UNIVERSITÀ DEGLI STUDI DI PADOVA

Dipartimento di Psicologia dello Sviluppo e della Socializzazione – DPSS

Corso di laurea Magistrale in Psicologia di comunità, della promozione del benessere e del cambiamento sociale

Tesi di laurea Magistrale

Il comportamento del faking analizzato tramite la tecnica del mouse tracking e l'algoritmo GB-SPM

The faking behavior analyzed using the mouse tracking technique and the GB-SPM algorithm

Relatore

Prof. Antonio Calcagnì

***Laureanda*:** Olha Dorosh

Matricola: 2016927

Anno Accademico 2021/2022

Un sentito grazie a tutte le persone che mi hanno permesso di arrivare fin qui e di portare a termine questo lavoro di tesi..

Ringrazio di cuore i miei genitori Petro e Nadiya. Grazie per avermi sempre sostenuto e per avermi permesso di portare a termine gli studi universitari.

Ringrazio il mio fidanzato Andrea per avermi trasmesso la sua immensa forza e la sua determinazione. Grazie perché ci sei sempre stato.

Un sentito grazie al mio relatore Calcagnì Antonio per la sua infinita disponibilità e tempestività ad ogni mia richiesta.

Indice

INTRODUZIONE

Al giorno d'oggi le valutazioni di costrutti non cognitivi (ad esempio di personalità, integrità e di dati anagrafici) sono utilizzate in modo crescente dagli addetti alle risorse umane per prevedere una varietà di risultati delle prestazioni lavorative e decidere quindi se assumere o meno il soggetto intervistato (Wegmeyer & Speer, 2022). Nel momento quindi in cui la selezione del personale si avvale ad esempio della valutazione di personalità, gli individui possono essere motivati a distorcere le loro risposte, attribuendo alla propria persona caratteristiche considerate desiderabili non veritiere, per fare una bella impressione e ottenere il posto di lavoro. Di conseguenza, su una scala Likert in cui vi sono quattro possibilità di risposta (fortemente d'accordo, d'accordo, disaccordo e fortemente disaccordo), all'item "Sono un lavoratore duro" vi è una maggiore propensione a selezionare "fortemente d'accordo" nonostante tale risposta non rappresenta eventualmente un tratto reale dell'intervistato.

Spostandoci nel campo clinico invece, i soggetti possono alterare le loro risposte per ottenere una cerca diagnosi o un determinato compenso (Frick, 2022). I precedenti comportamenti distorsivi possono essere raggruppati sotto la denominazione di *faking*, ovvero la tendenza a rispondere intenzionalmente in un modo che crea un'impressione favorevole di se stessi quando si è tenuti a rispondere in contesti ad alto rischio (Sun et al., 2022). Il faking può essere esercitato in due modalità opposte: i soggetti possono presentarsi maggiormente favorevoli (*faking good*) o in misura minore (*faking bad* o *malingering*) rispetto a quello che sono realmente (Walker et al., 2022).

Il faking non è un problema circoscritto alla psicologia, ma prende piede anche in altri contesti disciplinari quali le scienze economiche (Crawford, 2003), la medicina forense (Gray et al., 2003) e le frodi scientifiche (Marshall, 2000). Ad esempio, in ambito forense, un individuo può gonfiare i sintomi quando vi è di mezzo un risarcimento per danni psicologici (Walker et al., 2022).

Le ricerche passate dimostrano che il faking risulta prevalente sulle scale di valutazione e in situazioni ad alto rischio reale o simulato (Sun et al., 2022). Il faking inoltre deve essere distinto dalle distorsioni di risposta non intenzionali come ad esempio da risposte negligenti e da stili di risposta (Fick, 2022).

In una situazione ad alto rischio, spiega la studiosa Frick (2022), il numero di persone che si avvalgono del faking è stimato tra il 14 e il 40% e va sottolineato che gli

intervistati non falsificano necessariamente tutti gli item, rendendo doveroso distinguere quindi nel processo di risposta un comportamento onesto e uno falsificato. Più precisamente, in un contesto di selezione del personale il faking registra una prevalenza del 30-50%, mentre in ambiti forensi essa risulta essere del 30% (Mazza et al., 2020).

È provato empiricamente che le conseguenze del faking portano a punteggi gonfiati, validità ridotta e ordini di classifica distorti (Sun et al., 2022), alterando quindi l'affidabilità dello strumento utilizzato quando lo si usa in ambito inferenziale. Viste le ripercussioni negative del fenomeno in questione, sono stati messi a punto diversi approcci psicometrici per identificare e valutare il faking good e il malingering. Per prevenire tale comportamento gli studiosi hanno sviluppato diverse strategie da applicare ad anteriori, come il posizionamento randomizzato degli item, la ridotta trasparenza degli item e avvertimenti, ma nessun approccio è riuscito a sradicare il problema (Sun et al., 2022). Altri ricercatori invece hanno tentato di arginare l'impatto del faking includendo nei questionari item relativi alla desiderabilità sociale, ma anche tale proposta non è risultata del tutto efficace (Hu & Connelly, 2021). Come approcci a posteriori, si sono dimostrati invece validi a misurare l'effetto del faking l'approccio dell'analisi fattoriale, i modelli *factor mixture models*, le tecniche di diagnostica di casi e l'approccio probabilistico *Sample Generation by Replacement* (Lombardi et al., 2015). Un'altra procedura all'avanguardia che si pone come l'obiettivo quello di ricercare il faking nei questionari di personalità è la tecnica del *mouse tracking*. Essa, attraverso l'utilizzo del software *MouseTracker*, traccia la traiettoria della posizione del cursore del mouse 60-75 volte al secondo mentre l'intervistato compila il questionario. L'analisi delle traiettorie del mouse è utilizzata per esplorare il processo mentale e la sua evoluzione in tempo reale durante l'esecuzione di diverse decision-making tasks, tra le quali ad esempio proprio i questionari di personalità (Mazza et al., 2020). Il mouse tracking misura indici cinematici quali il tempo di reazione e si propone quindi di valutare l'atteggiamento implicito del soggetto, limitando quindi la sua possibilità di manipolare le risposte, si tratta quindi di un approccio che potrebbe rappresentare una svolta definitiva nel problema del faking e per questo motivo è l'approccio protagonista in questo elaborato. La spiegazione del funzionamento della procedura del mouse tracking verrà ripresa e approfondita nel §1.1.

Prima di passare alla lettura dell'elaborato vero e proprio, ritengo doveroso fare un'altra breve premessa riguardo all'algoritmo utilizzato per analizzare i dati grezzi rilevati dal MouseTracker. Spesso i dati grezzi sulle traiettorie sono rumorosi, presentano ridondanze, sono difficili da analizzare e sono privi di semantica (Wang et al., 2022). Per dare un senso a questi dati gli autori Wang et al. (2022) hanno usufruito di un concetto chiave: il *significant place*, ovvero l'ubicazione geografica nella quale l'oggetto movente risiede nella traiettoria per un ammontare di tempo significativo. Gli studiosi quindi hanno elaborato un algoritmo nuovo, denominato graph-based significant place mining (GB-SPM), combinando le conoscenze della teoria dei campi e dell'algoritmo Label Propagation (LPA) della community detection in una struttura a grafo, che individua i *significant places* in dati di traiettorie spaziotemporali dei GPS. Nonostante il GB-SPM nasce nel contesto del *mobile pattern mining*, nel mio elaborato ho adattato questo nuovo algoritmo per i dati del mouse tracking. La definizione e il funzionamento dell'algoritmo GB-SPM adattato ai dati rilevati con il MouseTracker verrà spiegato nel §1.3 e nel §2.2.

Lo scopo di questa tesi consiste nel verificare se i soggetti che si avvalgono del faking presentano un numero maggiore di *significant places* rispetto a coloro che rispondono sinceramente. In linea con i risultati della letteratura – vedi §1.1, §1.2 e §2.1 – ci aspettiamo che i soggetti che simulano presentino un maggior numero di *singnificant places* in quanto questo corrisponde indirettamente, come verrà spiegato nel §2.2, ad un tempo di esecuzione complessivo più ampio.

L'elaborato comprende tre capitoli. Il primo capitolo fornisce un'introduzione al mouse tracking applicato al campo del faking e all'algoritmo GB-SPM. Visto che il dataset sul quale è stato applicato l'algoritmo appartiene allo studio di Monaro et al. (2018), la prima parte del secondo capitolo (§2.1) comprende una breve descrizione dello studio in questione, la seconda parte (§2.2) mira a spiegare il funzionamento dell'algoritmo in dettaglio e la terza parte (§2.3) prevede l'applicazione del medesimo per dati relativi al faking. Il terzo capitolo, infine, è dedicato alle conclusioni e alla discussione dei problemi riscontrati durante l'analisi.

Capitolo I – Sguardo nella letteratura

1.1 Mouse tracking in letteratura

La letteratura sulla cognizione sociale ha ampiamente dimostrato che atteggiamenti e preferenze possono essere attivati in modo implicito, attraverso processi automatici e inconsci (Monaro et al., 2021). Considerando che la valutazione di costrutti non cognitivi fa un ampio uso di misure auto-riportate, ad esempio i questionari, ed è afflitta da diversi punti deboli, come la vulnerabilità dovuta ai pregiudizi di desiderabilità sociale, sono stati sviluppati diversi strumenti per misurare gli atteggiamenti e le preferenze degli intervistati a livello implicito, come ad esempio il noto Implicit Association Test (IAT) (Monaro et al., 2021). Lo IAT venne utilizzato in principio per comprendere le associazioni implicite di tipo razziale (Greenwald et al., 1998). Le risposte ai quesiti di questa tematica così sensibile infatti sono soggette alla desiderabilità sociale e quindi ad una manipolazione della risposta da parte dell'intervistato, che risulterebbe tuttavia meno efficace nel caso in cui venisse indagato l'atteggiamento implicito del soggetto. La tecnica dello IAT studia la forza delle associazioni automatiche tra diversi concetti tramite la misura delle latenze delle risposte e il numero degli errori in un compito di classificazione (Monaro et al., 2021). Di conseguenza, minore sarà la latenza di risposa, maggiore sarà la connessione dell'associazione inconscia tra i due concetti presentati. Ad esempio, riprendendo la tematica del razzismo, in un ipotetico IAT dove si richiede di accoppiare volti, bianchi o neri, con delle parole, positive o negative, un partecipante che impiega meno tempo ad associare volti bianchi con parole buone rispetto a volti neri con parole buone è molto probabilmente consciamente o inconsciamente razzista. La differenza tra la latenza della risposta tra categorie considerate compatibili, ad esempio la medesima categorizzazione di volti bianchi e parole positive nell'intervistato razzista del nostro esempio, e quelle incompatibili – la categorizzazione di volti neri con termini positivi sempre nel nostro esempio - è definito "effetto IAT" e indica la direzione dell'atteggiamento o la preferenza implicita del partecipante (Monaro et al., 2021). Recentemente, basandosi sull'analisi dell'interazione uomo-computer, il metodo di classificazione del paradigma classico dello IAT è stato modificato: invece di usare i tasti della tastiera, i partecipanti dovevano scegliere le risposte tramite il mouse. Questa nuova procedura, nota come

Mouse Tracking IAT (MT-IAT), non solo ha replicato gli effetti dello IAT classico ma ha consentito di esaminare i processi cognitivi relativi alle preferenze implicite in tempo reale grazie all'analisi cinematica dei movimenti del mouse (Monaro et al., 2021). Tale assunzione è supportata anche da recenti ricerche che studiano il mouse tracking in associazione ad altri dispositivi sperimentali consolidati come il fMRI o *eye tracking* (Calcagnì et al., 2019).

Il MT-IAT ad esempio può essere applicato nel campo della ricerca sui consumatori, come lo dimostra lo studio di Monaro et al. (2021). Monaro et al. (2021) hanno utilizzato l'analisi dei movimenti del mouse per indagare gli atteggiamenti impliciti degli utenti rispetto a due popolari social network: Facebook e Twitter. In primo luogo è stato somministrato un questionario preliminare di sei item a 40 individui; quest'ultimo ha misurato la frequenza di utilizzo dei due social network e la preferenza esplicita per Facebook o Twitter. In seguito è stato chiesto ai partecipanti di eseguire un MT-IAT, volto a misurare il loro l'atteggiamento implicito. Il paradigma del MT-IAT in questo caso comprendeva sette blocchi - ognuno contente 20 o 40 trials - di cui due critici. Il primo blocco critico comprendeva prove compatibili con l'atteggiamento esplicito espresso dal partecipante; quindi se il soggetto aveva espresso la preferenza di Facebook a Twitter, era prevista una classificazione compatibile con le sue preferenze, quindi Facebook doveva essere associato con immagini positive e Twitter con immagini negative. Il secondo blocco critico invece rappresentava le prove incompatibili, quindi sempre riprendendo l'esempio di una preferenza per Facebook rispetto a Twitter, l'individuo doveva classificare Facebook con le immagini negative e Twitter con le immagini positive. Inizialmente ai partecipanti veniva richiesto di premere un pulsante di start in basso al centro. In seguito, al centro dello schermo compariva un'immagine e il soggetto era tenuto a classificarla spostando il mouse verso l'etichetta a in alto a destra o a sinistra del monitor, come mostra la Figura 1.1. Per comprende maggiorente il processo cognitivo che si metteva in moto, i partecipanti sono stati sollecitati a muovere il mouse appena compariva lo stimolo anche se non erano sicuri della loro risposta. A questo punto il software MouseTracker registrava i movimenti spazio temporali del mouse.

