**Human Activity Recognition Using Smartphones**

**Reconhecimento de Padrões**

**2016/2017**

Gabriel Angel Amarista Rodrigues 2016211454

Maria José Mateus Branco 2013134952

**Introdução**

Atualmente, os smartphones são utilizados em diversas atividades do quotidiano, nomeadamente para monitorização da atividade humana. Assim, foi desenvolvido um trabalho com o objetivo de desenvolver classificadores para reconhecimento da atividade humana.

O projeto foi dividido em duas partes, na primeira é considerado apenas o caso binário (walking e not walking) e na segunda é estudado um caso com seis classes (walking, walking upstairs, walking downstairs, sitting, standing e laying).

**Procedimento**

Inicialmente, os dados foram carregados, estando divididos em dois conjuntos diferentes, um conjunto de treino e outro de teste. A cada um destes conjuntos corresponde uma matriz que contém todas as *features* (*X\_train* e *X\_test*) e um vetor com a classe correspondente a cada uma das instâncias (*y\_train* e *y\_test*). Após este passo, foi feito o *scaling* dos dados, de forma a que estes estejam todos contidos no intervalo de valores entre -1 e 1.

Uma vez que existem 561 *features* disponíveis para analisar, e que existe a possibilidade de algumas delas apresentarem informação insignificante para a classificação ou terem informação redundante, procedeu-se à redução da dimensionalidade dos dados, usando as técnicas de PCA e LDA, para além destas técnicas foi também testado o método de Kruskal Wallis para selecionar as *features* com maior importância na classificação. Ao longo desta primeira fase do projeto foram usadas 3 *features* para a realização de todos os testes. Todas as funções que foram usadas para redução de dimensionalidade, como PCA, Kruskal Wallis, encontram-se desenvolvidas no ficheiro *FeatureProcess.m.*

Antes de proceder à redução do número de *features*, verificou-se a existência de características muito correlacionadas e manteve-se apenas uma delas para proceder às análises seguintes.

Após a redução da dimensionalidade, foram testados apenas classificadores simples, o Fisher LDA, que é bastante usado em reconhecimento de padrões pois representa uma combinação linear de características originais que permitem realizar uma separação máxima entre duas populações, foram também usados dois classificadores baseados nas distâncias mínimas, sendo que é no conjunto de treino é definido um ponto médio que permite caracterizar cada classe, sendo que um novo ponto é classificado como a classe que apresenta o ponto mais próximo deste, as distâncias usadas foram a Euclidiana e a Mahalanobis. Assim, foram feitos diferentes testes para verificar qual a combinação que apresenta uma melhor performance no caso binário. No cenário *multiclass*, apenas foram desenvolvidos e testados classificadores baseados nas distâncias mínimas, uma vez que o não foi desenvolvido nenhum classificador do tipo Fisher LDA para distinguir as seis classes diferentes. As funções que permitem desenvolver estes classificadores, encontram-se no programa *Classifier.m*.

No desenvolvimento dos classificadores, foram feitos vários testes, dividindo o conjunto de treino em várias partes e fazendo validação cruzada, com o objetivo de avaliar a capacidade de generalização dos modelos, assim, para cada subconjunto foi obtida uma matriz de confusão, a soma destas permitiu obter uma matriz de confusão que permitiu verificar os classificadores com maior capacidade de generalização.

Após a escolha dos classificadores com melhor *performance*, estes foram aplicados ao conjunto de teste e analisados os resultados.

Para proceder à redução de dimensionalidade, criação e teste dos classificadores, as funções necessárias são chamadas no programa *testing\_script.m*.

Após a primeira fase do trabalho onde apenas foram testados classificadores simples, procedeu-se ao desenvolvimento de classificadores mais complexos. Desta forma decidiu-se implementar mais quatro classificadores, nomeadamente *Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Naive Bayes* e um classificador híbrido que implementa os dois últimos juntamente com o que se baseia na distância Euclidiana mínima e com o *Fisher One vs All*. Estes foram aplicados aos dois cenários combinando-os com as diferentes formas de seleção de dados usadas na primeira parte.

