Modelli Probabilistici

Andrico Michele 810946,
Miglio Diego 807408,
Porta Alessio 807457

appello Luglio 2019

Indice

1 Introduzione

L'Ambient Assisted Living (AAL) comprende un insieme di soluzioni tecnologiche destinate a rendere attivo, intelligente e cooperativo l'ambiente circostante ad un individuo. Questo permette di rendere soggetti con problematiche di mobilità il più indipendenti possibile, andando a fornire maggiore sicurezza e semplicità nello svolgimento delle loro attività quotidiane. I principali obiettivi dichiarati dall'Associazione Europea AAL sono:

- estendere il periodo in cui le persone possono vivere nel loro ambiente preferito, aumentando la loro autonomia, autosufficienza e mobilità;
- aiutare a mantenere la salute e le capacità funzionali delle persone anziane;
- Promuovere stili di vita migliori e più salutari per le persone a rischio;
- aumentare la sicurezza, prevenire l'esclusione sociale e mantenere la rete relazionale delle persone;
- sostenere gli operatori, i familiari e le organizzazioni dell'assistenza;
- migliorare l'efficienza e la produttività delle risorse nella società che invecchia.

Per lo sviluppo di questo tipo di ambienti è diventato indispensabile l'utilizzo di strumenti atti al tracciamento dei movimenti del soggetto. L'Human Activity Recognition (HAR) è diventata, quindi, una disciplina molto diffusa negli ultimi anni, come riportato in [1]. Le modalità principali con cui collezionare i dati per l'HAR risultano l'*image processing* e l'utilizzo di sensori posizionati sul corpo. Nel caso specifico su cui ci basiamo, ovvero il lavoro presentato in [1], i dati sono raccolti tramite il secondo metodo e, più precisamente, utilizzando accelerometri tri-assiali.
L'obiettivo del nostro lavoro è quindi testare se una rete bayesiana sia in grado di classificare correttamente le attività presenti nel dataset fornito. In particolare, è stata prima creata una rete imponendo dipendenze disegnate da noi in base a conoscenza pregressa, ovvero la posizione dei sensori, e quindi la dipendenza dei movimenti tra le parti del corpo. In secondo luogo, è stata indotta automaticamente la rete bayesiana utilizzando una libreria specifica di R: Bnlearn [2]. Per entrambi i modelli le features sono state considerate dapprima continue, come risultano originariamente, e successivamente discrete, andando a confrontare le performance ottenute in entrambi i casi. Infine, si sono confrontate le performance della nostra rete con quelle della rete indotta e con quelle del classificatore utilizzato in [1].

2 Dataset utilizzato

2.1 Descrizione dei dati

Il dataset utilizzato riporta i dati raccolti su 4 soggetti differenti per un totale di 8 ore di registrazione. Oltre ad età, sesso, altezza, peso e BMI (Body Mass Index) di ogni individuo, sono presenti i valori registrati da 4 accelerometri triassiali posizionati, come mostrato in Figura 1, rispettivamente su vita, coscia sinistra, caviglia destra e braccio destro dei soggetti esaminati. Le misurazioni sono state annotate durante il compimento di 5 attività fisiche differenti, ovvero:

- *sitting*;
- *sitting down*;
- *standing*;
- *standing up*;
- *walking*.

Due di esse risultano essere statiche, ovvero *sitting* e *standing*, mentre le restanti 3 dinamiche, ovvero in movimento.

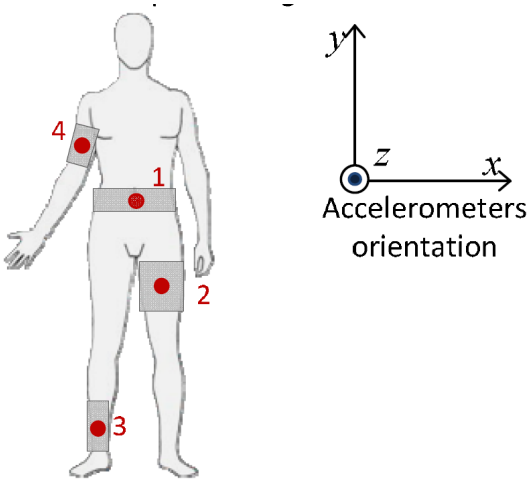


Figura 1: Posizione dei 4 accelerometri utilizzati.

2.2 Feature selection

Per poter realizzare una rete Bayesiana partendo dai dati descritti precedentemente si è deciso di considerare solamente le informazioni relative alle letture dei quattro accelerometri e, ovviamente, l'attività corrispondente. Le altre features,

ovvero sesso, peso, età, altezza e BMI, non sono state considerate in quanto risultano proprietà invarianti, dato il che ci si sta riferendo solamente a 4 soggetti.