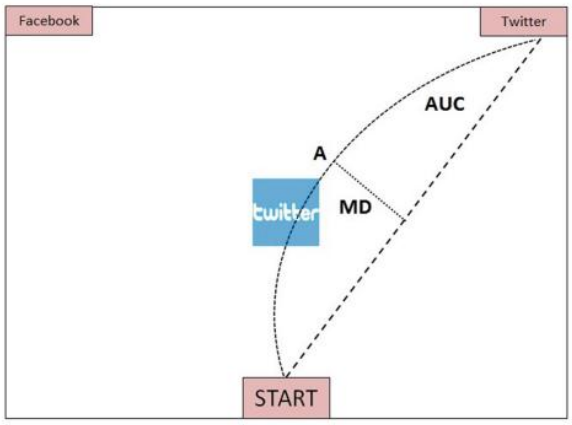

Figura 1.1 *Esempio di un trial di MT-IAT. Il soggetto è tenuto a classificare l'immagine posta al centro dello schermo in una delle due etichette ubicate una in alto a destra e l'altra in alto a sinistra. Durante il processo decisionale, il MouseTracker registra la traiettoria compiuta e sue coordinate x e y la riassume in una serie di indici cinetici, quali ad esempio l'area sotto la curva (AUC) o la massima deviazione (MD) – massima curvatura mostrata dalle traiettoria - (Monaro et al., 2021). La linea tratteggiata rappresenta la traiettoria ideale.*

Il questionario preliminare ha svelato che 30 soggetti mostravano una preferenza esplicita per Facebook mentre solo sei intervistati per Twitter, mentre quattro partecipanti non hanno espresso una preferenza. L'atteggiamento implicito invece è stato quantificato dal *D-Index*, calcolato sottraendo il tempo per raggiungere la massima deviazione (*MD-time*) del blocco compatibile dal *MD-time* del blocco incompatibile e dividendo tale differenza per la deviazione standard aggregata dei due blocchi. Il *D-Index* era positivo nel caso in cui la preferenza esplicita combaciava con quella implicita e negativa e viceversa. Solo quattro partecipanti su 36 hanno registrato un *D-Index* negativo, dimostrando che nel 89% dei casi i soggetti erano coerenti nella nei loro atteggiamenti espliciti e impliciti. Questo studio dimostra l'efficacia del MT-IAT nel campo del marketing, suggerendo quindi che le ricerche sui consumatori potrebbero essere condotte durante l'interazione tra le persone e un'ampia gamma di servizi e tecnologie, senza la necessità di indagare le preferenze esplicite tramite strumenti *self reported*. Il successo del mouse tracking è stato ampiamente dimostrato anche in altri

studi di ricerca cognitiva, quali la cognizione numerica, la decisione lessicale, la memoria e il rilevamento della menzogna (Calcagnì et al., 2021).

Il mouse tracking è stato utilizzato anche come *lie detector*. Ad esempio lo studio di Monaro et al. (2020) ha dimostrato che il MT-IAT è efficace nell'identificare le risposte false nei punteggi espressi su una scala di valutazione a cinque stelle e quindi poteva essere applicato anche alle recensioni di aziende che usufruiscono di E-commerce – ad esempio Amazon, TripAdvisor e Yelp - che utilizzano lo stesso strumento valutativo. Monaro et al. (2020) hanno chiesto a 59 partecipanti di valutare venti prodotti su una scala di valutazione a cinque stelle. Ai partecipanti veniva mostrata inizialmente la descrizione di un oggetto in base alla quale essi formulavano un atteggiamento. Tuttavia, per metà dei prodotti gli individui sono stati incoraggiare a mentire, scegliendo una recensione più positiva o più negativa rispetto alla preferenza che avevano formulato, per ottenere un compenso che variava in base all'item. Ad esempio un soggetto poteva ritrovarsi di fronte ad un tablet che veniva descritto come poco performante, lento e con software e funzionalità ridotti. Esso di conseguenza formulava un pensiero negativo rispetto al tablet. Ciononostante il venditore incoraggiava il partecipante a lasciare una recensione positiva in cambio di due anni di sottoscrizione gratuita ad Amazon Prime. Ogni intervistato quindi doveva affrontare quattro condizioni sperimentali differenti, ovvero quella in cui gli veniva richiesto di scegliere una recensione positiva o negativa che corrispondeva alla sua preferenza – rispettivamente *TP* e *TN* – e una recensione positiva o negativa falsa – rispettivamente *FP* e *FN*. Per ogni risposta, il software MouseTracker ha registrato la posizione del mouse dal punto di partenza al clic finale. L'analisi delle valutazioni mostra che i partecipanti danno punteggi in media più bassi quando si tratta di lasciare un *FP* piuttosto che un *FN*. Inoltre essi danno punteggi più alti quando hanno a che fare con una recensione *TP* piuttosto che una *FP*. In fine, essi sceglievano punteggi più bassi quando si trattava di selezionare una recensione *TN* piuttosto che un *FN*. L'analisi condotta sul movimento del mouse (vedi Figura 1.2) invece dimostra che coloro che dovevano lasciare una recensione falsa registravano delle traiettorie più corte, in quanto, come descritto in precedenza, i soggetti preferivano rimanere più neutrali quando gli veniva richiesto di mentire. Inoltre le condizioni *TP* e *TN* presentano traiettorie più diritte che collegano il punto di partenza con la risposta scelta, mentre le traiettorie di

FP e *FN* sono maggiormente irregolari. In fine, i partecipanti erano più svelti nel selezionare le risposte *TP* e *TN* rispetto a quelle false.

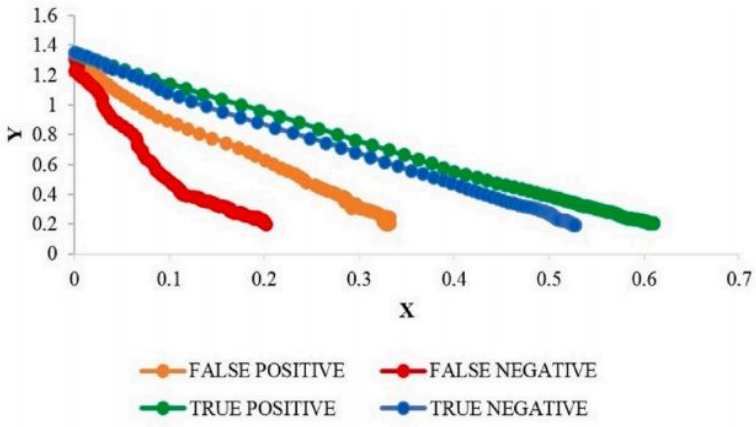

Figura 1.2 *Traiettoria media del mouse per ogni condizione sperimentale. X e Y rappresentano la posizione del mouse rispettivamente lungo l'asse delle x o y. Ogni traiettoria è composta da 101 punti che corrispondono ai 101 frame in cui le traiettorie sono state normalizzate. Le quattro condizioni sperimentali - FP, TP, FN, TN - sono rappresentate rispettivamente dal colore arancione, verde, rosso e blu (Monaro et al., 2020).*

L'analisi della varianza - vedi Tabella 1.1 - ha confermato le osservazioni grafiche in quanto evidenzia un *main effect* significativo dell'inganno (*Deception*) su tutte le caratteristiche spaziotemporali , ad eccezione dell' indice *y-flip*, ovvero il numero totale di modifiche della direzione del mouse lungo l'asse delle y lungo l'intera traiettoria. I risultati non riscontrano differenza di effetti significativi in alcuna caratteristica spaziotemporale per quel che riguarda la tipologia di valutazione (*Rating type*) – o più semplicemente valutazione positiva o negativa. Allo stesso modo le interazioni (*Interaction*) tra la presenza o meno dell'inganno – recensione vera o falsa – e la tipologia di valutazione – positiva o negativa – non mostrano nessun risultato significativo per tutte le variabili selezionate.

In conclusione quindi i partecipanti che lasciavano le valutazioni false impiegavano più tempo per iniziare a muovere il mouse (*IT*) e per scegliere la loro risposta (*RT*) rispetto a coloro che hanno dato valutazioni sincere. Inoltre le traiettorie *FP* e *FN* risultavano più ampie (*AUC* e *MD*) e maggiormente irregolari, con cambi di direzione più frequenti lungo l'asse delle x (numero più ampio di *x-flip*).

Feature	Effect	F, df, *p*-value, effect size
IT	Deception*	$F_{(1,58)} = 18.67, p < 0.007, \eta^2_G = 0.06$
	Rating type	$F_{(1,58)} = 0.01, p = 0.92, \eta^2_G < 0.02$
	Interaction	$F_{(1,58)} = 0.13, p = 0.72, \eta^2_G < 0.02$
RT	Deception*	$F_{(1,58)} = 39.72, p < 0.007, \eta^2_G = 0.16$
	Rating type	$F_{(1,58)} = 5.08, p = 0.03, \eta^2_G < 0.02$
	Interaction	$F_{(1,58)} = 0.50, p = 0.48, \eta^2_G < 0.02$
MD	Deception*	$F_{(1,58)} = 17.76, p < 0.007, \eta^2_G = 0.06$
	Rating type	$F_{(1,58)} = 1.02, p = 0.32, \eta^2_G < 0.02$
	Interaction	$F_{(1,58)} = 3.25, p = 0.08, \eta^2_G < 0.02$
AUC	Deception*	$F_{(1,58)} = 16.13, p < 0.007, \eta^2_G = 0.07$
	Rating type	$F_{(1,58)} = 0.008, p = 0.93, \eta^2_G < 0.02$
	Interaction	$F_{(1,58)} = 0.12, p = 0.73, \eta^2_G < 0.02$
MD-Time	Deception*	$F_{(1,58)} = 28.30, p < 0.007, \eta^2_G = 0.10$
	Rating type	$F_{(1,58)} = 4.75, p = 0.03, \eta^2_G < 0.02$
	Interaction	$F_{(1,58)} = 2.82, p = 0.10, \eta^2_G < 0.02$
x-flip	Deception*	$F_{(1,58)} = 10.72, p < 0.007, \eta^2_G = 0.02$
	Rating type	$F_{(1,58)} = 0.02, p = 0.88, \eta^2_G < 0.02$
	Interaction	$F_{(1,58)} = 0.87, p = 0.36, \eta^2_G < 0.02$
y-flip	Deception	$F_{(1,58)} = 2.76, p = 0.10, \eta^2_G < 0.02$
	Rating type	$F_{(1,58)} = 5.89, p = 0.02, \eta^2_G < 0.02$
	Interaction	$F_{(1,58)} = 0.39, p = 0.54, \eta^2_G < 0.02$

Tabella 1.1 *ANOVA degli indici cinematici di MouseTracker. Gli effetti (rispettivamente Inganno, Tipologia di valutazione e Interazione) che hanno raggiunto il livello significativo con un p-value <0.007, secondo la correzione di Bonferroni, sono contrassegnati da un asterisco. La terza colonna riporta F-score e suoi rispettivi gradi di liberta df, il p-value e l'effect size* η_G^2*. Rispetto alla magnitudo,* $\eta_G^2 = 0.02$ *indica un effetto piccolo,* $\eta_G^2 = 0.13$ *un effetto medio e* $\eta_G^2 = 0.26$ *un effetto grande (Monaro et al., 2020).*

Lo studio di Monaro et al. (2020) è stato sviluppato con l'idea di essere applicato per scovare le false recensioni nel contesto di E-Commerce. La metodologia di ricerca e i risultati tuttavia, come vedremo nel §1.2, possono essere estesi facilmente anche al fenomeno del faking, di cui abbiamo discusso nell'Introduzione.

1.2 Applicazione del mouse tracking al problema del faking in letteratura

Il seguente paragrafo sarà dedicato all'analisi della letteratura del faking rilevato attraverso il paradigma del mouse tracking e le peculiarità spaziotemporali che lo distinguono.

Lo studio di Mazza et al. (2020) indaga se gli indici cinematici possono migliorare il rilevamento di soggetti che usufruiscono del faking good quando

rispondono ai questionari di personalità. Il comportamento di faking good si manifesta nel momento in cui un individuo si presenta in una maniera particolarmente favorevole, sottolineando i tratti desiderabili e mascherando quelli indesiderabili (Mazza et al., 2020). Inizialmente 120 partecipanti sono stati assegnati a random in uno delle quattro condizioni sperimentali, definite dalla manipolazione di due variabili: Istruzioni (essere onesti: *H* o usufruire del faking good: *FG*) e Pressione del tempo (i soggetti erano sollecitati a rispondere rapidamente: *S* o non subivano alcuna pressione temporale:*U*). I primi due gruppi quindi risultavano essere composti da individui a cui veniva richiesto inizialmente di essere onesti e in seguito servirsi del faking good senza alcuna pressione temporale (*H-FG/U*) o con un incitamento di rapidità (*H-FG/S*). Gli alti due gruppi invece differivano in quanto inizialmente erano sollecitati ad utilizzare il faking good e in seguito rispondere onestamente, con lo stesso intreccio di pressione temporale dei precedenti (*FG-H/U* e *FG-H/S*).

I ricercatori hanno utilizzato tre scale di validità di Minnesota Multiphasic Personality Inventory-2 (MMPI-2): *Lie* (*L*), *Correction* (*K*) e *Superlative Self-Presentation* (*S*). Il MMPI-2 è uno dei più rinomati questionari clinici di autovalutazione che viene utilizzato per misurare la personalità e la psicopatologia e richiede una risposta dicotomica: vero o falso (Mazza et al., 2020). La scala *L* si riferisce a comportamenti che per quasi tutte le persone sono giudicati veri o falsi (Bertelloni, 2022) e misura la tendenza a offrire un'immagine di sé socialmente più accettabile in quanto la maggior parte degli item richiede "falso" come risposta (Mazza et al., 2020). Un esempio di item della scala *L* è: "Non dico sempre la verità". La scala *K* è stata progettata invece per calcolare l'atteggiamento difensivo tramite l'indagine sull'adattamento e sul controllo emotivo degli intervistati, ad esempio item può essere:"Le critiche e i rimproveri mi feriscono terribilmente" (Mazza et al., 2020). La scala *T* invece indaga un'autopresentazione eccessivamente virtuosa e ben adattata in qualsiasi contesto, come ad esempio: "Non mi sono mai sentito meglio in vita mia di adesso" (Mazza et al., 2020). Punteggi elevati su tutte e tre le scale indicano che vi è la possibilità che il soggetto presenti un'immagine di sé eccessivamente positiva. Inoltre i ricercatori hanno usufruito anche del questionario di personalità Psychopathic Personality Inventory-Revised (PPI-R). Più precisamente, è stata utilizzata la scala di validità *Virtuous Responding* (*VR*), i cui item prevedono la risposta su una scala di

quattro punti - vero, abbastanza vero, abbastanza falso, falso – e sono progettati per rilevare la sottostima (Mazza et al., 2020). Un item per esempio è: "Non ho mai desiderato ferire qualcuno". Il totale degli item era di 96 e i partecipanti erano tenuti a rispondere alle scale del MMPI-2 selezionando con il mouse "vero" o "falso" rispettivamente in alto a destra e a sinistra su uno schermo, con una procedura simile a quella descritta per lo studio di Monaro et al. (2021) (vedi §1.1) o selezionando una delle risposte a quattro punti per quel che riguarda il PPI-R, come il procedimento descritto da Monaro et al. (2020) (vedi §1.1). Anche in questo caso il MouseTracker raccoglieva le misure delle caratteristiche spaziotemporali del movimento del mouse.

