
Università di Verona
Dipartimento di Informatica
Laurea Magistrale in Ingegneria e Scienze Informatiche

**Apprendimento non supervisionato per
l'identificazione di contesti di
Freezing of Gait (FoG)
in pazienti affetti da morbo di Parkinson**

Candidato
Fuser Alessandro

Supervisore
Graziano Pravadelli

Assistente Supervisore
Florenc Demrozi

Sessione di Laurea del 21 Marzo 2018
Anno Accademico 2016/2017

Abstract

Ringraziamenti

Indice

Ringraziamenti	v
1 Introduzione	1
1.1 Contributo della Tesi	1
2 Letteratura	3
3 Conoscenze Preliminari	7
3.1 Freezing Of Gait	7
3.1.1 Le tipologie di Fog	8
3.1.2 Influenza del FoG nella camminata	8
3.2 Machine Learning	9
3.2.1 Cluster Analysis	12
3.2.2 Classificazione	19
3.2.3 Validazione della Classificazione	21
3.3 Riduzione della dimensionalità	22
3.3.1 Analisi Delle Componenti Principali	22
3.3.2 Analisi dei Discriminanti Lineari	24
4 Motivazioni ed obiettivi	27
5 Apprendimento non supervisionato	29
5.1 Dataset	29
5.2 Dimostrazione dell'esistenza effettiva delle 3 classi	30
5.2.1 Divisione dei dati	31
5.2.2 Algoritmo di discriminazione	31
5.2.3 Risultati	33
5.3 Determinazione del miglior intervallo di finestra	34
5.3.1 Pre-processamento dei dati	35
5.3.2 Definizione degli intervalli	36
5.3.3 Calcolo delle Feature	37
5.3.4 Clustering	40
5.3.5 Calcolo delle prestazioni	42
5.3.6 Risultati	43

5.4 Etichettatura dei Dati	46
6 Conclusione e Lavori Futuri	47
6.1 Lavori Futuri	47
Bibliography	49

Elenco delle tabelle

3.1	Misure di Similarità	14
5.1	Descrizione delle Feature Statistiche	38
5.2	Esempio di tabelle delle prestazioni del clustering	43
5.3	Esempio di tabelle delle prestazioni del clustering	44
5.4	Esempio di tabelle delle prestazioni del clustering	44
5.5	Rate Pazienti per l'algoritmo di clustering C-means	45
5.6	Rate Pazienti per l'algoritmo di clustering K-means	45
5.7	Rate Pazienti per l'algoritmo di clustering Self-Organizing Map	45

Elenco delle figure

3.1	Diversi tipi di Freezing esistenti e loro percentuale di incidenza su un certo gruppo di malati di Parkinson. [Fonte: J.M. Shine et al., 2012]	9
3.2	Quadro concettuale relativo al Freezing of Gait (FoG) riguardante aspetti mentali e motori.	10
3.3	Schema delle tipologie di Machine Learning	11
3.4	Esempio di Workflow tramite Machine Learning	12
3.5	Rappresentazione dei dati e dei gruppi ottenuti con la cluster analysis	13
3.6	Esempio di assegnazione dei dati tramite l'algoritmo k-means	15
3.7	Esempio di assegnazione dei dati tramite l'algoritmo k-medoids	16
3.8	Esempio di assegnazione dei dati tramite cluster gerarchico e la sua rappresentazione tramite dendogramma	17
3.9	Schema input-output per una SOM	18
3.10	Rappresentazione dei possibili approcci usando strategie di clustering	19
3.11	Rappresentazione della matrice di confusione per 2 classi	21
3.12	Rappresentazione della matrice di confusione per c classi	22
3.13	Rappresentazione del cambio di dimensionalità tramite PCA	23
3.14	Rappresentazione del cambio di dimensionalità tramite LDA	25
5.1	Rappresentazione del flusso della tesi	30
5.2	Grafico delle 3 classi in base alla discriminazione data dall'approccio di Fisher per il paziente 1	34
5.3	Schema generale di calcolo delle feature statistiche	35

Introduzione

La Malattia di Parkinson è una patologia neurodegenerativa che coinvolge in maniera elettiva la capacità di programmare ed eseguire il movimento, senza risparmiare altri aspetti dell'individuo come la sfera cognitiva e comportamentale. Questi aspetti, unitamente al decorso cronico e progressivo della malattia, determinano una compromissione delle attività di vita quotidiana e delle relazioni interpersonali.

Tra i sintomi della malattia di Parkinson, il Freezing of Gait (FOG) può sicuramente essere considerato uno dei più debilitanti. Il Freezing nella malattia di Parkinson, detto anche congelamento o semplicemente blocco motorio, è un'improvvisa, temporanea e involontaria incapacità di iniziare un movimento. È un disturbo che insorge nel corso dell'evoluzione della malattia di cui costituisce un sintomo indipendente e generalmente resistente al trattamento con levodopa. Tale fenomeno si può verificare in ogni momento e i pazienti che lo sperimentano affermano che: *«è come se i piedi rimanessero, per qualche istante, incollati al suolo con la conseguente impossibilità di eseguire il passo successivo»*. In realtà, il Freezing si può verificare anche durante azioni differenti dal cammino come ad esempio l'alzarsi da una sedia o il raccogliere un oggetto. Alcune persone sono più predisposte di altre a subire episodi di congelamento. Tali episodi, si possono verificare sia quando il soggetto è in astinenza da farmaci dopaminergici, in questo caso si parla di "Freezing off", sia quando il soggetto sta assumendo i farmaci, "Freezing on".

1.1 Contributo della Tesi

In questa lavoro di tesi, si è cercato di sviluppare una metodologia non supervisionata, ossia che non abbia bisogno dell'intervento dell'essere umano, per identificare gli episodi di FoG e preFoG, ossia una fase transitoria tra un'attività normale ed un episodio di Freezing. Questo è un problema molto debilitante per i malati di Parkinson che, allo studio attuale, non ha una soluzione efficace dal punto di vista medico e di cui non si conoscono a fondo

le cause.

Per sviluppare tale metodologia, è stato utilizzato un dataset in cui sono presenti 10 pazienti, tutti affetti dal morbo di Parkinson, ed 8 di loro presentano episodi di FoG. I contesti di FoG sono stati identificati dal dottore tramite l'utilizzo di videoregistrazioni ed i dati sono stati raccolti attraverso 3 accelerometri, posizionati sulla caviglia, sul ginocchio e sulla zona lombare del paziente.

Il primo passo è stato quello di verificare che esiste una distinzione tra le 3 classi del nostro dataset, ossia attività normale, preFoG e FoG tramite l'utilizzo di un Linear Discriminant Analysis (LDA). Una volta verificato, è stato condotto uno studio sugli intervalli con cui dividere i dati del dataset, cioè quanto deve essere grande la finestra temporale su cui calcolo le feature dai dati grezzi e quanta sovrapposizione è opportuno avere. Infine, sono stati usati algoritmi di clustering per etichettare i dati e, tali etichette, sono state confrontate con quelle del dataset originale per verificare quanto sono affidabili tali algoritmi di clustering.

Il resto della tesi è organizzata nel modo seguente: nel **Capitolo 2** si presentano i risultati dei principali lavori svolti sul FoG; nel **Capitolo 3** si presentano a livello teorico le problematiche del FoG e gli strumenti su cui si basa il lavoro svolto; nel **Capitolo 4** si presenta qual'è l'obiettivo finale del lavoro che viene svolto; nel **Capitolo 5** si presenta il lavoro svolto; nel **Capitolo 6** si riassume i risultati ottenuti e si presentano possibili miglioramenti al lavoro svolto.

Letteratura

Il problema del FOG è stato analizzato tramite una grande varietà di sistemi e sensori. Alcuni di questi, però, non sono utilizzabili durante la vita quotidiana dei pazienti poiché posso essere disponibili solo in ambienti di laboratorio. Esempi di questi sistemi sono le piattaforme di pressione[19], le quali sono non portatili, l'elettromiografia (EMG)[6], l'elettroencefalogramma (EEG)[22] o la conduttanza della pelle[26], il quale comporta il pizzamento di elettrodi sulla pelle in aggiunta ad sistema di rilevamento per raccogliere i dati. Altri sistemi invasivi sono i goniometri a ginocchio[1] o sistemi che fanno uso di camere e video, i quali hanno una bassa tolleranza del paziente in un'ambiente che non sia di laboratorio[1, 2, 5]. Quindi, dato che il monitoraggio del PD dovrebbe essere deambulatorio e durare diverse ore al fine di ricavare utili informazioni cliniche[9, 4], la maggior parte dei lavori si è basata su sistemi non invasivi come i dispositivi indossabili basati su circuiti microelettromeccanici (MEMS).

Nel 2003, Han et al. hanno usato MEMS basati su sistemi inerziali, come gli accelerometri, per esplorare le caratteristiche collegate agli episodi di FoG. Hanno trovato che la frequenza di risposta nei pazienti che indossavano gli accelerometri nella caviglia era intorno ai 6-8 Hz[18]. Nel 2008, Moore et al. hanno proposto una metodologia per identificare FoG con un'accelerometro posizionato nella caviglia nella quale hanno descritto il Freezing Index (FI), ossia il quoziente del rapporto della densità spettrale di potenza (PSD) tra 3 ed 8 Hz, chiamata Freezing Band (FB), con la PSD tra 0.5 e 3 Hz, denominata Walking Band (WB)[21]. Quando il FI supera una certa soglia (Freezing Threshold (FTH)), si considera che si sia verificato un episodio di FoG. A causa della presenza dei falsi positivi (FP) quando il paziente è a riposo, Bachlin et al. hanno introdotto il concetto di Power Index (PI), definito come la somma della WB e FB, il quale viene comparato con la Power Threshold (PTH) al fine di stabilire se c'era una quantità rilevante di movimento nel momento in cui il FI era alto, ossia oltre la soglia[21]. PI indica la quantità di movimento, perciò situazioni nelle quali il paziente non si stesse muovendo volontariamente sono state eliminate. In quest'ultima versione dell'algoritmo, quindi, un episodio di FoG è occorso se $FI > FTH$ e $PI > PTH$. Questo

metodo è il più avanzato nella detenzione di FoG dato il suo scarso costo computazionale e le sue buone performance[13].

L'algoritmo MBFA è stato ampiamente utilizzato nell'analisi del FoG, anche se di solito in condizioni di laboratorio e molto spesso con pochi pazienti. Jovanov et al. hanno implementato un algoritmo real time, anche se un solo volontario è stato usato per testare l'algoritmo. Inoltre, nessun risultato su sensibilità e specificità è stato riportato[13]. Zabaleta et al. hanno analizzato il FoG per mezzo di accelerometri a tre assi e giroscopi a due assi in differenti locazioni degli arti inferiori. La caratteristica principale ad essere stata analizzata è il FI in congiunzione con i cambiamenti della densità spettrale di potenza. Sono stati capaci di identificare correttamente l'82.7% degli episodi di FoG con i sensori inerziali posizionati su entrambe le caviglie, anche se in soli 2 pazienti[15].

Più recentemente, Niazmand et al. (2011) hanno presentato il Mimed-Pants[20], pantaloni da jogging lavabili con 5 accelerometri integrati. Hanno usato MBFA per identificare FoG, ottenendo un 88.3% in sensibilità e 85.3% in specificità con 6 pazienti in brevi e controllati test focalizzati nell'indurre FoG senza tenere conto dei FP. Nel 2012, Zhao et al.[30] hanno sviluppato un algoritmo embedded basato sull'approccio MBFA all'interno del sistema Mimed-Pants ottenendo un 81% in sensibilità con 8 pazienti usando dei test simili ai precedenti. Più recentemente, Mazilu et al. hanno proposto un nuovo algoritmo online usando 3 accelerometri ed comparando diversi classificatori di machine learning che sfruttavano le caratteristiche del MBFA, aggiungendone di nuovo, in 10 pazienti[27]. I risultati ottenuti sono stati migliori del 95% per specificità e sensibilità con differenti classificatori. Questi test, però, sono stati condotti in situazioni di controllo ed, inoltre, la metodologia di validazione sovrastimava le prestazioni delle misure poiché i classificatori erano allenati, iterativamente, con tutte le finestre del segnale disponibili da un paziente escludendone una, la quale veniva usata per ottenere le prestazioni citate. Inoltre, le sequenze di allenamento e di test erano molto simili, il che è molto diverso da normali situazioni. Quindi, ci si aspetta che le riportate specificità e sensibilità calino drasticamente in situazioni non controllate.

Nel 2013, Moore et al. hanno pubblicato il più recente lavoro focalizzato sul MBA. In questo, hanno confrontato differenti configurazioni applicando lo stesso algoritmo in 25 pazienti, dei quali 20 hanno avuto episodi di FoG. Diverse finestre di segnale, posizionamento dei sensori e valori per PTH e FTH sono stati valutati al fine di trovare le condizioni ottimali. I risultati migliori sono stati ottenuti con le finestre di segnale più lunghe, anche se con queste Moore et al. hanno riportato una rilevante perdita di sensibilità negli episodi brevi che, paradossalmente, sono quelli più frequenti nei pazienti affetti da PD[28]. In un test più complesso eseguito precedentemente[29] usando fino a 7 sensori ed un protocollo di test più lungo, sono stati ottenuti una sensibilità e specificità sopra al 70%, anche se, in certe configurazioni (finestra di segnale pari a 7.5s e il sistema installato nella zona lombare), sia

per la sensibilità che la specificità hanno raggiunto valori oltre l'80%. In un approccio differente, Tripoli et al. hanno testato diverse configurazioni e locazioni dei sensori al fine di trovare la migliore configurazione[14]. Il lavoro è stato svolto con 5 pazienti ed in condizioni controllate, usando uno specifico protocollo progettato per stimolare il FoG e senza test di FP. In tale lavoro, hanno integrato 2 giroscopi oltre a 6 accelerometri posizionati in posizioni differenti del corpo. Con tutti i sensori indossati, è stata ottenuta un'accuratezza del 96.11%, una specificità del 98.74% e, eseguendo i test su tutti i pazienti tranne uno, una sensibilità dell'81.94%. D'altra parte, con una IMU singola nella zona lombare hanno riportato una sensibilità del 75% ed una specificità del 95%, anche se l'algoritmo non è stato confrontato con nessun altro metodo usato sotto le stesse condizioni.