Come si nota in Figura 2, il numero di istanze per ciascuna delle classi risulta sbilanciato, soprattutto per *sittingdown* e *standingup* che risultano essere rispettivamente il 7.14% e il 7.5% del dataset, contro il 30.56% di *sitting*. Questa distribuzione pare sensata, in quanto sedersi e alzarsi sono effettivamente azioni brevi e poco ripetute rispetto alle altre. Si nota inoltre come vi è una leggera predominanza delle classi *sitting* e *standing*, che insieme risultano essere il 59.17% del dataset. Queste ultime rappresentano azioni statiche e quindi più spesso compiute dai soggetti, specialmente dai più anziani.

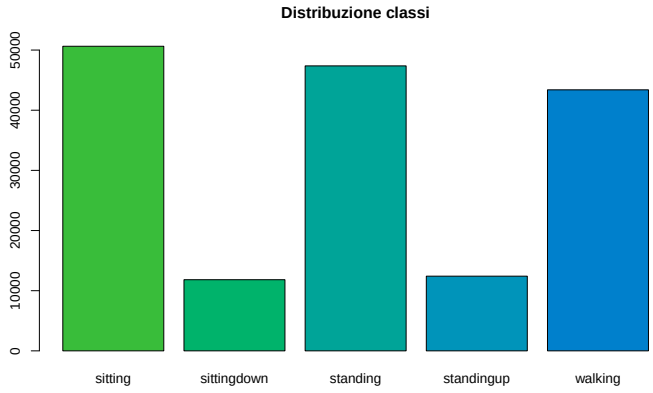


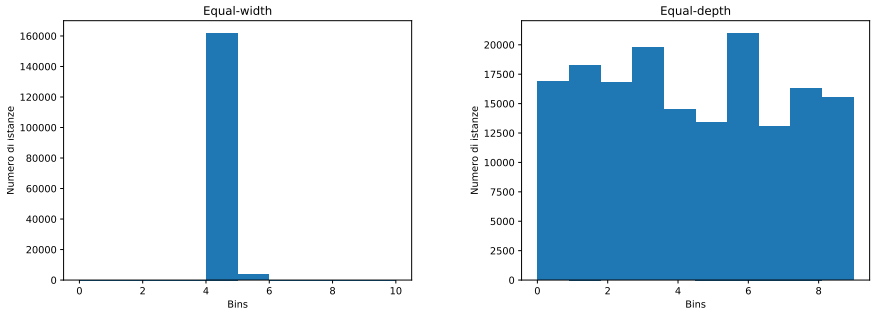
Figura 2: Distribuzione del numeri di istanze di ogni attività da classificare.

3 Reti Bayesiane

3.1 Discretizzazione

Il dataset rappresenta le letture dei vari sensori con valori continui, associati ad ognuno dei loro assi (x, y e z). Si è voluto discretizzare questi dati per verificare se questa operazione andasse ad impattare sulle performance del modello. La discretizzazione può permettere di attenuare il rumore nei dati attraverso la loro divisione in intervalli.
La strategia adottata è quella di *equal-depth* in quanto, come si evince dalla Figura 3, la distribuzione dei valori della maggior parte degli attributi si concentra in intorni molto ristretti.

Si è voluto generare una discretizzazione a 10 intervalli in quanto risulta essere un buon compromesso data la quantità di valori. Aumentando il numero di intervalli si ricadrebbe infatti in un problema computazionale dovuto al dominio delle realizzazioni, mentre con un numero minore gli intervalli risulterebbero troppo ampi, racchiudendo valori troppo distanti tra loro.



(a) (b)

Figura 3: (a) Discretizzazione *equal-width*. (b) Discretizzazione *equal-depth*.

3.2 Rete proposta

Si è voluto modellare, in primo luogo, una rete bayesiana che permetta di classificare quale delle 5 azioni si stia compiendo, basandosi sui valori di accelerazione registrati dagli accelerometri.
Le dipendenze tre le variabili sono state impostate basandoci sulle relazioni che le varie parti del corpo possiedono. Ad esempio, si è scelto di mantenere una dipendenza tra i movimenti della coscia con quelli del bacino, in quanto ragionevolmente, per la maggior parte delle attività essi si muoveranno insieme. Un'altra dipendenza impostata è stata quella tra la coscia e la caviglia perché, anche in questo caso, i movimenti si assume siano collegati nella maggior parte

delle situazioni. Si è scelto, inoltre, di rendere non direttamente dipendente agli altri sensori quello relativo ai movimenti del braccio, in quanto è logico pensare che l'arto superiore si muovi molto spesso indipendentemente da quelli inferiori. Occorre precisare che le dipendenze tra le variabili sono state impostate separatamente per ogni asse, perché si è deciso di rendere questi ultimi indipendenti, come intrinseco nella loro natura.

