di movimento si mantengano piuttosto costanti nel tempo, e che quindi il confronto possa essere significativo anche tra istogrammi ricavati da dati relativi a periodi di tempo non del tutto sovrapposti o anche completamente disgiunti, quindi statisticamente indipendenti. Mettendo a confronto due set di istogrammi, il primo riferito a utenti anonimi e il secondo a utenti con identità, il problema di trovare i match migliori viene

ricondotto a quello della ricerca di *matching* su un grafo bipartito. La funzione per calcolare il punteggio di similarità viene definita come:

$$D_{HIST} = -D_{KL}(\Gamma_u|\frac{\Gamma_u + \Gamma_v}{2}) - D_{KL}(\Gamma_v|\frac{\Gamma_u + \Gamma_v}{2}),$$

dove, γ_u e γ_v sono gli istogrammi relativi alle tracce da confrontare e D_{KL} è la funzione di divergenza Kullback-Leibler [19].

POIS In questo caso [10] la de-anonimizzazione viene basata principalmente sull'associazione di tracce (una anonima e una nota) in base agli eventi di "incontro" che intercorrono tra esse. Per poter riconciliare e confrontare traiettorie provenienti da dataset potenzialmente diversi per caratteristiche, soprattutto per quanto riguarda la frequenza di campionamento e l'accuratezza del posizionamento, viene effettuata una discretizzazione in regioni per quanto riguarda lo spazio e in slot rispettivamente al tempo. Il numero di visite di un utente in una regione in un certo periodo di tempo viene modellata seguendo una distribuzione di Poisson, mentre le azioni esterne (ad esempio un check-in) avvengono indipendentemente secondo una distribuzione bernoulliana. L'algoritmo si concretizza andando a computare per prima cosa il punteggio di ogni possibile coppia di traiettorie $(u, v)\,\epsilon\, U\, x\, V$, calcolabile come:

$$D_{POIS}(S_u, L_v) = \sum_{t\epsilon\tau} \sum_{r\epsilon R} \phi_{r,t}(S_u(t), L_v(t)),$$

dove ϕ misura l'importanza dell'evento di incontro nella regione r allo slot temporale t ed è calcolabile basandosi sul modello esposto precedentemente che assume distribuzioni di Poisson e Bernoulli. Fatto ciò viene definito un grafo bipartito su (U, V) dove il peso dell'arco tra (u, v) corrisponde al loro punteggio. Viene quindi calcolato il matching sul grafo bipartito che massimizza i pesi; al termine del processo l'algoritmo attribuisce i record collegati da un arco allo stesso utente.

CO-LOC In questo articolo [12] viene presentata una tecnica di identificazione che si diversifica dalle altre andando a basare la propria metodologia

sulle co-locazioni degli utenti in particolari luoghi nello stesso momento. Tale situazione è sempre più diffusa e rilevabile a causa dell'uso dei social network, in particolare quelli location-based, con i quali gli utenti possono condividere non solo la loro posizione ma anche quella delle persone in loro compagnia, ad esempio tramite un tag. Questo tipo di informazione permette di inferire, a partire dalla precisa posizione di un singolo utente, ad esempio tramite un check-in su un social network, anche quella di altri utenti per i quali non sarebbe possibile farlo in maniera diretta. Il metodo prevede quindi di costruire le traiettorie dei vari utenti noti (ad esempio relativi a profili social) basandosi sulle rispettive informazioni di locazioni e co-locazioni deducibili dalla compresenza con altri utenti. Per la de-anonimizzazione di un utente si cerca la traccia anonima con il match migliore, calcolato utilizzando il metodo della massima verosimiglianza. Per la definizione della funzione di verosimiglianza da massimizzare si utilizzano distribuzioni di probabilità bayesiane.

GM Questo lavoro [13] risulta essere il più recente tra quelli presi in considerazione. La possibilità degli autori di avere accesso a dati *ground-truth* ha permesso loro di formulare considerazioni e valutazioni più precise e fondate. Lo sviluppo del metodo di de-anonimizzazione proposto parte da un'assunzione precisa: le traiettorie che vengono memorizzate nei dataset presentano spesso discrepanze o mismatch, a volte anche piuttosto rilevanti, rispetto ai relativi e reali riferimenti spaziotemporali. In pratica, le locazioni con rispettivo riferimento temporale che vengono memorizzate e utilizzate per costruire le traiettorie possono presentare differenze sostanziali dall'effettivo punto spaziale e momento temporale in cui avvengono, principalmente a causa di errori o alterazioni dovute alla tecnologia di rilevamento o a comportamenti degli utenti. Un'altra principale problematica identificata dagli autori come causa di scarse performance degli algoritmi di de-anonimizzazione è quella della sparsità dei dati, in quanto raramente le tracce o le traiettorie che vengono estratte sono complete, non permettendo di conoscere la posizione degli utenti in ogni momento. Per ovviare a queste condizioni sfavorevoli l'approccio prevede l'utilizzo di un modello basato su *Gaussian Mixture* per gestire i mismatch

spaziotemporali e di uno markoviano per stimare le posizione dell'utente negli slot temporali in cui non sono disponibili osservazioni. Questo specifico approccio è stato scelto come caso di studio ed è quindi stato implementato e testato; nei prossimi capitoli verrà presentato ed analizzato in dettaglio.

Capitolo 3

Tecnologie

In questo capitolo si espongono le principali tecnologie utilizzate durante il lavoro di tesi.

3.1 Hadoop

Apache Hadoop è un *framework open-source* sviluppato per la memorizzazione ed elaborazione di dataset di grandi dimensioni in modo scalabile. Si pone come alternativa rispetto al modello di *Massively Parallel Processors (MPP)*, in cui più processori operano utilizzando risorse proprie (memoria, disco) su task che vengono suddivisi tra di essi. Rispetto alle architetture MPP, che utilizzano hardware di fascia alta e software proprietari, Hadoop è progettato per operare su commodity hardware e con strumenti open-source. In particolare questo tipo di hardware è rappresentato da componenti standard e commerciali, che possono essere reperiti senza legarsi ad un singolo *vendor*. Risulta inoltre più semplice sostituire i componenti del cluster in caso di eventuali rotture e fallimenti. Hadoop si compone di quattro moduli principali:

- *Common:* insieme di librerie e servizi di utilità per il sistema;
- *HDFS: file system* distribuito con altro *throughput* e replicazione gestita a livello di cluster;

- *YARN:* framework per lo *scheduling* dei *job* e la gestione ottimizzata delle risorse;

- *MapReduce:* sistema per processare parallelamente grandi dataset, servendosi di YARN.

3.1.1 HDFS

Il file-system di Hadoop è pensato per operare con file molto grandi, con accesso ai dati secondo un pattern streaming. Il design del sistema è maggiormente orientato a operare in *batch* piuttosto che in maniera interattiva. L'enfasi è posta sull'ottenere un alto throughput in fase di accesso ai dati, piuttosto che sulla bassa latenza. Le applicazioni che operano su HDFS necessitano di un modello di accesso *write-once-read-many*. Nello sviluppo del file system è stata inoltre dedicata particolare attenzione alla fase di recovery, poiché operando su commodity hardware il fallimento rappresenta la norma piuttosto che l'eccezione.

HDFS gestisce i file suddividendoli in blocchi, che rappresentano la minima quantità di dati che possono essere letti o scritti, con un range che può variare tra i 64 megabyte e un gigabyte: di *default* il dimensionamento del blocco è di 128 megabyte. Questo sistema permette di gestire file potenzialmente più grandi di un singolo disco. Se invece deve essere memorizzato un file di dimensioni inferiori a quelle di un blocco, si utilizza solo lo spazio necessario. Sono previsti due tipi di nodi:

- *NameNode:* svolge il ruolo di *master*, mantiene in maniera persistente l'albero del file system e tutti i metadati, memorizzando la locazione di ogni blocco relativo a un determinato file (*block pool*);

- *DataNode:* svolgono il ruolo di *slave* e si occupano di memorizzare e restituire i blocchi. Periodicamente vengono forniti report al NameNode contenenti la lista dei blocchi memorizzati e segnali per indicare lo stato di attività del nodo.

HDFS necessita di conoscere la topologia del cluster, in particolare della struttura dei nodi all'interno dei *rack* che compongono il *data center*: la struttura viene modellata in maniera simile ad un albero, computando e memorizzando le distanze tra i vari nodi.

Replicazione Ogni blocco è replicato in più DataNode per migliorare performance e robustezza, tenendo conto della topologia del cluster; per ogni blocco il NameNode mantiene la lista dei DataNode che lo memorizzano. Il fattore di replicazione standard è tre, con le due copie extra su due nodi diversi di uno stesso rack diverso dal primo. Hadoop applica il principio di *data locality*, ovvero il concetto secondo cui è preferibile portare il codice ai dati piuttosto che il contrario; per fare ciò vengono sfruttate la conoscenza della topologia del cluster e la replicazione. In questo contesto il NameNode può costituire un *single point of failure*: la soluzione consiste nel mantenimento di un nodo secondario su una macchina diversa che regolarmente si connette a quello primario per farne uno *snapshot*, in caso debba essere ricreato. La dimensione del block pool è limitata dalla memoria del NameNode: HDFS non è quindi adatto alla gestione di molti file di piccole dimensioni. Ogni NameNode gestisce una porzione del filesystem, detta un namespace, e mantiene tutti i relativi metadati.

3.1.2 YARN

YARN è il gestore delle risorse introdotto con Hadoop 2; originariamente sviluppato per migliorare le prestazioni di MapReduce, è abbastanza generale per essere utilizzato anche da altri paradigmi. Yarn fornisce API per richiedere e operare con risorse su infrastrutture a cluster. Viene utilizzato solo da framework e mai da codice utente. Implementa due demoni:

- *Resource manager (RM)* globale (uno per cluster), che gestisce le risorse tra più applicazioni;

- *Node Manager (NM)*, uno per ogni nodo slave, responsabile per i container, ovvero l'entità astratta usata per eseguire processi con un set limitato di risorse.

Quando riceve una richiesta da un'applicazione, il RM cerca un NM che possa lanciare l'*application master process (AMP)* in un container. Il Resource manager ha due componenti principali: lo scheduler, che alloca risorse alle applicazioni in esecuzione, e l'application manager, che accetta i job sottoposti e negozia il primo container per eseguire l'AMP e per fare il restart se fallisce. YARN fornisce:

- scalabilità: il RM non fa da bottleneck perché delega agli application master;
- apertura: è fornito il supporto ad altri framework oltre a MapReduce;
- alta disponibilità: i meccanismi usati su HDFS con i NameNode si possono applicare a RM e AMP.

3.1.3 Hive

Apache Hive è un progetto software orientato al *data warehousing* costruito su Hadoop, che permette di eseguire query e analisi dati su dataset di grandi dimensioni. Hive fornisce un'interfaccia *SQL-like*, chiamata *HiveQL*, per interrogare i dati memorizzati nei vari database e file system integrati con Hadoop. Inizialmente sviluppato da Facebook, Hive è attualmente utilizzato e sviluppato da altre importanti compagnie come Netflix e Amazon, che mantine una propria *fork* del progetto, inclusa nell'implementazione di MapReduce disponibile su *Amazon Web Services*.

3.2 Spark

Apache Spark è una piattaforma di esecuzione *general purpose*, naturale evoluzione di MapReduce, sviluppata per adattarsi ai cambiamenti avvenuti

nelle esigenze rispetto ad hardware, software e task degli utenti nel contesto dei big data. Per quanto riguarda il primo aspetto, i cambiamenti principali hanno riguardato l'aumento dei *core* a disposizione delle macchine e della maggior importanza acquisita dalla memoria RAM rispetto al disco come fonte principale di dati. Anche a livello software sono avvenuti cambiamenti sostanziali: il paradigma funzionale ha guadagnato rilevanza rispetto alla programmazione orienta agli oggetti, così come i sistemi NoSQL rispetto a quelli SQL; inoltre è cresciuta la necessità di implementare ottimizzazioni software per core multipli. Anche il mondo dei big data ha subito delle evoluzioni: mentre l'accento era inizialmente posto sul volume, si è poi spostato sulla velocità. Sono inoltre sorti nuovi casi d'uso come quelli legati al *machine learning*. MapReduce è un paradigma che non si presta a tutti i tipi di elaborazioni, essendo più adatto al batch che all'interattività e al data mining. Spark prevede due astrazioni: i Risilient Distributed Dataset (RDD) e i Direct Acyclic Graph (DAG). Le feature principali degli RDD sono:

- resilienza: vengono ricostruiti automaticamente in caso di fallimento;
- distribuzione: gli oggetti di una collezione sono divisi in partizioni sparse tra i nodi;
- immutabilità: una volta creati non si possono modificare, ciò comporta un aumento dello spazio occupato ma la parallelizzazione è facilitata;
- *lazy evaluation:* meccanismo di ottimizzazione prima dell'esecuzione, non si eseguono trasformazioni finché non sono necessarie (richiede immutabilità e assenza di *side effect*;
- *caching:* possono essere persistiti in memoria e le performance migliorano drasticamente;
- *type inference:* i tipi di dato non sono dichiarati ma inferiti.

Un RDD può essere creato caricando un dataset esterno o distribuendo una collezione di oggetti; di default non è reso persistente. Sugli RDD sono

possibili due tipi di operazioni: le trasformazioni, che costruiscono un RDD nuovo a partire da uno preesistente, e le azioni, che computano un risultato che è ritornato a un programma o salvato su un sistema di memorizzazione. La lazy evaluation prevede che gli RDD vengano computati solo quando usati in una azione, finché non ne viene eseguita una i dati non vengono acceduti. Le trasformazioni cambiano solo i metadati, in particolare le dependencies, ovvero la lista degli RDD preesistenti coinvolti.

Il DAG è un piano di esecuzione logica computato da Spark basandosi su *lineage graph* e sull'applicazione utente. Rappresenta una sequenza di computazioni effettuate sui dati come grafo diretto (trasformazioni da partizione A a B) e aciclico. Il piano di esecuzione è compilato in stage fisici i cui limiti sono definiti dalle operazioni di shuffle o caching di partizioni. L'unità di esecuzione fondamentale è il task, ne viene creato uno per ogni partizione nel nuovo RDD e vengono assegnati ai worker in base alla data locality.

Decomposizione dell'applicazione In Spark con applicazione si intende la singola istanza di *SparkContest* che schedula una serie di job, sequenzialmente o in parallelo. Il job è un set completo di trasformazioni su RDD che si chiude con un'azione o salvataggio, scatenate dal driver. Lo stage è invece un set di trasformazioni che possono essere messe in *pipeline* ed eseguite da un singolo worker. In ultimo, il task rappresenta l'unità base di scheduling, che esegue lo stage su una singola partizione.

Architettura Spark implementa un'architettura master-slave con un coordinatore centrale (driver) e worker distribuiti (executors), ovvero processi Java indipendenti che assieme formano un'applicazione Spark. Ogni applicazione Spark può avere un solo Driver che suddivide il programma utente in task, crea lo SparkContext e computa il DAG logico che converte poi in piano di esecuzione fisico. L'executor è il processo responsabile di eseguire il task ricevuto. Ogni applicazione tipicamente ha più executor e ciascun nodo worker ne ospita molti. Il Cluster Manager è il componente responsabile dell'assegnazione e gestione delle risorse del cluster, e può essere interpretato da YARN.