Mazilu et al.[10] hanno investigato un approccio di apprendimento non supervisionato per costruire un input ottimale per un classificatore ad albero di decisione con il dataset del progetto DAPHNET (10 pazienti PD). Il loro approccio è stato comparato ad un analogo basato su MBFA nel quale il FI e l'energia della banda spettrale tra 0.5 Hz e 8 Hz sono state valutate. L'allenamento ed i test erano dipendenti dall'utilizzatore e sotto condizioni controllate. I risultati superano l'approccio MBFA similare dell'8.1% in termini di punteggio dell'F1. Un altro approccio è stato presentato da Rodriguez et al., i quali hanno proposto un metodo per contestualizzare gli episodi di FoG tramite un algoritmo di riconoscimento dell'attività, il quale rifiutava i FP quando il paziente era seduto o eseguiva attività quali disegnare o digitare in un laptop. La specificità è stata aumentata in media del 5%, arrivando anche ad un 11.9% in certi casi[10]. Il metodo che aggiungeva la contestualizzazione, però, non ha contribuito a migliorare la sensibilità. Altri studi hanno la variabilità della camminata tra un episodio di FoG e condizioni normal. Anche se i risultati sono interessanti, hanno fallito nell'includere i falsi positivi ed un'affidabile classificazione non è stata eseguita[8, 3]. Un paper recente di Zach et al. presenta una nuova metodologia per suscitare FoG in condizioni di laboratorio controllate, le quali sono state valutate con l'algoritmo MBFA ottenendo una sensibilità del 75% ed una specificità del 76%[16].

Infine, Alrichs et al., all'interno del progetto REMPARK[25], usano una Support Vector Machines (SVM) per rilevare episodi di FoG in 8 pazienti con PD in ambiente casalingo. Il metodo include test in differenti test motori usando un singolo accelerometro nella zona lombare, raggiungendo un'accuratezza del 90%. La specificità, però, è stata calcolata solo non pazienti non FoG, il che può portare a predizioni non affidabili in quanto il modello non è stato testato con pazienti PD con FoG, i quali hanno movimenti molto diversi dai pazienti che non soffrono di FoG. Inoltre, la valutazione è stata eseguita su finestre di un minuto, tempo che è considerato troppo lungo per un'implementazione online[7]. Sempre all'interno del progetto REMPARK, Rodriguez et al. hanno presentato un lavoro che utilizza un algoritmo per

rilevare FoG tramite un approccio di machine learning basato su SVM ed un singolo accelerometro a 3 assi indossato nella zona lombare[11]. Il metodo è stato valutato su 21 pazienti affetti da PD in ambienti casalinghi sotto due condizioni: un modello generico testato su tutti i pazienti tranne uno ed un secondo modello personalizzato sull'utente che usa parte del dataset del paziente stesso. I risultati mostrano un significativo vantaggio del modello personalizzato rispetto a quello generico, portando ad un miglioramento in media, sia della sensibilità che della specificità, del 7.2%. Inoltre, l'approccio adottato è stato comparato con i metodi più utilizzati per la detenzione del FoG basati sull'algoritmo MBFA. I risultati del metodo generico mostrano un miglioramento in media dell'11.2% rispetto a metodi MBFA generici, mentre quello personalizzato porta ad un miglioramento del 10% rispetto ad altri metodi specifici sul paziente.

Conoscenze Preliminari

3.1 Freezing Of Gait

È stato dimostrato che il fenomeno del Freezing nella malattia di Parkinson è spesso collegato alle frequenti cadute a cui i soggetti malati vanno incontro. Le cadute nel Parkinson si verificano più spesso quando il soggetto si gira o cambia direzione e sono frequentemente legate a diversi episodi di Freezing. Non tutti i malati di Parkinson subiscono il fenomeno del Freezing, ma si pensa che coloro che lo provano abbiano una più alta probabilità di cadere a terra. L'imprevedibilità del Freezing, accompagnata dallo sforzo inutile a cui il soggetto si sottopone per cercare di muoversi in avanti, possono causare perdita di equilibrio e quindi cadute.

Nel tentativo di superare questo stato di forzata immobilità, i pazienti, talora con un aiuto esterno, cercano di mettere in atto adeguate strategie che si avvalgono di stimoli sensoriali di diversa natura (tattili, visivi oppure uditivi e verbali). Alcune tecniche di tipo motorio o sensoriale possono aiutare i pazienti a convivere con il problema del Freezing. Ad esempio, un paziente incapace di iniziare il primo passo potrebbe riuscire a superare il blocco motorio adottando una delle seguenti strategie:

- fare un passo in direzione di un bersaglio;
- fare un passo per superare un bastone posto sul pavimento;
- fare il primo passo marciando come un soldato.

L'idea che sta alla base di tali stratagemmi è mettere in atto un programma motorio volontario che sostituisca il programma motorio automatico malfunzionante nei malati di malattia di Parkinson. Episodi frequenti di Freezing possono avere pericolose conseguenze sia sullo stato fisico sia su quello psicologico del malato e compromettono ampiamente la qualità della vita di chi ne soffre privandolo spesso dalla propria indipendenza.

3.1.1 Le tipologie di Fog

Il FoG è un episodio transitorio che usualmente dura pochi secondi e di cui ancora non si conosce la patofisiologia, ossia la causa scatenante, ma è stato dimostrato che esistono più sottotipi di Freezing, che si differenziano per l'evento scatenante il fenomeno:

- necessità del paziente di girare su sè stesso per cambiare direzione (esitazione legata alla svolta);
- attraversamento di spazi stretti, come una porta od un corridoio;
- inizio del movimento di camminata;
- regolazione dei passi in prossimità della destinazione (come ad esempio una sedia su cui sedersi);
- stress, come lo squillo di un telefono o campanello o quando la porta dell'ascensore si apre.

Come la malattia progredisce, però, il FoG può apparire spontaneamente anche in uno spazio aperto, evidenziando così l'aspetto imprevedibile di questo fenomeno. Inoltre, fonti di distrazione, che possono distogliere l'attenzione del soggetto dal cammino, o il compimento contemporaneo di più azioni (dual-tasking) possono aumentare la probabilità che si verifichi un episodio di Freezing.

Dallo studio di Schaafsma et al.[12] emerge che gli episodi di Freezing possono anche essere suddivisi in ulteriori tre sottotipi andando ad osservare i movimenti delle gambe dei pazienti e applicando la classificazione di Thompson e Marsden (1995):

1. FoG associato a passi molto piccoli e strascicati con il minimo movimento in avanti (trascinamento con piccoli passi);
2. FoG caratterizzato da tremore alle gambe, ma nessun movimento in avanti efficace (tremore da fermo);
3. FoG caratterizzato da acinesia completa, vale a dire, nessun movimento osservabile delle gambe.

La necessità di dividere il FoG in sottogruppi dipende dal fatto che questi ultimi potrebbero avere origine differente e quindi essere provocati da cause separate.

3.1.2 Influenza del FoG nella camminata

Il Freezing of Gait influenza il pattern del cammino sia all'inizio della deambulazione sia a regime incrementando o diminuendo in modo evidente i valori

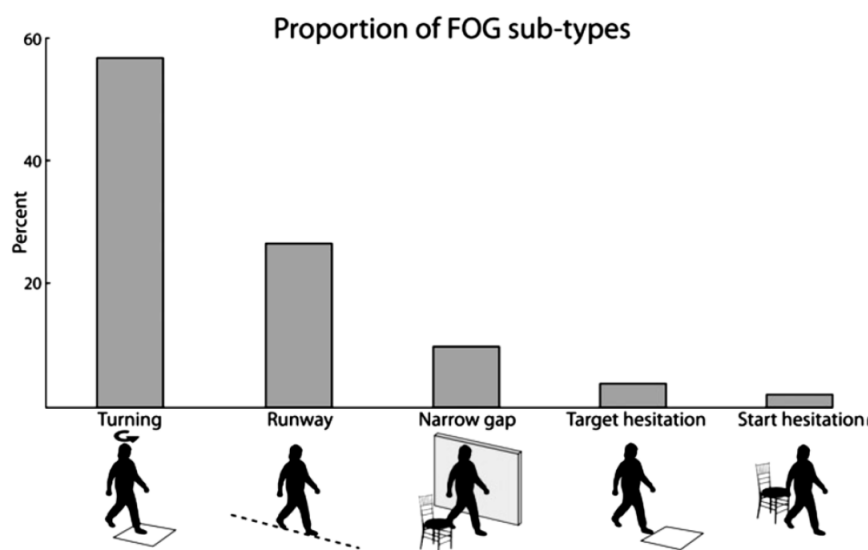


Figura 3.1: Diversi tipi di Freezing esistenti e loro percentuale di incidenza su un certo gruppo di malati di Parkinson. [Fonte: J.M. Shine et al., 2012]

dei parametri sopra riportati. Ad esempio, nei parkinsoniani che manifestano frequenti fenomeni di Freezing, la variabilità della durata del passo risulta maggiore e la lunghezza del passo minore rispetto alle situazioni in cui il Freezing è assente. Inoltre, la velocità e la lunghezza dei primi tre passi sono significativamente inferiori nei pazienti con malattia di Parkinson e con Freezing rispetto ai soggetti sani. Anche se il Freezing è tipicamente considerato un problema motorio, il fatto che spesso compaia quando il paziente si trova in spazi ristretti, suggerisce che la percezione dello spazio contribuisce in larga misura a scatenare il sintomo[24]. Inoltre, i pazienti con Freezing hanno velocità media del cammino minore rispetto a pazienti sani e subiscono una riduzione ulteriore della velocità nel momento in cui si trovano a dover attraversare una porta o uno spazio stretto. Depressione e ansia possono comportare un carico cronico sulla salute mentale, e la depressione è associata con i cambiamenti di andatura, tra cui una maggiore variabilità passo-a-passo. Il Dual-Tasking, l'ansia e la depressione possono incrementare la variabilità del passo, la di-sincronizzazione di gamba destra e sinistra e l'asimmetria nei pazienti con Parkinson riducendo così la soglia per il FoG.

3.2 Machine Learning

Il Machine Learning (ML) è un settore dell'Intelligenza Artificiale che si occupa di studiare le modalità in cui un computer può imparare (o migliorare le

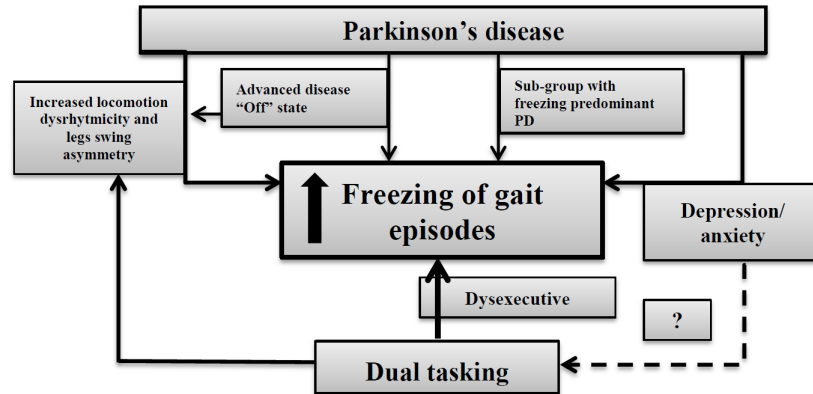


Figura 3.2: Quadro concettuale relativo al Freezing of Gait (FoG) riguardante aspetti mentali e motori.

sue prestazioni) dai dati[17]. Un'ampia area di ricerca tratta l'apprendimento automatico di sequenze complesse di dati e compiere decisioni intelligenti su di essi. Per esempio, un tipico problema di ML è quello di programmare un computer in modo tale che possa automaticamente riconoscere codici postali scritti a mano nelle lettere, dopo aver fatto allenare il computer stesso su dei dati. Gli algoritmi di apprendimento automatico sono tradizionalmente divisi in tre principali tipologie:

- **Apprendimento supervisionato:** quando l'utente fornisce esempi (e controesempi) di quello che si deve apprendere. E' il problema più studiato nel machine learning. Esso si pone l'obiettivo di prevedere, dato un elemento di cui si conoscono un insieme di parametri (features), il valore di un diverso parametro di output relativo all'elemento stesso;
- **Apprendimento non supervisionato:** parte da osservazioni non preclassificate;
- **Apprendimento con rinforzo:** tecnica di programmazione che si basa sul presupposto che l'algoritmo possa ricevere stimoli dall'esterno a seconda delle scelte fatte.