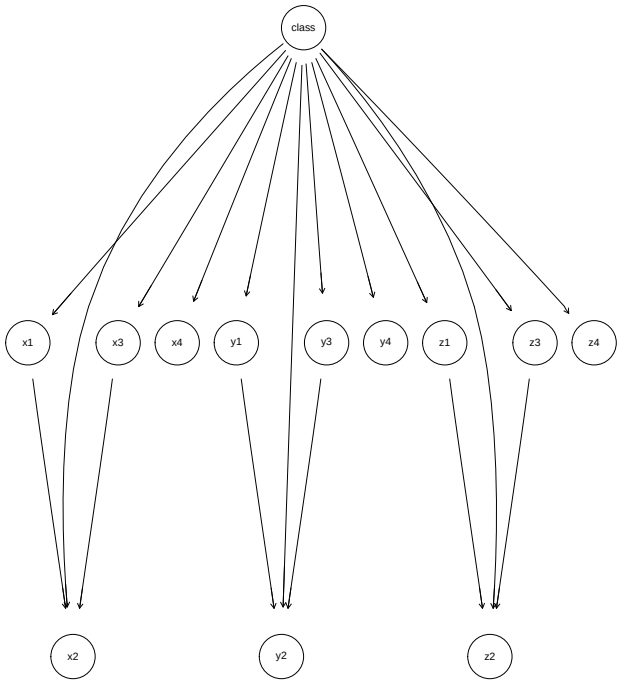


Figura 4: Rete definita sfruttando la nostra conoscenza pregressa.

3.3 Performance ottenute

Attraverso una 10-fold cross validation sono state stimate le performance del modello da noi realizzato. Per ogni fold, il training set è composto dal 90% del dataset, mentre il restante 10% rappresenta il test set. Il modello che utilizza valori continui è stato addestrato utilizzando il metodo *mle*, che si basa sulla *likelihood*, mentre per il caso discreto è stato usato *bayes* in quanto *mle* produceva valori nulli. In entrambi i casi la classificazione è stata fatta utilizzando la funzione *bayes-lw* che implementa la *Likelihood Weighting* su 500 sample.

Per il caso continuo, come si può vedere dalla Figura 5, sono state ottenute le seguenti performance:

- accuracy: 79.2%;
- precision: 68.5%;
- recall: 66.5%;
- f1measure: 67.5%.

Come si può notare dalla Tabella 1, la classe *standingup* risulta essere nella maggior parte dei casi classificata in modo errato, più precisamente, viene confusa con la classe *walking*. La classificazione migliore è ottenuta, invece, per la classe *sitting*.

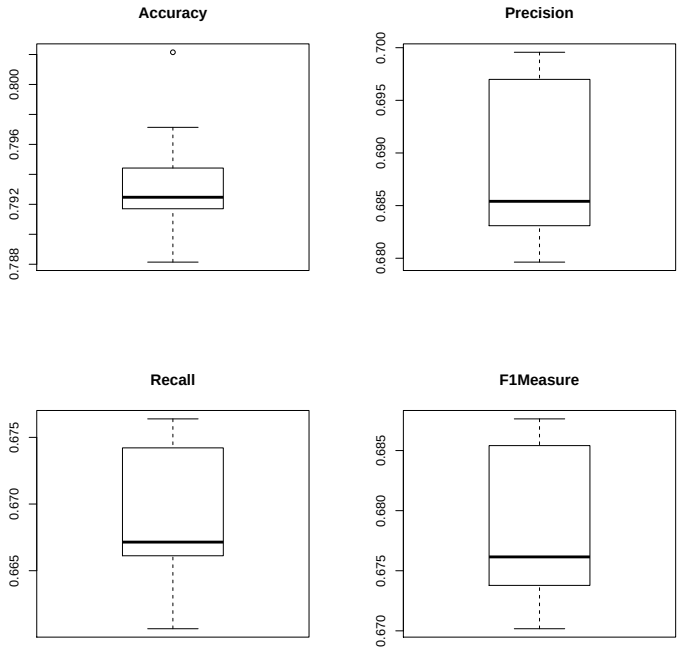


Figura 5: Performance ottenute con valori continui.

Tabella 1: Matrice di confusione di uno dei fold della rete proposta, caso continuo.

	sitting	sittingdown	standing	standingup	walking
sitting	4920	78	74	38	2
sittingdown	85	707	281	13	27
standing	42	73	4410	14	263
standingup	82	247	517	181	206
walking	111	182	837	261	2913

Per il caso discreto le performance risultano nettamente migliori rispetto al caso precedente (mostrato in Figura 5), con un aumento dell'accuratezza pari al 9%:

- accuracy: 88.2%;
- precision: 86.4%;
- recall: 88.6%;
- f1measure: 87.4%.

In Tabella 2 viene riportata una matrice di confusione relativa ad un fold. Si può notare come, in questo caso, è la classe *walking* a risultare la peggio classificata delle 5. La classificazione migliore è ottenuta, anche in questo caso, per la classe *sitting*, insieme a *sittingdown*.