SparkSQL SparkSQL è un modulo software costruito sopra a Spark Core, al pari di altri moduli come Spark Streaming e SparkML. L'introduzione di questo modulo vuole abilitare l'utilizzo dell'approccio relazionale nel contesto delle applicazioni big data. Il modello relazionale è infatti molto popolare ma carente in termini di performance; richiede inoltre di eseguire operazioni di *ETL* su varie sorgenti dati che potrebbero essere semi-strutturati. Le esigenze degli utenti sono cambiate, e nuovi task e analisi avanzate, ad esempio legate al machine learning, non si prestano ad essere realizzate in maniera relazionale. SparkSQL permette quindi di integrare API relazionali e procedurali. SparkSQL fornisce:

- integrazione: l'unione di SQL e Spark permette di affiancare query relazionali e algoritmi complessi (Python, Scala, Java);
- accesso uniforme: possono essere caricati ed interrogati dati provenienti da varie sorgenti, ad esempio Hive, formati colonnari, JSON, ecc.;
- compatibilità con Hive: le query scritte per Hive possono essere utilizzate direttamente sui warehouse esistenti, la compatibilità è completa;
- connettività standard: sono supportati gli standard industriali JDBC e ODBC;
- scalabilità: stesso motore per l'esecuzione sia di query lunghe che interattive, traendo vantaggio dal modello degli RDD di Spark.

La killer feature di SparkSQL è quindi la possibilità di integrare direttamente all'interno delle applicazioni Spark (Python, Scala, Java) elementi del modello relazionale, potendo combinare algoritmiche complesse con interrogazioni SQL. La principale astrazione di SparkSQL è il DataFrame, concettualmente equivalente a una tabella relazionale e manipolabile come un RDD Spark nativo. Questa struttura può essere costruita a partire dalle sorgenti gestite da SparkSQL: tabelle Hive, file strutturati, formati tabellari, database esterni, RDD, altri DataFrame, ecc.

GeoSpark GeoSpark è un'estensione di Spark e SparkSQL realizzata per la gestione di grandi quantità di dati spaziali. Fornisce le astrazioni di *Spatial Resilient Distributed Datasets* e *SpatialSQL*, che permettono di eseguire caricamento, elaborazione e analisi su larga scala di dati spaziali in modo efficiente. GeoSpark offre inoltre un set completo di API e funzionalità utili per l'elaborazione di dati spaziali, ad esempio per il calcolo di distanze, aree, centroidi e intersezioni. Viene inoltre fornita un'estensione dedicata alla visualizzazione, GeoSparkViz, che offre supporto nativo per il design cartografico e la rappresentazione di RDD e query spaziali. Possono essere gestite traiettorie e geometrie di diversi gradi di complessità (punti, poligoni, ecc.), garantendo la compatibilità con i formati più comuni, come CSV, TSV, WKT, WKB e GeoJSON. Sono inoltre fornite strutture con indice native, R-Tree e Quad-Tree, necessarie per la memorizzazione scalabile di grandi quantità di dati spaziali.

Capitolo 4

Progetto

In questo capitolo si espongono in dettaglio gli algoritmi implementati e testati sui dati a disposizione; si presenta inoltre il nuovo approccio alla de-anonimizzazione proposto con questo lavoro di tesi.

4.1 Approccio di Cecaj *et al.*

Il primo approccio analizzato in dettaglio e implementato è quello proposto in [7]. Come esposto precedentemente, questo algoritmo è stato presentato da Cecaj *et al.* scegliendo come caso di studio la de-anonimizzazione di tracce anonime derivanti da record telefonici. Queste sono state confrontate con traiettorie estratte a partire da dati estrapolati da social network, in particolare Twitter e Flickr. Si tratta quindi di uno scenario simile a quello oggetto di questo studio, pertanto l'approccio risulta applicabile e facilmente adattabile al problema. L'approccio risulta essere uno dei più diretti e meno elaborati, ma le comparazioni presenti in letteratura [13] dimostrano che le performance sono in linea con quelle di tecniche più sofisticate e complesse. L'algoritmo basa il processo di identificazione sulla ricerca per ogni traiettoria anonima della traccia collegata a un profilo noto per la quale è maggiore il numero di match spaziotemporali.

Risulta quindi fondamentale definire in cosa consiste un match spaziotemporale.

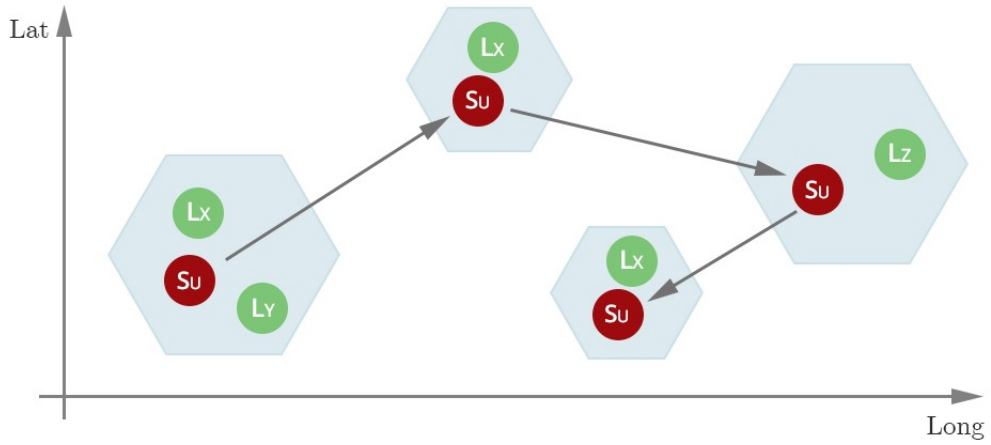


Figura 4.1: Intuizione dell'algoritmo: per la traccia social Su la corrispondenza migliore è con l'utente GPS Lx, essendo quello con il maggior numero di match spaziotemporali; la sua traiettoria viene preferita a quelle relative agli utenti anonimi Ly, Lz

Definizione 1. *Per ogni traccia social S_u (relativa al profilo u) e traccia CDR L_v (legata all'utente anonimo v), due eventi l_i^v e s_j^u costituiscono un match se $s_{dist}(l_i^v, s_j^u) < r_j^u$ e $t_{dist}(l_i^v, s_j^u) < \Delta t$, dove s_{dist} e t_{dist} sono le distanze spaziali e temporali rispettivamente.*

Come valore di partenza per la soglia temporale Δt gli autori propongo un'ora, mentre quella spaziale viene stimata in base all'area di copertura delle celle, ricavabile dai dati. La figura 4.1 esemplifica l'intuizione alla base dell'algoritmo.

Questa tecnica prevede inoltre una cosiddetta condizione di esclusione: qualora una coppia di utenti l_x e s_y generi due eventi l_j^x e s_i^y vicini nel tempo ma lontani nello spazio, si valuta come impossibile la corrispondenza tra i due soggetti in quanto fisicamente lontani in un certo momento. Qualora la condizione si verifichi, la coppia viene esclusa dall'analisi, anche se presenta altre combinazioni di eventi in match. Gli autori definisco il concetto di lontananza spaziale che verifica la condizione di esclusione come il doppio del raggio di copertura della cella telefonica. Nella figura 4.2 vengono rappresentate grafi-

Figura 4.2: Quando coppie di eventi si verificano in locazioni vicine o coincidenti, si genera un match spaziotemporale. Se nessun evento GPS è contemporaneo a un tweet, non si registra né un match né un'esclusione. Quando viene osservata una coppia di eventi distanti, come all'istante t_4, si attiva la condizione di esclusione: la traiettoria GPS e l'account Twitter non coincidono.

camente le possibili combinazioni di eventi, per esemplificare il comportamento dell'algoritmo nelle situazioni possibili.

Questo tipo di esclusione costituisce un meccanismo indubbiamente ragionevole ma per certi versi anche piuttosto stringente, possibile causa di errori, come verrà esposto in seguito.

Per formalizzare, il punteggio della coppia (L, S), dove L è la traccia anonima e t un istante temporale, può essere espresso come:

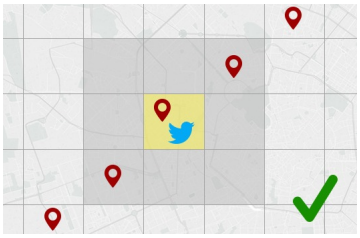
$$D_{ME} = \sum_{t \epsilon \tau} I(S(t) = L(t)),$$

dove $I(\cdot)$ è una funzione che ritorna uno in caso di match spaziotemporale, oppure zero. Data una traiettoria anonima l'algoritmo la associa quindi a quella con cui presenta il maggior numero di match spaziotemporali, che non abbia verificato la condizione di esclusione, tra tutte quelle collegate a profili social.

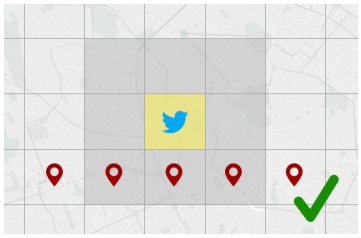
Implementazione L'implementazione di questo approccio è stata realizzata in Python. A livello spaziale i dati sono stati discretizzati applicando una griglia con celle di lato uguale a un millesimo di grado di coordinata (latitudine e longitudine), mentre per l'approssimazione dei riferimenti temporali si è scelta una granularità oraria. Data la struttura dei dati disponibili, l'occorrenza di un match spaziotemporale tra due eventi l_i^v e s_j^u si verifica qualora sussistano due condizioni:

- corrispondenza tra gli slot temporali di riferimento dei due eventi;
- corrispondenza tra le regioni di riferimento dei due eventi: il punto di localizzazione di s_j^u cade nella medesima cella di l_i^v o nel suo intorno, con raggio uguale a uno.

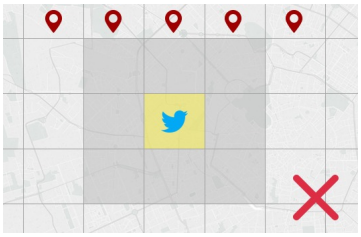
Nelle figure 4.3 viene esemplificato graficamente il procedimento di ricerca di un match.



(a) Match valido.



(b) Match valido.



(c) Esclusione.

Figura 4.3: In caso la traiettoria GPS abbia almeno un punto all'interno cella relativa a un tweet o in una adiacente nel time slot, si registra un match. Se invece la corrispondenza è solamente temporale ma non spaziale, si genera un'esclusione.

Per verificare la corrispondenza a livello spaziale il punto di localizzazione del tweet geo-referenziato viene confrontato con un'area di nove celle, centrata su quella corrispondente all'evento social s_j^u. Qualora due eventi siano contemporanei ma non corrispondenti a livello spaziale, anche considerando l'intorno di tolleranza, vengono considerati come distanti. Una coppia di eventi di questo tipo causa l'attivazione delle condizione di esclusione.

Una situazione particolare è quella in cui per un segmento temporale sono disponibili più campionamenti, sia per quanto riguarda la traccia GPS che quella social, che rendono possibili molteplici combinazioni di eventi tra le due traiettorie. In questo scenario è possibile che all'interno del medesimo segmento temporale siano presenti coppie di eventi in condizione di match spaziotemporale e altre che verificano la clausola di esclusione. In questo caso a livello implementativo si è scelto di pesare maggiormente le situazioni di match, principalmente per evitare di per eliminare dall'analisi coppie di tracce potenzialmente ad alto punteggio. Per questo, nel caso in cui all'interno di un certo segmento temporale si verifichi almeno una corrispondenza spaziale, eventuali combinazioni di punti che verificherebbero la condizione di esclusione vengono ignorate.

Di seguito si presenta una rappresentazione in pseudo-codice del funzionamento dell'algoritmo ad alto livello. Data una traiettoria anonima si cercano quelle relative ad account Twitter con il numero di match spaziotemporali maggiore. Per effettuare il confronto tra una traiettoria social e quella anonima, si scorre la sequenza di slot temporali contenuti nella prima e si verifica se per il medesimo riferimento temporale siano presenti osservazioni nella traccia GPS. Lo scorrimento parte dalle tracce social per motivi di efficienza, essendo queste tipicamente più corte di quelle GPS, e quindi con un numero minore di slot temporali osservati. Quando viene individuato uno slot orario che presenta campionamenti in entrambe le traiettorie, si procede al confronto dei relativi punti spaziali per verificare l'eventuale occorrenza di match, secondo il criterio di tolleranza espresso precedentemente. Qualora per uno slot temporale comune alle due traiettorie non venga riscontrata nemmeno una corrispondenza spaziale, viene verificata la condizione di esclusione. In questo caso alla coppia

```
1  Dati: una traccia anonima, una o più tracce social.
2  Risultato: Numero di match tra la traccia anonima e le tracce social.
3  Funzione CalcoloMatch {
4      Inizializza la lista di output
5          Per ogni traccia social:
6              Inizializza il contatore dei match
7              Per ogni slot temporale nella traccia GPS:
8                  Se lo slot temporale non è nella traccia:
9                      Continua
10                 Per ogni punto della traccia social al time slot:
11                     Per ogni punto della traccia GPS al time slot:
12                         Se c'è corrispondenza spaziale
13                             Incrementa contatore dei match
14                     Se c'è stata corrispondenza temporale ma non spaziale:
15                         Attiva condizione di esclusione
16                 Se la condizione di esclusione si è verificata, il punteggio è zero
17                 Altrimenti assegno il punteggio calcolato
18     Restituisce lista di output
19 }
```

di tracce viene assegnato un punteggio uguale a zero, anche in presenza di match precedentemente riscontrati, in quanto si valuta che i due soggetti si sono trovati in luoghi fisicamente distanti in un certo momento.

4.2 Approccio di Wang *et al.*

Questo approccio risulta essere il più recente tra quelli presi in considerazione durante la fase di studio preliminare. All'interno dell'articolo [13] gli autori espongono le problematiche riscontrate con gli approcci preesistenti, definendo poi quelle che considerano le cause principali di errori e scarse performance degli algoritmi di de-anonimizzazione.

La prima causa individuata è quella dei mismatch spaziotemporali che possono essere presenti all'interno dei dataset di traiettorie. Questo fenomeno consiste in una differenza, in termini spaziali e/o temporali, tra l'effettivo verificarsi di un evento e la relativa memorizzazione come dato di traiettoria. Tale differenza può in certi casi essere anche piuttosto rilevante, andando a inficiare le prestazioni di algoritmi che di fatto vengono applicati su dati fortemente rumorosi o incorretti. Questo tipo di alterazione nei dati, spesso non considerata

o marginalmente valutata nella costruzione di tecniche di de-anonimizzazione, può essere giustificata da alcune problematiche tecnologiche e pratiche, ad esempio:

- errore intrinseco del GPS: questa tecnologia, molto utilizzata per la localizzazione, può incorrere in malfunzionamenti dovuti a vari fattori, come errori del satellite, interferenze dell'atmosfera, fallimento del ricevitore;

- locazioni irraggiungibili dal GPS: non tutti i luoghi sono coperti e raggiungibili dai satelliti, ad esempio un dispositivo che transita per un tunnel della linea metropolitana potrebbe non potersi collegare al satellite per tutto il tempo trascorso al suo interno mantenendo però la connessione a una cella telefonica. In questo caso si riscontrerebbe un mismatch tra le traiettorie osservate dall'operatore telefonico e dal sistema satellitare per lo stesso soggetto;

- meccanismi di aggiornamento della localizzazione: alcune applicazioni e dispositivi mobili, principalmente per ragioni di efficienza della batteria, potrebbero aggiornare il posizionamento con una frequenza molto bassa o con una minore accuratezza.

- comportamento degli utenti: condizione che si verifica soprattutto per quel che riguarda le localizzazioni e i check-in effettuati tramite social network. Gli utenti possono infatti decidere di geo-localizzarsi presso una location quando si trovano anche solo nelle sue vicinanze o a diverse ore di distanza dall'effettiva presenza nel punto, costituendo un ulteriore fattore in grado di causare discrepanze e mismatch nelle traiettorie registrate. Un utente può mostrare questo tipo di comportamento per vari motivi, ad esempio per ricevere *badge* digitali o per competere con i propri amici utilizzando giochi o applicazioni.