Il problema del ML è definito a partire da un universo di elementi: ciascun elemento x è descritto dai valori assunti da un insieme di features considerate come input del problema. Ad ogni x è associato un valore y di output (o target). A partire dalla conoscenza di un insieme T di elementi (denominato training set) in cui ogni elemento è descritto da una coppia (x_i, y_i) , con x_i = vettore dei valori delle d features $x_{i1}, x_{i2}, \dots, x_{id}$ e y_i = valore di output, si vuole derivare un modello delle relazioni sconosciute tra features e valori di output, che, dato un nuovo elemento x , consenta di predire il corrispondente

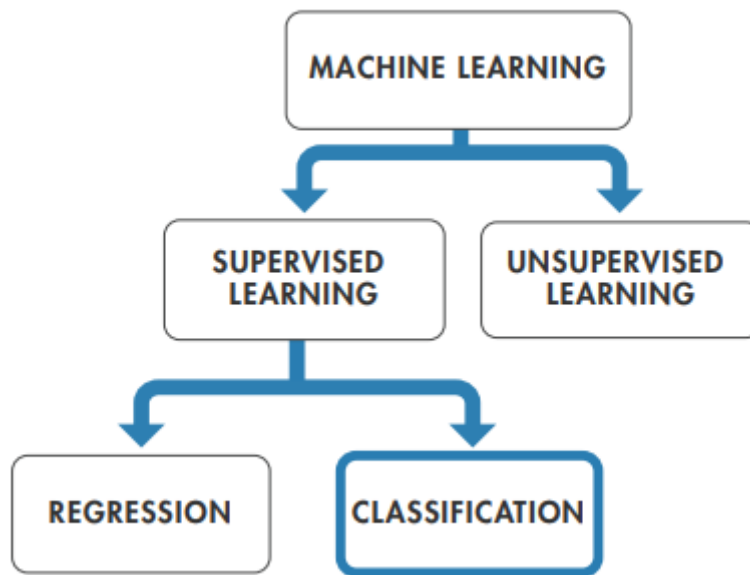


Figura 3.3: Schema delle tipologie di Machine Learning

valore di output y . Lo scopo dell'apprendimento supervisionato è di costruire un **modello di predizioni** basato su evidenze in presenza di incertezze. Un algoritmo di apprendimento supervisionato prende un insieme conosciuto di dati di input e di risposte ai dati (output) ed allena un modello al fine di generare predizioni ragionevoli a nuovi dati di input. Esistono due tecniche per questo approccio:

- **Classificazione:** tecnica per predire risposte discrete, classificando i dati di input in categorie;
- **Regression:** tecnica per predire risposte continue.

L'apprendimento non supervisionato, invece, trova pattern nascosti o strutture intrinseche nei dati. Tale tecnica è usata per identificare inferenze da un dataset consistente di dati di input senza classi già definite. Il clustering è l'approccio più diffuso di tale tecnica ed è utilizzato per trovare gruppi nei dati.

L'approccio che si usa con il ML è differente da problema a problema, per cui non sempre si ha a priori un procedimento fissato da seguire, ma piuttosto si procede a tentativi ed errori, molte volte provando diverse idee ed approcci che appartengono a tale contesto. Per questo è importante definire uno schema di lavoro generale ed evidenziare alcuni punti di decisione chiave lungo il percorso:

1. Accesso e caricamento dei dati: questi possono essere di tutte le forme e i tipi, incompleti o mescolati;

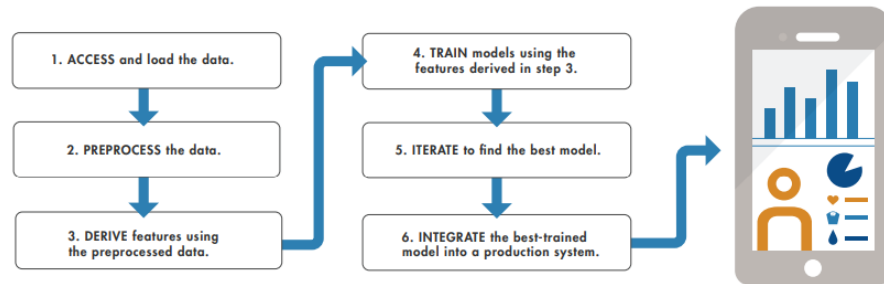


Figura 3.4: Esempio di Workflow tramite Machine Learning

2. Pre-processamento dei dati: applicazione di filtri o ri-campionamento;
3. Derivazione di feature: trovare caratteristiche peculiari a partire dai dati;
4. Allenamento del modello di ML: può essere un procedimento lungo, poichè dipende da molti parametri;
5. Integrazione del modello in un sistema di produzione.

La scelta della modalità supervised o unsupervised si basa sui vantaggi e svantaggi di entrambe: la modalità supervised riesce a predire la giusta classe per le istanze appartenenti al test set ma richiede una consistente quantità di istanze annotate e questo può rappresentare un processo costoso se effettuato manualmente. La modalità unsupervised tipica del clustering, invece, ha il vantaggio di non richiedere un training già annotato (situazione particolarmente frequente quando si ricorre al ML) ma difficilmente etichetta correttamente il cluster e ottiene una precisione più scarsa rispetto al primo metodo nell'associare le istanze ai cluster corretti.

3.2.1 Cluster Analysis

Con il termine Cluster Analysis, o analisi dei gruppi, si intendono le procedure che permettono di individuare, all'interno di un insieme di oggetti di qualsiasi natura, alcuni sottinsiemi, **clusters** appunto, mutuamente esclusivi e tendenzialmente omogenei al loro interno. Le tecniche di Cluster Analysis creano i gruppi in modo tale che ogni osservazione sia molto simile a tutte le altre che appartengono allo stesso gruppo, in funzione di alcuni criteri prestabiliti. Alla fine del procedimento, i cluster finali dovrebbero esibire un'alta omogeneità interna (intra-cluster) ed un'alta eterogeneità esterna (inter-cluster), quindi gli oggetti all'interno dei cluster saranno vicini tra loro, mentre gli oggetti che appartengono a differenti cluster saranno più lontani tra loro.

La cluster analysis rientra tra le tecniche di tipo esplorativo e pertanto non è

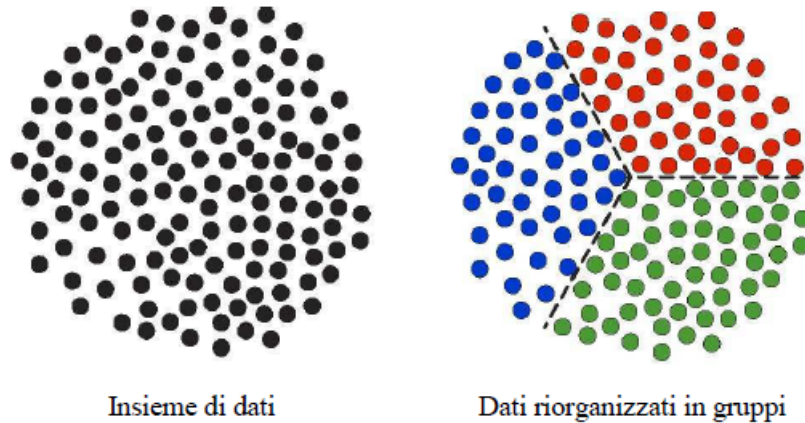


Figura 3.5: Rappresentazione dei dati e dei gruppi ottenuti con la cluster analysis

necessaria alcuna assunzione a priori, anche se impone una serie di decisioni durante l'analisi:

- Scelta delle variabili;
- Criteri di similarità o distanza;
- Tecniche di aggregazione;
- Numero dei gruppi da ottenere;
- Valutazione della qualità della soluzione;
- Scelta fra le diverse possibili soluzioni.

Per classificare e raggruppare gli elementi in gruppi omogenei, è necessario introdurre una nozione di prossimità o similarità. Un indice di prossimità tra due generici elementi x_i e x_j è definito come una funzione dei rispettivi vettori riga nella matrice dei dati: $IP_{ij} = f(x'_i, x'_j)$, $i, j = 1, 2, \dots, n$. Due individui sono vicini quando la loro dissimilarità o distanza è piccola o, equivalentemente, quando la loro similarità è grande. Le principali misure di similarità sono illustrate in tabella 3.1.

I principali algoritmi di clustering sono descritti nelle prossime sottosezioni.

K-Means

Il K-Means è un algoritmo di clustering partizionale che permette di suddividere un insieme di oggetti in K gruppi sulla base dei loro attributi. E' una

Tipo di dato	Misura	Formula
Binario	Coefficiente di Sokal	$S_{ij} = (a + d)/(a + b + c + d)$
Binario	Coefficiente di Jaccard	$S_{ij} = a/(a + b + c)$
Categorici non binari	Media delle variabili	$S_{ij} = (1/p) \sum_{k=1}^p s_{ijk}$
Dati Continui	Distanza Euclidea	$d_{ij} = [\sum_{k=1}^p (x_{ik} - x_{jk})^2]^{1/2}$
Dati Continui	Distanza City Block	$d_{ij} = \sum_{k=1}^p x_{ik} - x_{jk} $

Tabella 3.1: Misure di Similarità

variante dell'algoritmo di aspettativa-massimizzazione(EM) il cui obiettivo è determinare i K gruppi di dati generati da distribuzioni gaussiane. Si assume che gli attributi degli oggetti possano essere rappresentati come vettori e che quindi formino uno spazio vettoriale. L'obiettivo che l'algoritmo di prepone è di minimizzare la varianza intra-cluster, dove ogni cluster viene indentificato mediante un centroide o punto medio. L'algoritmo segue una procedura iterativa:

1. Crea K partizioni (dove K è il numero di cluster che si vogliono ottenere) ed assegna ad ogni partizione dei punti di ingresso (scelti casualmente o euristicamente);
2. Costruisce una nuova partizione associando ogni punto d'ingresso al cluster il cui centroide è più vicino ad esso;
3. Vengono ricalcolati i centroidi per i nuovi cluster e si itera al punto 2 finché l'algoritmo non converge.

L'algoritmo converge molto velocemente, infatti è stato osservato che il numero di iterazioni è minore del numero di punti da osservare. In termini di qualità delle soluzioni, però, l'algoritmo non garantisce il raggiungimento dell'ottimo globale poiché la soluzione dipende largamente dal set di cluster iniziale. Un rimedio a questo aspetto è di applicare l'algoritmo più volte e, fra tutte le soluzioni prodotte, scegliere quella più soddisfacente. Un altro svantaggio è che l'algoritmo richiede di scegliere a priori il numero di cluster da trovare, per cui se i dati non sono naturalmente partizionati si ottengono risultati strani. Inoltre, l'algoritmo funziona bene solo quando sono individuabili cluster sferici nei dati.

K-Medoids

K-medoids è un algoritmo di clustering partizionale correlato all'algoritmo K-means. Prevede in input un insieme di n oggetti e un numero k che determina quanti cluster si vogliono in output. Entrambi gli algoritmi sono partizionali (suddividendo il dataset in gruppi) ed entrambi cercano di minimizzare l'errore quadratico medio, la distanza tra punti di un cluster e il

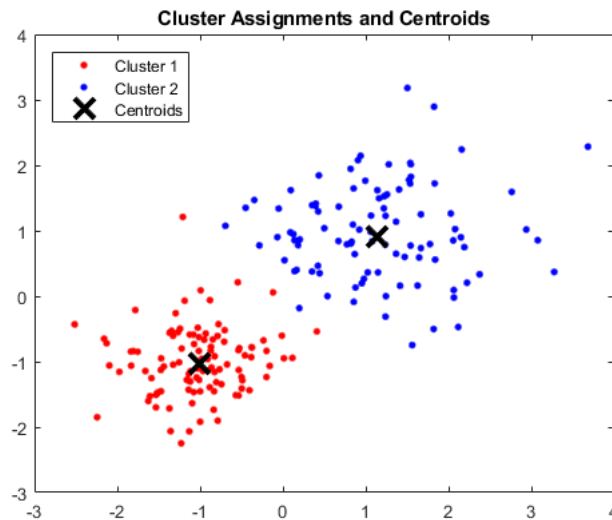


Figura 3.6: Esempio di assegnazione dei dati tramite l'algoritmo k-means

punto designato per esserne il centro. In K-means il punto è "artificiale" — è la pura media di tutti i punti nel cluster. Nel K-medoids è usato il punto collocato più centralmente, in questo modo il centro è uno dei datapoint attuali. K-medoids è più robusto al rumore e agli outlier rispetto al k-means. Un medoid può essere definito come un oggetto di un cluster la cui dissimilarità media rispetto a tutti gli oggetti nel cluster è minima, in questo modo esso sarà il punto più centrale di un dato dataset. L'algoritmo di clustering è il seguente:

1. Si scelgono arbitrariamente k oggetti (dove K è il numero di cluster che si vogliono ottenere) come punti medoid da un insieme di n data point ($n > k$);
2. In seguito alla selezione di k punti medoid, si associa ogni oggetto nel dato dataset al più simile medoid. La similarità è definita usando misure di distanza;
3. Si seleziona in modo casuale un oggetto non medoid O' ;
4. Si calcola il costo totale S per lo scambio del medoid iniziale nell'oggetto O' ;
5. Se $S < 0$, allora si scambia il medoid iniziale con il nuovo (se $S < 0$ allora ci sarà un nuovo insieme di medoid);
6. Si ripetono i passi dal 2 al 5 sino a quando si hanno cambiamenti nel medoid.