La prima fase dell'analisi ha visto l'implementazione dell'analisi della varianza sui *T-scores* di ogni scala del MMPI-2 e PPI-R e i risultati dimostrano un effetto significativo della variabile Istruzioni su ogni scala. In altre parole, coloro che usufruivano del faking good hanno ottenuto *T-scores* significativamente più alti rispetto a quelli che erano onesti. Inoltre è stato rilevato che la Pressione del tempo non ha nessun effetto significativo sulle scale utilizzate. Similarmente, l'interazione tra l'Istruzioni e la Pressione del tempo non apporta differenze significative.

Per quale che riguarda gli indici cinematici relativi alle caratteristiche temporali, i ricercatori hanno scelto di analizzare: *RT*, *MD-time*, vel_x e vel_y. *RT* è il tempo di latenza tra la presentazione della domanda e la selezione della risposta, *MD-time* è il tempo necessario per raggiungere la massima deviazione ed infine vel_x e vel_y corrispondono alla velocità del mouse lungo l'asse delle x o y tra due *time frames*. I risultati, mostrati in Tabella 1.2, rilevano un *main effect* della Pressione temporale su *RT* e *MD-time* su tutte le scale, ad eccezione del *MD-time* sulla scala *VR*. Ciò significa che effettivamente i partecipanti nella condizione *S* erano più rapidi nel rispondere rispetto a quelli che non erano sollecitati (*U*). Al contrario, la Pressione temporale non aveva alcun effetto significativo su vel_x e vel_y. Per quel che riguarda invece la variabile Istruzioni, i partecipanti che utilizzavano il faking good sono stati significativamente più lenti in termini di *RT* e *MD-time* solo sulla scala *L* e registravano un *main effect* significativo su vel_x su tutte le scale, ad eccezione della *VR*. Riguardo a vel_y, è stato registrato un effetto solo sulle scale *K* e *VR*. Questo significa che i partecipanti onesti del gruppo *H* erano più veloci quando si muovevano lungo l'asse delle x e delle y, rispettivamente sulle scale *L*, *K*, *S*, *K* e *VR*.

Temporal variable	Effect	F	*p*-value	η_G^2	95% CI
RT S scale	Time pressure	$F_{(1,118)} = 18.58$	$3.395e^{-05}$	0.09 (small)	[0.01, 0.18]
RT K scale	Time pressure	$F_{(1,118)} = 19.04$	$2.753e^{-05}$	0.10 (small)	[0.01, 0.18]
RT L scale	Time pressure	$F_{(1,118)} = 23.11$	$4.559e^{-06}$	0.12 (small)	[0.02, 0.20]
RT VR scale	Time pressure	$F_{(1,118)} = 10.36$	$1.661e^{-03}$	0.06 (small)	[0.00, 0.14]
MD-time S scale	Time pressure	$F_{(1,118)} = 18.78$	$3.097e^{-05}$	0.09 (small)	[0.01, 0.18]
MD-time K scale	Time pressure	$F_{(1,118)} = 20.60$	$1.374e^{-05}$	0.11 (small)	[0.01, 0.19]
MD-time L scale	Time pressure	$F_{(1,118)} = 19.27$	$2.481e^{-05}$	0.09 (small)	[0.01, 0.19]
RT L scale	Instructions	$F_{(1,118)} = 17.30$	$6.096e^{-05}$	0.05 (small)	[0.00, 0.14]
MD-time L scale	Instructions	$F_{(1,118)} = 9.21$	$2.962e^{-03}$	0.03 (small)	[0.00, 0.12]
velx S scale	Instructions	$F_{(1,118)} = 191.33$	$1.878e^{-26}$	0.28 (large)	[0.15, 0.42]
velx K scale	Instructions	$F_{(1,118)} = 140.99$	$7.097e^{-22}$	0.27 (large)	[0.14, 0.41]
velx L scale	Instructions	$F_{(1,118)} = 151.25$	$7.069e^{-23}$	0.32 (large)	[0.18, 0.46]
vely K scale	Instructions	$F_{(1,118)} = 6.76$	$1.050e^{-02}$	<0.02	[0.00, 0.08]
vely VR scale	Instructions	$F_{(1,118)} = 9.26$	0.003	0.02 (small)	[0.00, 0.10]

Tabella 1.2 *ANOVA degli indici temporali significativi per ogni scala. Nella seconda colonna sono mostrati gli effetti (rispettivamente Pressione temporale e Istruzioni) che hanno raggiunto il livello significativo con un p-value <0.0125, secondo la correzione di Bonferroni. Dalla terza colonna in poi sono riportai gli F-score e suoi rispettivi gradi di liberta df, il p-value e l'effect size* η_G^2 *e gli intervalli di confidenza al 95%. Rispetto alla magnitudo,* $\eta_G^2 = 0.02$ *indica un effetto piccolo,* $\eta_G^2 = 0.13$ *un effetto medio e* $\eta_G^2 = 0.26$ *un effetto grande (Mazza et al., 2020).*

Per quel che riguarda invece le caratteristiche spaziali del mouse tracking, vedi Tabella 1.3, sono stati analizzati la massima deviazione (*MD*) e l'area sotto la curva (*tAUC*). L'analisi della varianza mostra unicamente un *main effect* delle Istruzioni su *MD* e *tAUC* nella scala *L*. Ciò significa che coloro che usufruivano del faking good tratteggiavano traiettorie più ampie degli onesti sulla scala *L*, nonostante l'effetto sia molto piccolo (*tAUC L scale*: $F(1,118) = 5.43$, *p-value* = 0.021, $\eta_G^2 < 0.02$). Rispetto alla variabile Pressione temporale invece vi è l'evidenza che coloro che erano sottoposti alla versione accelerata (*S*) tendevano a compiere traiettorie maggiori su tutte e scale ad eccezione della scala *VR*.

Riassumendo quindi i risultati più significativi e in linea con la letteratura (Mazza et al., 2020) si possono trovare solo sulla scala *L*, per questa ragione le seguenti conclusioni saranno circoscritte solo a quest'ultima. I partecipanti che utilizzavano il faking good hanno impiegato più tempo, in quanto i valori dei loro *RT* e *MD-time* sono

significativamente maggiori (vedi Tabella 1.2). Essi inoltre hanno delimitato traiettorie più ampie, visto che i parametri *MD* e *tAUC* sono significativamente maggiori (vedi Tabella 1.3). Gli effetti registrati tuttavia sono complessivamente piccoli. In Figura 1.2 inoltre possiamo notare che le traiettorie degli onesti sono maggiormente regolari e diritte, esattamente come per gli onesti (*TP* e *TN*) nello studio di Monaro et al. (2020), vedi §1.1. Come nello studio di Monaro et al. (2020), i partecipanti che dovevano mentire hanno delimitato traiettorie più ampie ma più corte; in quanto la scala *L* richiede di rispondere su una scala di valutazione a quattro punti – mentre quella di Monaro et al. (2020) era a cinque punti- andrebbe analizzato se anche in questo caso i soggetti preferivano rimanere più neutrali quando gli veniva richiesto essere falsi.

Spatial variable	Effect	F	*p*-value	η_G^2	95% CI
MD S scale	Time pressure	$F_{(1,118)} = 6.62$	$1.130e^{-02}$	0.04 (small)	[0.00, 0.11]
MD K scale	Time pressure	$F_{(1,118)} = 5.15$	$2.506e^{-02}$	0.03 (small)	[0.00, 0.10]
MD L scale	Time pressure	$F_{(1,118)} = 8.72$	$3.792e^{-03}$	0.05 (small)	[0.00, 0.13]
MD L scale	Instructions	$F_{(1,118)} = 6.15$	$1.451e^{-02}$	<0.02	[0.00, 0.08]
tAUC L scale	Instructions	$F_{(1,118)} = 5.43$	$2.146e^{-02}$	<0.02	[0.00, 0.08]

Tabella 1.3 *ANOVA degli indici spaziali significativi per ogni scala. Nella seconda colonna sono mostrati gli effetti (rispettivamente Pressione temporale e Istruzioni) che hanno raggiunto il livello significativo con un p-value <0.025, secondo la correzione di Bonferroni. Dalla terza colonna in poi sono riportai gli F-score e suoi rispettivi gradi di liberta df, il p-value e l'effect size* η_G^2 *e gli intervalli di confidenza al 95%. Rispetto alla magnitudo,* $\eta_G^2 = 0.02$ *indica un effetto piccolo,* $\eta_G^2 = 0.13$ *un effetto medio e* $\eta_G^2 = 0.26$ *un effetto grande (Mazza et al., 2020).*

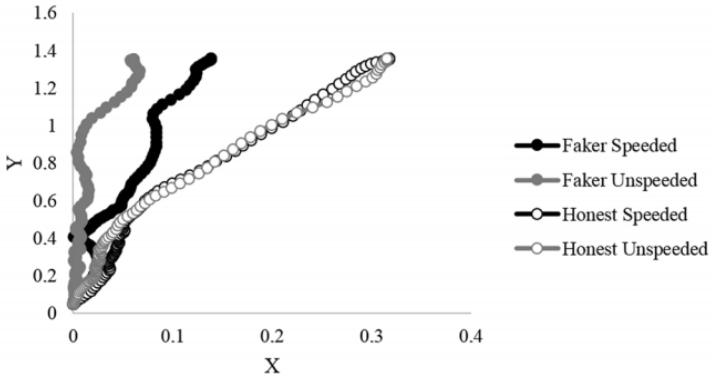

Figura 1.2 *Traiettoria media del mouse per ogni condizione sperimentale sulla scala L. Per consentire il confronto, tutte le traiettorie sono state rimappate orizzontalmente. X e Y rappresentano la posizione del mouse rispettivamente lungo l'asse delle x o y. La linea nera e grigia delineano le traiettorie dei soggetti in condizione di faking good (FG) rispettivamente nella situazione accelerata (S) e non accelerata (U). I punti neri e grigi invece rappresentano le traiettorie degli onesti (H) rispettivamente sollecitati a rispondere rapidamente (S) e non sollecitati (U) (Mazza et al., 2020).*

1.3 Algoritmo GB-SPM in letteratura

Negli scorsi paragrafi (§1.1 e §1.2) abbiamo analizzato il fenomeno del faking ed il paradigma del mouse tracking in letteratura. Il seguente paragrafo invece verrà dedicato al nuovo algoritmo proposto da Wang et al. (2022): il graph-based significant place mining (GB-SPM). Quest'ultimo nasce nel contesto del mobile pattern mining, utilizzato per supportare i servizi *location-aware*, ovvero i servizi che rilevano la posizione; tuttavia uno degli obiettivi di questa tesi è proprio quello di ricostruire questo algoritmo nel linguaggio di programmazione R e applicarlo successivamente ai dati rilevati dal MouseTracker. Il lavoro di ricostruzione sarà presentato nel §3.2, mentre il seguente paragrafo sarà incentrato su una breve introduzione all'algoritmo proposto da Wang et al. (2022).

I dati grezzi sulle traiettorie sono spesso rumorosi, presentano ridondanze, sono difficili da analizzare e sono privi di semantica (Wang et al., 2022). Per dare un senso a questi dati si può ricorrere al concetto di *significant place* (*SP*), ovvero l'ubicazione geografica dentro una traiettoria nella quale l'oggetto movente risiede per un ammontare di tempo significativo. Per rilevare i *SP* si utilizzano diversi approcci che fanno parte di una grande famiglia chiamata *significant place mining* (*SPM*). Si tratta di un data mining spaziotemporale molto impegnativo in quanto i dati sono generalmente ad alta densità e molto rumorosi. Gli approcci esistenti per individuare i *SP* applicano un algoritmo di clustering per raggruppare punti simili di una data traiettoria e il più utilizzato di questi è un clustering basato sulla densità in quanto può trovare cluster di qualsiasi forma. Il clustering basato sulla densità tuttavia presenta tre importanti limitazioni. In primis nel momento in cui i dati cambiano la loro densità i risultati risultano essere spesso innacurati; in secondo luogo l'utente è tenuto a fissare molte

soglie; in fine quando ha a che fare con punti di confine tra un cluster e un altro, può sbagliare la loro classificazione. Visti i precedenti problemi e l'elevata complessità temporale che caratterizza in generale i SPM, Wang et al. (2022) hanno deciso di creare un nuovo algoritmo – GB-SPM – che non usa la distanza di densità dei punti sulla traiettoria, migliorando di conseguenza la performance e l'accuratezza del significant place mining. L'algoritmo proposto deriva dall'ispirazione dal *data field theory* e dal community detection. Da quest'ultimo soprattutto riprende il Label Propagation Algorithm (LPA) in quanto è un algoritmo di apprendimento veloce semi-supervisionato con una complessità temporale approssimativamente lineare.

Il diagramma di flusso dell'algoritmo GB-SMP, vedi Figura 1.3, è composto da tre componenti principali. La prima consiste nell'estrazione dei *characteristic points* in base al *neighborhood velocity* – ovvero la velocità dei punti vicini di un dato punto - di ogni punto sulla traiettoria. La seconda converte i characteristic points in nodi all'interno di un grafo denominato *Characteristic Point Index Neighborhood* (*CPIN*) secondo un indice chiamato *index neighborhood*. La terza in fine compie un cluster dei nodi individuati avvalendosi dell'algoritmo LPA esteso con una nuova metrica degli archi.

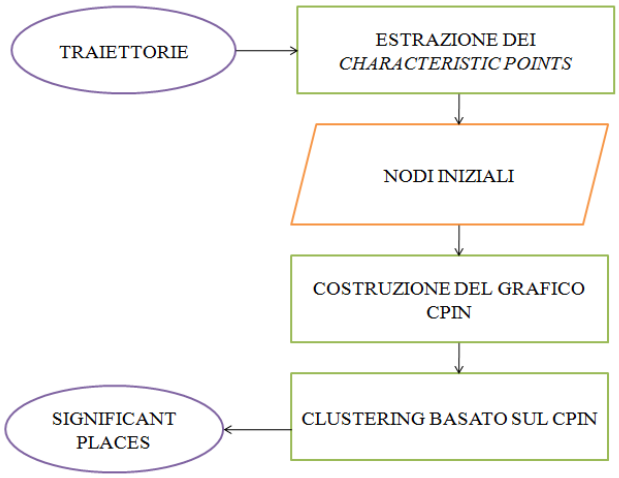

Figura 1.3 *Diagramma di flusso di GB-SPM (Wang et al., 2022).*

Più precisamente, ogni traiettoria inserita viene divisa dal GB-SPM in un set di punti – i *characteristic points*. In seguito esso calcola il *index neighborhood* di ogni *characteristic point* creando delle connessioni tra i punti vicini; tale procedura permette di mantenere la struttura locale dei dati spaziotemporali ad un costo relativamente

basso. Dopo il GB-SPM converte i *characteristic points* e le sue connessioni in una topologia a grafo. In fine esso applica l'algoritmo Label Propagation esteso allo scopo di individuare i *SPs*.