Foi escolhido o modelo *SVM* para testar o seu comportamento devido ao facto deste procurar a melhor forma de classificar uma dada instância na presença de *outliers* e de fornecer uma separação linear entre as classes, apesar destas vantagens, este é um processo que necessita de muito tempo de treino em conjuntos de dados muito grandes, pois primeiro classifica as classes, encontra uma linha de separação entre estas (hiperplano) que pretende maximizar a distância entre os pontos mais próximos em relação a cada classe.

O classificador *KNN* procura K elementos do conjunto de treino que estejam mais próximos de cada elemento de teste, sendo atribuída a este a classe com maior frequência.

O classificador *Naive* *Bayes* baseia-se na probabilidade de acontecimento de eventos, assumindo que dois eventos são independentes, ou seja, a existência de um tipo de característica não depende da existência de outras. A partir de um conjunto de treino, o algoritmo calcula a probabilidade de um dado D pertencer à classe C.

No cenário com seis classes, foi também testada uma abordagem diferente das outras, classificando os dados inicialmente em duas grandes classes (iguais às do caso binário) e procedendo depois à classificação da instância numa das três subclasses possíveis dentro da classe principal. Esta abordagem permite diminuir o erro de classificação uma vez que é possível classificar com grande exatidão uma instância quando apenas pode pertencer a uma das duas grandes classes (“Walking” ou “Not Walking”). Assim, após a determinação do grupo no qual se insere um determinado dado, a sua classificação só pode variar entre três classes em vez de seis, o que permite melhorar a *accuracy* do classificador.

Procedeu-se então à avaliação da capacidade de generalização de cada classificador através da técnica de validação cruzada selecionando apenas três features, como foi feito na primeira fase para os classificadores mais simples. Posteriormente, foram testados os que apresentaram melhores resultados nesta fase com os dados de teste e de seguida foi avaliada a variação da *performance* do melhor de acordo com o número de atributos usados.

**Resultados**

**Cenário Binário (Conjunto de Treino *cross validation*)**

**Kruskal Wallis + Fisher LDA**

|  |  |  |
| --- | --- | --- |
|  | Walking | Not Walking |
| Walking | 3490 | 0 |
| Not Walking | 30 | 3840 |

Accuracy=0.996

Precisão=0.9915

Especificidade=0.992

Sensibilidade/Recall=1.000

**Kruskal Wallis + Minimun Euclidean Distance**

|  |  |  |
| --- | --- | --- |
|  | Walking | Not Walking |
| Walking | 3260 | 0 |
| Not Walking | 10 | 4090 |

Accuracy=0.999

Precisão=0.997

Especificidade=0.998

Sensibilidade/Recall=1.000

**Kruskal Wallis + Minimun** **Mahalanobis Distance**

|  |  |  |
| --- | --- | --- |
|  | Walking | Not Walking |
| Walking | 3390 | 0 |
| Not Walking | 20 | 3950 |

Accuracy=0.997

Precisão=0.994

Especificidade=0.995

Sensibilidade/Recall=1.000

**PCA + Fisher LDA**

|  |  |  |
| --- | --- | --- |
|  | Walking | Not Walking |
| Walking | 3540 | 0 |
| Not Walking | 0 | 3820 |

Accuracy=1.000

Precisão=1.000

Especificidade=1.000

Sensibilidade/Recall=1.000

**PCA + Minimun Euclidean Distance**

|  |  |  |
| --- | --- | --- |
|  | Walking | Not Walking |
| Walking | 3650 | 0 |
| Not Walking | 10 | 3700 |

Accuracy=0.999

Precisão=0.996

Especificidade=0.997

Sensibilidade/Recall=1.000

**PCA + Minimun Mahalanobis Distance**

|  |  |  |
| --- | --- | --- |
|  | Walking | Not Walking |
| Walking | 3310 | 0 |
| Not Walking | 10 | 4040 |

Accuracy=0.999

Precisão=0.997

Especificidade=0.998

Sensibilidade/Recall=1.000

**LDA + Fisher LDA**

|  |  |  |
| --- | --- | --- |
|  | Walking | Not Walking |
| Walking | 3300 | 0 |
| Not Walking | 0 | 4060 |