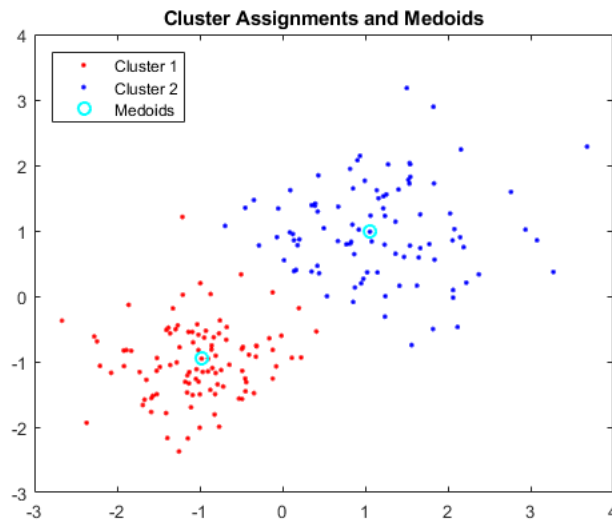


Figura 3.7: Esempio di assegnazione dei dati tramite l'algoritmo k-medoids

Hierarchical

Il clustering gerarchico è un approccio di clustering che mira a costruire una gerarchia di cluster. Le strategie per il clustering gerarchico sono tipicamente di due tipi:

- Agglomerativo: si tratta di un approccio "bottom up" (dal basso verso l'alto) in cui si parte dall'inserimento di ciascun elemento in un cluster differente e si procede quindi all'accorpamento graduale di cluster a due a due;
- Divisivo: si tratta di un approccio "top down" (dall'alto verso il basso) in cui tutti gli elementi si trovano inizialmente in un singolo cluster che viene via via suddiviso ricorsivamente in sotto-cluster.

Il risultato di un clustering gerarchico è rappresentato in un dendrogramma. Questo è uno strumento grafico per la visualizzazione del coefficiente di similarità quantificato dalle varie macchine e dai vari cluster nel processo di "raggruppamento". Il dendrogramma viene utilizzato per fornire una rappresentazione grafica del processo di raggruppamento delle istanze che esprime nell'asse delle ascisse la distanza logica dei clusters secondo la metrica definita, nell'asse delle ordinate il livello gerarchico di aggregazione (valori interi positivi). La scelta del livello gerarchico (del valore dell'asse Y) definisce la partizione rappresentativa del processo di aggregazione.

Per decidere quali cluster devono essere combinati (approccio agglomerativo) o quale cluster deve essere suddiviso (approccio divisivo) è necessario definire una misura di dissimilarità tra cluster. Nella maggior parte dei metodi di

clustering gerarchico si fa uso di metriche specifiche che quantificano la distanza tra coppie di elementi e di un criterio di collegamento che specifica la dissimilarità di due insiemi di elementi (cluster) come funzione della distanza a coppie tra elementi nei due insiemi. Le metriche di distanza più comuni sono: euclidea, Manhattan, uniforme, Mahalanobis, Hamming. Le metriche più diffuse sono: complete linkage (massimo distanza tra due punti), single linkage (massimo distanza tra due punti) o average linkage (massimo distanza tra due punti).

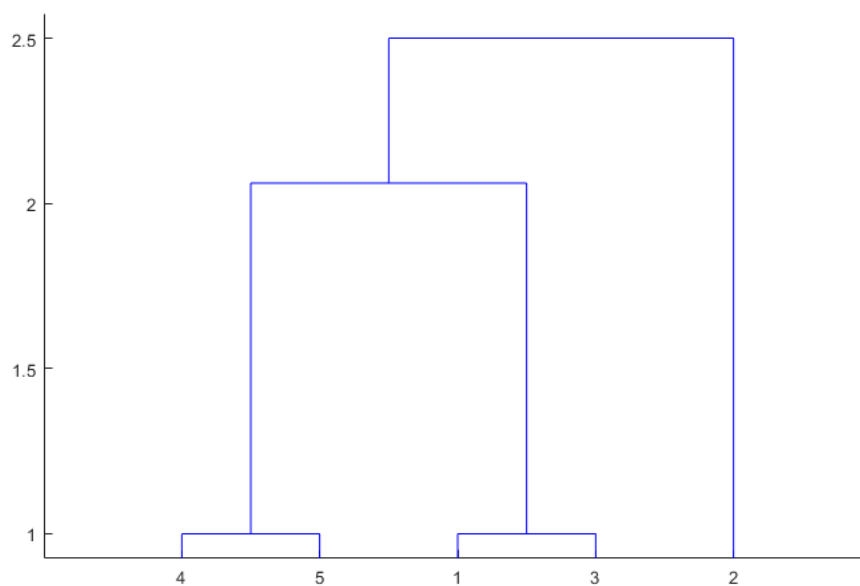


Figura 3.8: Esempio di assegnazione dei dati tramite cluster gerarchico e la sua rappresentazione tramite dendrogramma

Neural Networks

Le self-organizing map (SOM) sono una fattispecie di organizzazione di processi di informazione in rete analoghi alle reti neurali artificiali. È addestrata usando l'apprendimento non supervisionato per produrre una rappresentazione dei campioni di training in uno spazio a bassa dimensione preservando le proprietà topologiche dello spazio degli ingressi. Questa proprietà rende le SOM particolarmente utili per la visualizzazione di dati di dimensione elevata. Le self-organizing map sono reti neurali a connessioni laterali dove i neuroni di uscita sono organizzati in griglie di bassa dimensione (generalmente 2D o 3D). Ogni ingresso è connesso a tutti i neuroni di uscita. A ogni neurone è associato un vettore dei pesi della stessa dimensione dei vettori

d'ingresso. La dimensione del vettore d'ingresso è generalmente molto più alta della dimensione della griglia di uscita. L'obiettivo dell'apprendimento nelle self-organizing map è di specializzare parti differenti del reticolo SOM a rispondere similmente a particolari pattern d'ingresso.

L'algoritmo di clustering è il seguente:

1. Assegna ai vettori dei pesi valori casuali;
2. Prendi un vettore d'ingresso;
3. Attraversa ogni nodo della mappa ed usa la distanza euclidea per trovare somiglianze fra il vettore d'ingresso ed il vettore dei pesi di ogni singolo nodo della mappa, individuando il nodo a distanza minore (questo nodo verrà chiamato Best Matching Unit o BMU);
4. Aggiorna i nodi del vicinato di BMU "tirandoli" più vicino al vettore d'ingresso.

Ci sono due modi per interpretare una SOM:

- Dato che nella fase di addestramento i pesi di tutto il vicinato sono spostati nella stessa direzione, elementi simili tendono ad eccitare neuroni adiacenti. Perciò le SOM formano una mappa semantica dove campioni simili vengono mappati vicini e dissimili distanti;
- Un altro modo di considerare i pesi neuronali è di pensarli come punti distribuiti nello spazio degli ingressi. Questi formano un'approssimazione della distribuzione dei campioni d'ingresso. Più neuroni puntano a regioni con un'elevata concentrazione di campioni di addestramento, e meno in zone dove i campioni sono scarsi.

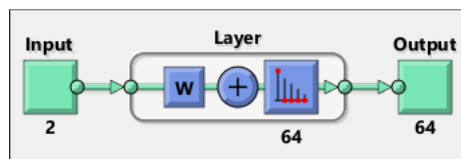


Figura 3.9: Schema input-output per una SOM

Fuzzy C-Means

Il Fuzzy C-means (anche riferito come soft clustering) è un algoritmo di clustering nel quale ogni punto può appartenere a più di un cluster. L'algoritmo è molto simile al k-means, ma in questo contesto viene assegnato ad ogni punto un grado di appartenenza. Questo indica il grado con cui un determinato punto appartiene ad un certo cluster. L'algoritmo di clustering è il seguente:

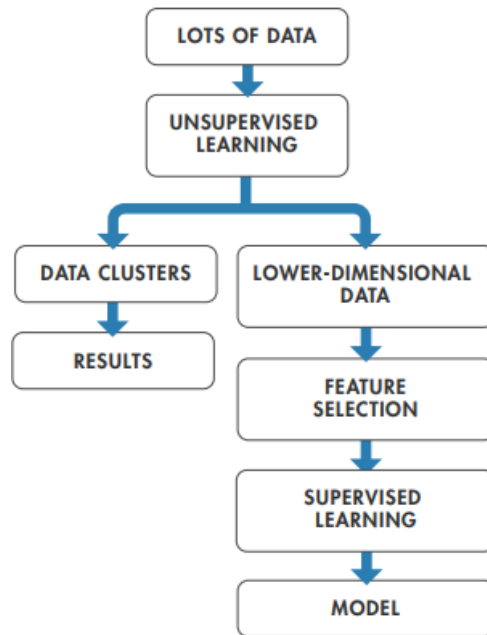


Figura 3.10: Rappresentazione dei possibili approcci usando strategie di clustering

1. Scegliere un numero K di cluster;
2. Assegna randomicamente dei coefficienti di appartenenza ad un cluster per ogni punto;
3. Calcola il centroide di ogni cluster;
4. Per ogni punto, calcola il suo grado di appartenenza ai clusters;
5. Itera fino ad avere convergenza.

3.2.2 Classificazione

I principali approcci di classificazione sono due. In un modello parametrico, il modello stesso è preventivamente caratterizzato da un vettore X di parametri: si suppone quindi che esista una relazione tra features e input e che tale relazione sia rappresentabile all'interno di una famiglia di relazioni parametriche rispetto a X che costituisce un modello; in altre parole, un'assegnazione di valori al vettore X definisce una specifica relazione della famiglia. Gli elementi nel training set sono utilizzati proprio per derivare tale assegnazione di valori ai parametri (o una distribuzione di probabilità

per tali valori), dopo di che non sono più utilizzati. In un modello non parametrico, invece, il numero di parametri cresce con la dimensione del training set: sostanzialmente, ogni singola previsione, in questo caso, richiede l'utilizzo dell'intero training set. Un esempio di approccio non parametrico sono i classificatori di tipo nearest neighbor, in cui si determina l'elemento x_i del training set più vicino a x e si impone il valore della nuova previsione y associata a x uguale al valore y_i dell'elemento x_i .

I principali algoritmi di classificazione sono:

- **Decisione Tree:** si tratta di un classificatore con struttura ad albero, in cui ogni nodo può essere o foglia o nodo interno: se foglia, indica il valore della classe assegnata all'istanza; se nodo interno, specifica il test effettuato su un attributo. Per ciascun valore assunto da un attributo in un test, l'algoritmo crea un ramo ed il relativo sottoalbero. Il focus principale dell'algoritmo di crescita del decision tree è come scegliere gli attributi da testare in ciascun nodo interno dell'albero. L'obiettivo è selezionare gli attributi più utili per classificare le istanze di training attraverso una strategia top down, che consiste in una ricerca greedy degli attributi senza tornare a riconsiderare le precedenti scelte. Il criterio di split (suddivisione) con cui crea un nuovo nodo si basa sul massimo guadagno di informazione (info gain). In pratica sceglie l'attributo che riesce a dividere "meglio" le istanze appartenenti a classi diverse (detto anche criterio di massima riduzione di incertezza). Quando tutti gli elementi in un nodo hanno la medesima classe, l'algoritmo non procede con ulteriore split (criterio di stopping).
- **K-Nearest-Neighbors:** esso memorizza le istanze del training, poi, basandosi su un criterio di vicinanza, mette in relazione l'istanza da classificare con alcune istanze del training set presenti nello spazio delle feature. In pratica, l'istanza è classificata "a maggioranza" in base alla classe più comune tra le k istanze più vicine del training. Tutto il lavoro è fatto dal classificatore in runtime. Data una nuova istanza x da classificare, il classificatore cerca i k esempi del training che sono più "simili" a x e guarda le loro etichette. Qualsiasi label ricorra più frequentemente tra le k label più vicine è scelta per assegnare la classe a x .
- **Support Vector Machine:** l'idea principale di questo classificatore consiste nel rappresentare gli esempi del training come punti nello spazio mappati in modo tale che punti appartenenti a classi differenti siano separati dal più ampio gap possibile. I punti che mappano il test set saranno assegnati ad una categoria o all'altra in base al lato del gap su cui cadono. Più specificatamente, SVM costruisce un iperpiano ed esegue una buona separazione quando l'iperpiano ha la più ampia distanza dai punti di training più vicini di ciascuna classe. Ci sono molti

iperpiani che potrebbero classificare il dato. La miglior scelta è quella di selezionare l'iperpiano che rappresenta la più ampia separazione, o margine, tra due classi, ossia l'iperpiano tale che la distanza tra esso e il punto più vicino su ciascun lato sia massima.