Un secondo aspetto che gli autori identificano come possibile causa di performance non ottimali della de-anonimizzazione è il forte grado di sparsità dei dati di traiettoria. Le traiettorie molto spesso non presentano osservazioni

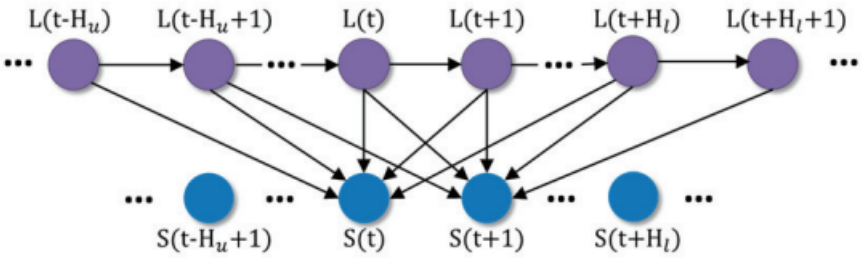


Figura 4.4: Relazione tra L(traiettoria GPS) e S (traiettoria Twitter).

ravvicinate. Questo può essere dovuto a diversi fattori, ad esempio: velocità di campionamento della posizione, errori di acquisizione, tipologia di dati da cui le traiettorie vengono estratte. Questa caratteristica è riscontrabile anche nei dati oggetto di questo studio, essendo le traiettorie non complete su tutto il periodo di tempo di interesse. Anche operando una discretizzazione della dimensione temporale in segmenti di un'ora, molti degli slot temporali risultano non coperti da osservazioni della posizione del soggetto, che per quel segmento rimane ignota.

Per ovviare a queste problematiche gli autori propongono un modello in cui due componenti specifiche vengono applicate per affrontarle adeguatamente:

- modello basato su Gaussian Mixture [23] per la gestione dei mismatch spaziotemporali;
- modello markoviano [24] per la stima delle locazioni ignote all'interno di tracce incomplete.

In figura 4.4 è presentata la relazione tra le variabili relative alla traccia anonima L e a quella social S. A partire da questo modello probabilistico, il punteggio di similarità può essere definito come:

$$D_{GM}(\mathbf{S}, \mathbf{L}) = \log p(\mathbf{S}|\mathbf{L}).$$

Per modellare i mismatch spaziotemporali nei dati utilizzati come informazioni esterne viene adottato un modello basato su Gaussian Mixture, ovvero

un insieme finito di densità gaussiane, chiamate componenti, singolarmente definibili come:

$$N(x|u_k, \sigma_k).$$

Ogni componente $N(\cdot)$ rappresenta una distribuzione caratterizzata dalla propria media u_k e covarianza σ_k. Facendo riferimento alla figura 4.4, la componente $N(x|u_p, \sigma_p)$ rappresenta la densità di probabilità attribuita al record esterno con mismatch temporale di p unità temporali.

Con L_C si rappresenta la traiettoria anonima completa, ovvero in cui $\forall t \in T$, $L_C \neq \emptyset$. Condizionata su di essa, la funzione di densità di probabilità (PDF) di un record esterno $S(t)$ di appartenere allo stesso utente può essere calcolata come:

$$p(S(t)|\mathbf{L}) = \sum_{p=-H_l}^{H_u} \pi(p) \cdot N(S(t)|L(t-p), \sigma^2(p)I_2),$$

dove $\pi(p)$ è la probabilità che il mismatch temporale sia di p unità di tempo e $\sigma(p)$ è la componente relativa alla distanza spaziale condizionata a tale valore di gap temporale. Infine essendo $S(t)$ e $L(t)$ costituite da coordinate geografiche (latitudine, longitudine) e quindi rappresentate come vettori bi-dimensionali, I_2 è la matrice identità 2×2. I valori dei parametri $\pi(p)$ e $\sigma(p)$ possono essere scelti empiricamente o stimati algoritmicamente a partire dai dati di interesse. L'utilizzo di un modello basato su distribuzioni gaussiane abilita la tolleranza di eventuali mismatch presenti all'interno dei dati. Questo meccanismo permette di individuare match temporali non necessariamente esatti, gestendo una finestra di tolleranza all'interno della quale possono essere ricercate le corrispondenze. Gli autori ad esempio utilizzano nei loro test diversi valori per la dimensione dell'intervallo di tolleranza, fissandolo poi a 24 ore. In questo modo è possibile individuare corrispondenze tra eventi distanti alcune ore, attribuendo questa differenza temporale a eventuali mismatch, che come esposto in precedenza possono essere causati da più fattori, che vengono così mitigati. Se a partire da un evento social viene riscontrata una corrispondenza all'interno della finestra temporale tra i campionamenti GPS, è quindi possibile calcolare un valore di probabilità che può essere aggregato al punteggio

complessivo tra le due traiettorie. In mancanza di una finestra di tolleranza, se la corrispondenza tra due eventi non risultasse precisa, questi non potrebbero contribuire.

Il modello mostrato in figura 4.4 evidenzia come data una traiettoria anonima completa L, i valori di $S(t)$ per t differenti siano tra loro indipendenti. La densità di probabilità di una intera traiettoria del dataset esterno può essere calcolata come:

$$p(\mathbf{S}|\mathbf{L}) = \prod_{DS(t)\neq\emptyset} p(S(t)|\mathbf{L}).$$

Come detto in precedenza le traiettorie degli utenti presentano molti slot temporali in cui le locazioni sono mancanti, ovvero $\exists\, t \in T$ tale che $L(t) = \emptyset$. Tra le conseguenze di questa condizione, i valori di $S(t)$ per t differenti diventano dipendenti, rendendo inapplicabile la formula per il calcolo della probabilità della traiettoria. Per ovviare a questo problema, gli autori propongono una soluzione: enumerare tutte le possibili traiettorie complete per L e applicare la formula della probabilità totale rispetto ad esse. Si denota con $C(L)$ l'insieme di tutte le possibili traiettorie complete di L. A questo punto la densità di probabilità di $S(t)$ condizionata L può essere calcolata come segue:

$$p(\mathbf{S}|\mathbf{L}) = \sum_{\mathbf{L_C}\epsilon C(\mathbf{L})} p(\mathbf{L_C}|\mathbf{L}) \prod_{S(t)\neq\emptyset} p(S(t)|\mathbf{L_C}).$$

Per calcolare la probabilità $p(\mathbf{L_C}|\mathbf{L})$ si utilizza un modello markoviano; gli autori presentano due soluzioni alternative utilizzano modelli di ordine zero o uno.

Questo modello si applica qualora per un elemento della traiettoria social analizzata non si trovino corrispondenze tra gli eventi della traccia GPS, che viene però considerata completa. Per calcolare i valori di probabilità da aggiungere al punteggio complessivo si considera la rappresentazione del comportamento del soggetto. Per ognuna delle location presenti nella storia dell'utente anonimo, si calcola il valore di probabilità in funzione della distanza spaziale e del mismatch temporale, come nel caso precedente, applicando un peso dato dalla probabilità con cui è stimata la presenza del soggetto in quella locazione. Sostanzialmente si effettua un confronto stimando la posizione

reale del soggetto, ignota per quel preciso slot temporale, a partire dalle osservazioni effettivamente disponibili all'interno della traiettoria relative agli altri momenti. All'aumentare della frequenza con cui un soggetto è stato osservato in un luogo, conseguentemente crescerà la probabilità con la quale verrà stimato in quella posizione in mancanza di campionamenti reali. Ad esempio una traccia composta da campionamenti registrati in un solo punto distinto viene modellata con quell'unica locazione associata a una probabilità uguale a uno; in questo caso per tutti gli slot temporali per i quali non sono disponibili osservazioni si stima la presenza del soggetto in quello specifico punto con una probabilità del 100%. Una traiettoria anonima composta da dieci campionamenti in altrettanti punti distinti viene invece modellata associando a ciascuna locazione osservata per il soggetto una probabilità uguale a 0.1. Per gli slot temporali non coperti da rilevazioni la posizione del soggetto viene stimata su tutte e dieci le locazioni con la stessa probabilità.

Nel caso di modello markoviano di ordine zero, le locazioni ad ogni slot temporale si assumono tra loro indipendenti. La distribuzione marginale dell'utente $E(r)$, definita come:

$$E(r) := p(L(t) = r) = \frac{\sum_{t \epsilon T} I(L(t) = r = +\alpha(f)}{\sum_{t \epsilon T} I(L(t) \neq \emptyset + \sum_{r \epsilon R} \alpha(r)},$$

dove $I(\cdot)$ rappresenta una funzione equivalente all'espressione logica $I(true) = 1$ e $I(false) = 0$. Infine $\alpha(r)$ è un parametro utile ad eliminare le probabilità zero. Può essere valorizzato in diversi modi, ad esempio utilizzando lo *smoothing* di Laplace.

Per concludere, date le definizioni precedenti la probabilità di una traiettoria completa $\mathbf{L_C} \, \epsilon \, C(\mathbf{L})$ condizionata $\mathbf{L}$ si può calcolare come:

$$p(\mathbf{L_C}|\mathbf{L}) = \prod_{t \epsilon T, L(t) = \emptyset} E(L_C(t)).$$

Implementazione Per l'implementazione di questo algoritmo si è inizialmente fatto riferimento a quella (in Python) disponibile in [21], per poi adattarla alle esigenze e caratteristiche del caso di studio. Si presenta l'implementazione dell'algoritmo ad alto livello in pseudo-codice.

```
1  Dati: una traccia anonima, una o più tracce social.
2  Risultato: Punteggio di similarità tra la traccia anonima e le tracce social.
3  Funzione CalcoloPunteggio = {
4      Inizializza parametri
5      Effettua e memorizza la stima della traiettoria completa del soggetto anonimo
6          Per ogni traccia social:
7              Inizializza punteggio
8              Per ogni slot temporale nella traccia social:
9                  Inizializza probabilità iniziale
10                 Per ogni valore nella finestra di tolleranza:
11                     Se lo slot corrente è presente nella traiettoria GPS:
12                         Calcola distanza spaziale tra i punti
13                         Aggiorna la probabilità iniziale in funzione della distanza spaziale e
                             del valore di scorrimento della finestra
14                     Altrimenti:
15                         Per ogni locazione nella traiettoria completa:
16                             Calcola la distanza spaziale tra i punti
17                             Aggiorna la probabilità iniziale proporzionalmente alla
                                 frequenza della locazione
18                 Aggiorna il punteggio con la probabilità calcolata, pesandola logaritmicamente
19     Restituisce Punteggi di similarità tra la traccia anonima e quelle social
20 }
```

Ogni traiettoria anonima viene confrontata con le tracce derivate dai tweet per individuare i profili per i quali il punteggio di similarità risulta migliore. Prima di calcolare i punteggi delle coppie di tracce, vengono eseguite alcune operazioni di inizializzazione. Si definiscono le probabilità di occorrenza dei diversi valori di mismatch temporale. Per l'implementazione, seguendo quanto suggerito in [13], si sono utilizzati i valori già definiti all'interno dell'articolo per quanto riguarda le probabilità dei mismatch. Anche per la definizione delle funzioni gaussiane da applicare si sono utilizzati i valori di media e varianza proposti. Viene poi effettuata la stima del comportamento abituale del soggetto legato alla traiettoria anonima, per poter poi simulare la completezza della stessa. La modellazione è realizzata costruendo una struttura che memorizza tutte le locazioni distinte occupate dal soggetto, associandole alla relativa probabilità di occorrenza all'interno della traiettoria, secondo quanto esposto precedentemente nella trattazione teorica dell'algoritmo.

Il confronto tra due tracce procede scorrendo gli slot temporali per i quali la traccia social presenta osservazioni. A questo punto si verifica l'esisten-

za una corrispondenza temporale con un'osservazione della traiettoria GPS, ovvero si verifica se sono presenti campionamenti relativi allo specifico slot temporale, esplorando una finestra di tempo che copre anche le 24 ore precedenti; l'ampiezza della finestra può essere variata, ma si è utilizzato il valore di base proposto dagli autori. L'introduzione di un intervallo di tolleranza permette di individuare corrispondenze anche non esatte a livello temporale, imputando eventuali discrepanze a mismatch nei dati memorizzati. Quando viene riscontrata una corrispondenza temporale, si calcola un valore di probabilità da aggiungere al punteggio complessivo. Tale valore è dato dal prodotto tra la probabilità del mismatch temporale in uso (definito in fase di inizializzazione) e il risultato della funzione di distanza spaziale pesato utilizzando una funzione gaussiana, per permettere di tollerare eventuali mismatch spaziali.

Qualora invece non si trovino corrispondenze tra lo slot temporale correntemente analizzato della traccia social e quelli della traiettoria GPS, che viene però considerata come completa, i valori di probabilità da aggiungere al punteggio complessivo si calcolano utilizzando come termine di paragone la rappresentazione del comportamento dell'utente. Si calcolano i valori di probabilità in funzione della distanza spaziale e del mismatch temporale rispetto a ogni locazione presente nella struttura. Il contributo di ciascun valore viene pesato in base alla probabilità con cui l'utente viene stimato nella locazione, che risulta alta qualora il soggetto sia stato frequentemente osservato in quel punto.

L'algoritmo prevede quindi che ogni evento osservato per la traiettoria social esterna apporti un contributo al punteggio di similarità finale. In caso per l'evento social sia presente una corrispondenza temporale con un reale campionamento della traiettoria anonima, il contributo è dato dalla loro relazione; qualora invece non vi siano corrispondenze temporali, nemmeno considerando la finestra di tolleranza, esso viene calcolato confrontando l'evento con l'abitudine complessiva del soggetto. Ogni volta che un campionamento della traccia social viene analizzato e il suo punteggio rispetto alla traccia GPS viene calcolato, questo si somma al punteggio globale, attribuendogli un peso logaritmico. Una volta scorsi tutti i punti della traiettoria legata a un profi-

lo Twitter, si ottiene il punteggio di similarità rispetto alla traccia anonima. Dopo aver calcolati i punteggi di tutte le traiettorie social rispetto alla singola traccia anonima, quella con il valore migliore è quella che l'algoritmo assegna con maggiore probabilità all'identità ignota.

4.3 Approccio basato su staypoint

Dopo l'analisi dei possibili approcci alla de-anonimizzazione presenti in letteratura e l'implementazione di due algoritmi profondamente diversi ma assimilabili allo stato dell'arte, si è studiata una nuova proposta per la risoluzione del problema. Nel fare ciò si è fatto riferimento ad alcune osservazioni derivanti dall'analisi approfondita delle tecniche citate precedentemente e dei loro risultati sui dati reali, in particolare:

- un approccio basato su match/corrispondenze spaziotemporali può risultare efficace nonostante la sparsità di dati, soprattutto per quanto riguarda quelli relativi ad eventi esterni (check-in, tweet, ecc.);

- un criterio di match stringente come quello proposto da [7] può risultare però eccessivamente rigido: a causa di possibili imprecisioni o mancanze nei dati, possono verificarsi errori sia in positivo (falsi match) che in negativo (false esclusioni). In particolare un solo mismatch spaziotemporale, potenzialmente dovuto a un errore o distorsione dei dati, può causare l'esclusione di una possibile coppia di tracce da parte dell'algoritmo;

- applicare un criterio di esclusione particolarmente selettivo e diretto può portare a risultati incorretti anche in relazione al fatto che una persona, per i motivi esposti in precedenza, può decidere di localizzarsi in luoghi diversi da quello in cui si trova effettivamente;

- l'analisi dei risultati di [7] ha dimostrato come non tutti i match spaziotemporali abbiano la stessa valenza in termini di potere informativo: una corrispondenza avvenuta in un punto e un momento densi di match ri-

sulta meno discriminante rispetto ad una spazialmente e temporalmente isolata;

- l'analisi dei risultati di [13] evidenzia come la stima della traiettoria completa di un utente basata sulle sue posizioni storiche, nel caso in cui queste siano legate a un numero basso di punti o coprano un'area geografica limitata, possono causare una modellazione distorta del comportamento;

- non valutare la lunghezza delle tracce quando si calcola un punteggio di similarità può influire in maniera rilevante sui risultati. Quando i punteggi di similarità rispetto a una traiettoria anonima dei profili social vengono calcolati aggregando i contributi degli eventi che le compongono, non prevedere fattori di normalizzazione può portare a incongruenze nei risultati di traiettorie lunghe rispetto ad altre più corte, o viceversa;

- all'interno delle tracce GPS anonime molti punti campionati possono risultare poco rilevanti per l'analisi e inficiare le prestazioni di un algoritmo di de-anonimizzazione.

