

# Riconoscimento di Attività Umane con Perceptron Intelligenza Artificiale

Lapo BARTOLACCI

January 10, 2020

## 1 Introduzione

I moderni smartphone hanno la possibilità di sfruttare i molti sensori che hanno integrati. Dato il ruolo sempre più centrale di questi dispositivi nella vita di tutti i giorni, si prevede che questi dispositivi terranno continuamente traccia delle nostre attività, imparando da esse per aiutarci a prendere decisioni migliori. Il Riconoscimento di Attività ha lo scopo di identificare le azioni eseguite da un essere umano date delle informazioni percepite da esso o dall'ambiente. In questo elaborato vengono utilizzate le informazioni ricavate dai sensori presenti negli smartphone per classificare un insieme di attività (standing, walking, laying, walking upstairs e walking downstairs) utilizzando l'algoritmo di apprendimento supervisionato Perceptron.

## 2 Metodo

### 2.1 Data Set

Il data set utilizzato è disponibile qui: <http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones>

I dati sono stati raccolti con un gruppo di 30 volontari di età compresa tra i 19 e i 48 anni. Ogni persona ha eseguito le 6 attività con uno smartphone alla vita, registrando i dati del giroscopio e dell'accelerometro ad una frequenza di 50 Hz. I dati sono stati partizionati in modo casuale in due insiemi, il 70% è stato utilizzato per il training e il 30% per il test.

Ulteriori informazioni su come sono stati processati i dati sono disponibili nella pagina del data set.

## 2.2 Implementazione

Il linguaggio di programmazione utilizzato è il Python, in particolare è stata utilizzata l'implementazione dell'algoritmo Perceptron fornita dalla libreria **Scikit-learn**. Dopo aver creato un oggetto perceptron la funzione fit esegue il training con 7352 istanze ognuna con 561 features.

```
ppn = Perceptron(max_iter=1000, eta0=0.1, random_state=0)
ppn.fit(features_train, target_train)
```

La funzione predict viene utilizzata per testare il perceptron appena allenato con 2947 istanze di test.

```
target_pred = ppn.predict(features_test)
```

## 3 Risultati

		Predette						
		Walking	Upstairs	Downstairs	Standing	Sitting	Laying	Recall %
Effective	Walking	<b>488</b>	1	5	2	0	0	98.39
	Upstairs	9	<b>454</b>	7	1	0	0	96.39
	Downstairs	2	3	<b>411</b>	4	0	0	97.86
	Standing	0	0	0	<b>524</b>	8	0	98.5
	Sitting	0	2	0	85	<b>404</b>	0	82.28
	Laying	0	0	0	23	0	<b>514</b>	95.72
	Precision %	97.8	98.7	97.16	82.0	98.06	100.0	<b>94.84</b>

Table 1: Matrice di Confusione. Le righe rappresentano le classi effettive e le colonne le classi predette. La diagonale mostra le istanze classificate correttamente.

I risultati della classificazione di Perceptron per i dati di test sono riportati tramite matrice di confusione nella Table 1, dove sono anche mostrati l'accuratezza complessiva, la precisione e il recall. Sono state valutate 2947 istanze di test con circa lo stesso numero di istanze per classe. La maggior parte delle predizioni false si hanno per le classi sitting e laying a differenza dei risultati riportati da Anguita et al. [1] nella Figure 1, dove le attività statiche hanno performato meglio rispetto a quelle dinamiche.

Method	MC-SVM							MC-HF-SVM $k = 8$ bits						
Activity	Walking	Upstairs	Downstairs	Standing	Sitting	Laying	Recall %	Walking	Upstairs	Downstairs	Standing	Sitting	Laying	Recall %
Walking	<b>109</b>	0	5	0	0	0	95.6	<b>109</b>	2	3	0	0	0	95.6
Upstairs	1	<b>95</b>	40	0	0	0	69.8	1	<b>98</b>	37	0	0	0	72.1
Downstairs	15	9	<b>119</b>	0	0	0	83.2	15	14	<b>114</b>	0	0	0	79.7
Standing	0	5	0	<b>132</b>	5	0	93.0	0	5	0	<b>131</b>	6	0	92.2
Sitting	0	0	0	4	<b>108</b>	0	96.4	0	1	0	3	<b>108</b>	0	96.4
Laying	0	0	0	0	0	<b>142</b>	100	0	0	0	0	0	<b>142</b>	100
Precision %	87.2	87.2	72.6	97.1	95.6	100	<b>89.3</b>	87.2	81.7	74.0	97.8	94.7	100	<b>89.0</b>

Figure 1: Matrice di confusione dei risultati riportati da Anguita et al. [1] utilizzando due versioni di un classificatore diverso.

## References

- [1] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz. Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In *International workshop on ambient assisted living*, pages 216–223. Springer, 2012.