Accuracy=1.000

Precisão=1.000

Especificidade=1.000

Sensibilidade/Recall=1.000

**LDA + Minimun Euclidean Distance**

|  |  |  |
| --- | --- | --- |
|  | Walking | Not Walking |
| Walking | 1950 | 1310 |
| Not Walking | 50 | 4050 |

Accuracy=0.815

Precisão=0.975

Especificidade=0.988

Sensibilidade/Recall=0.598

**LDA + Minimun Mahalanobis Distance**

|  |  |  |
| --- | --- | --- |
|  | Walking | Not Walking |
| Walking | 1710 | 1500 |
| Not Walking | 500 | 3650 |

Accuracy=0.728

Precisão=0.774

Especificidade=0.880

Sensibilidade/Recall=0.533

**LDA + SVM**

|  |  |  |
| --- | --- | --- |
|  | Walking | Not Walking |
| Walking | 3220 | 0 |
| Not Walking | 0 | 4140 |

Accuracy=1.000

Precisão=1.000

Especificidade=1.000

Sensibilidade/Recall=1.000

**Cenário Binário (Conjunto de Teste)**

Após a verificação dos resultados obtidos aplicando os classificadores ao conjunto de treino foi possível escolher os classificadores desenvolvidos através das combinações PCA+Fisher LDA, PCA + Minimun Mahalanobis Distance e Kruskal Wallis + Minimun Euclidean Distance para proceder à sua aplicação ao conjunto de teste e assim verificar a sua *performance,* uma vez que foram os que apresentaram melhores resultados na fase de treino*.* Após a eliminação das *features* correlacionadas foram testados os classificadores para os dados de treino e foi obtido um classificador com uma *performance* superior aos selecionados anteriormente (classificador baseado na combinação LDA + Fisher LDA).

Na segunda fase, quando aplicado o classificador *SVM* em conjunto com o método *LDA* para selecionar *features* foi obtida uma *performance* ótima, conseguindo classificar corretamente todos os dados de treino. Assim este método foi também aplicado ao conjunto de teste como forma de avaliar a sua capacidade de classificação destes dados.

Como foram encontrados classificadores com um desempenho perfeito não foram testadas as outras combinações possíveis no treino e no teste.

**PCA+Fisher**

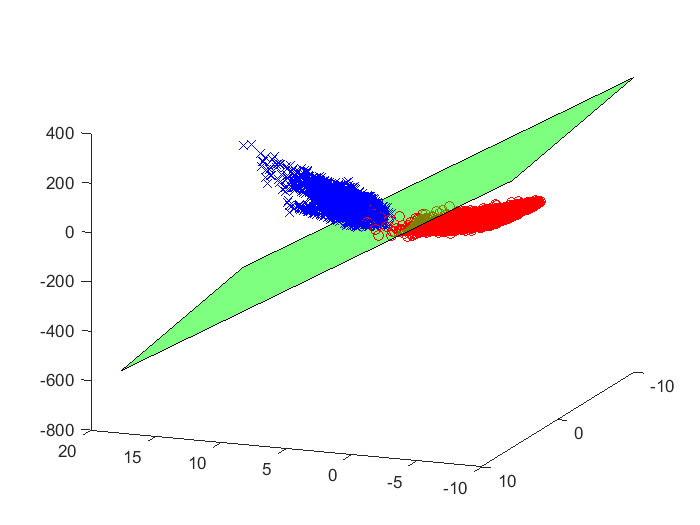


Figura 1- Decison boundary

|  |  |  |
| --- | --- | --- |
|  | Walking | Not Walking |
| Walking | 1387 | 0 |
| Not Walking | 5 | 1555 |

Accuracy=0.998

Precisão=0.996

Especificidade=0.997

Sensibilidade/Recall=1.000

**PCA + Minimun Mahalanobis Distance**

|  |  |  |
| --- | --- | --- |
|  | Walking | Not Walking |
| Walking | 1387 | 0 |
| Not Walking | 9 | 1551 |

Accuracy=0.997

Precisão=0.994

Especificidade=0.994

Sensibilidade/Recall=1.000

**Kruskal Wallis + Minimun Euclidean Distance**

|  |  |  |
| --- | --- | --- |
|  | Walking | Not Walking |
| Walking | 1387 | 0 |
| Not Walking | 6 | 1554 |