Questi fattori, emersi durante la fase di test degli algoritmi, verranno esposti in dettaglio successivamente nei capitoli 5.3 e 5.4.

4.3.1 Nozione di staypoint

Tra i fattori individuati come possibili cause di performance non ottimali degli algoritmi di de-anonimizzazione si è citata la scarsa utilità nel dare rilevanza a tutti i punti di una traiettoria anonima, senza indagare se all'interno della traccia stessa alcuni campionamenti possano essere considerati più rilevanti o rappresentativi. Una possibilità per gestire questa problematica che può essere esplorata è quella dell'utilizzo degli staypoint. Prima di definire con precisione la nozione di staypoint, è utile introdurre alcuni concetti correlati, in particolare quello di *personal gazzetteer* e di algoritmo di clustering basato sulla densità.

Personal gazetteer I personal gazetteer [17] registrano le locazioni o luoghi rilevanti per un soggetto. Individuare e identificare location piuttosto che punti nello spazio permette di attribuire maggior significato ai risultati ottenuti, ad esempio associando un valore semantico alle locazioni. Un personal gazetteer può quindi rappresentare luoghi importanti per un soggetto come la sua abitazione, posto di lavoro, luoghi ricreativi e abitudinari, che possono poi risultare utili per applicazioni che utilizzano la conoscenza dell'utente per personalizzare i propri servizi.

Clustering basato su densità È possibile estrarre questo tipo di rappresentazioni anche dai dati di traiettoria, utilizzando un algoritmo di clustering basato sulla densità. In generale, gli algoritmi di clustering hanno lo scopo di individuare, in modo non supervisionato, gruppi di elementi tra loro simili (*cluster*) all'interno di una collezione di dati. Una possibile soluzione per risolvere il problema è individuare i cluster in base alla densità dei punti. Un riferimento tra gli algoritmi di questa tipologia è *DBSCAN* [20]. Questo algoritmo assegna i punti ai cluster basandosi sulla densità dell'area circostante; in particolare il concetto di densità viene definito in funzione di due parametri:

- *Eps*, ovvero il raggio dell'area da considerare attorno al punto;
- *MinPts*, ovvero il numero minimo di punti all'interno dell'area per raggiungere la soglia di densità richiesta.

Definita la densità, l'algoritmo opera classificando i punti in tre tipologie:

- *core*, ossia punti il cui vicinato ha densità superiore alla soglia minima, e che sono interni a un cluster;
- *border*, ossia punti il cui vicinato non raggiunge la soglia di densità minima ma che si trovano vicini ad almeno un punto core (a distanza inferiore a *Eps*), venendo assegnati al suo stesso cluster;
- rumore, ossia punti che si trovano in aree a bassa densità e distanti da un punto core, non vengono assegnati a nessun cluster.

I punti rumorosi vengono scartati dall'algoritmo DBSCAN, che non li assegna ad alcun raggruppamento: la tecnica è quindi definita come incompleta. In letteratura sono presenti anche proposte alternative per il calcolo dei gazetteer tramite algoritmi di clustering basati su densità, come *DJ-Cluster* [17].

Staypoint Il concetto di staypoint è strettamente legato a quello di personal gazetteer. Lo staypoint è infatti un'elaborazione della traiettoria complessiva di un soggetto che permettere di far emergere eventuali punti per esso rilevanti. In particolare per l'estrazione degli staypoint precedentemente effettuata sui dati oggetto di questo studio si era scelto di selezionare i punti spaziali in cui il soggetto aveva trascorso complessivamente più di venti minuti nell'arco temporale coperto dai dati a disposizione. Si era inoltre effettuata una discretizzazione dello spazio in celle di dimensione approssimabile a 110×80 metri. L'estrazione degli staypoint a partire dalle traiettorie era poi stata concretizzata in un'implementazione in Spark di *PatchWork* [18], algoritmo di clustering distribuito pensato per gestire grandi quantità di dati. L'algoritmo definisce un'area di interesse, ovvero particolarmente densa: si sceglie una cella iniziale e la si assegna al cluster. Si espande poi il cluster aggiungendo celle limitrofe con densità sufficiente. Una volta terminata la fase di espansione del cluster il relativo staypoint viene calcolato come punto medio di tutti quelli compresi nelle celle incluse nel cluster. Attraverso delle apposite regole i punti eventualmente individuati sono stati etichettati come:

- *LivesIn*: punto in cui viene trascorsa la quantità maggiore di tempo nella fascia oraria 2-6, che viene quindi classificato come luogo in cui il soggetto vive e abita;
- *WorksIn*: punto in cui viene trascorsa la quantità maggiore di tempo nella fascia oraria 8-12, che viene identificato come luogo di lavoro del soggetto;
- *Frequents*: zero, uno o più punti identificati come staypoint senza però ricadere nelle due definizioni precedenti, vengono riconosciuti come punti frequentati dal soggetto e quindi rilevanti.

Si tratta quindi di un'elaborazione volta ad estrarre punti rilevanti da una traiettoria attribuendo loro anche un preciso valore semantico. A prescindere dalla specifica etichetta attribuita alla singola location, tutti gli staypoint estratti elaborando una traccia, rappresentando i suoi punti frequenti, possono anche essere considerati come punti salienti in grado di caratterizzare la traccia stessa.

4.3.2 Caratteristiche dell'algoritmo

Si espongono ora le caratteristiche principali della tecnica di de-anonimizzazione proposta, a cui d'ora in avanti si farà riferimento come *Distributed Deanonimization of Trajectories (DDT).*

Utilizzo degli staypoint Come discusso precedentemente, la considerazione di tutti i campionamenti che compongono un traiettoria è un fattore da considerare con attenzione in fase di de-anonimizzazione. Questo nuovo approccio si basa quindi sull'assunzione che i rilevamenti avvenuti in punti poco rilevanti o infrequenti possano costituire sostanzialmente rumore ai fini della risoluzione del problema. Si prevede quindi di considerare solo i campionamenti relativi a punti frequenti per il soggetto osservato. Per fare ciò si è deciso di utilizzare come punti rilevanti ai fini della de-anonimizzazione gli staypoint relativi a una traiettoria, escludendo quindi dall'analisi tutti i campionamenti non collegati ad essi. Come riportato precedentemente gli staypoint vengono estratti tramite un processo di clustering, che aggrega campionamenti frequenti e ne classifica il punto medio come staypoint: ognuno di questi punti rilevanti è quindi associato a più campionamenti singoli. L'algoritmo modella la traiettoria anonima complessiva rappresentandola tramite i campionamenti relativi a staypoint, scartando tutti gli altri punti, che vengono considerati poco rilevanti essendo collegati a locazioni temporanee o transitorie. Questa assunzione permette di escludere singoli punti che però potrebbero condizionare la de-anonimizzazione, ad esempio come per la condizione di esclusione in Cecaj *et al.* [7] o la stima della traccia completa in Wang *et al.* [13].

Tolleranza dei mismatch spaziotemporali Si è precedentemente analizzato come i dati di traiettoria possano essere caratterizzati da incongruenze e mismatch spaziotemporali, e come questo aspetto possa essere rilevante in fase di design di un algoritmo di de-anonimizzazione [13]. Il nuovo approccio prevede un certo grado di tolleranza di eventuali mismatch presenti nelle informazioni a disposizione dell'attaccante. In particolare i valori di distanza spaziale e temporale utilizzati per valutare la similarità tra due traiettorie vengono applicati a un modello gaussiano che permette di gestire intervalli di tolleranza variabili.

Modellazione del comportamento abituale Una problematica già citata in precedenza è quella della sparsità dei dati, soprattutto per quanto riguarda le traiettorie estratte a partire da un numero limitavo di eventi, ad esempio tweet geo-localizzati. In fase di calcolo del punteggio di similarità tra una traiettoria social e una anonima è quindi possibile che un evento della prima non trovi nessun elemento della seconda con una corrispondenza temporale, anche prevedendo un intervallo di tolleranza. In questo caso l'algoritmo confronta l'evento con il comportamento abituale di quel soggetto, che viene stimato attraverso tutte le osservazioni relative a staypoint che compongono la traccia del soggetto. Questo meccanismo permette, estremizzando, di calcolare un punteggio di similarità anche tra due tracce senza alcuna sovrapposizione temporale, ovvero relative a due periodi completamente disgiunti.

Rilevanza degli staypoint È stato osservato come le corrispondenze spaziotemporali tra eventi di traiettorie diverse possano avere rilevanze diverse. Si sono infatti evidenziati dei casi in cui l'attribuzione di eguale importanza a tutti i match spaziotemporali [7] non permette di interpretare adeguatamente situazioni in cui più soggetti si aggregano in un'area limitata, generando molti eventi in un intervallo di tempo relativamente breve. L'approccio prevede quindi, quando viene calcolato un punteggio tra una coppia di eventi, di attribuirgli una rilevanza legata al contesto spaziale in cui è avvenuta. Questa è legata alla frequenza della locazione in cui il match è avvenuto, calcolata in

funzione di quanto essa è comune e ripetuta all'interno di tutte le traiettorie analizzate.

4.3.3 Formalizzazione

Date le caratteristiche generali esposte, si definisce ora nel dettaglio la struttura dell'algoritmo. Data una traiettoria GPS anonima L e una relativa a un profilo Twitter, indicata con S, con $l_i \in L$ e $s_i \in S$, il punteggio di similarità complessivo è dato da:

$$Sc(S, L) = \sum_{s_i \in S} argmax_{l_i \in L}(Score(s_i, l_i)),$$

dove la funzione $Score(\cdot)$ calcola il valore di similarità tra una coppia di eventi, e per ogni evento social viene considerata la corrispondenza con il punteggio massimo ai fini dell'aggregazione finale. Dato un tweet geo-localizzato s_i si calcola il punteggio di similarità tra esso e tutti i ping GPS l_j relativi a staypoint vicini, in funzione di una soglia di distanza spaziale precedentemente definita. Data una coppia di eventi, il punteggio di similarità è definito come:

$$Score(s_i, l_j) = ts + (1 - t)\frac{s + sw}{2},$$

dove s è funzione della distanza spaziale, t di quella temporale e w rappresenta il valore del comportamento dell'utente GPS riferito alla finestra temporale.

Nelle figure 4.5 vengono mostrati i grafici relativi alle funzioni s e t, che ritornano un valore tra zero e uno in funzione della distanza spaziale o temporale. Il ruolo centrale appartiene a t, che definisce l'influenza della componente spaziale e di quella legata all'abitudine. Infatti se il valore di t derivante dalla distanza temporale è alto, il termine contenente sw influirà meno, viceversa se è basso il secondo termine avrà un peso maggiore. Ciò avviene perché se il punteggio di t è alto conseguentemente la distanza temporale sarà ridotta: esiste quindi una buona corrispondenza temporale tra i due eventi e non è necessario pesare in modo significativo l'abitudine dell'utente. Al contrario, se il valore di t è basso significa che la distanza temporale è alta, e non esiste quindi un buon match a livello temporale tra i due eventi: in questo caso ha

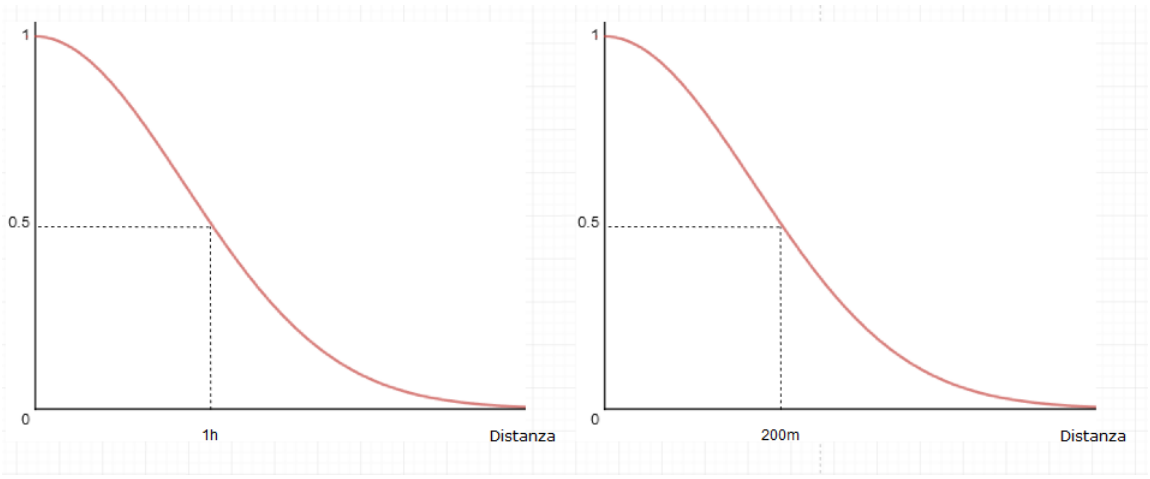


Figura 4.5: Funzioni w e t. Un valore di distanza spaziale di 200 metri equivale a 0,5. Un valore di distanza temporale di un'ora equivale a 0,5.

senso pesare maggiormente la componente che coinvolge l'abitudinarietà del soggetto.

Il valore del comportamento abituale del soggetto è definito come:

$$w = \sum_{h} w_h,$$

mentre w_h è dato da $w_{sp} \cdot gauss_{tw,h}$. Il valore di w_{sp} è dato dal rapporto tra il numero di campionamenti GPS dell'utente nello staypoint nelle fascia oraria h e il totale dei sui ping relativi a quell'ora. Questo valore ha lo scopo di modellare l'abitudine dell'utente, e in particolare la frequenza della sua presenza in uno staypoint in una certa fascia oraria. Come detto, il contributo di questo elemento viene pesato in maniera inversa rispetto a t.

Per evitare i possibili errori dovuti a meccanismi di esclusione troppo stringenti, non sono state introdotte condizioni che portino a scartare in modo immediato una possibile coppia di tracce. La distanza spaziale e quella temporale sono però pesate applicando funzioni gaussiane, che annullano il contributo dell'elemento qualora la distanza sia alta, ammettendo un intervallo di tolleranza all'interno del quale il contributo decresce all'aumentare della distanza. Questa soluzione permette di gestire eventuali e limitati mismatch spaziotem-

porali senza penalizzare troppo l'elemento considerato, e viene applicata anche al calcolo del contributo del comportamento abituale.