I principali contributi tecnici del GB-SPM si possono riassumere in tre punti. In primis è stato elaborata una nuova metodologia per estrarre i *characteristic points* in base a un indice calcolato secondo la velocità dei loro punti vicini (*index neighborhood velocity*) e un nuovo metodo per costruire la mappa topologica dei *characteristic points* e dei loro *index neighborhoods*; il *index neighborhood velocity* ha un costo computazionale minore e può attenuare il rumore rispetto alle tecniche precedenti basate sulla *co-location probability*. In secondo luogo l'algoritmo di clustering basato sul *CPIN*, che considera le caratteristiche spaziotemporali dei *characteristic points* nella mappa topologica, evita i calcoli delle distanze tra le densità. In terzo luogo la combinazione in una struttura grafica delle conoscenze di data field theory (Li et al., 2017) e del LPA rappresenta l'innovazione più grande in quanto permette di ridurre le soglie che devono essere specificate a priori dall'utente, traducendosi quindi in prestazioni migliori.

L'algoritmo di Wang et al. (2022) verrà approfondito maggiormente nel §2.2. Nel §2.2 inoltre verranno fornite le definizioni di tutti gli elementi caratteristici e il procedimento necessario per implementare il GB-SPM. Prima di concludere questo paragrafo tuttavia ritengo necessario soffermarsi su alcune definizioni del significant place mining fornite da Niu et al. (2021):

1. *Traiettoria* (*TR*): La traiettoria *TR* di un oggetto in movimento è rappresentata da una sequenza di punti spaziotemporali rilevati in determinati intervalli temporali ed è indicata nel seguente modo:

   $$TR = \{p_1, p_2, \ldots, p_i, \ldots, p_n\} \quad (1.1)$$

   Ogni punto p_i è formato da una tripletta:

   $$p_i = (Lat_i, Long_i, t_i) \quad (1.2)$$

   Lat_i e $Long_i$ rappresentano la latitudine e la longitudine dell'oggetto movente nel tempo t_i. Nel paradigma del mouse tracking invece la latitudine e la longitudine sono sostituite dalle coordinate cartesiane, $p_i = (x_i, y_i, t_i)$. I punti sulla traiettoria

sono ordinati per tempo crescente, quindi per tutti gli $i = 1, 2, \ldots, n$ vale la relazione $t_1 < t_2 < \ldots < t_n$.

2. *Stop Region* (*SR*): Un *stop region SR* è rappresentato da una tripletta:

   $$SR = (R_s, t_{ins}, t_{outs}) \quad (1.2)$$

   Dove $R_s = \{p_{ins}, p_{ins+1}, \ldots, p_{outs}\}$ è una sub- traiettoria della traiettoria *TR* mentre t_{ins} e t_{outs} sono il primo e l'ultimo marcatore temporale di R_s. Inoltre, la durata di un *stop region*, definita come $t_{outs} - t_{ins}$, non deve essere minore di una soglia chiamata *minDuration*, ovvero deve valere la relazione $t_{outs} - t_{ins} \geq minDuration$.

3. *Moving Region* (*MR*): Un *moving region MR* è una rappresentato da una tripletta:

   $$MR = (R_m, , t_{inm}, t_{outm}) \quad (1.3)$$

   Dove $R_m = \{p_{inm}, p_{inm+1}, \ldots, p_{outm}\}$ è la massima sub-traiettoria costituita da un set di punti spaziotemporali contigui di una traiettoria *TR*, mentre t_{inm} e t_{outm} rappresentano il primo e l'ultimo marcatore temporale di R_m.

Ogni punto della traiettoria deve essere assegnato o ad un *stop region* o ad un *moving region*.

4. *Significant Place (SP)*: un *significant place* è rappresentato da:

   $$SP = (R_{SP}, \Delta_{SP}) \quad (1.4)$$

   Dove R_{SP} è un poligono topologicamente chiuso che raffigura la posizione e la forma di un luogo e Δ_{SP} è il minimo ammontare di tempo che l'oggetto movente deve passare in quel luogo. Più precisamente $R_{SP} = \{p_i, p_{i+1}, \ldots, p_{i+j}\}$ è costituito da una serie di punti della traiettoria e il tempo totale che l'oggetto deve spendere in SP deve essere maggiore di Δ_S, ovvero deve valere la condizione $|t_{i+j} - t_i| \geq \Delta_S$.

I *significant places* sono identificati in base al *stop region*, ma non tutti gli *SR* sono degli *SP*. Di conseguenza bisogna filtrare tutti quei *SR* che non soddisfano le condizioni per diventare dei *SP*, riducendo in questo modo i costi di ricerca.

Capitolo II – Malingering e GB-SPM in dettaglio

2.1 Studio di riferimento per l'analisi dei dati

Analogamente alo studio di Mazza et al. (2020) presentato nel §1.2, lo studio di Monaro et al. (2018) indaga il fenomeno del faking secondo il paradigma del mouse tracking; esso tuttavia esamina l'altra faccia di questo comportamento, ovvero il faking bad o malingering. Il faking bad rappresenta quell'insieme di condotte messe in atto deliberatamente dal soggetto per presentare un'immagine più negativa di sé rispetto alla sua reale condizione. Tale comportamento è facilmente rintracciabile anche nella sfera clinica; la sintomatologia del disturbo depressivo maggiore infatti può essere emulata e quindi falsificata in ordine di ottenere un compenso economico dall'assicurazione aziendale (Monaro et al, 2018). Lo studio di Monaro et al. (2018) propone un nuovo metodo basato sul mouse tracking per scovare automaticamente la falsificazione dei sintomi. I movimenti cinematici del mouse, come già dimostrato nel §1.1, hanno il vantaggio di non essere coscientemente controllabili dal soggetto e quindi la manipolazione delle risposte risulta essere quasi impossibile.

La ricerca di Monaro et al. (2018) ha utilizzato due campioni indipendenti, entrambi composti da partecipanti sani e pazienti con diagnosi di disturbo depressivo maggiore. Ai volontari sani inoltre è stato somministrato il Beck Depression Inventory (BDI) allo scopo di scovare partecipanti con sintomi depressivi non diagnosticati. Il campione finale del primo gruppo era composto da venti pazienti depressi, venti soggetti a cui era stato richiesto di essere onesti e venti partecipanti sollecitati a fingere di soffrire di depressione. Il secondo gruppo era caratterizzato dalla medesima suddivisione: nove pazienti, nove onesti e nove malingering.

Gli stimoli adottati comprendevano 30 domande semplici e 46 complesse sui sintomi della depressione e sulla condizione sperimentale e richiedano un "sì" o un "no" come risposta. Gli item sui sintomi depressivi sono stati estratti dal *Depression Questionnaire* (*QD*), ripreso dalla batteria di scale di Cognitive Behavioural Assessment 2.0 (CBA 2.0), e da Structured Clinical Interview for Mood Spectrum (SCI MOODS). Gli item complessi comprendevano nella stessa domanda due o più informazioni. I partecipanti dovevano rispondere "sì" nel momento in cui tutte le informazioni erano considerate vere e "no" se anche solo una delle informazioni era falsa. Le domande

complesse appartengono a una metodologia per scovare i bugiardi in quanto maggiore è la mole di informazioni da gestire per riuscire ad elaborare una risposta falsa coerente con la situazione che si vuole fingere, maggiore è il sovraccarico cognitivo di coloro che mentono (Monaro et al., 2018).
Le trenta domande semplici erano composte da:

- Cinque item riferiti alla condizione sperimentale (*EX*), come ad esempio: "Stai indossando le scarpe?". Tutti i partecipanti erano tenuti a rispondere sinceramente a queste domande in quanto sono semplici domande di controllo.
- Dieci item che si richiamavano ai sintomi depressivi (*DS)*, come: "Pensi più lentamente del solito?".
- Quindici item che indagavano sintomi atipici (*VAS*), ad esempio: "Ridi raramente?". Queste domande sono state riprese dalla scala *Affective Disorders* (*AF*) della batteria Structured Inventory of Malingered Symptomatology (SIMS). Il SIMS è un questionario progettato per rilevare il malingering attraverso il numero di esperienze bizzarre e sintomi depressivi e ansiosi altamente atipici descritti dall'intervistato. Un soggetto è categorizzato come bugiardo se riporta più di cinque sintomi atipici.

I 46 stimoli complessi invece erano formati da:

- Quindici item formati da due sintomi discordanti (*2DS-d*), come ad esempio: "Hai difficoltà a concentrarti al lavoro e sei pieno di energia?".
- Quindici item composti da due sintomi concordanti (*2DS-c*), per esempio: "Ti senti abbandonato dagli altri e ti senti triste per tutto il giorno?".
- Cinque item costruiti in base a due informazioni discordanti: un sintomo depressivo tipico e una domanda controllo (*DS&EX-d*). Il contrasto nasceva in quanto una domanda prevedeva un "sì" come risposta, mentre l'altra l'opposto. Ad esempio: "Hai difficoltà a concentrarti e ti trovi a Parigi?".
- Cinque item composti da due informazioni concordanti (*DS&EX-c*), come: "Sei spesso triste e stai seduto su una sedia?".
- Tre item formulati da due informazioni discordanti relativi alla condizione sperimentale (*2EX-d*): "La domanda che stai leggendo è scritta in rosso e stai indossando le scarpe?".

- Tre item formati da due informazioni concordanti relativi alla condizione sperimentale (*2EX-c*): "Stai rispondendo con il mouse e stai indossando le scarpe?".

Gli item complessi invece comprendevano nella stessa domanda due o più informazioni. I partecipanti dovevano rispondere "sì" nel momento in cui tutte le informazioni erano considerate vere e "no" se anche solo una delle informazioni era falsa. Le domande complesse appartengono a una metodologia per scovare i bugiardi in quanto maggiore è la mole di informazioni da gestire per riuscire ad elaborare una risposta falsa coerente con la situazione che si vuole fingere, maggiore è il sovraccarico cognitivo di coloro che mentono (Monaro et al., 2018).

Poco prima dell'esperimento è stato richiesto ai partecipanti che dovevano fingere di simulare uno stato depressivo di mentire sul loro umore. Per aumentare la complicità è stato descritto a questo gruppo di soggetti uno scenario in cui essi venivano esaminati da una commissione di polizza assicurativa a causa di un danno psicologico subito. Lo scopo era quello di fingere che il danno avesse causato una grave depressione che doveva essere risarcita.

I partecipanti erano tenuti a rispondere a 76 domande casuali mostrate nella parte superiore dello schermo di un computer. Inoltre in alto a destra e a sinistra del monitor vi erano due etichette contenenti la risposta "sì" o "no". Inizialmente i soggetti dovevano premere il pulsante start, situato nella parte centrale e inferiore dello schermo, e in seguito alla comparsa della domanda, essi erano tenuti a rispondere selezionando con il mouse una delle due etichette, analogamente al procedimento di Monaro et al. (2021) riportato nel §1.1.

Per ogni risposta il software MouseTracker ha provveduto a registrare i movimenti compiuti con il mouse. Per consentire il calcolo della media e il confronto tra le prove, il software ha normalizzato il tempo; più precisamente ogni traiettoria è stata normalizzata in 101 intervalli temporali attraverso l'interpolazione lineare. Risulta quindi che ogni intervallo temporale corrisponde a specifiche coordinate x e y in un piano cartesiano. In altre parole MouseTracker ha derivato la posizione del mouse lungo gli assi sui 101 intervalli temporali (X_n,Y_n). Il software calcola diversi indici spaziotemporali, come abbiamo già visto nel §1.1. Negli indici spaziali troviamo la massima deviazione (*MD*), area sotto la curva (*AUC*), *x-flip*, *y-flip*. Il *MD* corrisponde

alla più elevata distanza perpendicolare tra la traiettoria ideale e quella reale. Il *AUC* è l'area geometrica confinata tra la traiettoria ideale e quella reale. Gli indici *x-flip* e *y-flip* rappresentano invece in numero di cambiamenti di direzione rispettivamente lungo l'asse delle x e delle y. Per quel che riguarda le caratteristiche temporali invece il software misura il tempo di inizio (*IT*) e di reazione (*RT*), massima deviazione temporale (*MD-time*). Il IT corrisponde a quel lasso di tempo tra la comparsa della domanda e l'inizio del movimento del mouse mentre il *RT* misura la latenza temporale che intercorre tra la comparsa della domanda e la selezione di un'etichetta. In fine, *MD-time* rappresenta il tempo necessario per raggiungere la massima deviazione (*MD*).

Figura 2.1 *Procedura sperimentale dello studio di Monaro et al (2018). Nella figura è rappresentato un esempio di item ("Ti stanchi facilmente?"), le etichette "sì" e "no" e il pulsante di START.*

Per ogni indice spaziotemporale è stato calcolato il valore medio delle risposte nelle differenti tipologie di domande - *EX*, *DS*, *VAS*, *2DS-d*, *2DS-c*, *DS&EX-d*, *DS&EX-c*, *2EX-d*, *2EX-c*. Inoltre è stata misurata anche la velocità media (*v*) e l'accelerazione (*a*) del movimento del mouse tra due intervalli temporali. Sono stati calcolati anche il numero dei sintomi segnalati dai partecipanti (*DS*, *2DS-d*, *2DS-c*, *DS&EX-d*, *DS&EX-c* e *VAS*) e il numero degli errori nelle domande di controllo (*EX*, *2EX-d*, *2EX-c*).

In primis è stata effettuata un'analisi visiva preliminare delle traiettorie medie dei tre gruppi sperimentali in base alle risposte fornite alle 76 domande, vedi Figura 2.2. Il pattern visivo è simile a quello osservato in altri studi che mirano a individuare il faking tramite il mouse tracking (Monaro et al., 2017; Mazza et al., 2020; Monaro et al.,

2021). Le traiettorie dei bugiardi e degli onesti differiscono sia nei parametri di *AUC* che di *MD*. Le traiettorie di coloro che sono stati sinceri, sia gli intervistati sani che depressi, sono più diritte. Coloro che hanno usufruito del faking invece hanno trascorso inizialmente più tempo a muoversi lungo l'asse delle y e hanno deviato verso la l'etichetta scelta con un ritardo rispetto a coloro che dicevano il vero.