Tabella 2: Matrice di confusione di uno dei fold della rete proposta, caso discreto.

	sitting	sittingdown	standing	standingup	walking
sitting	4711	0	132	2	267
sittingdown	51	1018	20	0	24
standing	114	161	3926	285	316
standingup	3	0	1	1094	135
walking	50	0	315	28	3911

Al fine di indagare se le performance dei due modelli fossero statisticamente differenti, sono stati eseguiti dei test di significatività basati sulla *paired t-student* per ogni misura di performance, impostando un livello di significatività α pari a 0.05. I risultati ottenuti "come ci si poteva aspettare" hanno mostrato che le performance medie dei due modelli sono statisticamente differenti in favore del modello che utilizza valori discreti.

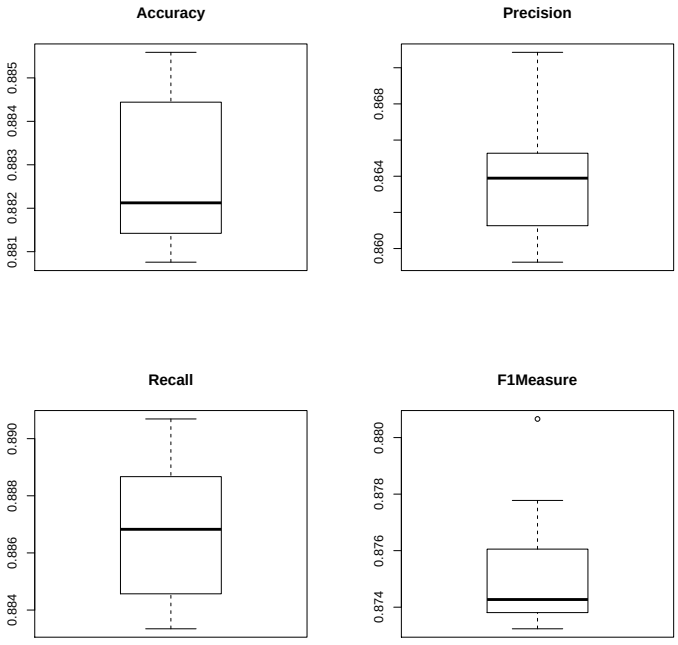


Figura 6: Performance ottenute con valori discreti.

3.4 Rete indotta

La seconda rete è stata, invece, indotta da un processo automatico fornito dalla libreria *Bnlearn* di R. In particolare si è scelto di utilizzare la funzione *tabu* che utilizza un algoritmo di apprendimento greedy basato su score, ovvero *Tabù search*. Per il caso continuo, dopo varie prove, si è scelto di utilizzare lo score *bic-g*, risultato più performante. Per il caso discreto, invece è stato utilizzato lo score *k2*. Come per il modello proposto da noi, anche in questo caso la classificazione è stata eseguita utilizzando la funzione *bayes-lw*.

3.4.1 Performance ottenute

Per il caso continuo, la rete è composta da 13 nodi e 78 archi, come si può vedere in Figura 7, le performance ottenute risultano:

- accuracy: 90.4%;
- precision: 84.9%;
- recall: 83%;
- f1measure: 84.1%.

Dalla matrice di confusione in Tabella 3 si nota come la classe classificata peggio risulti ancora una volta *standingup*, mentre quella classificata meglio rimane *sitting*.

Tabella 3: Matrice di confusione di uno dei fold della rete indotta, caso continuo.

	sitting	sittingdown	standing	standingup	walking
sitting	4990	73	0	49	0
sittingdown	64	891	111	24	23
standing	0	101	4614	21	66
standingup	17	307	158	636	115
walking	0	121	310	38	3835

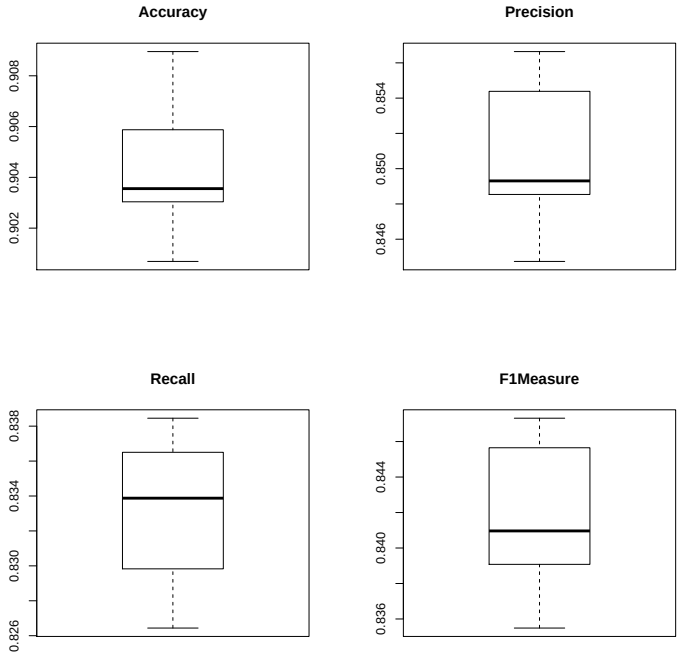


Figura 7: Performance ottenute dalla rete indotta con valori continui.