Implementazione Per l'implementazione dell'algoritmo è stata utilizzata la piattaforma Spark, volendo realizzare un'implementazione del nuovo approccio alla de-anonimizzaione scalabile e prestante. Si presenta una rappresentazione di alto livello in pseudo-codice. Nella fase preliminare si calcola la rilevanza

```
1  Dati: una traccia anonima, una o più tracce social.
2  Risultato: Numero di match tra la traccia anonima e le tracce social.
3  Funzione CalcoloPunteggio = {
4      Calcola e memorizza la rilevanza degli staypoint
5      Calcola e memorizza il comportamento abituale del soggetto anonimo
6      Per ogni punto di una traiettoria social vicino a uno staypoint:
7          Calcola e memorizza la distanza spaziotemporale tra i punti
8      Per ogni combinazione di punti:
9          Calcola e memorizza i punteggio di similarità della coppia
10         Applica la rilevanza dello staypoint di riferimento
11     Per ogni tweet:
12         Estrai la corrispondenza con uno staypoint con il punteggio massimo
13     Per ogni traccia social:
14         Somma i punteggi delle migliori corrispondenze
15     Restituisce il punteggio complessivo di ogni utente social rispetto alla traiettoria anonima
16 }
```

degli staypoint. In particolare ogni staypoint viene associato a un valore che è dato da uno fratto il numero di elementi nel vicinato. Il vicinato di uno staypoint viene calcolato estraendo tutti gli altri punti rilevanti che si trovano entro una soglia di distanza, definita in fase di inizializzazione.

Si procede poi al calcolo della rappresentazione del comportamento abituale del soggetto associato alla traiettoria GPS anonima. In particolare per ogni ora del giorno si vuole modellare l'abitudine in termini di posizionamento spaziale. I riferimenti temporali relativi a singoli punti vengono riportati a una granularità oraria, poi per ogni ora si calcola la frequenza degli staypoint come rapporto tra il numero di campionamenti in quel punto e il totale di quella fascia oraria.

Una volta calcolati questi valori preliminari, vengono estratte tutte le combinazioni di eventi rilevanti per l'algoritmo, ovvero per ogni evento social si

cercano i campionamenti della traccia anonima effettuati presso staypoint vicini, in funzione di una soglia di distanza spaziale definita. Di queste combinazioni vengono memorizzate le distanze spaziali e temporali. Dopo aver estratto tutte le combinazioni di eventi, a ognuna viene assegnato un punteggio di similarità, secondo la funzione precedentemente definita, che considera la distanza spaziale, quella temporale, e la componente derivante dal comportamento abituale del soggetto. Il punteggio calcolato viene ulteriormente elaborato, moltiplicandolo per il peso dato dalla rilevanza dello staypoint presso cui la corrispondenza si è verificata. Una volta calcolato il punteggio finale di ogni combinazione di eventi, tramite operazioni di aggregazione si ottengono i risultati complessivi di similarità tra la traiettoria anonima e quelle relative ai profili social. In particolare per ogni tweet viene mantenuta la corrispondenza con l'elemento della traccia anonima con il punteggio più alto. Queste corrispondenze, che rappresentano le migliori combinazioni tra tutti i singoli eventi social e i campionamenti GPS, vengono poi aggregate per profilo Twitter, sommando i punteggi, e ottenendo quindi quello complessivo dell'account rispetto alla traiettoria anonima.

Capitolo 5

Testing

5.1 Analisi dei dati

Per tutte le sperimentazioni eseguite nell'ambito del lavoro di tesi sono stati utilizzati dataset appartenenti a due macro-categorie:

- traiettorie anonime campionate tramite dispositivi mobili dotati di GPS;
- traiettorie collegate a profili Twitter, costruite a partire dai tweet geolocalizzati effettuati dagli utenti.

Il periodo di tempo di interesse dello studio coincide con quello coperto dalle traiettorie GPS. Esse erano già disponibili presso il gruppo di ricerca, e riferite a un arco di tempo di circa tre mesi, tra settembre e dicembre del 2017. Per la costruzione delle traiettorie legate agli utenti Twitter si è quindi fatto riferimento al medesimo periodo. Per quanto riguarda i confini spaziali dello studio, si è considerata l'area della città di Milano, giudicata di particolare interesse per la ricerca anche in relazione alla composizione del dataset di traiettorie. I confini precisi dell'area di interesse sono stati infine stabiliti empiricamente a partire da alcune analisi sui dati. In particolare si è osservata la distribuzione degli staypoint sull'area della provincia di Milano, considerata un buon indicatore della distribuzione generale delle traiettorie nello spazio e visibile in figura 5.1. Come prevedibile la maggior concentrazione di punti viene raggiunta nel

centro dell'area urbana di Milano, ma sono evidenti anche altre aggregazioni rilevanti di punti, spesso in corrispondenza degli altri centri urbani nei dintorni della città e della provincia. Volendo includere nell'analisi anche le traiettorie relative a questi punti e consci della tendenza della popolazione a concentrarsi nella zona urbana di Milano anche a partire dalle aree limitrofe, si è quindi deciso di estendere i confini dell'area di interesse oltre a quelli della città. Per la precisione è stata scelta l'area corrispondente al quadrilatero compreso tra i punti di coordinate (latitudine, longitudine): (45.3, 8.8), (45.6, 8.8), (45.3, 9.5), (45.6, 9.5).

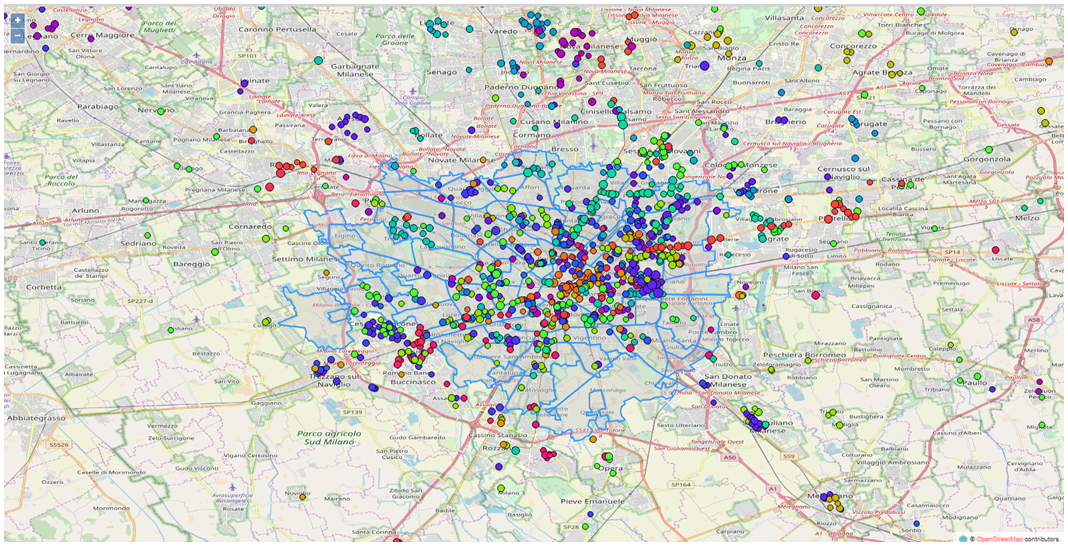


Figura 5.1: Distribuzione degli staypoint sull'area di Milano. Le aree caratterizzate da punti di colore rosso e arancio sono più dense di staypoint, mentre quelle in verde e blu hanno una densità inferiore.

5.1.1 Dataset di traiettorie GPS

Per la costruzione del dataset di traiettorie anonime da utilizzare nello studio, si è quindi estratta una porzione del dataset complessivo di tracce, eseguendo una selezione su base geografica per considerare solo quelle comprese all'interno dell'area di interesse appena descritta. Una volta stabiliti

gli estremi dell'area geografica, si è definita la granularità dei dati, anche in relazione alle caratteristiche degli algoritmi da eseguire su di essi. In termini spaziali, seguendo un approccio comune in letteratura, è stata effettuata una discretizzazione sui riferimenti spaziali arrotondando al terzo decimale le coordinate, suddividendo di fatto lo spazio in una griglia con celle di lato uguale a un millesimo di grado di coordinata. Sono quindi state definite delle celle di estensione di circa 110 metri sull'asse latitudinale e 80 metri su quello longitudinale. Anche a livello temporale si è deciso di operare una discretizzazione, stabilendo una granularità oraria.

5.1.2 Dataset Twitter

Il dataset di traiettorie esterne da collegare a quelle anonime già disponibili è stato costruito a partire da dati relativi ad attività sui social network. In particolare si è deciso di estrarre i dati di partenza da Twitter. Questa scelta è dovuta principalmente al fatto che Twitter risulta essere, tra i social network più utilizzati nell'area di interesse, quello che permette di raccogliere una quantità maggiore di dati tramite le API ufficiali o strumenti esterni. In figura 5.2 è possibile osservare come Twitter sia utilizzato da circa un terzo degli utenti presi in considerazione dall'indagine statistica. Un ulteriore dato da considerare è quello dell'utilizzo della geo-localizzazione quando viene postato un tweet, poiché è solo da questi eventi che possono essere estratte delle traiettorie. Risulta più complesso trovare un'indicazione univoca riguardo a questo dato, in [22] si riporta una percentuale di utilizzo della localizzazione tra il $0,5\%$ e il 3% . Si tratta quindi di una porzione limitata del numero totale di tweet. A questi eventi geo-referenziati generati da tweet postati direttamente sulla piattaforma possono però essere aggiunti quelli originati da altre applicazioni e poi condivisi su Twitter. È infatti possibile collegare gli account di altri social network e applicazioni al proprio profilo Twitter, prevedendo che eventuali azioni eseguite su altre piattaforme vengano condivise anche tramite esso. Questo permette di poter estratte da Twitter eventi geo-localizzati legati anche ad altre piattaforme social location-based, ad esempio Instagram,

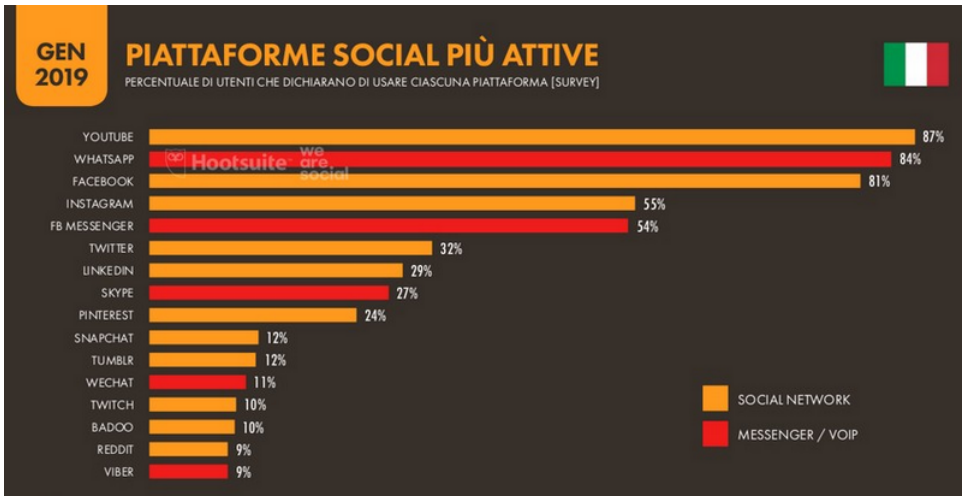


Figura 5.2: Percentuale di utenti che dichiarano di utilizzare ciascuna piattaforma social, fonte Hootsuite.

Foursquare e Swarm. Per l'effettivo download dei dati Twitter necessari per costruire le traiettorie è stato utilizzato Twint, uno strumento realizzato in Python che permette di semplificare la formulazione delle query rispetto alle API ufficiali disponibili. Sono quindi stati scaricati tutti gli eventi localizzati all'interno dell'area geografica di interesse, considerando una finestra temporale compresa tra il 15 settembre 2017 e il 31 dicembre dello stesso anno. I riferimenti spaziali legati ai tweet sono poi stati arrotondati al terzo decimale, discretizzandoli quindi in una griglia con le stesse caratteristiche esposte in precedenza. Fatto ciò gli eventi sono stati aggregati in base al profilo Twitter di riferimento, ottenendo così le traiettorie relative ai singoli account.

5.2 Osservazioni sui dati

Dopo aver eseguito le operazioni di selezione e discretizzazione sul dataset di traiettorie GPS, e aver costruito quelle relative ai profili social, sono state effettuate alcune esplorazioni preliminari dei dati. Alcune informazioni riguardati i dataset sono presentate in tabella 5.1.

Tabella 5.1: Statistiche principali dei dataset utilizzati. La dicitura "utenti mobili" fa riferimento a profili Twitter con eventi localizzati in almeno tre punti spazialmente distinti.

Parametro	Valore
Traiettorie anonime totali	69 237
Traiettorie lunghezza > di 20	58 076
Staypoint	104 361
Tweet nel periodo	73 601
Utenti Twitter nel periodo	11 515
Tweet di utenti mobili	43 106
Utenti Twitter mobili	2 701

Un primo aspetto che è stato considerato è quello della lunghezza delle traiettorie. Per quanto riguarda quelle GPS, osservando la distribuzione dei dati, si è deciso di escludere dall'analisi quelle di lunghezza inferiore a venti, non indicative e con poche informazioni utili per gli algoritmi. Questo ha portato all'esclusione di circa undicimila traiettorie, come si evince dalla tabella.

La figura 5.3 mostra l'istogramma che rappresenta la distribuzione degli utenti Twitter in base al numero di tweet geo-localizzati rilevati nell'area di interesse, e di conseguenza la lunghezza delle traiettorie dedotte dai dati ed associate ai profili social. Appare evidente come gran parte degli utenti siano caratterizzati da un numero molto basso di eventi, spesso inferiore a tre. Tenendo conto del numero di segmenti orari in cui è suddivisibile la finestra temporale oggetto di studio, di circa tre mesi, risulta chiaro come le traiettorie estratte dai dati Twitter siano fortemente incomplete. Un altro fattore rilevante è quello legato alla staticità degli account Twitter: analizzando i dati, presentati in figura 5.4, si può notare come molti utenti tendano a produrre i propri tweet geo-referenziati da un numero limitato di posizioni.

Eseguendo ulteriori indagini sui dati è emerso che spesso questo tipo di comportamento, con un rapporto molto sbilanciato tra il numero di tweet eseguiti e quello dei punti distinti dai quali vengono condivisi, è legato nella

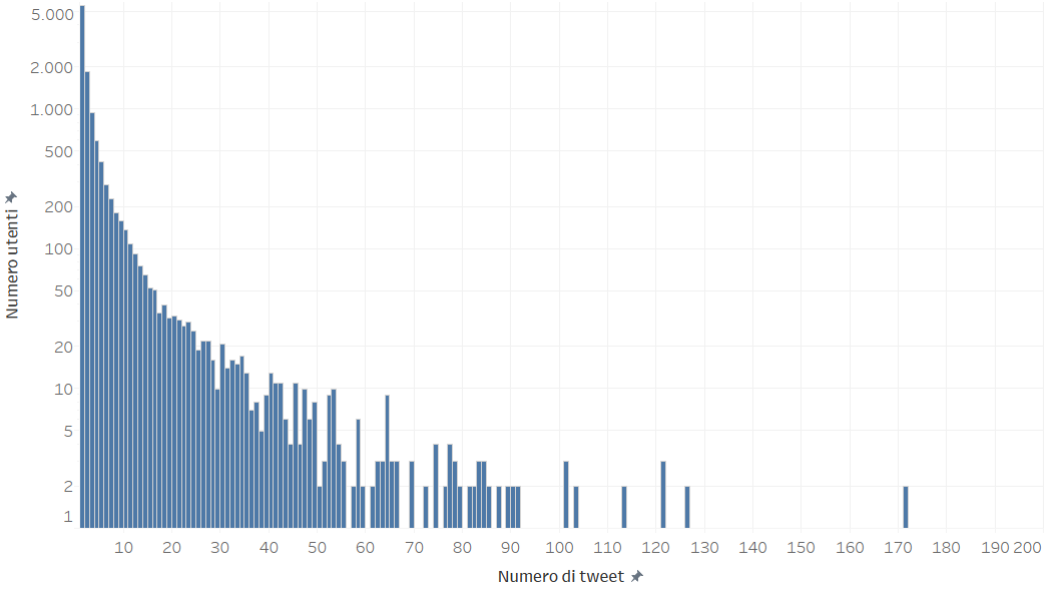


Figura 5.3: Istogramma che mostra la distribuzione degli utenti Twitter in base al numero di tweet geo-localizzati osservati nel periodo di tempo di interesse.