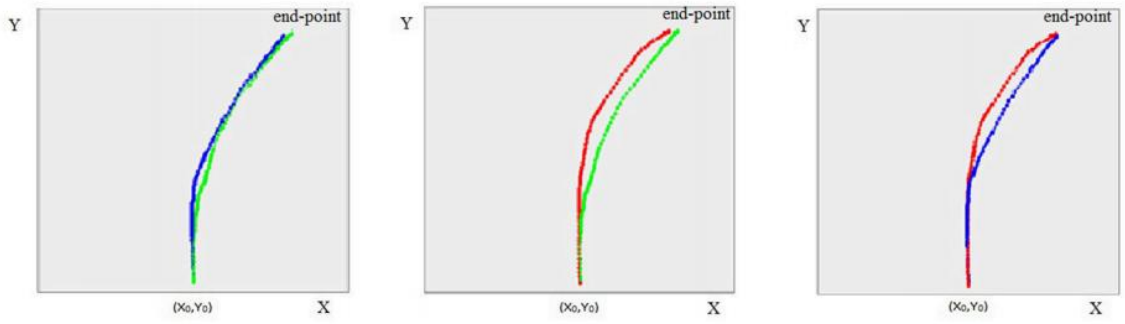

Figura 2.2 *Confronto tra le traiettorie medie dei tre gruppi sperimentali: linea rossa per i bugiardi, verde per gli onesti sani e blu per i pazienti depressi (Monaro et al., 2018).*

2.2 Algoritmo GB-SPM in dettaglio

Come anticipato nel §1.3, l'algoritmo è composto da tre fasi: estrazione dei *characteristic point*, costruzione del grafo *CPIN* e clustering basato sul grafo *CPIN*.

Nella prima fase l'algoritmo seleziona i *characteristic point* da una data traiettoria. Per capire meglio il procedimento è necessario prima fornire alcune definizioni:

5. *Index Neighborhood (IN)*: Data una traiettoria $TR = \{p_1, p_2, ..., p_i, ... , p_n\}$, il *index neighborhood* di un punto p_i, $i \in [1, n]$, è denotato come:

$$IN(p_i) = \{p_k | \, |i-k| \leq indexR, \ \forall p_k \in TR\} \tag{2.1}$$

Dove *indexR* è un numero intero e positivo che controlla il raggio di *IN*.

6. *Dist*(p_l, p_j): La distanza di due punti p_l e p_j di una traiettoria TR è definita come:

$$Dist(p_l, p_j) = \sum_{i=l}^{j-1} Dist(p_i, p_{i+1}) \tag{2.2}$$

Dove $l < j$ e $Dist(p_i, p_{i+1})$ denota la distanza euclidea tra due punti p_i e p_{i+1}.

7. *Neighborhood Velocity (NV)*: Dato un *index neighborhood* $IN(p_i) = \{p_l, p_{l+1}, \ldots, p_{j-1}, p_j\}$, il *neighborhood velocity* di un punto p_i, $i \in [1, n]$, è definito come:

$$NV(p_i) = \frac{Dist(p_l, p_j)}{|p_j.T - p_l.T|} \quad (2.3)$$

Dove $p_j.T$ e $p_l.T$ sono marcatori temporali di p_j e p_l rispettivamente.

8. *Characteristic Point (CP)*: Data una traiettoria $TR = \{p_1, p_2, \ldots, p_i, \ldots, p_n\}$, il *characteristic point* CP_i è un punto spaziotemporale p_i, $i \in [1, n]$, che soddisfa il vincolo $NV(p_i) \leq minVelocity$, dove *minVelocity* rappresenta una velocità minima di soglia specificata dall'utente.

In questa prima fase l'algoritmo calcola il *neighborhood velocity* di tutti i punti della traiettoria *TR*. In seguito vengono filtrati tutti quei punti che possiedono un *neighborhood velocity* uguale o minore a una velocità di soglia *minVelocity* fissata dall'utente. I punti che soddisfano tale condizione sono definiti *characteristic points*. Tale processo aiuta a ridurre la dimensionalità dei dati, a individuare i punti più lenti e a lavorare su un insieme di dati più ristretto all'intera traiettoria.

Tale procedimento viene effettuato da un algoritmo ad hoc chiamato *CPExtraction*, vedi Algoritmo 1.1. Esso richiede come input una traiettoria *TR* e i parametri *indexR* e *minVelocity*, fissati dal soggetto, e ritorna come output una serie di punti *CPS*, che corripondono ai *characteristic points*. L'algoritmo *CPExtraction* in primo luogo esegue la funzione *index_neighbor* (p_i, *indexR)*, allo scopo di individuare per ogni punto p_i il suo index neighborhood (vedi definizione 2.1). In seguito, per ogni punto p_i viene calcolato il *neighborhood velocity* tramite la funzione *neighborhood_velocity(indexNeighbor_*p_i*)*, la quale richiede come input il *index neighborhood* precedentemente quantificato. Se la velocità del punto p_i calcolata risulta essere minore o uguale alla soglia *minVelocity*, allora il punto viene inserito nell'insieme dei *characteristic points CPS.* Tale procedimento viene eseguito per ogni punto della traiettoria. Infine l'algoritmo ritorna il vettore contenente tutti i *characteristic points*.

Algoritmo 2.1: *CPExtraction*

Require *TR, indexR, minVelocity*

Ensure *CPS*

1: $CPS \leftarrow \varnothing$;
2: **for all** each point p_i in *TR* **do**
3: $indexNeighbor\_p_i \leftarrow index\_neighbor\ (p_i, indexR)$;
4: $neighborhoodVelocity\_p_i \leftarrow neighborhood\_velocity(indexNeighbors\_p_i)$
5: **if** $neighborhoodVelocity\_p_i \leq minVelocity$ **then**
6: Place point p_i in *CPS*;
7: **end if**
8: **end for**
9: **Return** *CPS*

La seconda fase ha lo scopo di costruire i grafi *CPIN*. Rappresentare i *characteristic points* come un grafo è maggiormente efficace rispetto gli altri algoritmi di clustering, come quello basato sulla densità visto nel §1.3, per i motivi elencati in seguito:

- La modellazione dei dati secondo un grafo *CPIN* permette di isolare i punti eccessivamente lontani tra di loro nel tempo e nello spazio, riducendo la complessità computazionale. Inoltre l'uso del *index neighborhood* piuttosto che la distanza tra le densità per il calcolo della vicinanza tra i punti risolve il problema delle variazioni della frequenza di campionamento.
- Il grafo *CPIN* cattura la struttura locale dei dati e può regolare di conseguenza la strategia di connessione tra i punti in modo adattivo. Più precisamente quando la densità locale dei punti è grande, i *characteristic points* vengono collegati ai punti più vicini, mentre quando la densità locale è piccola, ovvero quando i punti sono scarsi, i *characteristic points* vengono collegati a punti più lontani. La dinamicità dei *CPIN* permette di risolvere in parte il problema della differenza di densità di densità locale che colpisce gli altri algoritmi di clustering, i quali basano la divisione in cluster di diverse densità locali su dei parametri globali statici.

- Nei grafi *CPIN* i *significant places* vengono estratti attraverso gli algoritmi di community detection, i quali possono clusterizzare i dati senza conoscere la dimensione, il numero e la densità delle comunità. Tale peculiarità permette all'algoritmo GB-SPM di ridurre notevolmente il numero di parametri da fissare.

Il grafo *CPIN* è costituito da una tripletta $G_{CPIN} = (V, E, W)$, dove V è costituito da una serie di nodi che rappresentano i *characteristic points* e E è un set di archi dove un arco connette due nodi se e solo se almeno uno dei nodi si trova tra il *index neighborhood* dell'altro. Nel grafico *CPIN* ogni arco è etichettato da un attributo di peso che riflette la somiglianza degli attributi nei nodi corrispondenti. Prima di passare al calcolo degli attributi di peso tuttavia è necessario fornire alcune definizioni.

9. *Characteristic Point Index Neighborhood (IN_{CP})*: Dato un set di *characteristic points* $CPS = \{cp_1, cp_2, \dots, cp_i, \dots, cp_m\}$, il *characteristic point index neighborhood* di un *characteristic point* cp_i, $i \in [1, m]$, è definito come

$$IN_{CP}(cp_i) = \{cp_k \mid |i-k| \leq indexR, \ \forall cp_k \in CPS\} \tag{2.4}$$

Dove *indexR* è un numero intero e positivo che controlla il raggio di IN_{CP}.

10. *Characteristic Point Neighborhood Stay Time (NST_{CP})*: Dato un *characteristic point index neighborhood* $IN_{CP}(cp_i) = \{cp_l, cp_{l+1}, \dots, cp_i, \dots, cp_m\}$, il *characteristic point neighborhood stay time* di un un *characteristic point* cp_i, $i \in [l, j]$, è definito come

$$NST_{CP}(cp_i) = cp_j.T - cp_l.T \tag{2.5}$$

Dove $cp_l.T$ e $cp_l.T$ sono i marcatori temporali rispettivamente dell'ultimo e del primo *characteristic point* di $IN_{CP}(cp_i)$.

Come accennato in precedenza, gli algoritmi utilizzati nel significant place mining utilizzano solitamente il numero di punti per raggio unitario – la densità – per rappresentare la coesione dei punti in un *sinificant place*. Tuttavia risulta complesso lavorare con la densità quando i punti sono sparsi. Come introdotto nel §1.3 l'algoritmo GB-SPM prende ispirazione anche dal data field theory (Liu et al., 2017) ispirato al campo della fisica. Tale teoria descrive il campo fisico delle

particelle in interazione. Li et al. (2017) hanno proposto di considerare ogni oggetto movente come una particella fisica dotata di massa e di usare lo spazio dei dati per descrivere la relazione tra di loro. Contemporaneamente, essi usano la funzione potenziale per calcolare il valore potenziale nello spazio dei dati e descrivono quantitativamente l'influenza di ciascun punto sugli altri punti nel campo. Wang et al. (2022) hanno pensato di applicare i concetti elaborati da Li et al. (2017) all'algoritmo GB-SPM, ottenendo in questo modo la definizione del potenziale di *characteristic point*.

11. *Characteristic Point Potential (P_{CP})*: Data una traiettoria Data una traiettoria $TR = \{p_1, p_2, \ldots, p_i, \ldots, p_n\}$ e un *characteristic point index neighborhood* $IN_{CP}(cp_i) = \{cp_l, cp_{l+1}, \ldots, cp_{j-1}, cp_j\}$, il *characteristic point potential* di un *characteristic point* cp_i, $i \in [l, j]$, è definito come:

$$P_{CP}(c\text{-}p_i) = \sum_{i=l}^{j-1} e^{-\frac{Dist(p_i, p_{i+1})}{\sigma_{TR}}} \quad (2.6)$$

Dove $Dist(p_i, p_{i+1})$ denota la distanza euclidea tra due punti vicini p_i e p_{i+1} e σ_{TR} è la deviazione standard della distanza tra due punti adiacenti nella traiettoria *TR*.

Molti algoritmi SPM affrontano il problema dell'attribuzione del punto limite, ovvero dove assegnare i punti di confine tra due cluster. Questo problema è illustrato in Figura 2.3, dove sono raffigurati 15 punti in un piano cartesiano. Per intuizione potremo raggruppare i cinque punti rossi nello steso cluster C_1 ed i restanti nove punti blu nel C_2. Tuttavia, la maggior parte degli algoritmi considera la distanza di densità come unico criterio per creare un cluster e ignora di conseguenza i punti circostanti attorno ad un punto specifico. Di conseguenza alcuni punti limite saranno distribuiti in modo contro intuitivo. Gli algoritmi che considerano solo la distanza di densità infatti classificano erroneamente il punto *c* al cluster C_1 mentre intuitivamente esso dovrebbe appartenere a C_2. Per risolvere questo problema Wang et al. (2022) hanno utilizzato il concetto di potenziale ripreso dal data field theory (Li et al., 2017) per descrivere i punti circostanti ad un punto specifico e proporre la seguente definizione di potenziale relativo $RP(v_i, v_j)$:

$$RP(v_i, v_j) = \begin{cases} P_{CP}(v_i) - P_{CP}(v_j), \\ se\ P_{CP}(v_i) \leq P_{CP}(v_j) \\ P_{CP}(v_j) - P_{CP}(v_i), \\ se\ P_{CP}(v_i) > P_{CP}(v_j) \end{cases} \quad (2.7)$$

Dove *RP* riflette la differenza di potenziale tra due punti v_i e v_j e $RP(v_i, v_j)$ è minore o uguale a zero. Inoltre v_i e v_j avranno una maggiore probabilità di appartenere allo stesso cluster quando $RP(v_i, v_j)$ è più vicino allo zero. Riprendendo la Figura 2.3 possiamo notare che *RP(b, a)* è più vicino allo zero rispetto a *RP(b, c)*. Di conseguenza il punto *b* deve essere annesso al cluster C_1 piuttosto che al C_2.