3.2.3 Validazione della Classificazione

Al fine di valutare la bontà della classificazione effettuata, viene introdotto il concetto di **Matrice di Confusione**. Essa è una matrice $n \times n$, con n =numero di classi, nella quale vengono messe a confronto le etichette reali con quelle del classificatore proposto. La figura 3.11 rappresenta la matrice di confusione con 2 classi possibili:

- Il **Vero Positivo** indica quante volte ho classificato giusto l'etichetta che io considero come positiva, ad esempio la classe 1;
- Il **Vero Negativo** indica quante volte ho classificato giusto l'etichetta che io considero come negativa, ad esempio la classe 2;
- Il **Falso Positivo** indica quante volte ho classificato sbagliato l'etichetta che io considero come positiva, ossia il mio classificatore ha etichettato il mio campione con la classe 1, ma in realtà quella misurazione appartiene alla classe 2;
- Il **Falso Negativo** indica quante volte ho classificato sbagliato l'etichetta che io considero come negativa, ossia il mio classificatore ha etichettato il mio campione con la classe 2, ma in realtà quella misurazione appartiene alla classe 1;

		Indici predetti	
Indici reali		Classificazione positiva	Classificazione negativa
	Presenza di condizione	Vero positivo <i>tp</i>	Falso negativo <i>fn</i> (<i>type II error</i>)
	Assenza di condizione	Falso positivo <i>fp</i> (<i>type I error</i>)	Vero negativo <i>tn</i>

Figura 3.11: Rappresentazione della matrice di confusione per 2 classi

La condizione ideale di classificazione sarebbe quella in cui la matrice di confusione è diagonale, ossia non presenta falsi positivi o falsi negativi. Da questa si possono ricavare varie misure, tutte nell'intervallo $[0,1]$:

- **Accuratezza:** $\frac{tp + tn}{tp + tn + fp + fn}$
- **Precisione:** $\frac{tp}{tp + fp}$
- **Sensitività:** $\frac{tp}{tp + tn}$
- **F1-measure:** $\frac{2 * precisione * sensitività}{sensitività + precisione}$
- **Specificità:** $\frac{tn}{fp + tn}$

Nel caso di più classi, si possono generalizzare le misure di precisione e sensibilità associate alle singole classi, mentre il concetto di accuratezza resta sempre lo stesso. La figura 3.12 rappresenta la generalizzazione delle formule suddette.

$$precisione_c = \frac{tp}{tp + fp} = \frac{Conf(c, c)}{\sum_d Conf(d, c)}$$

$$sensitività_c = \frac{tp}{tp + fn} = \frac{Conf(c, c)}{\sum_d Conf(c, d)}$$

Figura 3.12: Rappresentazione della matrice di confusione per c classi

3.3 Riduzione della dimensionalità

3.3.1 Analisi Delle Componenti Principali

L'analisi in componenti principali o **PCA** è una tecnica per la semplificazione dei dati, con lo scopo primario di ridurre un numero più o meno elevato di variabili in alcune caratteristiche latenti. Questo procedimento prende il nome di **feature reduction**. Ciò avviene tramite una trasformazione lineare delle variabili che proietta quelle originarie in un nuovo sistema cartesiano nel quale la nuova variabile con la maggiore varianza viene proiettata sul primo asse, la seconda variabile per dimensione della varianza sul secondo

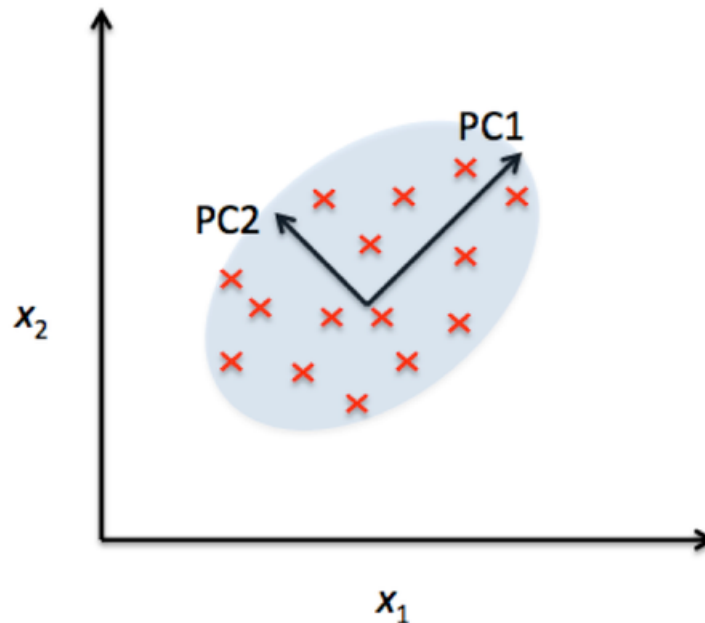


Figura 3.13: Rappresentazione del cambio di dimensionalità tramite PCA

asse e così via. La riduzione della complessità avviene quindi limitandosi ad analizzare le principali (per varianza) tra le nuove variabili. Diversamente da altre trasformazioni lineari, in questa tecnica sono gli stessi dati che determinano i vettori di trasformazione.

Un metodo per calcolare la componente w_i (ossia quella che effettua la trasformazione per la variabile i) utilizza la **matrice delle covarianze** di x , mentre un altro metodo possibile usa la matrice dei coefficienti di correlazione. Innanzitutto, si devono trovare gli autovalori della matrice di covarianza: Si ottengono così tanti autovalori quante sono le variabili x . L'autovalore con il maggiore valore corrisponde alla dimensione w che ha la maggior varianza: esso sarà dunque la varianza della componente principale numero 1. Per ciascun autovalore viene calcolato il corrispondente autovettore, ossia la matrice dei coefficienti che moltiplicano le vecchie variabili x nella combinazione lineare per l'ottenimento delle nuove variabili w . La matrice degli autovettori degli autovettori viene definita anche matrice di rotazione V e, eseguendo l'operazione $W = V * X$, dove W è il vettore colonna avente come elementi le nuove variabili w_1, w_2, \dots e X è il vettore colonna avente come elementi le vecchie variabili x_1, x_2, \dots , si possono trovare le coordinate di ciascun punto nel nuovo spazio vettoriale. Alla fine, quindi, si tengono solo le componenti le quali, sommate tra di loro, esprimono una certa varianza (es. 90% della varianza dei dati), mentre le altre vengono ignorate. In questo modo, par-

tendo da n variabili iniziali, posso arrivare a $(n-k)$ nuove variabili, dove k è il numero di componenti che non mi servono per raggiungere la soglia di varianza prefissata.

3.3.2 Analisi dei Discriminanti Lineari

Mentre la PCA è utile per la rappresentazione dei pattern (es. riconoscimento di immagini), l'analisi dei discriminanti lineari o **LDA** viene usata per discriminare tali pattern, ossia per trovare delle misure che mi permettano di dividere in più classi i miei dati. Entrambe vengono usate per la riduzione della dimensionalità delle variabili di input).

L'obiettivo della LDA è quello di trovare un vettore w , di trasformazione, tale per cui classi differenti siano ben separate, mentre la diffusione di ogni classe sia ridotta il più possibile. In pratica, si tratta di trovare una soluzione al cosiddetto criterio di Fisher:

$$J_F = \frac{w^T S_B w}{w^T S_w w} \quad (3.1)$$

, nella quale S_B ed S_W sono rispettivamente la matrice di dispersione tra le classi e la matrice di dispersione all'interno della classe. Nel caso di due classi, la S_B si calcola come:

$$S_B = (m_1 - m_2)(m_1 - m_2)^T \quad (3.2)$$

, dove m_1 rappresenta la media della classe 1 e m_2 quella della classe 2. S_w , invece, si calcola come

$$S_W = \sum_{i=1}^c \sum_{x \in w_i} (x - m_i)(x - m_i)^T \quad (3.3)$$

con m_i = media della classe i e c = numero di classi.

La soluzione del criterio di Fisher viene chiamata anche **Problema dell'Autovalore Generalizzato** e viene rappresentata da tale equazione:

$$S_B w = \lambda S_w w \quad (3.4)$$

e, se S_w è invertibile, diventa

$$S_w^{-1} S_B w = \lambda w \quad (3.5)$$

, corrispondente al problema degli autovalori regolari che coinvolge $S_w^{-1} S_B$. Una volta che w viene trovato, le feature cercate vengono calcolate:

$$y = w^T x \quad (3.6)$$

le quali possono essere usate per allenare l'algoritmo di classificazione scelto e procedere alla predizione su nuovi dati.

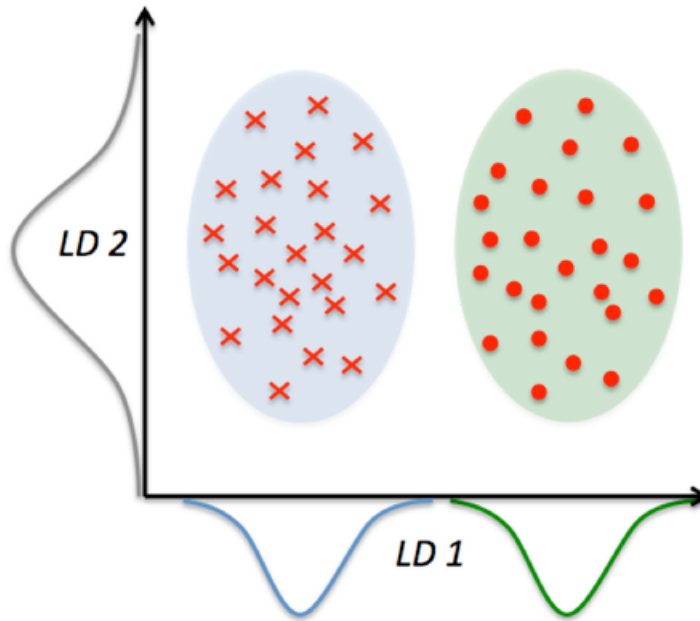


Figura 3.14: Rappresentazione del cambio di dimensionalità tramite LDA

Nel caso si abbiano più di 2 classi, la matrice di dispersione tra le classi, la quali misura la separazione tra le classi, diventa:

$$S_B = \sum_{i=1}^c n_i (m_i - m)(m_i - m)^T \quad (3.7)$$

con n_i = numero di campioni di training appartenenti alla classe i ed m = media di tutti i campioni di training. Ci saranno quindi $C-1$ autovettori, ognuno proveniente da una soluzione di (3.5), che potrebbero non essere ortogonali tra loro ma formano uno sottospazio lineare tale per cui il criterio di Fisher è massimizzato. Questi vengono inseriti in una matrice W e si calcolano le feature come in (3.6).

Motivazioni ed obiettivi

Quello del FoG, come introdotto in precedenza, è un problema che debilita in modo anche grave il paziente affetto da Parkinson. L'obiettivo della nostra metodologia è quello di riconoscere le situazioni di FoG, attraverso l'identificazione di pattern o condizioni che portano il paziente a bloccarsi ed incorrere nel rischio di cadere. A tale scopo, è necessario sviluppare una metodologia che permetta di identificare tali avvenimenti, quindi fornisca, nel modo più accurato possibile, l'informazione sulla presenza o meno di FoG, per poi definire anche gli istanti immediatamente precedenti al blocco, che chiamiamo preFoG. Una volta che si riconosce tali intervalli, tramite uno stimolo uditorio dato nella fase di preFoG si può evitare l'occorrenza del FoG e quindi permette al paziente di continuare nell'attività che stava svolgendo senza che abbia blocchi o rischi di cadere.

Quello che, dunque, si vuole progettare è un algoritmo automatizzato non supervisionato che, attraverso l'uso di accelerometri posizionati sul paziente, permetta di identificare gli istanti di pre-FoG e di FoG del paziente. Questo permetterebbe di sostituire il dottore, o fisioterapista, nell'identificazione di episodi di FoG nella fase di test.

Infatti, fino ad adesso, al paziente venivano fatti fare dei percorsi ed il dottore, tramite l'utilizzo di registrazioni, registrava i FoG ed etichettava dunque tutti gli istanti della registrazione in base all'occorrenza del blocco oppure no. Con questo approccio, invece, tutto questo verrebbe effettuato dall'algoritmo, senza il bisogno dell'intervento umano.

Apprendimento non supervisionato per l'identificazione di contensti di FoG

I lavori che sono stati svolti e presentati nel capitolo 2 sul Freezing Of Gait sono basati sull'identificazione di 2 classi, ossia quando il paziente e' in una normale attivita' quotidiana, che chiameremo NoFoG, oppure quando il paziente soffre di un episodio di FoG. Il nostro lavoro, invece, vuole basarsi sull'identificazione anche di una nuova classe rispetto alle 2 precedenti, che chiamiamo preFoG. Questa dovrebbe rappresentare la fase di transizione da uno stato di NoFoG ad uno di FoG, per cui identificare tale classe permetterebbe di anticipare l'occorrenza di un FoG e, quindi, dare in anticipo uno stimolo al paziente per evitare il verificarsi del FoG. La durata di ogni preFoG e' stata fissata a 2 secondi in quanto, secondo noi, questo intervallo di tempo e' sufficiente per raccogliere abbastanza informazioni al fine di identificare le occorrenze di preFoG e quindi quelle di FoG in anticipo rispetto al suo verificarsi. Il flusso della tesi viene rappresentato in figura 5.1.

Lo sviluppo della procedura e' stato svolto usando software Matlab.

5.1 Dataset

L'approccio che andiamo a proporre è stato testato sul dataset DAPHNET¹, il quale contiene dati collezionati da 10 pazienti parkinsoniani, dei quali 8 presentano contesti di FoG, mentre 2 di loro non ne presentano. I dati sono stati registrati usando 3 accelerometri 3D attaccati alla caviglia, al ginocchio e nella zona lombare del paziente, usando una frequenza di campionamento di 64 Hz, ossia vengono raccolti 64 campioni ogni secondo.