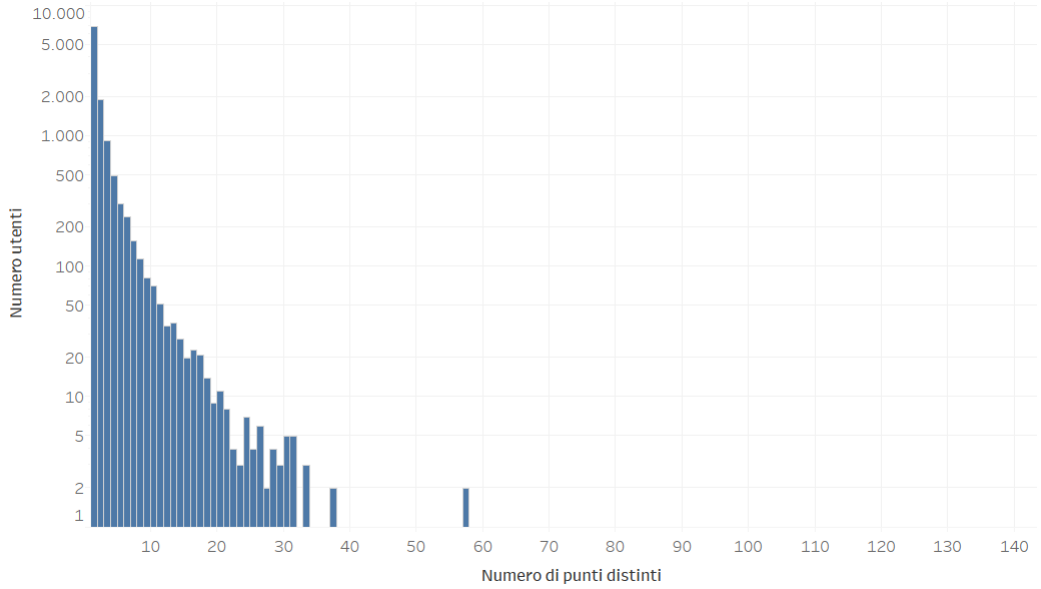


Figura 5.4: Istogramma che mostra la distribuzione degli utenti Twitter in base al numero di punti distinti da cui tweet geo-localizzati sono stati effettuati.

maggior parte dei casi ad account non personali e statici. Si fa quindi riferimento a profili pubblici, aziendali, commerciali. Questo tipo di profilo tende a presentare tweet, anche in quantità elevate, localizzati sempre in un numero limitato di punti, spesso anche uno solo. Questo si può spiegare, ad esempio, con l'utilizzo di un account relativo a un'attività commerciale, che viene quindi impiegato per condividere informazioni ed eventi relativi al negozio, sempre dalla stessa posizione o indicando manualmente la location quando si carica il tweet. Tramite Twitter è infatti possibile geo-localizzarsi non in modo puntuale ma posizionandosi presso una location specifica, che restituirà quindi sempre le medesime coordinate: si tratta di una modalità di utilizzo del social network frequente da parte degli utenti, ad esempio quando vogliono condividere la propria presenza in un locale o un altro luogo pubblico. La stessa distorsione è stata riscontrata relativamente a particolari account informativi, che pubblicano molti tweet, ad esempio con informazioni relative alle condizioni meteorologiche o a notizie locali. Questi account presentano un numero molto elevato di tweet nel periodo di tempo (centinaia o anche migliaia) localizzati sempre nello stesso punto, molto probabilmente perché pubblicati con la generica localizzazione su Milano, che il sistema posiziona automaticamente sempre nello stesso punto che è stato associato alla città.

A partire da queste considerazioni si è quindi valutato come gli account Twitter con eventi osservati da un numero di punti distinti inferiore a tre, anche in presenza di un numero elevato di tweet, non avessero particolare rilevanza per la de-anonimizzazione, trattandosi spesso di account commerciali, informativi o comunque non personali. Si è quindi deciso di escluderli dall'analisi, andando di conseguenza a considerare solo traiettorie composte da almeno tre campionamenti. Ciò ha portato, come si evince dalla tabella, all'esclusione di molti profili Twitter, che però erano associati a traiettorie di lunghezza uguale a uno o due, che possono dunque essere considerate poco indicative relativamente al problema della de-anonimizzazione.

5.3 Test di Cecaj *et al.*

Questo algoritmo si è dimostrato sufficientemente efficiente, nella sua implementazione, per essere testato su tutti i dati disponibili. Dall'analisi dei risultati sono emerse alcune considerazioni interessanti:

- per un numero elevato di traiettorie GPS (66%) l'algoritmo non restituisce account Twitter con un numero di match maggiore di zero: questo è spiegabile con la vera e propria mancanza di esatte corrispondenze spaziotemporali oppure con la presenza di tracce con cui si sono verificati match che sono poi state scartate per l'applicazione della condizione di esclusione;

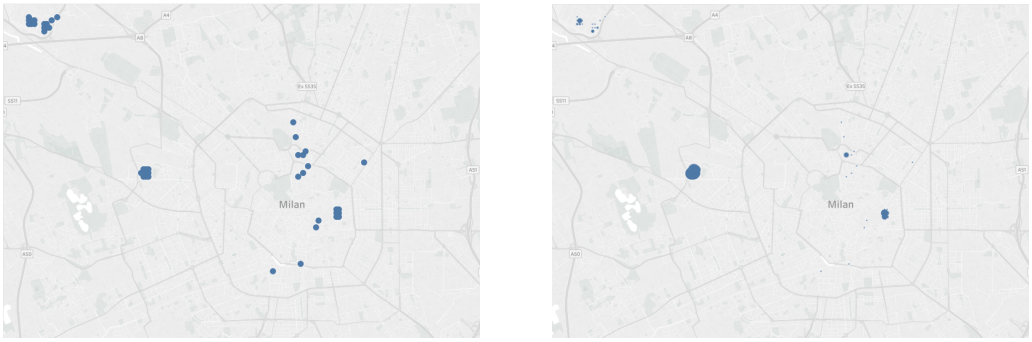
- in molti casi, anche in presenza di account con punteggi positivi, il numero è limitato e quindi poco rilevante in termini probabilistici: solo lo 0.9% delle traiettorie anonime viene collegata ad almeno un profilo Twitter con un numero di match spazio temporali maggiore o uguale a tre, valore indicato dagli stessi autori [7] come significativo;

- un insieme limitato di tracce GPS (0.08%) viene associato dall'algoritmo a un solo account Twitter con un punteggio ancora più rilevante (maggiore o uguale a quattro): questi casi sono quelli probabilisticamente più certi e di interesse per eseguire ulteriori analisi sui risultati;

- si evidenzia un gruppo di tracce che vengono associate con un numero rilevante di match e valori simili o uguali a più account Twitter: in questa situazione il risultato risulta difficilmente interpretabile, ma appare statisticamente improbabile che più account social, collegati a persone diverse, possano presentare valori di match spaziotemporali così rilevanti su un periodo di tempo limitato.

Nello specifico l'ultima osservazione sui risultati ha portato ad ulteriori analisi sui dati per trovare una spiegazione più precisa al fenomeno. Per prima cosa è stato selezionato un campione limitato di traiettorie caratterizzate da un risultato non univocamente interpretabile, ovvero con più account Twitter

con un numero uguale o molto simile di match spaziotemporali. Osservando i risultati nel complesso, è emerso che per buona parte di queste traiettorie incerte gli account Twitter coinvolti fossero i medesimi, costituendo un campione di circa dieci profili ricorrenti.



(a) Posizionamento dei match. (b) Quantità dei match.

Figura 5.5: Posizionamento dei match spaziotemporali sul campione di traiettorie incerte. Nella seconda figura la dimensione degli indicatori è legata al numero di match riscontrati nel punto geografico.

L'algoritmo è stato eseguito di nuovo su queste traiettorie specifiche, memorizzando però anche i punti geografici in cui avvenivano match spaziotemporali. Questi punti sono stati poi visualizzati sulla mappa, visibile in figura 5.5. Appare evidente come i match tendano a raggrupparsi in un numero limitato di aree. Da una semplice ricerca è emerso come quelle locazioni GPS coincidessero con luoghi non casuali, come lo stadio Giuseppe Meazza, la fiera di Milano, Piazza Duomo e la basilica di San Babila. Analizzando ancor più nel dettaglio queste corrispondenze, è emerso come un numero rilevante di esse (superiore a 500) avvenisse nell'area dello stadio e in fasce orarie molto limitate. Continuando l'esplorazione dei dati si è evidenziato un picco del numero di match, con un valore di alcune centinaia, avvenuti in un'area limitata di nove celle, equivalenti a un'area di circa 330×240 metri. Queste celle, esattamente corrispondenti allo stadio, presentavano una considerevole quantità di match spaziotemporali in una fascia oraria di sole tre ore nello stesso giorno, molto limitata rispetto all'intero periodo di tempo considerato. Indagando nello spe-

cifico sulla data è emerso che in quella precisa fascia oraria si giocava la partita tra le due squadre di Milano, evento in grado di concentrare circa 80 000 persone in un'area relativamente ridotta per un intervallo di tempo molto limitato, compatibile con la durata dell'incontro e del tempo necessario per l'afflusso e il deflusso del pubblico dallo stadio. Un fenomeno simile è emerso anche esplorando più nel dettaglio i dati relativi ai match spaziotemporali avvenuti nell'area della fiera, avvenuti per la maggior parte in fascia oraria diurna e in giornate in cui erano effettivamente attive iniziative presso il polo fieristico. Da questa analisi approfondita è quindi emersa la tendenza di pochi luoghi particolarmente popolari a generare molti tweet geo-localizzati e match spaziotemporali. Alcune di queste location, come le già citate Piazza Duomo e Piazza San Babila, tendono ad avere una distribuzione del numero di match su fasce orarie abbastanza ampie: ciò appare ragionevole in quanto questi luoghi risultano essere affollati sia nelle ore diurne che in quelle serali. Esistono poi luoghi in grado di raccogliere molte persone, come la fiera e lo stadio, e generare match in fasce orarie limitate, presentando una densità di campionamenti GPS molto bassa negli altri momenti. Intuitivamente questo è spiegabile con la scarsa presenza di persone in questi luoghi quando non sono previsti eventi. Date queste evidenze, si possono fare alcune considerazioni:

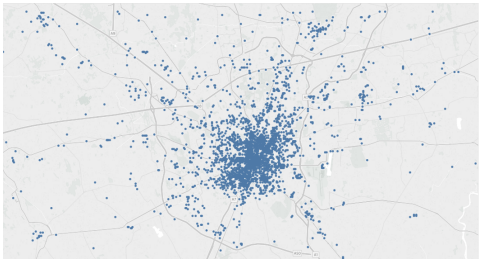
- geo-localizzarsi in location dove anche molti altri utenti lo fanno, potenzialmente in un lasso di tempo limitato, rende più difficile essere collegati a una traccia GPS anonima quando vengono utilizzati algoritmi che basano il risultato non su un'analisi complessiva della traiettoria ma solo sulle corrispondenze di alcuni punti;

- volendo pensare a un approccio che basa il proprio risultato sulle corrispondenze spaziotemporali tra eventi, può essere utile attribuire valori diversi a match avvenuti in luoghi diversi, a seconda della popolarità di questi ultimi

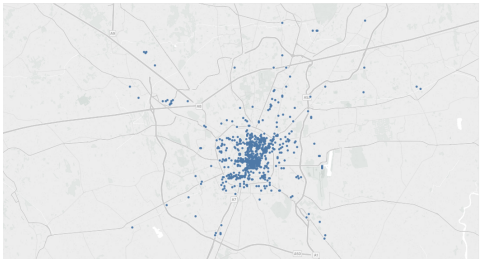
- l'algoritmo di Cecaj *et al.* può presentare situazioni controverse in cui una traccia anonima viene associata con probabilità simili o identiche a

più tracce, portando di fatto a nessuna identificazione significativa: facendo riferimento agli esempi analizzati nel dettaglio, non presentandosi altre situazioni di match spaziotemporale in punti isolati, non è possibile osservando i risultati attribuire con cognizione di causa una delle possibili identità alla traccia anonima: si può solo supporre che sia relativa a una persona che frequenta un luogo popolare, ad esempio lo stadio o Piazza Duomo.

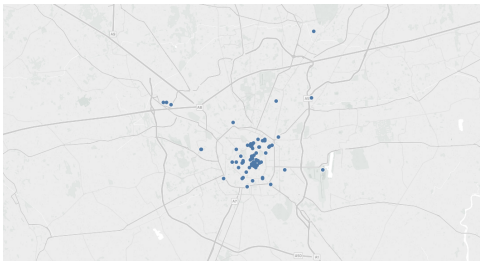
Per ottenere una comprensione più generale di questo tipo di tendenza all'interno dei dati, si è eseguito nuovamente l'algoritmo su tutti i dati, memorizzando i punti geografici dove venivano riscontrati match spaziotemporali. In figura 5.6 sono visibili i risultati di questa analisi.



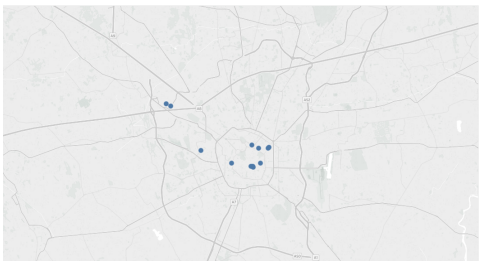
(a) Celle con almeno un match, 100%.



(b) Almeno 100 match, 23.8%.



(c) Almeno 1000 match, 2.8%



(d) Più di 5000 match, 0.44%

Figura 5.6: Mappe delle location in cui si verificano match spaziotemporali, con soglie crescenti ed indicazione della quantità evidenziata rispetto al totale.

Come si evince dalle mappe, quantità significative di match spaziotemporali si verificano in un numero limitato di celle. Analizzandole nello specifico, queste coincidono sempre con luoghi pubblici in grado di riunire grandi quantità di persone: lo stadio, la fiera, Piazza Duomo, San Babila, Piazza Gae

Aulenti, Stazione Centrale, l'aeroporto di Linate, ecc. Ciò avvalora ulteriormente le considerazioni fatte riguardo alla tendenza dei match a costituire dei cluster ben definiti, rendendo di fatto queste corrispondenze poco distintive rispetto agli utenti.