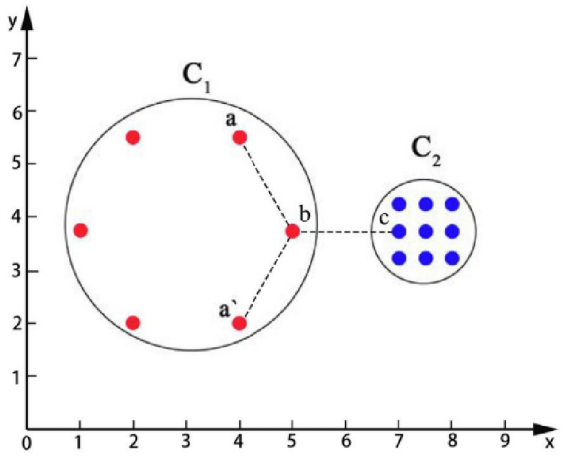

Figura 2.3 *Illustrazione del problema dell'attribuzione del punto limite. La distanza tra il punto b e a è uguale alla distanza tra il punto b e c. Intuitivamente il punto b dovrebbe essere assegnato al cluster C_1 e il punto c al cluster C_2. Inoltre per intuito il cluster C_1 è composto da sei punti rossi a bassa densità, mentre il cluster C_2 da nove punti blu ad alta densità (Wang et al., 2022).*

Il $RP(v_i, v_j)$ considera la concentrazione di punti nel *index neighborhood* da una prospettiva spaziale per risolvere il problema dell'assegnazione dei punti limite. Inoltre per rappresentare al meglio la relazione tra due nodi nel grafo *CPIN* da una prospettiva temporale, Wang et al. (2022) introducono il concetto di $MST(v_i, v_j)$ per descrivere il tempo di permanenza reciproca di due nodi adiacenti v_i e v_j.

$$MST(v_i, v_j) = \frac{1}{2}(NST_{CP}(v_i) + NST_{CP}(v_j)) \quad (2.8)$$

Per clusterizzare i nodi nel grafo *CPIN* viene utilizzato l'algoritmo LPA. Il peso di default dell'arco considerato dal LPA è *1* e l'algoritmo propaga le etichette assegnando ad un nodo l'etichetta che ha il maggior numero di vicini. Pertanto, LPA considera la topologia di rete del grafo ma ne ignora gli attributi. Wang et al. (2022) quindi hanno proposto una nuova metrica per calcolare i pesi degli archi in modo da catturare la somiglianza tra i nodi nei *significant place*. Nella maggior parte dei casi, i risultati dell'estrazione di *significant places* saranno più ragionevoli se si considera che un nodo e la maggioranza dei suoi nodi vicini dovrebbero essere collocati nello stesso *significant place* dal punto di vista della struttura del grafo; inoltre i nodi nel medesimo *significant place* dovrebbero avere attributi uguali o simili. Sulla base di queste considerazioni quindi Wang et al. (2022) hanno esteso l'algoritmo LPA perché questo estragga i *SP* tenendo conto sia della struttura del grafo - la topologia del grafo *CPIN* - che degli attributi del nodo – *neighborhood velocity NV*, potenziale relativo *RP* e il tempo di permanenza reciproca *MST*. Tutto ciò avviene grazie alla definizione di una nuova metrica di peso degli archi:

12. *Edge weight (w)*: Dato un grafo *CPIN* $G_{CPIN} = (V, E, W)$, il *edge weight* di qualunque nodo v_i e v_j è definito come:

$$w(v_i, v_j) = MST(v_i, v_j) \times RP(v_i, v_j) \times e^{-Dist(v_i, v_j)} \quad (2.9)$$

Dove $Dist(v_i, v_j)$ denota la distanza euclidea tra due punti v_i e v_j adiacenti e $w \in W$.

La Figura 2.4 mostra l'esempio di un grafo *CPIN* per un set di *characteristic points* estratti dall'Algoritmo 1.1. Data una serie di *characteristic points CPS* essa viene convertita in un set di vertici $V = \{v_1, v_2, v_3, v_4, v_5\}$. Inizialmente, prima del processo di clusterizzazione, viene assegnata un'unica etichetta ad ogni nodo. Ad esempio nella Figura 2.4 i nodi $\{v_1, v_2, v_3, v_4, v_5\}$ sono contrassegnati da cinque etichette sequenziali {1, 2, 3, 4, 5}. Dal momento che i nodi che possiedono una densità maggiore hanno più probabilità di essere centri di un cluster, le etichette vengono aggiornate in ordine decrescente in base al *characteristic point potential* P_{CP} in modo da considerare l'influenza di ciascun nodo sul cluster.

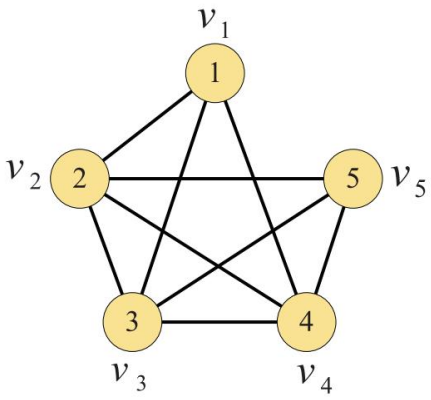

Figura 2.4 *Illustrazione dei nodi iniziali. I nodi v_1, v_2, v_3, v_4, v_5 rappresentano i nodi nel grafo CPIN mappati secondo un set di characteristic points CPS = {cp_1, cp_2, cp_3, cp_4, cp_5}. Prima del processo di clusterizzazione, un'unica etichetta 1, 2, 3, 4, 5 viene assegnata ad ogni nodo (Wang et al., 2022).*

Finito tale processo si passa alla fase finale dell'algoritmo GB-SPM, ovvero quella che prevede il clustering sul grafo *CPIN*.

Arrivati a questo punto, l'obiettivo è quello di estrarre i *significant places* attravrso l'algoritmo LPA con la nuova metrica proposta. A tal propositivo Wang et al. (2022) hanno proposto un nuovo algoritmo denominato *SignificantPlaceMining* - Algoritmo 2.2. Esso richiede come input una traiettoria *TR*, un set di *characteristic points CPS* ricavati dal'algoritmo 2.1 e i parametri *indexR* e *minDration*. L'output invece è composto da un set di *significant plaes* $SPS = \{SP_1, SP_2, \dots, SP_N\}$. In primo luogo l'algoritmo *SignificantPlaceMining* collega ogni punto contenuto in *CPS* ad un nodo nel set dei vertici *V* in G_{CPIN}. In l'algoritmo assegna un'etichetta specifica ad ogni nodo in *V*, rappresentato da un numero dell'insieme {1, 2, … , *Nv*}. In seguito la funzione *setUpdateOrder(V)* organizza i nodi in *V* in ordine decrescente affidandosi al *characteristic point potential* P_{CP}. Dopo le etichette dei nodi in *V* vengono propagati interattivamente seguendo l'ordine della lista *orderList* fino a quando l'ordine delle etichette cessa di cambiare. Durante ogni interazione vengono utilizzate le definizioni 2.7 e 2.8 per calcolare rispettivamente il potenziale relativo $RP(v_i, v_j)$ e il tempo di permanenza reciproca $MST(v_i, v_j)$ per ogni nodo v_i e v_j nel *index neighborhood* di v_i. In seguito viene applicata la nuova metrica del peso degli archi secondo la definizione 2.9. In fine l'algoritmo determina il nodo vicino v_j che risulta più simile ad ogni nodo v_i in base al massimo valore di $w(v_i, v_j)$ e

aggiorna l'etichetta di v_i a quella dell'etichetta vicina – ad esempio l'etichetta di v_j. Quando l'etichetta viene propagata in tutto il grafo *CPIN*, il gruppo di nodi connesso spaziotemporalmente raggiunge un consenso sulla sua etichetta. In seguito, i nodi portatori della medesima etichetta vengono raggruppati nello stesso cluster, che forma il risultato finale del *stop region* (definizione 1.2) . Per tutti i punti della traiettoria *TR* che non sono posizionati in nessun cluster – *stop region* SR_i – viene generato e ammesso un *moving region* (definizione 1.3) nel set finale del *moving region MR*. Successivamente i *stop region* in *SR* relativamente vicini nel tempo e nello spazio vengono unite per ottenere i *stop region* con durata maggiore o uguale al parametro *minDuration*. Infine, l'algoritmo filtra i *stop regions* con troppi pochi punti – ad esempio meno di uno – o quelli con raggio eccessivamente grande e forma il set finale dei *significant places SPS*.

Algoritmo 2.2: *SignificantPlaceMining*

Require *TR, CPS,indexR, minDuration*

Ensure *SPS*

1: map cp_i in *CPS* to v_i in node set $V \leftarrow \{v_1, v_2, \ldots, v_N\}$;
2: isChange ← True;
3: set a unique label for each node in *V*;
4: orderList ← setUpdateOrder(*V*)
5: **repeat**
6: **for all** each node v_i in orderList **do**
7: count ← 0;
8: record $label\_v_i$ of v_i
9: $indexNeighbor\_v_i \leftarrow index\_neighbor(v_i, indexR)$
10: **for all** each node v_i in $indexNeighbor\_v_i$ **do**
11: W←∅
12: calculate $MST(v_i,v_j)$;
13: calculate $RP(v_i,v_j)$;
14: $w(v_i, v_j) \leftarrow MST(v_i, v_j) \times RP(v_i, v_j) \times e^{-Dist(v_i,v_j)}$;
15: add $w(v_i, v_j)$ into *W*;
16: **end** for

```
            select node v_i with max w(v_i, v_j) in W;
            updateLabel ← label of node v_i;
            if updateLabel ≠ label_v_i then
                change label_v_i to updateLabel;
                count ← count + 1;
            end if
        end for
        if count == 0 then
            isChange ← False;
         end if
    until isChange ← False
    place the nodes with the same label in a cluster SR_i;
    place all the SR_i into stop region set SR;
    get move region set MR by TR and SR;
    SPS ← mergeAndFilterCluster(TR, SR, minDuration);
    Return SPS
```

In Figura 2.5 viene illustrato il processo di propagazione dell'etichetta per il grafo CPIN della Figura 2.4. In primis vengono calcolati i *characteristic point potential* P_{CP} dei cinque nodi, i quali vengono disposti in ordine decrescente, come mostrato nella Tabella 2.1.

	v_4	v_5	v_2	v_1	v_3
P_{CP}	5.3	4.8	4.2	3.9	3.8

Tabella 2.1 *Characteristic point potential P_{CP} di ogni nodo in Figura 2.4 (Wang et al., 2022).*

In quanto $P_{CP}(v_4)$ è il più elevato, v_4 è il primo nodo in attesa di un aggiornamento della sua etichetta. Come mostrato in Figura 2.5a, l'arco di colore blu connette v_4 con i nodi $\{v_1, v_2, v_3, v_5\}$ nel suo *index neighborhood* mentre i numeri posti su ogni arco rappresentano *edge weight* w – per esempio $w(v_4, v_1) = 4.87$, $w(v_4, v_2) = 4.33$, $w(v_4, v_3) = 5.87$ e $w(v_4, v_5) = 6.33$. Il valore $w(v_4, v_5) = 6.33$ rappresenta il massimo

peso degli archi di v_4 e per questo è raffigurato in rosso. Di conseguenza l'etichetta di v_4 viene aggiornata con quella di v_5, come in Figura 2.5b. I nodi rimanenti aggiornano le loro etichette in modo simile, come mostrato in Figura 2.5b-e. Quando termina il processo di aggiornamento, i nodi associati con la stessa etichetta vengono raggruppati in un *stop region* – vedi Figura 2.5f. Si può osservare che l'intero grafo CPIN così ottenuto è diviso in due *stop regions*, rappresentate dai nodi verdi e blu.

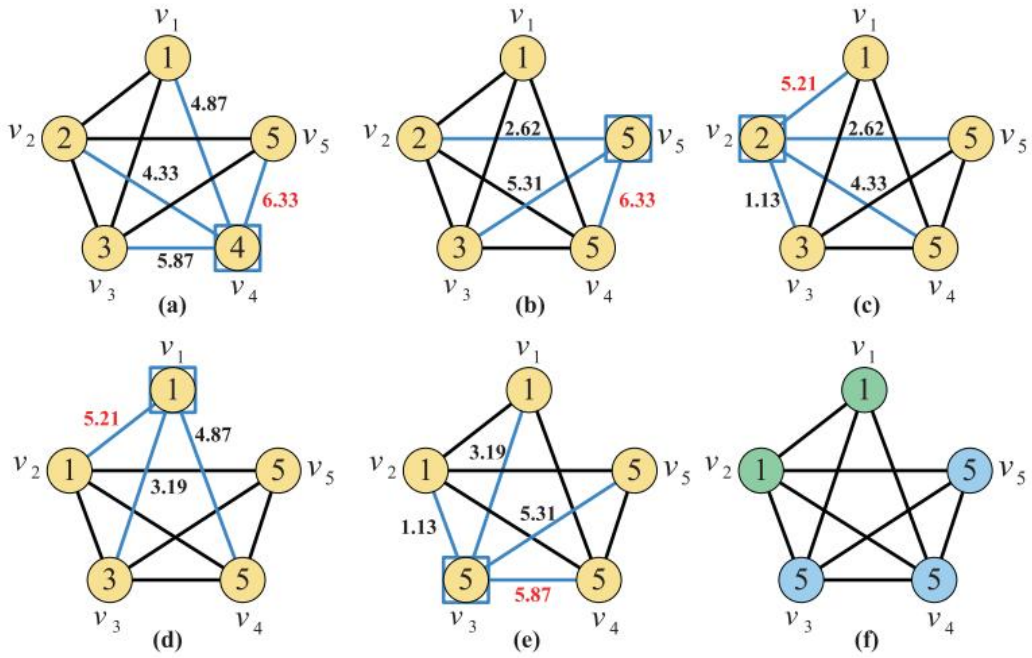

Figura 2.5 *Il processo di propagazione dell'etichetta di Wang et al.(2022).* ***(a)*** *Mostra il nodo selezionato v_4 e i suoi archi blu che lo collegano agli altri punti {v_1, v_2, v_3, v_5}. I valori sugli archi rappresentano i pesi e il massimo peso è raffigurato in rosso.* ***(b)*** *Illustra l'aggiornamento dell'etichetta di v_4 in quella corrispondente al nodo di v_5.* ***(c)*** *–* ***(e)*** *Mostrano i risultati intermediari.* ***(f)*** *Rappresenta il set finale di stop regions dove i nodi {v_1, v_2} vengono raffigurati in verde mentre i nodi {v_3, v_4, v_5} in blu.*

2.3 Analisi dei dati tramite il GB-SPM

Il dataset sul quale è stato applicato l'algoritmo GB-SPM è relativo alla ricerca di Monaro et al. (2018) e contiene 2052 osservazioni relative a 209 variabili. L'analisi è stata condotta nel software R nella versione 4. 0. 4. Di seguito si riporta la lista delle informazioni disponibili:

- *Subject*: il numero identificativo del soggetto.
- *Trial*: variabile categoriale che identifica il item.
- *Condition*:variabile categoriale dove *condition* = 1 indica un paziente realmente depresso oppure un paziente sano ma sincero e *condition* = 2 indica un soggetto che si avvale del faking.
- *Init_time*: tempo di latenza tra la comparsa dello stimolo e l'inizio del movimento del mouse.
- *RT*: tempo totale impiegato dal soggetto per rispondere al item.
- *Vel_x*: velocità media del mouse lungo l'asse delle x.
- *Acc_x:* accelerazione media del mouse lungo l'asse delle x.
- Le variabili comprese tra la colonna 6 e 106 rappresentano le posizioni normalizzate in 101 frame lungo l'asse delle x per ogni soggetto e trial, vedi §1.1.
- Le variabili comprese tra la colonna 107 e 207 rappresentano le posizioni normalizzate in 101 frame lungo l'asse delle y per ogni soggetto e trial.

Tale dataset fa riferimento al secondo campione indipendente dello studio di Monaro et al. (2018) (§2.1) e comprende in totale 27 soggetti, di cui 18 individui sinceri e nove soggetti costretti a simulare. In questo dataset il gruppo di individui sinceri sani è stato accorpato al gruppo di depressi, formando un'unica condizione, quella relativa all'etichetta pari a 1. Inoltre nel dataset mancavano i tempi relativi ad ogni coordinata x e y e di conseguenza è stata creata una matrice contente i tempi calcolati suddividendo gli *RT* per le coordinate campionate.