I soggetti hanno completato sessioni da 20-30 minuti ciascuno, consistenti di 3 fasi di camminata:

1. Camminata avanti ed indietro lungo una linea retta, con delle rotazioni di 180 gradi;

¹www.wearable.ethz.ch/resources/Dataset

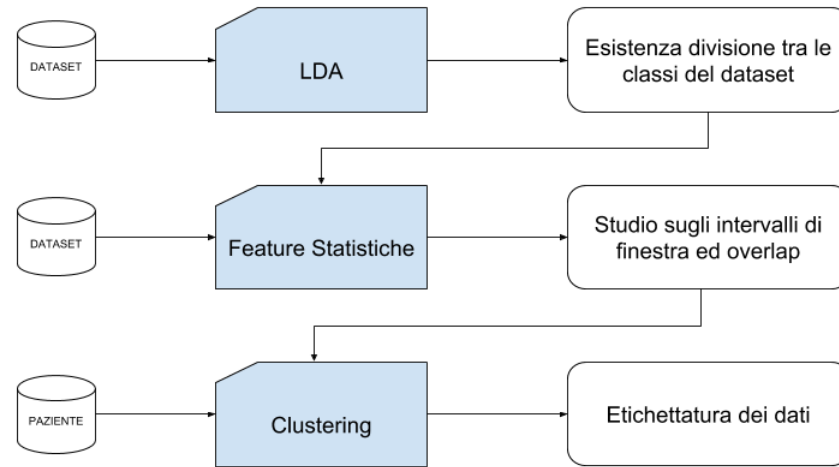


Figura 5.1: Rappresentazione del flusso della tesi

2. Camminata casuale con una serie di fermate volontarie e rotazioni di 360 gradi;
3. Camminata che simula attività di vita quotidiana, tra le quali entrare in stanze ed uscirne, camminare nella cucina, prendersi un bicchiere d'acqua e tornare al punto di partenza.

Le prestazioni motorie variano molto tra i pazienti. Mentre alcuni soggetti hanno mantenuto una camminata regolare durante gli episodi di non FoG, altri hanno camminato molto lentamente ed in modo instabile. L'intero dataset contiene in totale 237 episodi di FoG; la durata di ognuno di essi è tra i 0.5s ed i 40.5s. Il 50% degli episodi di FoG è durato meno di 5.4s ed il 93.2% è più corto di 20s. Gli episodi di FoG sono stati identificati da fisioterapisti usando registrazioni video sincronizzate. L'inizio di un episodio di FoG è stato definito come il punto dove la sequenza normale di camminata è stata interrotta, mentre la fine del FoG è stata definita come il momento in cui tale sequenza riprende.

5.2 Dimostrazione dell'esistenza effettiva delle 3 classi

Il lavoro svolto in questa tesi, come scritto precedentemente, si differenzia dagli altri già citati in 2 in quanto si vogliono usare 3 classi invece delle 2 fornite dal dataset. Il primo passo dunque è quello di dimostrare che effettivamente le 3 classi sono distinte tra loro e che quindi sviluppare un approccio basandosi su tali classi è possibile.

Per fare questo, si e' usato un approccio di Discriminazione Lineare. Un discriminante e' una funzione che prende in ingresso un vettore x e lo assegna ad una tra le K classi fornite. Il metodo di discriminazione che si vuole usare e' quello di Fisher, il quale viene spiegato nel Capitolo 3.

5.2.1 Divisione dei dati

Per ogni paziente, prendiamo in input la matrice di dati grezzi degli accelerometri fornita dal dataset, la quale contiene anche un campo relativo al tempo di ogni campione ed un'etichetta, che mi indica lo stato in cui si trova il mio paziente. Dividiamo i dati degli accelerometri da quelli relativi al tempo ed allo stato e calcoliamo delle feature su tali dati. Queste vengono calcolate vettorizzando finestre di dati grezzi, dove ognuna di esse dura 2 secondi, in accordo con quanto posto per la durata del pre-FoG. Ogni nuova finestra non comincia dalla fine della precedente, ma presenta una certa sovrapposizione. Per tenere traccia dell'etichetta reale a cui ogni finestra temporale appartiene, abbiamo usato il vettore relativo all'etichetta fornita dal dataset e, per ogni finestra temporale, abbiamo tenuto quella che si presenta piu' volte all'interno di essa. Il codice implementativo di quanto appena descritto e' fornito in 5.1.

Listing 5.1: Vettorizzazione dei dati degli accelerometri

```
for i=1:size_windows_sample-size_overlap_samples:m -  
    size_windows_sample  
    B = A(i:i+size_windows_sample-1,:);  
    % vettorizzo ogni finestra  
    B=B(:);  
    F(number_sample,:)=B';  
  
    %salvo la classe di ogni finestra  
    class(number_sample)=mode(FREEZE(i:i+  
        size_windows_sample-1,:));  
  
    %go to next sample  
    number_sample = number_sample + 1;  
  
end
```

5.2.2 Algoritmo di discriminazione

Una volta vettorizzato l'intero dataset, si procede ad applicare l'algoritmo di discriminazione, il quale per prima cosa divide le feature appena calcolate

tramite le classi di appartenenza. Calcola quindi la media per ogni classe e determina la dimensione di ognuna di esse. A questo punto, determina la within class covariance e la between class covariance, ossia quanta similarita' esiste tra gli elementi della stessa classe e quanta dissimilarita' e' presente tra classi differenti. Una volta trovate tali misure, prende gli autovalori ed autovettori della soluzione del problema dell'autovalore generalizzato e teniamo solo quelli piu' rappresentativi, che nel caso dell'approccio di Fisher sono pari al numero di classi presenti meno 1. Questi autovalori vengono poi moltiplicati per le feature calcolate precedentemente per ottenere una riduzione della dimensionalita' ed avere 2 feature rappresentative. Il codice della discriminazione viene fornito in 5.2.

Listing 5.2: LDA

```
A=F';

%quindi ho A 1152x1449 double
%1152 finestre
%ogni finestra ha 1449 features
[d,N] = size(A);

K = max(class); % numero classi in gioco;

% 1. Divido le feature tramite le classi Ck
for k = 1:K
a = find (class == k);
Ck{k} = A(:, a);
end

% 2. Calcolo la media per ogni classe per ogni
    finestra
for k = 1:K
mk{k} = mean(Ck{k},2);
end

% 3. Determino la grandezza di ogni classe
for k = 1:K
[d, Nk(k)] = size(Ck{k});
end

% 4. determino le within class covariance
for k = 1:K
S{k} = 0;
for i = 1:Nk(k)
S{k} = S{k} + (Ck{k}(:,i)-mk{k})*(Ck{k}(:,i)-mk{k})';
end
S{k} = S{k}./Nk(k);
```

```
end
Swx = 0;
for k = 1:K
Swx = Swx + S{k};
end

% 5. determino la between class covariance
% 5.1 determino la media totale
m = mean(A,2);
Sbx = 0;
for k=1:K
Sbx = Sbx + Nk(k)*((mk{k} - m)*(mk{k} - m)');
end
Sbx = Sbx/K;

MA = inv(Swx)*Sbx;

% eigenvalues/eigenvectors
[V,D] = eig(MA);

% 5: transform matrix
if (k > 1)
W = V(:,1:K-1);
end
if (k == 1)
W = V(:,1:1);
end

% 6: transformation
Y = W'*A;
```

5.2.3 Risultati

Una volta calcolate le feature discriminanti, per visualizzare se effettivamente c'è una distinzione tra le classi, facciamo un grafico con queste due grandezze ed ogni punto viene colorato in base alla reale classe di appartenenza. La figura 5.2 mostra tale grafico. Si può notare immediatamente come le due feature utilizzate riescono a discriminare effettivamente tra le 3 classi e quindi un approccio basandosi su tali componenti è giustificato.

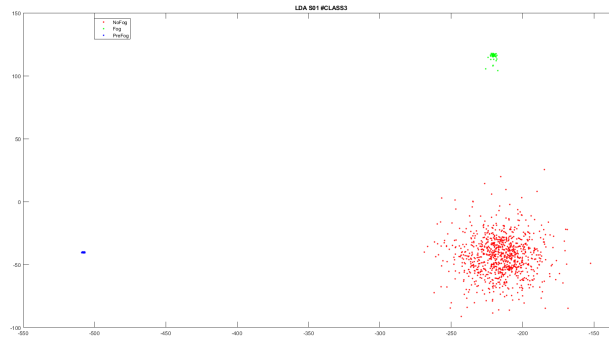


Figura 5.2: Grafico delle 3 classi in base alla discriminazione data dall'approccio di Fisher per il paziente 1

5.3 Determinazione del miglior intervallo di finestra

In letteratura, la finestra temporale su cui vengono calcolate le feature e' sempre stata presa a priori, basandosi su delle ipotesi. Non e' mai stato condotto quindi uno studio su quale possa essere il miglior intervallo di divisione dei dati per calcolare le feature. Quello che ci prefissiamo in questa sezione e' dunque testare diverse combinazioni di durata delle finestre con sovrapposizione al fine di trovare quella che meglio divide i nostri dati. Per fare questo, abbiamo utilizzato algoritmi di clustering in quanto non volevamo che il risultato fosse influenzato dall'appartenenza o meno di un dato ad una determinata classe, come nel caso dell'LDA o di algoritmi di classificazione. L'approccio che è stato preso in considerazione è basato sul calcolo di feature (o proprietà) derivanti dalla matematica statistica. Il flusso di tale approccio è:

1. Pre-processamento dei dati degli accelerometri, al fine di eliminare il rumore presente ed identificare eventuali punti di outline, ossia campioni che non presentano affinità col resto dei dati poiché dovuti a movimenti non consoni;
2. Finestramento dei dati in base ad intervalli variabili al fine di calcolare le feature, dove ogni intervallo presenta una certa sovrapposizione con l'intervallo precedente;
3. Calcolo effettivo delle feature statistiche;
4. Applicazione degli algoritmi di clustering;
5. Calcolo di metriche che indicano quanto bene gli algoritmi di clustering hanno diviso i dati in relazione alle label fornite dal dataset.

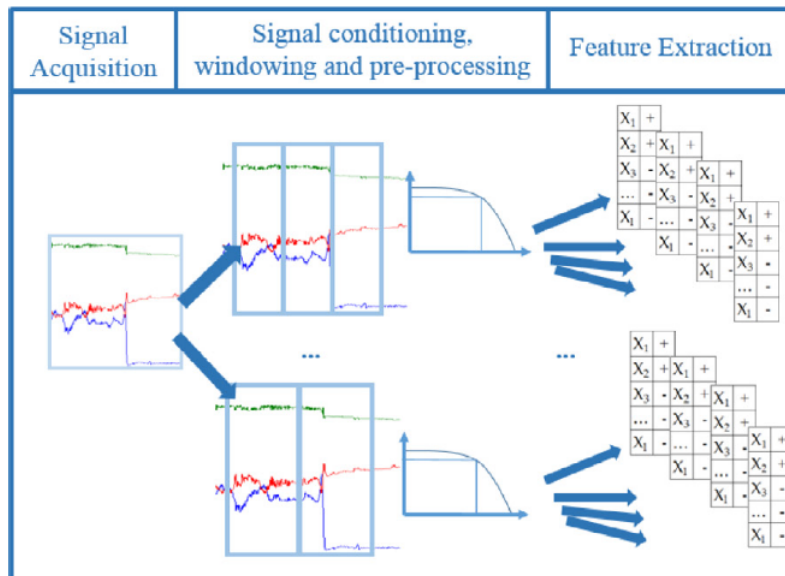


Figura 5.3: Schema generale di calcolo delle feature statistiche

5.3.1 Pre-processamento dei dati

Gli accelerometri sono dispositivi che misurano le vibrazioni o l'accelerazione del movimento di una struttura. La forza generata dalle vibrazioni o una variazione del movimento (accelerazione) fa in modo che la massa "comprima" il materiale piezoelettrico, che genera una carica elettrica proporzionale alla forza esercitata su di esso. Dato che la carica è proporzionale alla forza e che la massa è una costante, la carica è proporzionale anche all'accelerazione. Come tutti i dispositivi che misurano delle grandezze presentano dell'incertezza strumentale che può portare ad avere rumore, ossia un segnale non desiderato, di origine naturale o artificiale, che si sovrappone all'informazione degli accelerometri stessi. Questo rumore porta ad avere dei punti definiti outlier, ossia campioni che non presentano affinità col resto dei dati poiché dovuti a movimenti non consoni.

Al fine di rimuovere tali punti, che altererebbero in modo negativo il calcolo delle nostre feature, si rende necessario rimuoverli dal nostro dataset. Per identificarli, è stato implementato un filtro passa-alto che eliminano tutte le frequenze inferiori a 0.5Hz, le quali non appartengono al normale movimento umano ma indicano appunto la presenza di rumore, come evidenziato in [21]. L'implementazione del filtro viene fornita in 5.3.

Listing 5.3: hpfilter

```
% All frequency values are in Hz.
Fs = 64; % Sampling Frequency
```

```

Fstop = 0.4;      % Stopband Frequency
Fpass = 0.8;      % Passband Frequency
Astop = 60;       % Stopband Attenuation (dB)
Apass = 1;        % Passband Ripple (dB)
match = 'passband'; % Band to match exactly

% Construct an FDESIGN object and call its ELLIP
method.
h = fdesign.highpass(Fstop, Fpass, Astop, Apass, Fs)
;
Hd = design(h, 'cheby2', 'MatchExactly', match);

```

5.3.2 Definizione degli intervalli

Una volta filtrati i dati, il passo successivo dell'algoritmo consiste nel dividerli in intervalli temporali, al fine di poter computare le caratteristiche degli stessi. Un intervallo contiene un determinato movimento, che però verrebbe "interrotto" con la fine della finestra stessa e quindi potresti perdere informazioni su tale movimento. Per evitare il più possibile tale perdita, è usato un approccio che sfrutta le sovrapposizioni tra intervalli, per cui ogni nuova finestra non inizia subito dopo la fine della precedente, ma all'interno di essa. La durata degli intervalli che sono stati presi in considerazione sono:

- Da 1 secondi a 5 secondi per la finestra temporale, con un incremento di 0.5 secondi;
- Da 0.5 a 4.5 secondi per l'intervallo di overlap, con un incremento sempre di 0.5 secondi.