Come per la rete da noi proposta, il caso discreto ottiene performance superiori rispetto a quello continuo, inoltre il numero di archi si riduce notevolmente arrivando a 45, ottenendo un modello più semplice, il quale riportato in Figura 8. Al contempo in Figura 9 sono riportati i boxplot riguardanti ciascuna delle misure di performance seguenti:

- accuracy: 92.7%;
- precision: 89.3%;
- recall: 92.4%;
- f1measure: 90.8%.

Dalla matrice di confusione in Tabella 4 si evince come la classificazione, in questo caso, sia buona per tutte le classi, senza uno sbilanciamento in favore di *sitting* a differenza degli altri modelli. Anche in questo caso per valutare se le performance dei due modelli fossero statisticamente differenti, sono stati eseguiti dei test di significatività basati sulla *paired t-student* per ogni misura di performance, impostando un livello di significatività α pari a 0.05. I risultati hanno mostrato che le performance medie dei due modelli sono statisticamente differenti in favore, ancora una volta, del modello discreto.

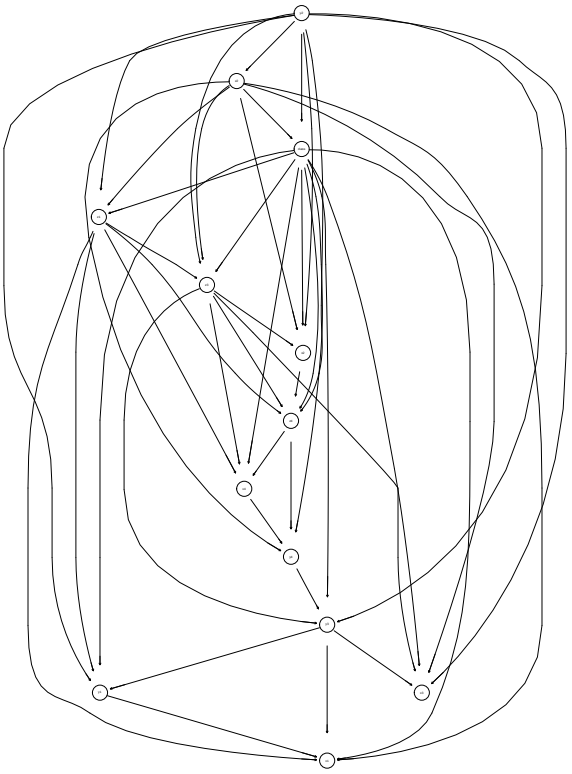


Figura 8: Rete indotta dai dati discretizzati.

Tabella 4: Matrice di confusione di un fold della rete indotta, caso discreto.

	sitting	sittingdown	standing	standingup	walking
sitting	4791	79	75	88	79
sittingdown	11	1040	55	5	2
standing	58	68	4455	114	107
standingup	1	2	94	1103	33
walking	50	60	86	92	4016