Lo scopo di questa analisi è di verificare se i soggetti che si avvalgono del faking presentano un numero maggiore di *significant places* rispetto a coloro che rispondono sinceramente. In linea con i risultati della letteratura – vedi §1.1, §1.2 e §2.1 – ci aspettiamo che i soggetti che simulano presentino un maggior numero di *significant*

places in quanto questo corrisponde indirettamente ad un tempo di esecuzione complessivo più ampio.

Rispetto all'Algoritmo 2.2 di Wang et al. (2022) è stata apportata una modifica nel calcolo dei pesi *Edge weight (w)*, ovvero inspiegabilmente alcuni pesi risultavano negativi e per questo motivo è stato aggiunto un valore assoluto nella formula della Definizione 2.9.

Per consultare i codici di programmazione R dell'algoritmo GB-SPM invito a vedere Appendice A.

Come abbiamo già spiegato nel §1.3, il GB-SPM risulta efficace anche grazie al fatto che richiede di fissare poche costanti di soglia. Per quel che riguarda i parametri di soglia, tramite un approccio a mano, sono stati fissati *minVelocity* = 0.005, *indexR* = 2 e *minDuration* = 2.

Prima di procedere con l'applicazione dell'algoritmo è risultato necessario creare tre matrici contenenti rispettivamente le coordinate x e y e i tempi per ogni posizione. Dopo aver inserito come input le tre matrici e i parametri precedentemente citati, l'algoritmo restituisce come output una lista di componenti: la lista dei *characteristic points*, la lista dei *significant places*, gli archi, i pesi degli archi *w* e la quantità totale dei *characteristic points* calcolati per tutti i soggetti e trial del dataset. Ad esempio il soggetto *subject* = 1 e il item *trial* = 10 possiede 41 *characteristic points* e 41 *significant places* ed in Figura 2.6 è possibile vedere la sua traiettoria.

In seguito il numero dei *significant places* per ogni soggetto e trial è stato isolato in una nuova variabile allo scopo di verificare se effettivamente i soggetti che si avvalgono del faking presentano un maggior numero di *significant place* rispetto ai soggetti depressi e sinceri.

Il dataset finale, composto da 2052 osservazioni relative a sei variabili, è stato di conseguenza modificato e risulta composto dalle variabili *subject*, *init_time*, *RT*, *trial*, *condition* e *num_sps*, ovvero il numero di *significant places*. Il numero dei *significant places* varia tra 1 e 91. In Figura 2.7a è possibile osservare la distribuzione marginale della variabile risposta – *num_sps*; la Figura 2.7b invece mostra i boxplot relativi alle due condizioni al variare di *num_sps*. L'analisi grafica suggerisce che il numero dei *significant places* non è distribuito normalmente e che la condizione *condition* = 2 presenta meno *significant places* rispetto alla condizione *condition* = 1. Di conseguenza,

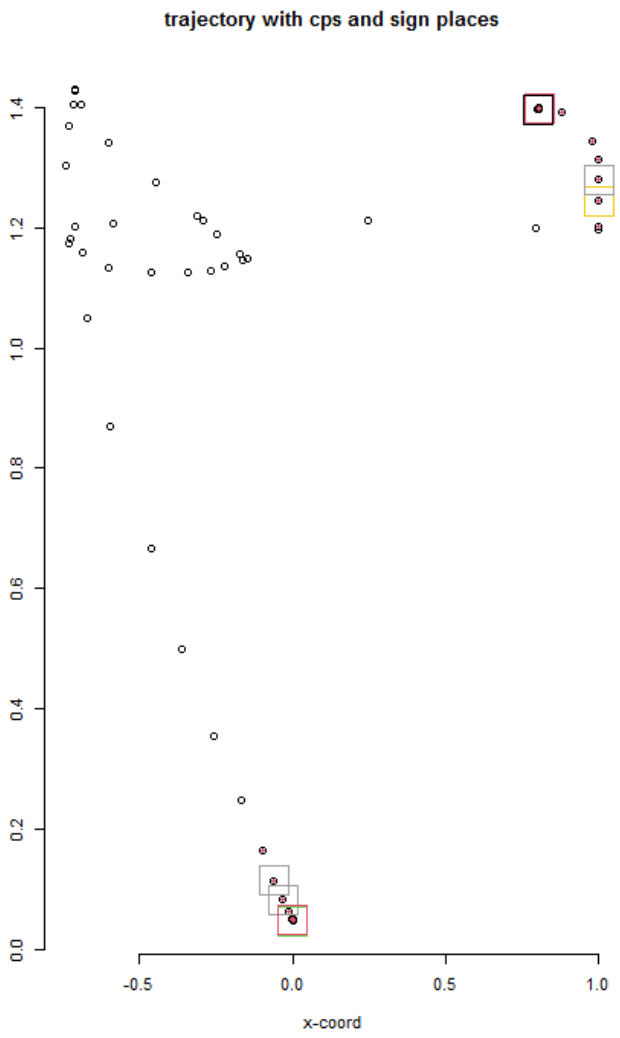

Figura 2.6 *Illustrazione della traiettoria del soggetto subject = 1 relativo al item trial = 10. I punti rossi corrispondono ai characteristic points e i quadrati corrispondono ai significant places.*

almeno graficamente, la mia ipotesi non è confermata in quanto i sinceri presentano un maggior numero di *significant places* rispetto a coloro che si avvalgono del faking.
Per verificare questa ipotesi è stato implementato un modello di regressione di Poisson, vista la natura della variabile dipendente. Per selezionare il modello migliore è stato utilizzato il metodo *stepwise backward*, che seleziona tutte le variabili che contribuiscono a spiegare la variabile dipendente, ovvero il numero di *significant places*. Il modello migliore risulta essere formato dalle variabili: *condition*, *trial*, *subject*, *init_time* e *RT* e presenta un *AIC* pari a 35510. Come accennato nell'analisi

grafica, la variabile condizione *condition* = 1, ovvero la categoria dei sinceri, presenta significativamente un numero maggiore di *significant places* rispetto a coloro che hanno

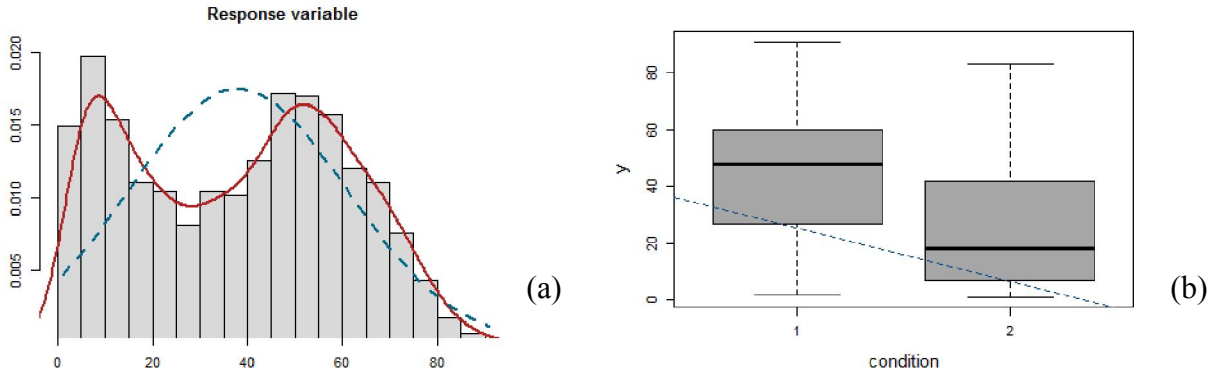

Figura 2.7 *(**a**) Distribuzione marginale della variabile risposta num_sps. La linea tratteggiata blu corrisponde alla distribuzione normale mentre la linea rossa corrisponde alla distribuzione reale dei significant places. (**b**) Boxplot relativi alle due condizioni al variare di num_sps, dove condition = 1 corrisponde ai sinceri mentre condition = 2 raggruppa coloro che usufruivano del faking.*

utilizzato il faking, vedi Figura 2.9. L'intercetta β_0 del modello codifica *condition* = 1 come livello di riferimento e il numero dei *sinificant places* decresce significativamente in funzione dei *condition* = 2 ($\beta_{condition2}$= *-3.740e-01*, $\sigma_{\beta condition2}$= *1.012e-02*, $z_{\beta condition2}$= *-36.95*).

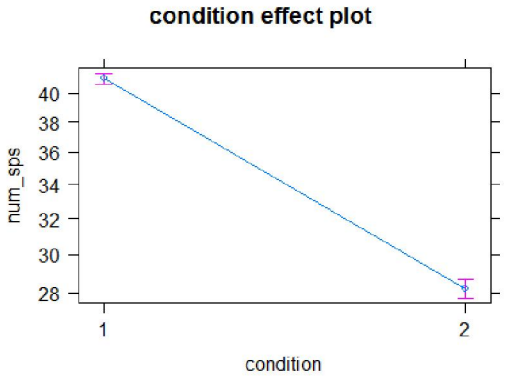

Figura 2.9 *Grafico degli effetti marginali della variabile condition rispetto alla variabile dipendente numero dei significant places. Condition = 1 indica i sinceri mentre condition = 2 coloro che usufruiscono del faking.*

L'esecuzione delle diagnostiche relative al modello stimato tuttavia ci suggerisce che il problema potrebbe essere la scelta di un modello sbagliato. Infatti il calcolo dell'indice R^2 ci dice che solo il 18% della variabilità della variabile dipendente è spiegata dal modello. Inoltre sono presenti dei punti influenti (osservazioni 2010 e 2005) l'osservazione 2001 risulta essere un *outlier*, vedi Figura 2.10a.

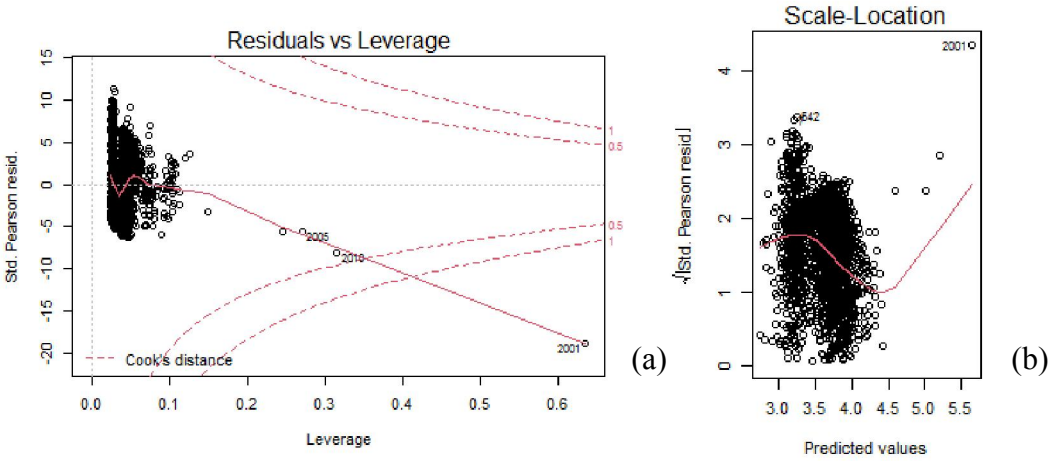

Figura 2.10 ***(a)*** *La figura mostra i residui (in ordinata) in funzione del valore di leverage (leva, in ascissa).****(b)*** *Grafico dei residui (in ordinata) in funzione dei valori attesi (in ascissa).*

I residui inoltre non sono distribuiti intorno allo zero, sono elevati e si possono osservare dei pattern, vedi Figura 2.10b. Infatti la variabile dipendente è affetta dalla sovradispersione (ϕ = 11.61).

Capitolo III – Conclusioni

3.1 Discussione e conclusioni

L'analisi dei dati tramite il modello di regressione di Poisson ha dimostrato che il numero di *significant places* nei soggetti sinceri sono maggiori rispetto ai soggetti costretti a mentire. L'intercetta β_0 del modello codifica come livello di riferimento *condition* = 1 e il numero dei *significant places* decresce significativamente in funzione di *condition* = 2 ($\beta_{condition}$= *-3.740e-01*, $\sigma_{\beta condition}$= *1.012e-02*, $z_{\beta condition}$= *-36.95*), mostrando un decremento del 69%. Tale risultato non conferma l'ipotesi iniziale che prevedeva un numero maggiore di *significant places* per i soggetti costretti ad avvalersi del faking.

Ciononostante le diagnostiche del modello non sono ottimali e solo il 18% della variabilità del numero di *significant places* è spiegata da quest'ultimo. Inoltre è stata rilevata sovradispersione per il numero di *significant places*, con un parametro di dispersione ϕ pari a 11.61. Un ϕ maggiore di 1 indica che il modello di Poisson non è il modello ideale per la variabile dipendente. Per aggirare il problema in futuro si possono provare ad utilizzare diversi approcci quali il metodo della verosomiglianza, la regressione di Poisson zero-inflated e il modello negativo-binomiale.

Un altro problema potrebbe essere rappresentato dallo scarso numero del campione utilizzato, ovvero di soli 27 soggetti, o dalla selezione errata dei parametri *indexR*, *minVelocity* e *minDuration*.

Concludendo, i risultati delle mie analisi indicano che i soggetti che rispondono sinceramente sono caratterizzati da un numero più elevato di *significant places* rispetto a coloro che mentono.

3.2 Ricerche future

Visti i problemi riscontrati, sarebbe opportuno in futuro condurre ulteriori ricerche. In primo luogo occorre capire perché durante l'implementazione di Algoritmo 2.2 di Wang et al. (2022) in R sono stati riscontrati dei risultati dei pesi *w* negativi.

In secondo luogo bisognerebbe condurre una simulazione con il metodo di Monte Carlo per assegnare dei valori ideali per i parametri *indexR*, *minVelocity* e *minDuration*.

In terzo luogo bisognerebbe condurre ricerche analoghe con campioni maggiori.

In quarto luogo sarebbe opportuno implementare più modelli stocastici al fine di individuare quello che presenta l'adattamento migliore.

Una volta individuato il modello migliore per analizzare i *significant places*, l'analisi delle traiettorie raccolte tramite i movimenti del mouse potrebbe essere compiuta tramite l'algoritmo GB-SPM nei contesti organizzativi di diversa natura. Ad esempio tale approccio potrebbe essere utilizzato nella compilazione dei questionari suscettibili alla manipolazione da parte dei soggetti, come suggerito nel §2.1.