La condizione fondamentale per usare le sovrapposizioni è che l'intervallo delle stesse non sia mai maggiore della durata delle finestre temporali, per cui è stata posta la condizione che la finestra temporale sia sempre almeno 0.5 secondi più lunga dell'intervallo di sovrapposizione. Per capire da dove parte la nuova finestra, quindi, prendiamo la posizione a cui siamo arrivati e togliamo la durata della sovrapposizione. L'implementazione viene fornita in 5.4.

Listing 5.4: Definizione degli intervalli

```

for k = 5:5:45

Y = k/10;

for i = (Y+0.5):0.5:5
%dimensione della finestra in secondi
size_windows_sec = i;

```



```

%dimensione della finestra in campioni
size_windows_sample = Fs * i;

%dimensione dell'overlap in secondi
size_overlap_sec = Y;
%dimensione dell'overlap in campioni
size_overlap_samples = Fs * Y;

number_sample = 1;

%for each sample window, compute the features
for i=1:size_windows_sample-size_overlap_samples:m -
    size_windows_sample
    B = A(i:i+size_windows_sample-1,:);
    ...

```

5.3.3 Calcolo delle Feature

Per ogni possibile combinazione di finestra temporale e sovrapposizione, a questo punto vengono calcolate le feature per ogni intervallo. Le feature prese in considerazione nel nostro studio sono descritte in tabella 5.1.

Nel dataset, ogni campione è etichettato. Poiché le nostre feature sono composte da molti campioni messi assieme, si rende necessario trovare un metodo per unire, sbagliando il meno possibile, tutte le etichette della finestra che prendiamo in considerazione. Per fare ciò, si è deciso di utilizzare la funzione di moda, ossia il numero che si ripete più spesso all'intervallo dell'intervallo considerato. L'etichetta risultante viene inserita nella tabella e verrà utilizzata come criterio di confronto, al fine di valutare le prestazioni degli algoritmi di clustering. L'implementazione del codice è fornito in 5.5

Listing 5.5: Calcolo delle Feature

```

%time sample
F(number_sample, 1) = TIME(i,:);
%min --> minimum value for each accelerometer
F(number_sample, 2:10) = min(B);
%max --> maximum value for each accelerometer
F(number_sample, 11:19) = max(B);
%median --> median signal value
F(number_sample, 20:28) = median(B);
%mean --> average value
F(number_sample, 29:37) = mean(B);
%ArmMean --> harmonic average of the signal
F(number_sample, 38:46) = harmmean(B);

```

N	Feature	Descrizione
1	Minimo	Valore minimo del segnale
2	Massimo	Valore massimo del segnale
3	Mediana	Valore mediano del segnale
4	Media	Valore medio del segnale
5	Media Armonica	Media armonica del segnale
6	Errore Quadratico Medio	Valore Quadratico medio del segnale
7	Media Geometrica	Media geometrica del segnale
8	Varianza	Radice della deviazione standard
9	Deviazione Standard	Deviazione media del segnale rispetto alla media
10	Curtosi	Allontanamento dalla normalità distributiva del segnale
11	Simmetria	Grado di asimmetria della distribuzione del segnale
12	Moda	Il numero che appare più volte nel segnale
13	Media Tagliata	Media tagliata del segnale nella finestra
14	Entropia	Misura della di distruzione delle componenti in frequenza
15	Range	Differenza tra il valore minimo e massimo del segnale
16	Magnitudine	Somma della norma euclidea di tre assi normalizzato sulla lunghezza del segnale
17	Area Magnitudine	Accelerazione della magnitudine di tre assi normalizzato sulla lunghezza del segnale
18	Autovalori delle direzioni dominanti	Autovalori della matrice di covarianza di tre assi
19	Accelerazione media dell'energia	Valore medio dell'energia sui 3 assi

Tabella 5.1: Descrizione delle Feature Statistiche

```

%root mean square --> quadratic mean value of the
    signal
F(number_sample, 47:55) = rms(B);
%variance --> square of the standard deviation
F(number_sample, 56:64) = var(B);
%standard deviation --> mean deviation of the signal
    compared to the
%average
F(number_sample, 65:73) = std(B);
%kurtosis --> degree of peakedness of the sensor

```

```

    signal distribution
    %(allontanamento dalla normalita distributiva)
    F(number_sample, 74:82) = kurtosis(B);
    %skewdness --> degree of asymmetry of the sensor
    signal distribution
    F(number_sample, 83:91) = skewness(B);
    %mode --> number that appears most often in the
    signal
    F(number_sample, 92:100) = mode(B);
    %trim mean --> trimmed mean of the signal in the
    window
    F(number_sample, 101:109) = trimmean(B,10);
    %range --> difference between the largest and the
    smallest values of
    %the signal
    F(number_sample, 110:118) = range(B);
    %signal magnitude vector --> sum of the euclidean
    norm over the three
    %axis over the entire window normalized by the
    windows lenght
    F(number_sample, 119) = svmn(B(:,1:3), length(B));
    F(number_sample, 120) = svmn(B(:,4:6), length(B));
    F(number_sample, 121) = svmn(B(:,7:9), length(B));
    %normalized signal magnitude area --> acceleration
    magnitude summed
    %over three axes normalized by the windows length
    F(number_sample, 122) = sman(B(:,1:3), length(B));
    F(number_sample, 123) = sman(B(:,4:6), length(B));
    F(number_sample, 124) = sman(B(:,7:9), length(B));
    %eigenvalues of dominant directions --> eigenvalues
    of the
    %covariance matrix of the acceleration data along x,
    y and z axis
    F(number_sample,125) = eigs(cov(B(:,1:3)),1);
    F(number_sample,126) = eigs(cov(B(:,4:6)),1);
    F(number_sample,127) = eigs(cov(B(:,7:9)),1);
    %averaged acceleration energy --> mean value of the
    energy over
    %three acceleration axes
    F(number_sample,128) = energyn(B(:,1:3),length(B));
    F(number_sample,129) = energyn(B(:,4:6),length(B));
    F(number_sample,130) = energyn(B(:,7:9),length(B));
    %is freezing?
    F(number_sample,131) = mode(FREEZE(i:i+

```

```

        size_windows_sample-1,:));

%go to next sample
number_sample = number_sample + 1;
...

```

5.3.4 Clustering

Avendo il nuovo dataset composto dalle caratteristiche dei dati originali scomposti in intervalli, si possono applicare gli algoritmi di clustering. Quelli che sono stati presi in considerazione nella nostra analisi sono: K-means, Fuzzy C-Means e Neural Network. Il K-means viene testato in 4 varianti diverse, in base alle seguenti metriche di distanza: cityblock, correlation, cosine, euclidean.

Ogni algoritmo di clustering viene applicato in tutte le possibili combinazioni di intervalli ed ognuno di essi restituisce un vettore di numeri, che corrispondono alle etichette che assegnano ad ogni vettore di feature. Questi vengono salvati e verranno poi utilizzati per calcolare le prestazioni degli algoritmi stessi in confronto alla classificazione originale del dottore. L'implementazione del codice è fornita in 5.6

Listing 5.6: Uso degli algoritmi di clustering

```

%% k-means %%

% choose of parameter
means1 = 'sqeuclidean';
means2 = 'correlation';
means3 = 'cityblock';
means4 = 'cosine';
for q=1:4
    if q == 1
        dist_k = means1;
    end
    if q == 2
        dist_k = means2;
    end
    if q == 3
        dist_k = means3;
    end
    if q == 4
        dist_k = means4;
    end
    options_km = statset('UseParallel', false);
    maxiter = 100000;

```

```
% cluster
kidx = kmeans(bonds, numClust, 'distance', dist_k, 'options', options_km, 'MaxIter', maxiter);

P = array2table([A(:,n) kidx]);
writetable(P, [datadir 'versus_kmeans_' dist_k '_' fileruns(r).name] );
display([datadir 'versus_kmeans_' dist_k '_' fileruns(r).name]);
end

%%% neural networks - Self organizing Maps %%%

% Create a Self-Organizing Map
dimension1 = 3;
dimension2 = 1;
net = selforgmap([dimension1 dimension2]);

% Train the network
net.trainParam.showWindow = 0;
[net,tr] = train(net,bonds');

% Test the network
nidx = net(bonds');
nidx = vec2ind(nidx)';

P = array2table([A(:,n) nidx]);
writetable(P, [datadir 'versus_net_' fileruns(r).name] );
display([datadir 'versus_net_' fileruns(r).name]);

%      %%% FUZZY C-MEANS %%%
options(1) = 2;
options(2) = 10000;
options(3) = 1e-5;
options(4) = 0;
% Hide iteration information by passing appropriate options to FCM
[centres,U] = fcm(bonds,numClust,options);
[~, fidx] = max(U);
fidx = fidx';
```

```
P = array2table([A(:,n) fidx]);
writetable(P, [datadir 'versus_cmeans_' fileruns(r).
    name] );
display([datadir 'versus_cmeans_' fileruns(r).name]);
...
```

5.3.5 Calcolo delle prestazioni

Ogni algoritmo di clustering, per tutte le combinazioni di finestra ed overlap, fornisce un vettore di etichette. Questo rappresenta la divisione in cluster effettuata dall'algoritmo, i quali andranno confrontati con le etichette reali del dataset. Per fare questo, procediamo al calcolo di metriche quali, in ordine, accuratezza, precisione, sensitività e F1-measure. Per quanto riguarda la precisione e la sensitività, esse sarebbero relative ad ogni classe, ma poiché ci interessa indovinare la maggior parte delle volte tutte e 3 le classi, quelle che saranno riportate sono la media della precisione e sensitività delle classi. Il codice implementativo è fornito in 5.7

Listing 5.7: Calcolo delle prestazioni

```
% Per tutti i file del paziente $isubject
for r = 1:length(fileruns)

% Etichette del dataset
filename = [datadir fileruns(r).name];
T1 = readtable(filename);
[m1,n1] = size(T1);
A1 = table2array(T1(:,n1));

% Etichette cluster
filename2 = [datadir fileruns2(r).name];
T2 = readtable(filename2);
[m2,n2] = size(T2);
A2 = table2array(T2(:,2));
% Etichette cluster e dataset
D = [A2 A1];

% Matrice di confusione
[C,order] = confusionmat(D(:,2),D(:,1));
accuracy = c_accuracy(C);
precision = c_precision(C);
recall = c_recall(C);
F1measure = c_F1measure(precision,recall);

B = [accuracy precision recall F1measure];
```

```

E = [E B];

end
Q = [Q ; [e E]];
e = [e [0 0 0 0]];
...
function accuracy = c_accuracy(C)
giusti = sum(diag(C));
totale = sum(sum(C));
accuracy = giusti/totale;
end

function precision = c_precision(C)
precision1 = C(1,1)/(C(1,1)+C(2,1)+C(3,1));
precision2 = C(2,2)/(C(1,2)+C(2,2)+C(3,2));
precision3 = C(3,3)/(C(1,3)+C(2,3)+C(3,3));
precision = mean([precision1;precision2;precision3]);
end

function recall = c_recall(C)
recall1 = C(1,1)/(C(1,1)+C(1,2)+C(1,3));
recall2 = C(2,2)/(C(2,1)+C(2,2)+C(2,3));
recall3 = C(3,3)/(C(3,1)+C(3,2)+C(3,3));
recall = mean([recall1;recall2;recall3]);
end

function F1measure = c_F1measure(precision,recall)
F1measure = 2*precision*recall/(precision+recall);
end

```

Viene creata dunque una tabella che riassume, per ogni possibile intervallo, le prestazioni dell'algoritmo, come si vede in 5.2, 5.3, 5.4.

Tabella 5.2: Esempio di tabelle delle prestazioni del clustering

	Secondi: 1				Secondi: 1.5				Secondi: 2			
Ov 0.5	0,7	0,6	0,3	0,4	0,8	0,3	0,3	0,3	0,91	0,5	0,5	0,5
Ov 1	0	0	0	0	0,7	0,3	0,3	0,3	0,8	0,6	0,5	0,55
Ov 1.5	0	0	0	0	0	0	0	0	0,91	0,72	0,6	0,6

5.3.6 Risultati

Dopo che sono state generate tutte le tabelle per tutti i pazienti in cui vengono espressi i valori di accuratezza, precisione, sensitività e F1-measure,

Tabella 5.3: Esempio di tabelle delle prestazioni del clustering

	Secondi: 2.5				Secondi: 3				Secondi: 3.5			
Ov 0.5	0,7	0,3	0,3	0,3	0,4	0,1	0,2	0,17	0,46	0,32	0,4	0,35
Ov 1	0,05	0,3	0,2	0,3	0,46	0,3	0,4	0,38	0,46	0,32	0,4	0,35
Ov 1.5	0,75	0,2	0,2	0,2	0,8	0,3	0,3	0,3	0,46	0,32	0,4	0,35
Ov 2	0,8	0,3	0,3	0,3	0,6	0,3	0,1	0,21	0,46	0,32	0,4	0,35
Ov2.5	0	0	0	0	0,81	0,26	0,24	0,25	0,7	0,3	0,2	0,26
Ov 3	0	0	0	0	0	0	0	0	0,6	0,3	0,31	0,31

Tabella 5.4: Esempio di tabelle delle prestazioni del clustering

	Secondi: 4				Secondi: 4.5				Secondi: 5			
Ov 0.5	0,8	0,36	0,35	0,35	0,81	0,32	0,34	0,33	0,75	0,24	0,26	0,25
Ov 1	0,76	0,21	0,28	0,26	0,79	0,32	0,34	0,31	0,79	0,3	0,3	0,3
Ov 1.5	0,46	0,36	0,33	0,35	0,8	0,36	0,35	0,35	0,59	0,24	0,33	0,29
Ov 2	0,41	0,33	0,44	0,39	0,76	0,21	0,28	0,26	0,26	0,79	0,32	0,34
Ov2.5	0,59	0,24	0,33	0,29	0,46	0,36	0,33	0,35	0,35	0,8	0,36	0,35
Ov 3	0,44	0,11	0,12	0,11	0,41	0,33	0,44	0,39	0,8	0,36	0,35	0,35
Ov 3.5	0,78	0,23	0,33	0,28	0,75	0,24	0,26	0,25	0,76	0,21	0,28	0,26
Ov 4	0	0	0	0	0,79	0,3	0,3	0,3	0,75	0,24	0,26	0,25
Ov 4.5	0	0	0	0	0	0	0	0	0,79	0,3	0,3	0,3

siamo interessati all'intervallo migliore, ossia quello che ci fornisce i valori maggiori delle metriche suddette. Per fare questo, analizziamo ogni tabella e, per ogni algoritmo, salviamo l'intervallo ed i valori che presentano le metriche migliori, come nel caso delle tabelle 5.5, 5.6, 5.7.