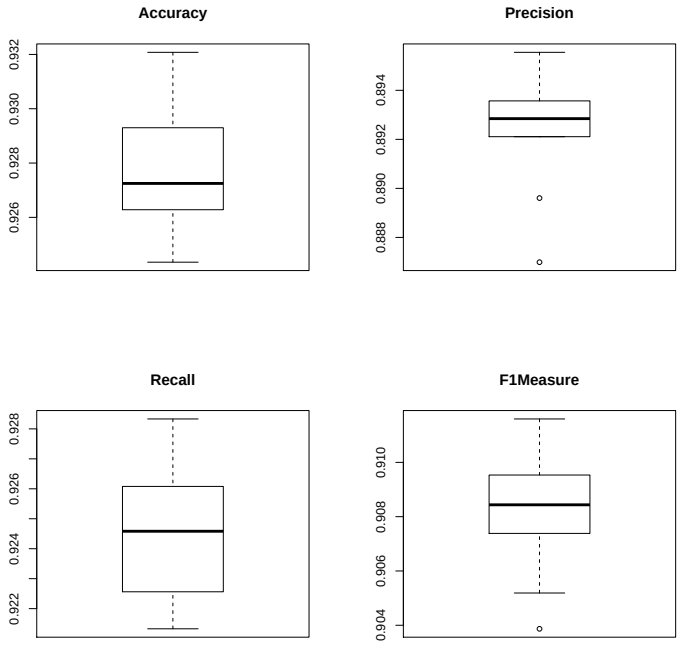


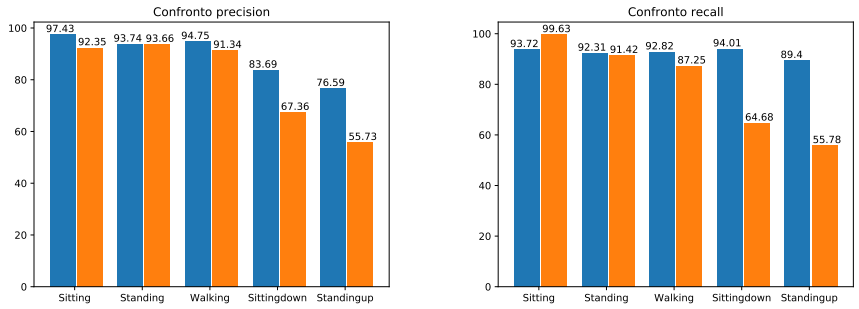
Figura 9: Performance ottenute dalla rete indotta con valori discreti.

3.4.2 Oversampling

Tramite la funzione *Smote* (*Synthetic Minority Over-sampling Technique*) del tool *Weka*, la cui documentazione è disponibile qui, abbiamo realizzato *oversampling*, prima di entrambe le classi *sittingdown* e *standingup*, poi solamente della seconda. La funzione *Smote* è una tecnica supervisionata di *random oversampling* che permette di generare campioni sintetici per una classe di minoranza all'interno di un dataset. La generazione segue questo procedimento: per ogni istanza della classe minoritaria vengono identificati i primi k sample geometricamente più vicini ad essa (nel nostro caso 5 vicini), dopodiché, preso un sottoinsieme d'essi, per ogni elemento di questo sottoinsieme viene generato un nuovo *sample* i cui attributi sono calcolati come la media tra i suoi attributi e quelli dell'istanza. Grazie a questo metodo è stato possibile raddoppiare gli elementi delle due classi di minoranza, riducendo così la differenza rispetto le altre 3 classi.
Nel primo approccio, effettuando l'*oversampling* su entrambe le classi di minoranza, si può notare dalla Tabella 5, come le due classi di minoranza *sittingdown* e *standingup* si dimostrino quelle classificate peggiormente dalla rete come nel caso originario. Andando a verificare le performance di *precision* e *recall* per ogni classe, prima e dopo quest'operazione, si nota un calo significativo di entrambe, come si può vedere in Figura 10). Vi è infatti un calo del 16.33% e del 20.86% nella *precision* per *sittingdown* e *standingup* rispettivamente, mentre un calo del 29.33% e del 33.62% nella *recall*. Questo calo di performance è dato probabilmente dal fatto che, identificando due attività molto simili tra loro, quali alzarsi e sedersi, l'introduzione di molti esempi sintetici per entrambe le classi abbia aumentato l'incertezza del modello invece di diminuirla.

Tabella 5: Matrice di confusione della rete indotta dopo *oversampling* sulle due classi di minoranza *sittingdown* e *standingup*.

	sitting	sittingdown	standing	standingup	walking
sitting	5029	2	8	5	1
sittingdown	50	1528	75	663	62
standing	153	78	4366	115	71
standingup	85	594	167	1377	233
walking	117	40	43	348	3778



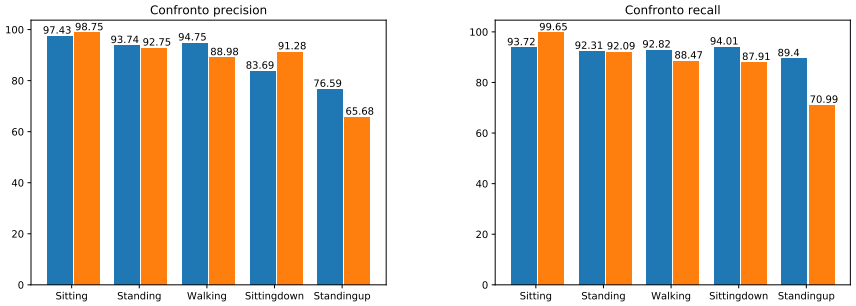
(a) (b)

Figura 10: Misure di *precision* (a) e di *recall* (b) per ogni classe prima e dopo *oversampling* sulle classi *sittingdown* e *standingup*. In azzurro le performance della rete indotta prima di *oversampling*, mentre in arancio le performance calcolate dopo *oversampling*.