Bibliografia

Frick, S. (2022). Modeling faking in the multidimensional forced-choice format: the faking mixture model. *Psychometrika*, 87, 773–794.

Sun, T., Zhang, B., Cao, M., Drasgow, F. (2022). Faking Detection Improved: Adopting a Likert Item Response Process Tree Model. *Organizational Research Methods*, 25(3) 490–512.

Wegmeyer, L. J., Speer, A. B. (2022). Understanding, detecting and deterring faking on interest inventories. *International Journal of Selection and Assesment*.

Hu, J., Connelly, B. S. (2021). Faking by actual applicants on personality tests: A meta-analysis of within-subjects studies. *International Journal of Selection and Assesment*.

Crawford, V. P. (2003). Lying for strategic advantage: Rational and boundedly rational misrepresentation of intentions. *The American economic review* , 93 (1), 133–149.

Gray, N., MacCulloch, M., Smith, J., Morris, M., & Snowden, R. (2003). Violence viewed by psychopathic murderers. *Nature* , 423 (6939), 497–498.

Marshall, E. (2000). Scientific misconduct - how prevalent is fraud? that's a million-dollar question. *Science*, 290 (5497), 1662–1663.

Walker, S. A., Double, K. S., Birney, D. P., MacCann, C. (2022). How much can people fake on the dark triad? A meta-analysis and systematic review of instructed faking. *Personality and Individual Differences*, 193, 111622.

Lombardi, L., Pastore, M., Nucci, M., Bobbio, A. (2015). SGR Modeling of Correlational Effects in Fake Good Self-report Measures. *Methodol Comput Appl Probab*, 17, 1037–1055.

Mazza, C., Monaro, M., Burla, F., Colasanti, M., Orrù, G., Ferracuti, S., Roma, P. (2020). Use of mouse-tracking software to detect faking-good behavior on personality questionnaires: an explorative study. *Scientific Reports*, 10(4835).

Monaro, M., Negri, P., Zecchinato, F., Gamberini, L., Sartori, G. (2021). Mouse Tracking IAT in Customer Research: An Investigation of User's Implicit Attitudes Towards Social Networks. In: Russo, D., Ahram, T., Karwowski, W., Di Bucchianico, G., Taiar, R. (Eds.) Intelligent Human System Integration 2021. IHSI 2021. *Advances in Intelligent Systemand Computing*, 1322

Greenwald, A. G., McGhee, D. E., & Schwartz, J. L. (1998). Measuring individual differences in implicit cognition: the implicit association test. *Journal of personality and social psychology*, 74(6), 1464.

Calcagnì, A., Lombardi, L., D'Alessandro, M., Freuli, F. (2019). A State Space Approach to Dynamic Modeling of Mouse-Tracking Data. *Frontiers in Psychology*, 10(2716).

Monaro, M., Cannonito, E., Gamberini, L., Sartori, G. (2020). Spotting faked 5 stars ratings in E-Commerce using mouse dynamics. *Computers in Human Behavior*, 109 (106348)

Monaro, M., Toncini, A., Ferracuti, S., Tessari, G., Vaccaro, M. G., De Fazio, P., Pigato, G., Meneghel, T., Scarpazza, C., Sartori, G. (2018). The Detection of Malingering: A New Tool to Identify Made-Up Depression. *Frontiers in Psychology*, 9(249)

Bertelloni, D. (2022, 23 febbraio). Come funziona l'MMPI-2: scale, valutazione e impiego. *Unopsicologo.it – Psicologo per gli studenti*. Disponibile in: https://www.unopsicologo.it/come-funziona-lmmpi-2-scale-valutazione-e-impiego/

Wang, S., Niu, X., Fournier-Viger, P., Zhou, D., Min, F. (2022). A graph based approach for mining significant places in trajectory data. *Information Sciences*, 609, 172–194

Monaro, M., Gamberini, L., Sartori, G. (2017). The detection of faked identity using unexpected questions and mouse dynamics. *PLoS ONE*, 12(e0177851).

Niu, X., Wang, S., Wu, C. Q., Li, Y., Wu, P., Zhu, J. (2021). On a clustering-based mining approach with labeled semantics for significant place discovery, *Information Sciences*, 578, 37–63.

Li, D., Du, Y. (2017). *Artificial intelligence with uncertainty*, CRC Press.

Appendice A

Codici di programmazione R

```
rm(list=ls())
datax = read.csv2(file = "C:\\Users\\Dorosh_Olga\\Desktop\\data.csv",header = TRUE,sep = ",")

CPExtraction = function(x,y,tt,indexR=3,minVelocity=3.0){
  n = length(x)
  Cps = rbind()
  IID = mapply(function(i)abs(i-(1:n))<=indexR,1:n)
  for(ii in 1:n){
    print(ii)
    last = sum(IID[ii,])
    d_pi = sqrt((y[IID[ii,]][1] - x[IID[ii,]][1])^2 + (y[IID[ii,]][last] - x[IID[ii,]][last])^2)
    nv_pi = d_pi / abs(tt[IID[ii,]][last]-tt[IID[ii,]][1])
    if(nv_pi<=minVelocity){
      Cps = rbind(Cps,c(x[ii],y[ii],tt[ii]))
    }
  }
  return(Cps)
}

compute_rpcp = function(pc_vi,pc_vj){
  if(pc_vi<=pc_vj){
    return(pc_vi-pc_vj)
  }else{
    return(pc_vj-pc_vi)
  }
}

compute_mst = function(nst_vi,nst_vj){
  return(0.5*(nst_vi+nst_vj))
}

refine_cluster = function(X=NULL,minVelocity_sps=0.05){
  m = NROW(X)
  dj = cl = rep(NA,m); cl = X[,4]
  for(j in 2:m){
```

```
    dj[j] = as.numeric(sqrt((X[j,1]-X[j-1,1])^2 + (X[j,2]-X[j-1,2])^2) / abs(X[j,3]-X[j-1,3]))
    if(is.nan(dj[j]) | is.infinite(dj[j])){dj[j]=0}
    if(dj[j]<=minVelocity_sps){
      cl[j] = X[j-1,4]
    }
  }
  return(cl)
}


significant_place_mining =
function(x=NULL,y=NULL,tt=NULL,indexR_cps=3,indexR_incp=2,minVelocity_cps=3.0,minVelocity_
sps=0.05,refine_cluster=TRUE,plotx=FALSE){
  Cps=NULL;num_cps=NULL;E=NULL;W=NULL;N=NULL

  Out = CPExtraction(x = x,y = y,tt = tt,indexR = indexR_cps,minVelocity = minVelocity_cps)
  num_cps = NROW(Out)
  if(!is.null(Out)){
    Cps = Out

    m = num_cps
    v = 1:m #nodes
    IID = mapply(function(j)abs(j-(1:m))<=indexR_incp,1:m) #characteristic point index neighborhood
INcp
    E = IID*1 #edges
    rownames(E)=colnames(E)=1:m

    sigma_tr = sd(mapply(function(i)sqrt((x[i+1]-y[i])^2 + (x[i+1]-y[i])^2),1:(length(x)-1)))

    # Compute Characteristic Point Potential (Def.11) and Characteristic Point Neighborhood Stay Time
(Def.12)
    pcp = matrix(NA,m,1); nstcp = matrix(NA,m,1)
    for(i in 1:m){
      Cp_current = Cps[which(E[i,]==1),1:2]; p = NROW(Cp_current)
      pcp[i] = sum(exp(-mapply(function(j)sqrt((Cp_current[j+1,1]-Cp_current[j,1])^2 +
(Cp_current[j+1,2]-Cp_current[j,2])^2),1:(p-1))/sigma_tr)) #Def.11
      last = sum(E[i,])
      nstcp[i] = Cps[which(E[i,]==1),3][last] - Cps[which(E[i,]==1),3][1] #Def.12
    }
```

```
  Iid = expand.grid(1:m,1:m)
  W = matrix(data = NA,nrow = m,ncol = m) # weights
  for(k in 1:(m*m)){
   h = Iid[k,1]; l = Iid[k,2]
   W[h,l] = compute_mst(nstcp[h],nstcp[l]) * compute_rpcp(pcp[h],pcp[l]) * exp(-sqrt((Cps[l,1] -
Cps[h,1])^2 +  (Cps[l,2] - Cps[h,2])^2))
  }
  W = abs(W)

  # Relabelling
  N = matrix(NA,m,4); colnames(N)=c("original node","updated label","pcp","current node")
  N[,c(1,2,4)] = order(pcp,decreasing = TRUE) #order nodes according to largest pcp
  N[,3] = pcp[N[,1]]

  while(sum(is.na(N[,4]))!=m){
   j=which(!is.na(N[,4]))[1]; #current node index
   neigh_j = which(E[N[j,4],]!=0); neigh_j = setdiff(neigh_j,N[j,4]) #find neighbors of current node
   jjd = neigh_j[which.max(W[N[j,4],neigh_j])] #find maximum connection with current node
   N[j,2] = jjd; N[j,3] = pcp[jjd]
   N[c(j,which(N[,4]==jjd)),4] = NA
  }

  Q = unique(N[,2]); gps = 1:length(Q)
  for(q in 1:length(Q)){N[N[,2]==Q[q],4] = gps[q]};
  colnames(N)[4] = "significant places (clusters)"

  if(refine_cluster==TRUE){
   X = cbind(Cps[sort(N[,2]),],N[sort(N[,2],index.return=TRUE)$ix,4])
   cl = refine_cluster(X,minVelocity_sps)
   N[,4] = cl
  }

  if(plotx==TRUE){
   plot_output(x = x,y = y,Cps = Cps,E = E,N = N)
  }
 }else{
  message("Error: Significant Places cannot be determined for the current data")
  }
```

```
  return(list(Cps=Cps,num_cps=num_cps,Edges=E,Weights=W,SignPlaces=N))
}


plot_output = function(x=NULL,y=NULL,Cps=NULL,E=NULL,N=NULL,onlyTraj=FALSE){
  require(igraph)
   require(Polychrome)
   glasbey.colors(max(N[,4]))
   cls = createPalette(max(N[,4]),  c("#ff0000", "#00ff00", "#0000ff"))
            cls_iid = as.numeric(mapply(FUN = function(k)strsplit(x = names(cls),split =
"NC")[[k]][2],1:length(cls)))
   for(k in 1:length(cls)){N[N[,4]==cls_iid[k],4] = cls[k]}

  cpin_graph = as.undirected(graph = graph.adjacency(adjmatrix = E, diag = FALSE,weighted = NULL))

  if(onlyTraj==FALSE){
    par(mfrow=c(1,3))
    plot(x,y,bty="n",xlab="x-coord",ylab="y-coord",main="trajectory with cps and sign places")
    points(Cps[,1],Cps[,2],col=2,pch=20) #plot cps over the recorded trajectory
    points(Cps[N[,2],1],Cps[N[,2],2],col=N[,4],pch=0,cex=4)

    plot(cpin_graph,layout=Cps[,1:2],main="CPIN graph")

plot(cpin_graph,vertex.color=N[,4],vertex.shape="square",vertex.label.color="black",vertex.label.font=2,
cex=3,edge.curved=0.2,layout=Cps[,1:2],
        main="Significant places")
  }else{
    par(mfrow=c(1,1))
    plot(x,y,bty="n",xlab="x-coord",ylab="y-coord",main="trajectory with cps and sign places")
    points(Cps[,1],Cps[,2],col=2,pch=20) #plot cps over the recorded trajectory
    points(Cps[N[,2],1],Cps[N[,2],2],col=N[,4],pch=0,cex=4)

  }
}
head(datax)
str(datax)
sbjs = unique(datax$subject)
X = rbind(); Y = rbind(); TMS = rbind(); design = rbind()
```

```
for(i in 1:length(sbjs)){
  print(i)
  X = rbind(X,datax[datax$subject==sbjs[i],paste0("X_",1:101)])
  Y = rbind(Y,datax[datax$subject==sbjs[i],paste0("Y_",1:101)])
  TMS =
rbind(TMS,t(mapply(function(j)seq(from=0,to=datax[datax$subject==sbjs[i],"RT"][j],by=(datax[datax$s
ubject==sbjs[i],"RT"]/100)[j]),1:NROW(datax[datax$subject==sbjs[i],paste0("X_",1:101)]))))
  design = rbind(design,datax[datax$subject==sbjs[i],1:3])
}
NN = NROW(X) #total number of observations/trajectories
CPS = array(data = NA,dim = c(NN,101,3),dimnames = list(NULL,NULL,c("x-coord","y-
coord","timestamp"))) #multi-dimensional array (sbj,x,y,time) for characteristic points
num_cps = matrix(data = NA,nrow = NN,ncol = 1)
num_sps = matrix(data = NA,nrow = NN,ncol = 1)
SPS = array(data = NA,dim = c(NN,101,4),dimnames = list(NULL,NULL,c("original node","updated
label","pcp","significant places (clusters)"))) #multi-dimensional array (sbj,x,y,time) for significant places
Es = array(data = NA,dim = c(NN,101,101))
for(i in 1:NN){
  print(paste0("Trajectory no.: ",i))
  out = significant_place_mining(x = as.numeric(X[i,]),y = as.numeric(Y[i,]),tt =
as.numeric(TMS[i,]),indexR_cps = 2,indexR_incp = 2,minVelocity_cps = 0.005)
  if(!is.null(out$Cps)){
    num_cps[i] = out$num_cps
    CPS[i,1:num_cps[i],] = out$Cps
    SPS[i,1:num_cps[i],] = out$SignPlaces
    Es[i,1:num_cps[i],1:num_cps[i]] = out$Edges
    num_sps[i] = max(out$SignPlaces[,4])
  }
}
datax = cbind(datax[,1:5],num_sps)
head(datax) #num_sps is the outcome variable (number of sign places)
datax$condition=as.factor(datax$condition)
datax$trial=as.factor(datax$trial)
exploratory_plots(y=datax$num_sps, X=datax[,c(1,2,3,4,5)])
psych::describe(x=datax)
mod0 = glm(data=datax, formula = num_sps~1, family="poisson")
step(mod1,direction = "backward")
mod1= glm(data=datax, formula = num_sps~., family="poisson")
IRR=exp(mod1$coefficients)
```

```
IRR
R2 = 1-logLik(mod1)/logLik(mod0)
R2
summary(mod1)
plot(effects::allEffects(mod1))
plot(mod1)
mod1a= glm(data=datax, formula = num_sps~., family="quasipoisson")
summary(mod1a)
```