Ogni tabella descrive:

- l'algoritmo di clustering che e' stato utilizzato;
- il paziente del test;
- i valori delle metriche migliori trovate;
- la combinazione overlap-secondi in cui risultano tali metriche.

Osservando le tabelle, si nota come il valore di overlap e dimensione della finestra che piu' frequentemente mi portano ad avere metriche migliori sono:

- 1 secondo di sovrapposizione e 2 secondi di finestra;
- 1 secondo di sovrapposizione e 1,5 secondi di finestra.

Il nostro studio degli intervalli quindi ci porta a concludere che il miglior modo con cui si possono dividere i dati in finestre per calcolare le feature e' scegliere 1 secondo di sovrapposizione tra le finestre e 1,5 o 2 secondi per quanto riguarda la lunghezza della finestra temporale.

Tabella 5.5: Rate Pazienti per l'algoritmo di clustering C-means

	CMEANS					
Paziente	Accuratezza	Precisione	Recall	F1	Overlap	Finestra
S01	0,92	0,31	0,33	0,32	1	2
S02	0,44	0,46	0,63	0,53	1	2
S03	0,7	0,51	0,57	0,54	1	2
S04	0,7	0,33	0	0	0,5	1
S05	0,53	0,49	0,46	0,47	1	1,5
S06	0,57	0,36	0,32	0,34	1	1,5
S07	0,48	0,37	0,43	0,4	1	1,5
S08	0,41	0,39	0,43	0,41	1,5	2
S09	0,59	0,35	0,37	0,36	1	1,5
S10	0,73	0,33	0	0	0,5	1

Tabella 5.6: Rate Pazienti per l'algoritmo di clustering K-means

	KMEANS					
Paziente	Accuratezza	Precisione	Recall	F1	Overlap	Finestra
S01	0,92	0,72	0,34	0,46	1,5	2
S02	0,82	0,61	0,34	0,44	1	2
S03	0,75	0,78	0,54	0,64	1,5	2
S04	1	0,33	0	0	0,5	1
S05	0,67	0,56	0,33	0,42	1	2
S06	0,91	0,3	0,33	0,32	1	1,5
S07	0,92	0,31	0,33	0,32	1	2
S08	0,7	0,57	0,34	0,42	1	1,5
S09	0,81	0,6	0,34	0,43	1	2
S10	1	0,33	0	0	0,5	1,5

Tabella 5.7: Rate Pazienti per l'algoritmo di clustering Self-Organizing Map

	NEURAL NETWORK					
Paziente	Accuratezza	Precisione	Recall	F1	Overlap	Finestra
S01	0,92	0,31	0,33	0,32	1	1,5
S02	0,81	0,44	0,34	0,38	1	1,5
S03	0,75	0,78	0,54	0,63	1	2
S04	0,64	0,33	0	0	0,5	1
S05	0,58	0,42	0,46	0,44	1	2
S06	0,57	0,32	0,31	0,31	1	1,5
S07	0,63	0,3	0,26	0,28	1	2
S08	0,56	0,69	0,38	0,49	1,5	2
S09	0,57	0,37	0,41	0,39	1	2
S10	1	0,33	0	0	0,5	1

5.4 Etichettatura dei Dati

Nella sezione precedente, lo studio degli intervalli ci ha portato a scegliere, come dimensioni delle finestre temporali, 1,5 o 2 secondi con 1 secondo di overlap. A questo punto, usando tali valori, ritestiamo per ogni paziente gli algoritmi di clustering ed andiamo ad analizzare direttamente le matrici di confusione, le quali ci diranno esattamente come etichettano tali algoritmi a confronto con le etichette reali.

Conclusione e Lavori Futuri

This thesis proposes a methodology

6.1 Lavori Futuri

Bibliography

Riferimenti bibliografici

- [1] Delval A, Snijders AH, Weerdesteyn V, Duysens JE, Defebvre L, Giladi N e et al. “Objective detection of subtle freezing of gait episodes in Parkinson’s disease.” In: *Mov Disord.* (2010), 25: 1684–1693.
- [2] Nieuwboer A, De Weerd W, Desloovere K, Fieuws S e Broens kaucsik E. “Abnormalities of the Spatiotemporal Characteristics of Gait at the Onset of Freezing in Parkinson ‘ s Disease.” In: *Mov Disord.* (2001), 16: 1066–1075.
- [3] Tay A, Yen SC, Lee PY, Wang CY, Neo A, Phan SW e et al. “Freezing of Gait (FoG) detection for Parkinson Disease.” In: *10th Asian Control Conference (ASCC)* (2015), pp. 1–6.
- [4] Weiss A, Herman T, Giladi N e Hausdorff JM. “Objective assessment of fall risk in Parkinson’s disease using a body-fixed sensor worn for 3 days.” In: *PLoS One* (2014).
- [5] Takač B, Català A, Rodriguez Martin D, van der Aa N, Chen W e Rauterberg M. “Position and Orientation Tracking in a Ubiquitous Monitoring System for Parkinson Disease Patients With Freezing of Gait Symptom.” In: *JMIR mhealth uhealth* (2013), 1: e14.
- [6] Cole BT, Roy SH e Nawab SH. “Detecting Freezing-of-Gait During Unscripted and UNconstrained Activity”. In: *33rd Annual International Conference of the IEEE EMBS* (2011), pp. 5649–5652.
- [7] Ahlrichs C, Samà A, Lawo M, Cabestany J, Rodriguez-Martin D, Pérez-López C e et al. “Detecting freezing of gait with a tri-axial accelerometer in Parkinson’s disease patients.” In: *Med Biol Eng Comput.* (2016), 54: 223–233.
- [8] Coste C, Sijobert B, Pissard-Gibollet R, Pasquier M, Espiau B e Geny C. “Detection of Freezing of Gait in Parkinson Disease: Preliminary Results.” In: *Sensors* (2014), 14: 6819–6827.

- [9] Pérez-López C, Samà A, Rodríguez-Martin D, Català A, Cabestany J e et al. de Mingo E. “Monitoring Motor Fluctuations in Parkinson’s Disease Using a Waist-Worn Inertial Sensor.” In: *International Work Conference on Artificial Neural Networks Advances in Computational Intelligence Lecture Notes on Computer Science*. (2015), pp. 461–474.
- [10] Rodríguez-Martin D, Samà A, Pérez-López C, Català A, Cabestany J e Rodríguez-Molinero A. “A. Posture Detection with waist-worn Accelerometer : An application to improve Freezing of Gait detection in Parkinson ‘ s disease patients.” In: *European Conference on Ambient Assisted Living*. (2014), pp. 1–6.
- [11] Rodríguez-Martin D, Samà A, Pérez López C, Català A, Moreno ArosteGUI JM, Cabestany J e et al. “Home detection of freezing of gait using support vector machines through a single waist-worn triaxial accelerometer”. In: *PLoS ONE* (2017).
- [12] Schaafsma J. D., Gurevich T. Balash Y and, Bartels A. L., Hausdorff J. M. e Giladi N. “Characterization of freezing of gait subtypes and the response of each to levodopa in Parkinson’s disease”. In: *European Journal of Neurology* (2003), 18: 25–29.
- [13] Jovanov E, Wang E, Verhagen L, Fredrickson M e Fratangelo R. “de-FOG: A real time system for detection and unfreezing of gait of Parkinson’s patients.” In: *31st Annual International Conference of the IEEE EMBS*. (2009), pp. 5151–5154.
- [14] Tripoliti EE, Tzallas AT, Tsipouras MG, Rigas G, Bougia P, Leontiou M e et al. “Automatic detection of freezing of gait events in patients with Parkinson’s disease.” In: *Comput Methods Programs Biomed. Elsevier Ireland Ltd* (2013), 110: 12–26.
- [15] Zabaleta H, Keller T e Fimbel EJ. “Gait analysis in frequency domain for freezing detection in patients with Parkinson’s disease.” In: *Gerontechnology*. (2008).
- [16] Zach H, Janssen AM, Snijders AH, Delval A, Ferraye MU, Auff E e et al. “Identifying freezing of gait in Parkinson’s disease during freezing provoking tasks using waist-mounted accelerometry.” In: *Parkinsonism Relat Disord. Elsevier Ltd* (2015), 21: 1362–1366.
- [17] Jiawei Han, Jian Pei e Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.
- [18] Han JH, Lee WJ, Tae Beom A, Beom Suk J e Kwang Suk P. “Gait analysis for freezing detection in patients with movement disorder using three dimensional acceleration system.” In: *25th Annual International Conference of the IEEE EMBS*. (2003), pp. 1863–1865.

- [19] Hausdorff JM, Balash Y e Giladi N. “Time series analysis of leg movements during freezing of gait in Parkinson’s disease:akinesia,rhyme or reason?” In: *Physica A*. (2003), 321:656–570.
- [20] Niazmand K, Tonn K, Zhao Y, Fietzek UM, Schroeteler F, Ziegler K e et al. “Freezing of Gait detection in Parkinson’s disease using accelerometer based smart clothes.” In: *IEEE Biomedical Circuits and Systems Conference (BioCAS)*. (2011), pp. 201–204.
- [21] Bachlin M, Roggen D, Plotnik M, Hausdorff JM, Giladi N e Troster G. “Online Detection of Freezing of Gait in Parkinson’s Disease Patients: A Performance Characterization.” In: *Proc 4th Int ICST Conf Body Area,Networks*. (2009).
- [22] Handojoseno a MA, Shine JM, Nguyen TN, Tran Y, Lewis SJG e Nguyen HT. “The detection of Freezing of Gait in Parkinson’s disease patients using EEG signals based on Wavelet decomposition”. In: *34th AnnuInt Conf IEEE Eng Med Biol Soc*. (2012), 2012:69–72.
- [23] Sinziana Mazilu, Alberto Calatroni, Eran Gazit, Daniel Roggen, Jeffrey M. Hausdorff e Gerhard Troster. “Feature Learning for Detection and Prediction of Freezing of Gait in Parkinson’s Disease”. In: *Machine Learning and Data Mining in Pattern Recognition* (2013).
- [24] Almeida Q.J. e Lebold C. A. “Freezing of Gait in Parkinson’s Disease: A Perceptual Cause for a Motor Impairment?” In: *Neurol Neurosurg Psychiatry* (2010), 81: 513–518.
- [25] REM PARK. “Personal Health Device for the Remote and Autonomous Management of Parkinson’s Disease.” In: *FP7-ICT-2011-7-287677*. (2011-2014).
- [26] Mazilu S, Calatroni A, Gazit E, Mirelman A, Hausdorff J e Troester G. “Prediction of Freezing of Gait in Parkinson’s from Physiological Wearables: An Exploratory Study.” In: *IEEE J Biomed Heal Informatics*. (2015), 6:1–1.
- [27] Mazilu S, Hardegger M, Zhu Z, Roggen D, Troster G, Plotnik M e et al. “Online Detection of Freezing of Gait with Smartphones and Machine Learning Techniques.” In: *Proc 6th Int ICST Conf Pervasive Comput Technol Healthc*. (2012), pp. 123–130.
- [28] Moore ST, Yungher Da, Morris TR, Dilda V, MacDougall HGa, Shine JM e et al. “Autonomous identification of freezing of gait in Parkinson’s disease from lower-body segmental accelerometry”. In: *Neuroeng Rehabil. Journal of NeuroEngineering and Rehabilitation* (2013).
- [29] Moore ST, MacDougall HG e Ondo WG. “Ambulatory monitoring of freezing of gait in Parkinson’s disease”. In: *J Neurosci Methods* (2008), 167: 340–348.

- [30] Zhao Y, Tonn K, Niazmand K, Fietzek UM, D'Angelo LT, Ceballos-Baumann a. e et al. "Online FOG Identification in Parkinson's disease with a time-frequency combined Algorithm." In: *Proc 2012 IEEE-EMBS Int Conf Biomed Heal Informatics*. (2012), pp. 192–195.

Acronyms

FI	Freezing Index
MEMS	MicroElectroMechanical Systems
PSD	Power Spectral Density
FB	Freezing Band
WB	Walking Band
EMG	Elettromiografia
EEG	Elettroencefalogramma
FTH	Freezing Threshold
FP	False Positive
PI	Power Index
PT	Power Threshold
MBFA	Moore-Bächlin FoG Algorithm
PD	Parkinson's disease