Nel secondo caso, ovvero eseguendo *oversampling* solamente alla classe *sittingdown*, come si evince dalla tabella 6, vi è un miglioramento della classificazione per questa classe, ma per *standingup* le performance rimangono scarse. Dai grafici riportati in Figura 11 si può infatti notare come la *precision* di *sittingdown* sia migliorata del 7.59% rispetto al dataset originale, mentre per *standingup* si registra un peggioramento del 10.91%. Per quanto riguarda la *recall*, vengono ottenuti risultati peggiori rispetto all'utilizzao del dataset di partenza, ma meno significativi rispetto al primo caso. L'accuratezza, utilizzando questi 2 approcci, risulta quindi minore rispetto al caso iniziale, attestandosi rispettivamente all'84.7% e al 91.3%. L'utilizzo dell'*oversampling*, in questo caso, è risultato poco efficace e quindi da evitare per entrambe le classi minori. Per ottenere un bilanciamento, l'unica via rimane quella di eseguire nuove rilevazioni sul campo in quanto, dato il ridotto numero di istanze del dataset, non è possibile effettuare con successo nè l'operazione di oversampling, nè la sua "complementare" (i.e., undersampling)

Tabella 6: Matrice di confusione di uno dei fold della rete indotta dopo *oversampling* sulla classe di minoranza *sittingdown*.

	sitting	sittingdown	standing	standingup	walking
sitting	5091	7	2	2	4
sittingdown	22	2092	107	30	130
standing	7	80	4329	140	148
standingup	3	8	185	836	195
walking	21	95	85	280	3847



(a) (b)

Figura 11: Misure di *precision* (a) e di *recall* (b) per ogni classe prima e dopo *oversampling* sulla classe *sittingdown*. In azzurro le performance della rete indotta prima di *oversampling*, mentre in arancio le performance calcolate dopo *oversampling*.

3.5 Modifiche alla rete indotta

3.5.1 Unione della rete indotta con quella proposta

Si è voluto combinare il modello proposto da noi con quello risultato migliore tra quelli esaminati precedentemente, ovvero quello indotto che utilizza dati discreti. L'obiettivo è stato quello di creare una rete che mantenga performance alte pur non essendo eccessivamente complessa come quella indotta automaticamente mediante l'algoritmo *Tabu*. In particolare, si è voluta modificare quest'ultima introducendo l'indipendenza tra i 3 assi x, y e z utilizzata nel modello da noi proposto inizialmente. Il risultato di questo procedimento ha portato alla realizzazione della rete mostrata in Figura 12 che, rispetto a quella originale, possiede un numero molto più ristretto di archi, e quindi di dipendenze tra le feature. Più precisamente, quest'ultimo è diminuito del 53.3%, passando da 45 a 21.
Rispetto alla rete proposta da noi, la classe ora dipende da x2 e y2, mentre ogni altra variabile riferita alle accelerazioni dipende da essa. L'accuratezza di questo nuovo modello si attesta intorno al 90.9%, contro il 92.7% della rete indotta originale. Si è deciso di operare test di significatività basati sulla *paired t-student* per ogni misura di performance, impostando un livello di significatività α pari a 0.05, la differenza tra i due modelli si è dimostrata statisticamente significativa, ma si può comunque ritenere il nuovo modello una valida alternativa perché riduce notevolmente la complessità del modello.

3.5.2 Eliminazione del sensore 4

Per ultimo, abbiamo voluto verificare il cambiamento delle performance andando a rimuovere i dati relativi al sensore 4, ovvero quello posizionato sul braccio destro dei soggetti. Questa modifica è stata apportata come da consiglio di Ugulino *et al.* in [3], dove tali dati sono stati rimossi dal dataset dopo una feature selection.
L'accuratezza della rete senza le informazioni del sensore 4 si attesta all'89% e quindi ad un peggioramento del 1.9% rispetto alla rete completa. La rete completa rimane quindi da preferirsi, in quanto, considerato il costo ridotto degli accelerometri e dei microcontrollori necessari all'acquisizione dei dati, si riescono ad ottenere performance migliori. Si potrebbero incrementare ulteriormente le performance andando a posizionare il sensore del braccio in zone del corpo più significative, come per esempio quelle proposte in [4].

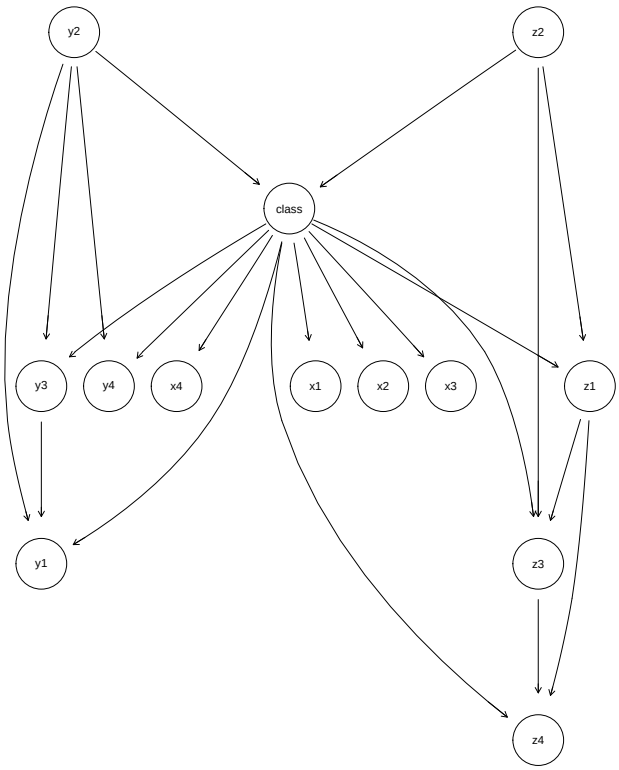


Figura 12: Rete indotta modificata.