Esclusione Analizzando i risultati dell'algoritmo, una problematica emersa è legata alla condizione di esclusione. È stato eseguito un test per evidenziare i casi in cui a fronte di un numero rilevante di match spaziotemporali, una coppia di traiettorie venisse comunque esclusa dall'analisi a causa di una o più occorrenze della condizione di esclusione. Da questa analisi sono emersi casi in cui un numero sostanziale di match (superiore a cinque) venga annullato da una sola coppia di eventi considerati come incompatibili dall'algoritmo. In figura 5.7 viene presentato un caso limite, che vuole essere rappresentativo della potenziale debolezza di un criterio di esclusione troppo rigido. In questo caso due eventi osservati durante lo stesso slot temporale non risultano essere avvenuti nella stessa cella o in due celle adiacenti: il campionamento GPS si posiziona esattamente al di fuori dell'intorno di raggio uno del tweet. La distanza effettiva è modesta (circa 230 metri), e come trattato in precedenza sono molteplici i fattori che potrebbero aver causato una lieve alterazione dei dati, anche di pochi metri, banalmente un errore del satellite. In questo caso l'algoritmo esclude in modo netto una coppia potenzialmente valida per la mancata coincidenza di due eventi a una distanza modesta. Questo caso specifico potrebbe essere gestito semplicemente aumentando la dimensione delle celle o prevedendo un intorno di tolleranza più ampio, ma l'esempio è volto a evidenziare come un meccanismo di esclusione rigido, anche a fronte di un numero rilevante di corrispondenze, possa portare a risultati intuitivamente errati.

Figura 5.7: Due eventi contemporanei (tweet di colore giallo e campionamento GPS in blu) ma spazialmente separati da una cella: nonostante la relativa vicinanza, si genera un'esclusione totale della coppia.

Riassumendo, i test effettuati con Cecaj *et al.* evidenziano che:

- un punteggio basato solo su precise corrispondenze spaziotemporali porta a non ottenere risultati per la maggior parte delle traiettorie social;
- i punteggi molto bassi, solitamente inferiori a tre, non si rivelano particolarmente rilevanti;
- data la tendenza delle persone a concentrarsi in luoghi specifici e in fasce orarie limitate, esistono situazioni dense di eventi in cui l'algoritmo non riesce a distinguere chiaramente gli utenti, producendo risultati difficilmente intellegibili;
- il criterio di esclusione risulta particolarmente rigido e diretto, portando ad escludere risultati potenzialmente rilevanti a causa di mismatch modesti.

5.4 Confronto tra Cecaj *et al.* e Wang *et al.*

L'elevato costo computazionale dell'algoritmo di Wang *et al.* [13] ha reso proibitiva l'esecuzione dello stesso su tutti i dati a disposizione. La principale

motivazione è probabilmente da ricondurre all'elevato numero di combinazioni possibili tra gli eventi dei due dataset, considerando anche il meccanismo utilizzato per completare le traiettorie anonime, che richiede di considerare per ogni slot temporale senza osservazioni tutte le locazioni storiche dell'utente. Sulle performance viene quindi a pesare l'elevato numero di distanze e valori di probabilità da calcolare. Va inoltre sottolineato che l'implementazione di Wang *et al.* è sequenziale, e che probabilmente possiede margini di ottimizzazione importanti. Il testing e l'analisi dei risultati dell'algoritmo sono quindi stati concentrati su un sottoinsieme delle traiettorie anonime disponibili. Questa porzione del dataset è stata individuata a partire dai risultati ottenuti dall'algoritmo di Cecaj *et al.*.

Per comprendere maggiormente il comportamento e le caratteristiche dell'algoritmo si sono analizzati nel dettaglio i risultati relativi a traiettorie anonime già emerse e considerate durante la valutazione dei risultati di Cecaj *et al.* In particolare, analizzando i risultati delle tracce con maggior confidenza secondo Cecaj *et al.*, si è riscontrato come spesso il risultato atteso non coincidesse con quello prodotto da Wang *et al.*. Il secondo algoritmo calcola un punteggio relativo a una traiettoria anonima per ogni profilo social disponibile, producendo poi un elenco di account ordinati secondo la probabilità di corrispondenza associata. In molti casi si è constatato che il profilo più probabile secondo Cecaj *et al.* non figurasse tra le prime posizioni del risultato del secondo algoritmo, principalmente perché preceduto da altri account Twitter, spesso ricorrenti, riconducibili a un gruppo limitato di profili. Analizzando più in dettaglio questi profili frequenti, per la maggior parte di essi è emersa una caratteristica comune. Le traiettorie relative a questi profili era effettivamente composte da eventi localizzati in almeno tre punti spaziali distinti, come stabilito in precedenza, ma tutti generati all'interno dello stesso slot temporale, ovvero nell'arco di una singola ora di un solo giorno. Questa particolare situazione non aveva mostrato ripercussioni sul funzionamento dell'algoritmo precedente, e non era quindi stata osservata. L'algoritmo di Wang *et al.*, come proposto anche nell'implementazione fornita dagli autori [21], considera un solo evento social per ogni slot temporale, selezionando di fatto tra più eventi os-

servati nella stessa ora solo uno di essi. Questi casi limite, con più osservazioni all'interno dello stesso riferimento orario, si traducevano quindi nella costruzione di traiettorie social con un solo elemento. Un'analisi più approfondita del comportamento dell'algoritmo ha evidenziato come questo possa portare a distorsioni nel risultato. L'algoritmo prevede infatti di calcolare il punteggio come somma di valori di probabilità (quindi compresi tra zero e uno), pesati logaritmicamente, che producono valori negativi. Questi valori vengono sommati, e il punteggio migliore risulta essere quello più vicino a zero. L'algoritmo non prevede alcuna normalizzazione rispetto al numero di termini che concorrono alla somma. Per questo i profili con un solo evento presentano un unico valore che concorre alla somma complessiva, con maggiori probabilità di ottenere un punteggio vicino a zero rispetto a una traiettoria caratterizzata da più eventi, e quindi con più elementi partecipanti alla somma. Per esemplificare questo meccanismo, si propone in figura 5.8 un caso reale, graficando i singoli campionamenti GPS della traiettoria anonima e posizionamenti dei tweet.

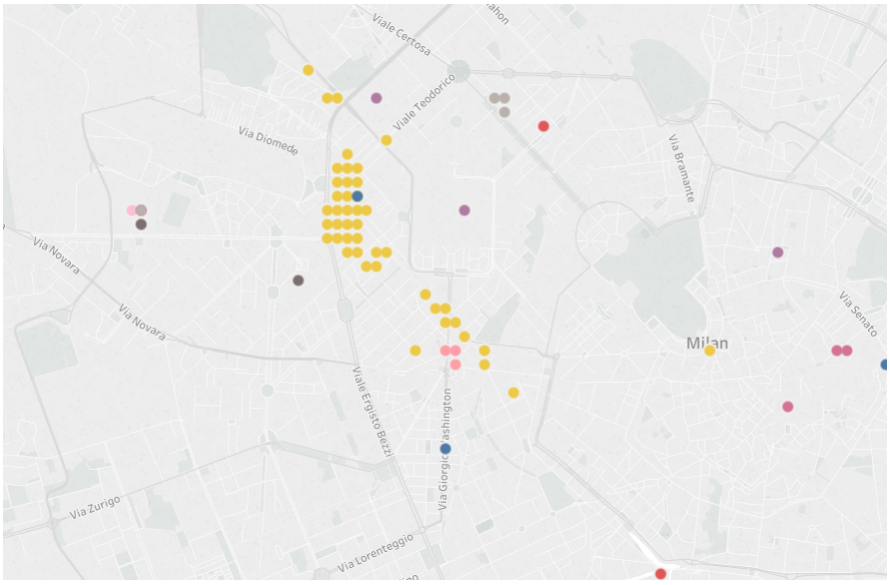


Figura 5.8: In giallo vengono indicati i campionamenti GPS anonimi. L'utente in rosso, considerando un solo campionamento, viene preferito a quello in blu, che però presenta molti match spaziotemporali.

Applicando Cecaj *et al.* l'utente rappresentato in blu evidenzia nove match spaziotemporali, tutti localizzati nello stesso punto ma in slot temporali diversi, con la traiettoria anonima in giallo. L'algoritmo di Wang *et al.* lo classifica però come diciassettesimo profilo più probabile per la traiettoria anonima. A esso vengono preferiti, tra gli altri, tutti i profili mostrati in figura, caratterizzati da un numero limitato di eventi concentrati nel stessi slot temporali. Il profilo evidenziato in rosso, ad esempio, risulta essere il più probabile, in quanto viene considerato dall'algoritmo solo l'evento più vicino alla traiettoria gialla, con un solo valore a concorrere alla somma. Questa analisi ha quindi evidenziato come l'algoritmo posso portare a risultati fortemente influenzati dalla lunghezza delle traiettorie social, non prevedendo alcun fattore di normalizzazione.

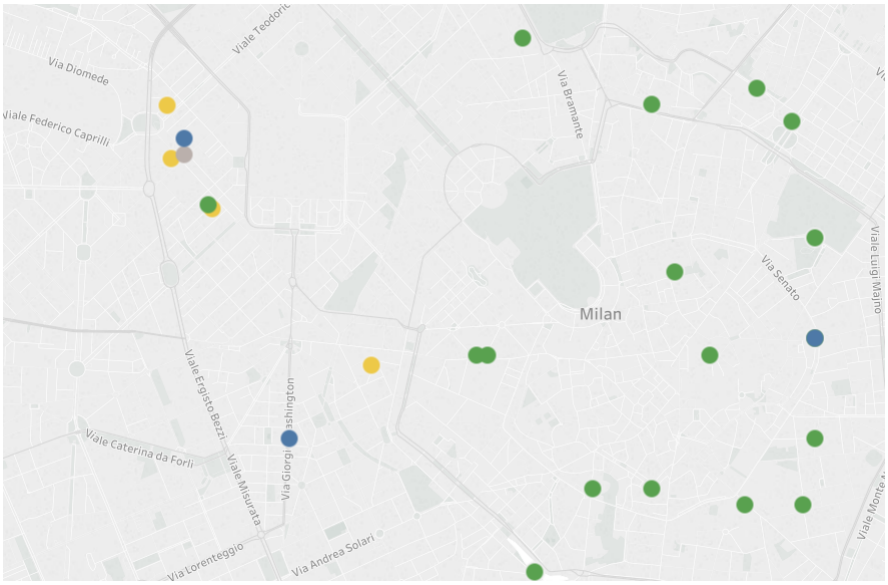


Figura 5.9: L'utente in verde possiede un tweet spazialmente molto vicino a uno degli staypoint della traiettoria GPS, in giallo. Viene quindi preferito al profilo Twitter evidenziato in blu, miglior risultato secondo Cecaj *et al.*

In figura 5.9 si mostra il comportamento del nuovo approccio rispetto al medesimo caso reale: vista la natura dell'algoritmo non vengono graficati i singoli campionamenti come in precedenza, ma gli staypoint relativi alla tra-

iettoria anonima, oltre ai tweet che compongono le tracce social. Rispetto alla traiettoria GPS in giallo, l'algoritmo classifica il risultato atteso, ovvero l'utente in blu, come il secondo più probabile. Il miglior punteggio viene attribuito invece all'utente rappresentato in verde. Analizzando con più attenzione i risultati emerge che la preferenza per questo utente deriva dalla maggior vicinanza spaziale a uno staypoint rispetto all'utente blu. Se ne deduce quindi che l'influenza della componente spaziale risulti molto forte, condizionando il risultato a favore dell'utente.

Un'ulteriore problematica dell'approccio di Wang *et al.* è legata alla stima della traiettoria anonima completa. Come esposto in precedenza, l'algoritmo prevede di stimare la posizione del soggetto anonimo per gli slot temporali non coperti da osservazioni a partire dalle quelle storiche dello stesso, con un grado di probabilità derivato dalla loro frequenza. Se però la traiettoria anonima presenta un numero limitato di locazioni storiche con frequenza molto alta, l'utente otterrà corrispondenze con punteggi elevati anche quando non presenta osservazioni reali, per una considerazione eccessiva del comportamento storico del soggetto. Estremizzando, se un utente GPS evidenzia campionamenti relativi ad un solo punto, che costituisce il suo comportamento abituale con probabilità uguale a uno, quando viene confrontato con eventi social ottiene il medesimo punteggio che otterrebbe da una esatta corrispondenza spaziotemporale.

Per riassumere, i test di Wang *et al.* mostrano che:

- costruire un modello particolarmente elaborato e che richiede di calcolare molte distanze, senza adeguate ottimizzazioni, può portare a prestazioni scarse e soprattutto poco scalabili;

- la mancanza di un fattore di normalizzazione legato al numero di eventi che compongono le traiettorie può influire negativamente sui risultati;

- considerare un solo evento social per time slot può portare a distorsioni ulteriori dei risultati;

- la stima della traiettoria completa di un utente anonimo basata sulla frequenza delle locazioni precedentemente visitate può non rivelarsi ottimale per la modellazione del comportamento di soggetti che vengono osservati in un numero limitato di location.

5.5 Analisi del nuovo approccio proposto

Per valutare il comportamento del nuovo approccio proposto, e relazionarlo agli algoritmi precedentemente implementati, è stato effettuato un test comparativo tra tutti e tre gli approcci. Non avendo a disposizione dati *ground truth*, quindi un effettivo e verificato collegamento tra traiettorie anonime e profili social, è stato individuato un numero limitato di dati da utilizzare come campione per il confronto. Questo sottoinsieme di traiettorie è stato costruito selezionando quelle con i risultati migliori secondo l'algoritmo di Cecaj *et al.* Nello specifico sono state selezionate le cento traiettorie con il grado di confidenza maggiore. Essa è stata calcolata, per ogni traiettoria anonima, come rapporto tra il numero di match spaziotemporali ottenuti dal primo e dal secondo miglior risultato prodotti dall'algoritmo. Le traiettorie con il rapporto maggiore, evidenziando una significativa distanza tra il primo e il secondo classificato, si possono considerare le più certe a livello probabilistico. Le cento traiettorie anonime individuate sono state date in input anche agli altri due algoritmi implementati, per poter poi confrontare i risultati aggregati delle tre tecniche. I risultati esposti nelle tabelle 5.2 sottostanti sono stati ottenuti nel seguente modo. Preso per ogni traiettoria anonima il primo risultato secondo Cecaj *et al.*, si è verificato con che frequenza lo stesso account è presente anche nelle soluzioni fornite da Wang *et al.* e DDT. Questa verifica è stata effettuata ricercando la corrispondenza con diversi intervalli di tolleranza: nei primi cento, cinquanta e dieci risultati, e infine confrontando in modo diretto i primi classificati secondo le due tecniche. Lo stesso approccio è stato applicato per tutte le diverse combinazioni di algoritmi: sono quindi stati estratti i migliori risultati di Wang *et al.* e DDT per ogni traiettoria del campione, per poi confrontarli con quelli ottenuti dagli altri approcci e verificarne le coincidenze.

Tabella 5.2: Confronto tra i risultati dei tre algoritmi implementati. Si valuta il posizionamento degli account più probabili per ciascun algoritmo nei risultati degli altri. Nelle intestazioni: ME = Cecaj *et al.*, GM = Wang *et al.*

	ME
Top100 GM	25%
Top50 GM	22%
Top10 GM	16%
Top1 GM	11%
Top100 DDT	70%
Top50 DDT	61%
Top10 DDT	57%
Top1 DDT	30%

	GM
Top100 ME	14%
Top50 ME	14%
Top10 ME	14%
Top1 ME	11%
Top100 DDT	22%
Top50 DDT	15%
Top10 DDT	12%
Top1 DDT	0%

	DDT
Top100 ME	30%
Top50 ME	30%
Top10 ME	30%
Top1 ME	30%
Top100 GM	4%
Top50 GM	1%
Top10 GM	0%
Top1 GM	0%

Dall'analisi di questi risultati si possono dedurre alcune considerazioni. La profonda differenza tra i primi due approcci implementati si evidenzia anche nei risultati, piuttosto dissimili, con una corrispondenza esatta tra i migliori risultati dei due algoritmi che si attesta all'11%. Emerge inoltre una certa regolarità nei risultati degli algoritmi rispetto a quelli di Cecaj *et al.:* questo è spiegabile dal fatto che le traiettorie scelte per i test sono quelle con maggiore confidenza secondo l'algoritmo, quindi nella larga maggioranza dei casi presentano un solo risultato con un numero di match spaziotemporali maggiore di zero e nessun altro valore positivo, rendendo sostanzialmente indifferente allargare l'intervallo di ricerca della corrispondenza fino a cento o anche oltre. Il nuovo approccio proposto presenta risultati molto superiori a Wang *et al.* rispetto al campione scelto, lasciando intuire che probabilmente il tipo di tecnica definita sia concettualmente più simile a quella di Cecaj *et al.*, che è stata utilizzata per la scelta del campione di test. I risultati raffrontati di Wang *et al.* e DDT si dimostrano invece profondamente diversi evidenziando una certa incompatibilità tra le due tecniche. Infine, occorre sottolineare come DDT non produca un risultato per ogni traiettoria anonima. Nel caso in

cui una traiettoria non possa essere rappresentata estraendo degli staypoint, l'algoritmo non è in grado di calcolare un punteggio di similarità con le tracce social, non potendo sussistere corrispondenze temporali effettive o basate sulla componente abituale.