4 Conclusioni

In questo lavoro si è voluto realizzare un modello di apprendimento bayesiano che potesse riconoscere l'attività svolta da un individuo conoscendo i valori registrati da accelerometri posizionati su varie parti del suo corpo. Come prima operazione, si è deciso di rimuovere dal dataset gli attributi relativi ad età, sesso, peso e BMI dei vari soggetti in quanto ritenuti ininfluenti per la classificazione essendo costanti per ogni persona. Successivamente, si è deciso di discretizzare secondo il metodo di *equal-depth* le letture dei vari sensori per verificare se questa operazione influisse sulle performance.
Il primo modello che si è deciso di realizzare considera le relazioni tra le varie parti del corpo su cui sono posizionati i sensori e l'indipendenza tra gli assi x, y e z. In seguito, si sono misurate le performance ottenute da questo modello sia con valori discreti che continui. Dai risultati ottenuti si è potuto constatare che utilizzando valori discreti il modello risulti essere più efficiente ed accurato rispetto alla controparte continua. Successivamente, si è voluto indurre un modello utilizzando l'algoritmo di apprendimento automatico *Tabu search*, anche in questo caso, utilizzando sia il dataset continuo che quello discreto. In entrambi i modelli indotti le performance risultano migliori rispetto a quelle ottenute dal modello da noi proposto e, nuovamente le performance migliori sono ottenute dal modello indotto dai dati discretizzati, che riesce ad ottenere il 92.7% di accuratezza.
Notando che le istanze delle varie classi risultano sbilanciate, si è deciso di eseguire un'operazione di *oversampling* per ripopolare le classi minorate in base a due approcci differenti:

- ripopolando sia la classe *sittingdown* che quella *standingup*, essendo entrambe carenti di istanze;
- ripopolando solo la classe *sittingdown*, essendo quella minoritaria.

In entrambi i casi si sono riscontrati cali nelle prestazioni del modello, specialmente eseguendo l'*oversampling* ad entrambe le classi. Nel secondo caso, però, si è ottenuta una precisione più alta nella classificazione della classe *sittingdown* rispetto al caso originario e un calo di accuratezza solamente dell'1.4%. L'*oversampling* è risultato quindi inefficace in questo caso, per ciò si è voluto mantenere il dataset sbilanciato.
Infine, dato che la struttura della rete indotta risultava essere elaborata, avendo 13 nodi collegati da un totale di ben 45 archi, si è deciso di realizzare un terzo modello partendo da quello indotto dai dati discreti, ovvero quello risultato migliore. Per semplificare quest'ultimo, sono state imposte le indipendenze tra gli assi x, y e z indotte nel modello proposto inizialmente da noi. Grazie a questa operazione, la rete semplificata ottenuta è composta dal 53.3% di archi in meno rispetto a quella indotta, ovvero solamente 21. Per quanto riguarda le performance, tale modello riesce ad ottenere il 90.9% di accuratezza, risultando quindi un buon compromesso tra complessità strutturale e performance.
Per ultimo, si è deciso di rimuovere i valori relativi al sensore posizionato sul

braccio del soggetto, in quanto risultavano poco influenti nelle predizioni del modello, come riportato in [3]. L'accuratezza ottenuta da quest'ultima rete risulta peggiorata solamente dell'1.9% rispetto a quella completa. Da questo dato si può confermare che il sensore sul braccio sia poco significativo ai fini della classificazione e che, probabilmente, si potrebbe sacrificare tale posizione del sensore in favore di altre [4] in modo da ottenere un incremento maggiore delle performance.

Riferimenti bibliografici

[1] Wallace Ugulino et al. “Wearable computing: Accelerometers’ data classification of body postures and movements”. In: *Brazilian Symposium on Artificial Intelligence*. Springer. 2012, pp. 52–61.

[2] *Libreria bnlearn*. URL: http://www.bnlearn.com/ (visitato il 25/06/2019).

[3] Wallace Ugulino et al. “Human Activity Recognition using On-body Sensing”. In: *Proceedings of III Symposium of the Brazilian Institute for Web Science Research (WebScience)*. Vol. 1. 2012.

[4] Louis Atallah et al. “Sensor positioning for activity recognition using wearable accelerometers”. In: *IEEE transactions on biomedical circuits and systems* 5.4 (2011), pp. 320–329.