5.5.1 Analisi delle performance

Si espone in tabella 5.3 una rappresentazione delle performance dei tre algoritmi implementati, applicati su un numero variabile di traiettorie.

Tabella 5.3: Analisi delle performance degli algoritmi implementati, confrontando un numero crescente di traiettorie anonime con il dataset Twitter. Tempi espressi in minuti.

	100	1 000	10 000	20 000
Cecaj *et al.*	< 1	1-2	3-4	4-6
Wang *et al.*	4-6	> 60	> 300	> 600
DDT	3-5	7-8	10-12	12-14

Come prevedibile, Cecaj *et al.* si è dimostrato l'approccio più semplice ed efficiente, in grado di operare su tutti i dati in poche ore. Wang *et al.* ha mostrato performance molto peggiori, rendendo sostanzialmente proibitivo applicare l'algoritmo sull'intero dataset a disposizione: questo fa supporre che il modello markoviano utilizzato per la stima della traiettoria completa dei soggetti anonimi porti a un numero eccessivamente elevato di confronti, che unito ai calcoli di distanze e probabilità da effettuare, rendono la tecnica particolarmente inefficiente. Raffrontare le prestazioni di DDT risulta difficile, trattandosi di un approccio strutturalmente molto diverso dai precedenti e soprattutto per via della sua implementazione sulla piattaforma Spark. Si può però osservare come, nonostante l'utilizzo di una piattaforma così ottimizzata e di un cluster prestante, l'algoritmo presenti comunque prestazioni peggiori di Cecaj *et al.* Ciò è probabilmente da imputare all'elevato numero di

possibili combinazioni che i campionamenti relativi agli staypoint e gli eventi social possono produrre, con un conseguentemente alto numero di distanze e punteggi di similarità da calcolare. Sicuramente l'attuale implementazione dell'algoritmo può essere ottimizzata riducendo il numero di combinazioni da calcolare, soprattutto tenendo conto che per ogni evento social viene poi considerata solo la corrispondenza con il punteggio più alto, mentre tutte le altre vengono scartate nonostante il notevole costo computazionale richiesto per il loro calcolo.

5.5.2 Test della componente abituale

Un ultimo test è stato eseguito per valutare, almeno intuitivamente, una caratteristica di DDT. In particolare si è testata la capacità dell'algoritmo, tramite il proprio meccanismo di valutazione del comportamento abituale rispetto agli staypoint, di trovare buone corrispondenze tra traiettorie relativi a periodi di tempo non sovrapposti, per le quali non è quindi possibile trovare esatte corrispondenze temporali, anche considerando intervalli di tolleranza. È stato applicanto l'algoritmo per de-anonimizzare il campione di traiettorie testato in precedenza, utilizzando però un dataset di tweet relativo all'arco di tempo compreso tra il 15 settembre 2018 e il 30 novembre 2018, esattamente un anno dopo a quello di riferimento delle traiettorie anonime. In questo caso i risultati hanno mostrato punteggi numericamente inferiori in termini assoluti, come prevedibile, essendo le relazioni unicamente basate sulla componente abitudinaria e non su corrispondenze temporali. In più della metà dei casi si sono però evidenziati risultati simili, in termini di ordinamento degli account social proposti in associazione a una traiettoria anonima, a quelli ottenuti utilizzando dati sovrapposti nel tempo. Tale risultato lascia intendere che l'algoritmo sia effettivamente in grado, grazie agli staypoint, di modellare in maniera efficace il comportamento dell'utente GPS, permettendo poi di ottenere risultati sensati anche confrontando il modello con periodi di tempo non coincidenti. Questa capacità è spiegabile con il fatto che il comportamento in termini di luoghi frequenti (abitazione, posto di lavoro, luoghi ricreativi,

ecc.) delle persone risulti particolarmente stabile nel tempo, permettendo a una modellazione di queste abitudini di mantenere una certa validità nel tempo. Un'ottimizzazione di questa feature permetterebbe di aumentare il numero di possibili scenari di de-anonimizzazione, potendo confrontare dati relativi a periodi di tempo diversi: ciò non può essere fatto con la maggior parte delle tecniche di de-anonimizzazione note in letteratura. Un'altra osservazione che può essere fatta è che la perdita di efficacia del modello potrebbe far supporre un cambiamento sostanziale in termini di luoghi frequentemente visitati da una persona, ad esempio a fronte di eventi quali cambi di residenza, di posto di lavoro o di relazioni sociali e affettive. Questo aspetto risulta probabilmente uno degli spunti più interessanti e innovativi per la continuazione dello sviluppo di questo nuovo approccio.

Conclusioni

Il contributo apportato da questo lavoro di tesi si spinge principalmente in due direzioni: l'analisi approfondita dello stato dell'arte per quanto riguarda la de-anonimizzazione di dati di traiettoria e la formulazione di un nuovo approccio per la risoluzione del problema. Per quanto riguarda il primo aspetto, è stato profuso uno sforzo significativo nello studio e nell'analisi delle tecniche note in letteratura, in modo da poterne comprendere i punti di contatto e le differenze. Sono stati individuati due approcci strutturalmente diversi, poi implementati e testati in modo approfondito per analizzarne il comportamento su un dataset reale di notevoli dimensioni. Questa analisi ha permesso di identificare problematiche ancora non del tutto risolte che possono inficiare le prestazioni degli algoritmi di de-anonimizzazione.

A partire dalle considerazioni fatte sugli algoritmi noti in letteratura e sui fattori che ne limitano le performance, sono state isolate alcune caratteristiche utili per la risoluzione di questo tipo di problema. Esse sono state combinate e formalizzate in un nuovo approccio alla de-anonimizzazione, che è poi stato implementato sulla piattaforma Spark, allo scopo di ottenere le migliori prestazioni possibili in termini di efficienza e scalabilità. L'algoritmo è stato testato sui dati a disposizione, ed i risultati sono stati messi in relazioni, nella maniera più oggettiva possibile, con quelli prodotti dalle tecniche precedentemente implementate. Si è infine prodotto un ulteriore sforzo per analizzare e comprendere il comportamento dell'algoritmo, al fine di identificarne i punti di forza e di debolezza, utili per sviluppare ulteriormente la tecnica di de-anonimizzazione.

Volendo fare alcune considerazioni finali di carattere generale, questo lavoro

evidenzia, se mai ce ne fosse bisogno, quanto annoso si riveli il problema della tutela della privacy degli utenti. Nonostante le tecniche di anonimizzazione esistenti, per mantenere almeno una parte del prezioso potere informativo dei dati, in particolare di quelli di traiettoria, si devono accettare i rischi conseguenti alla diffusione di queste informazioni. Dalle analisi effettuate si deduce che, nella maggior parte dei casi, un attaccante sufficientemente motivato possa ottenere le risorse necessarie per effettuare un attacco di de-anonimizzazione con buone probabilità di successo, almeno su parte degli utenti. È stato però osservato come il comportamento consapevole e responsabile degli utenti possa rendere più difficile per un attaccante identificarli in maniera univoca. Questo lavoro può forse portare a fare considerazioni più ampie rispetto all'uso quotidiano che viene fatto dei social network e di Internet, soprattutto per quanto riguarda la condivisione di posizionamento e localizzazione, e più in generale di dati potenzialmente sensibili.

Valutando i possibili sviluppi futuri di quanto svolto nel lavoro di tesi, permangono alcune questioni ancora aperte. Il nuovo approccio alla de-anonimizzazione proposto, dopo le prime sessioni di test effettuate, presenta ancora margini di miglioramento che possono essere ulteriormente formalizzati. In particolare l'influenza delle singole componenti (spazio, tempo, comportamento abituale) può essere ulteriormente raffinata, così come potrebbe essere utile l'introduzione di un fattore di normalizzazione legato al numero di eventi che compongono le traiettorie. Si potrebbe inoltre valutare l'applicazione di un meccanismo che penalizzi, senza però portare ad esclusioni nette, le coppie di traiettorie qualora evidenzino eventi vicini nel tempo ma lontani nello spazio. I risultati ottenuti sui dati a disposizione sono comunque considerati soddisfacenti, vista anche la natura innovativa della tecnica proposta in relazione all'attuale stato dell'arte. Inoltre, per quantificare in maniera maggiormente accurata le performance dell'algoritmo rispetto alle tecniche note in letteratura, l'algoritmo dovrà essere testato su dati per i quali è disponibile un'effettiva ground-truth. Ottenere dataset con queste caratteristiche e dimensioni sufficienti risulta però essere un problema non banale. La caratteristica probabilmente più interessante ed innovativa dell'algoritmo è la possibilità di

costruire una modellazione del comportamento degli utenti basata sui luoghi frequentemente visitati che sembra poter mantenere un certa validità nel tempo, permettendo il confronto anche fra dati relativi a periodi non coincidenti. Sarebbe pertanto molto interessante esplorare ulteriormente questo scenario eseguendo nuovi test e cercando di raffinare ulteriormente il modello. In conclusione, il nuovo approccio alla de-anonimizzazione si è quindi dimostrato idoneo al proseguimento delle attività di ricerca.

Ringraziamenti

Il primo e più grande ringraziamento va ancora una volta ai miei genitori, Mauro e Milena, che con i loro sacrifici hanno reso possibile non solo questi ultimi due anni, ma il mio intero percorso di studi.

Un altro grande ringraziamento va a Beatrice, per avermi sopportato e supportato negli ultimi e particolarmente impegnativi mesi di lavoro, e per essermi sempre stata di sprono durante tutto il percorso, talvolta anche con motivata durezza.

Non posso non ringraziare Daniele e Matteo V. per aver condiviso con me altri due anni di fatiche e studio, durante i quali ci siamo però molto divertiti e che hanno portato momenti di grande soddisfazione; questo risultato è anche frutto del lavoro fatto insieme. Se "anche così è stato breve il nostro lungo viaggio", lo devo soprattutto a loro.

Un ringraziamento speciale va a Enrico e Matteo F., per tutto l'aiuto, il supporto e la pazienza, ma soprattutto per la piacevole compagnia e l'amicizia dimostratami negli ultimi mesi.

Infine, ringrazio il Prof. Golfarelli per avermi dato la possibilità di lavorare a un progetto interessante e stimolante e per tutto l'aiuto, il supporto e la cordialità ricevuti.

Bibliografia

[1] L. Sweeney, *k-anonymity: A model for protecting privacy*, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems,* vol. 10, no. 05, pp. 557–570, 2002.

[2] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, *ACM Transactions on Knowledge Discovery from Data (TKDD),* vol. 1, no. 1, p. 3, *Proc. IEEE ICDE*, 2007.

[3] N. Li, T. Li, and S. Venkatasubramanian, *t-closeness: Privacy beyond k-anonymity and l-diversity*, *Proc. IEEE ICDE*, 2007.

[4] A. Narayanan, V. Shmatikov, *Robust de-anonymization of large sparse datasets*, *Proc. IEEE SP*, 2008.

[5] R. Shokri, G. Theodorakopoulos, J.-Y. Le Boudec, and J.-P. Hubaux, *Privacy vulnerability of published anonymous mobility traces*, *IEEE/ACM Transactions on Networking (TON), vol. 21, no. 3, pp. 720–733*, 2013.

[6] C. Y. Ma, D. K. Yau, N. K. Yip, N. S. Rao, *Quantifying location privacy*, *Proc. IEEE SP*, 2011.

[7] A. Cecaj, M. Mamei, F. Zambonelli, *Re-identification and information fusion between anonymized cdr and social network data*, *Journal of Ambient Intelligence and Humanized Computing, vol. 7, no. 1, pp. 83–96*, 2016.

[8] F. M. Naini, J. Unnikrishnan, P. Thiran, M. Vetterli, *Where you are is who you are: User identification by matching statistics*, *IEEE Transactions on*

Information Forensics and Security (TIFS), vol. 11, no. 2, pp. 358–372, 2016.

[9] Shan Chang, Chao Li, Hongzi Zhu, Ting Lu, Qiang Li, *Revealing Privacy Vulnerabilities of Anonymous Trajectories*, *IEEE Transactions on Vehicular Technology, vol. 6*, 2015.

[10] C. Riederer, Y. Kim, A. Chaintreau, N. Korula, and S. Lattanzi, *Linking users across domains with location data: Theory and validation*, *Proc. WWW*, 2016.

[11] L. Rossi, M. Musolesi, *It's the way you check-in: identifying users in location-based social networks*, *Proc. ACM WOSN*, 2014.

[12] Youssef Khazbak, Guohong Cao, *Deanonymizing Mobility Traces With Co-Location Information*, *IEEE Conference on Communications and Network Security (CNS)*, 2017.

[13] Huandong Wang, Chen Gao, Yong Li, Gang Wang, Depeng Jin, Jingbo Sun, *De-anonymization of Mobility Trajectories: Dissecting the Gaps between Theory and Practice*, *Network and Distributed Systems Security (NDSS) Symposium 2018*, 2018.

[14] C. Dwork, *Differential privacy: A survey of results*, *Proc. TAMC*, 2008.

[15] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, C. Palamidessi, *Geo-indistinguishability: Differential privacy for location-based systems*, *Proc. ACM CCS*, 2013.

[16] Zhenni Feng, Yanmin Zhu, *A Survey on Trajectory Data Mining: Techniques and Applications*, *IEEE Access, vol. 4*, 2016.

[17] Zhou, Changqing, Frankowski, Dan, Ludford, Pamela, Shekhar, Shashi, Terveen, Loren, *Discovering Personal Gazetteers: An Interactive Clustering Approach*, *Proceedings of the 12th Annual ACM International Workshop on Geographic Information Systems*, 2004.

[18] Gouineau, Frank & Landry, Tom & Triplet, Thomas, *PatchWork, a scalable density-grid clustering algorithm*, *ACM Symposium*, 2016.

[19] J. A. Thomas, T. M, *Elements of information theory*, *John Wiley & Sons*, 2006.

[20] Ester, Martin, Kriegel, Hans-Peter, Sander, Jorg, Xu, Xiaowei, *A Density-based Algorithm for Discovering Clusters a Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*, *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 1996.

[21] Huandong Wang, *De-anonymization-of-Mobility-Trajectories*, *GitHub repository*, 2018.

[22] Hootsuite, *https://wearesocial.com/it/blog/2018/01/global-digital-report-2018*, *Accessed: 13/03/2019*.

[23] Douglas A. Reynolds, *Gaussian Mixture Models*, *Encyclopedia of Biometrics*, 2009.

[24] L. R. Rabiner and B. H. Juang, *An introduction to hidden Markov models*, *IEEE ASSp Magazine*, 1986.