Accuracy=0.998

Precisão=0.996

Especificidade=0.996

Sensibilidade/Recall=1.000

**LDA + Fisher LDA**

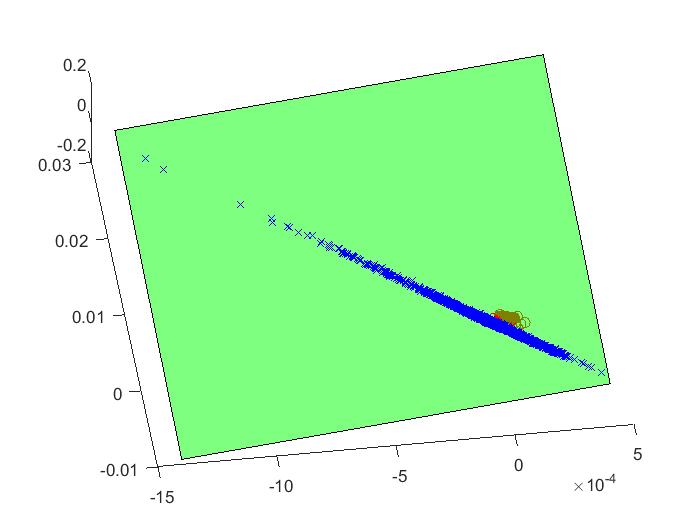


Figura 2- Decision boundary

|  |  |  |
| --- | --- | --- |
|  | Walking | Not Walking |
| Walking | 1387 | 0 |
| Not Walking | 0 | 1560 |

Accuracy=1.000

Precisão=1.000

Especificidade=1.000

Sensibilidade/Recall=1.000

**LDA + SVM**

|  |  |  |
| --- | --- | --- |
|  | Walking | Not Walking |
| Walking | 1387 | 0 |
| Not Walking | 0 | 1560 |

Accuracy=1.000

Precisão=1.000

Especificidade=1.000

Sensibilidade/Recall=1.000

**Cenário com 6 classes (Conjunto de Treino *cross validation*)**

**Kruskal Wallis + Minimun Euclidean Distance**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Walking | Walking Upstairs | Walking Downstairs | Sitting | Standing | Laying |
| Walking | 930 | 250 | 100 | 0 | 0 | 0 |
| Walking Upstairs | 200 | 780 | 150 | 0 | 0 | 0 |
| Walking Downstairs | 100 | 130 | 630 | 0 | 0 | 0 |
| Sitting | 0 | 0 | 0 | 1060 | 190 | 10 |
| Standing | 0 | 0 | 0 | 1030 | 310 | 10 |
| Laying | 0 | 10 | 0 | 1190 | 280 | 0 |

Accuracy=0.505

**Kruskal Wallis + Minimun Mahalanobis Distance**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Walking | Walking Upstairs | Walking Downstairs | Sitting | Standing | Laying |
| Walking | 510 | 350 | 270 | 0 | 0 | 0 |
| Walking Upstairs | 380 | 590 | 180 | 0 | 0 | 0 |
| Walking Downstairs | 120 | 80 | 1050 | 0 | 0 | 0 |
| Sitting | 0 | 0 | 0 | 1070 | 210 | 10 |
| Standing | 0 | 20 | 0 | 920 | 350 | 30 |
| Laying | 0 | 0 | 0 | 980 | 220 | 20 |

Accuracy=0.488

**PCA + Minimun Euclidean Distance**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Walking | Walking Upstairs | Walking Downstairs | Sitting | Standing | Laying |
| Walking | 580 | 270 | 280 | 0 | 0 | 0 |
| Walking Upstairs | 160 | 770 | 200 | 0 | 0 | 0 |
| Walking Downstairs | 240 | 100 | 750 | 0 | 0 | 0 |
| Sitting | 0 | 0 | 0 | 800 | 430 | 50 |
| Standing | 0 | 0 | 0 | 520 | 750 | 0 |
| Laying | 0 | 10 | 0 | 10 | 0 | 1440 |

Accuracy=0.692

**PCA + Minimun Mahalanobis Distance**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Walking | Walking Upstairs | Walking Downstairs | Sitting | Standing | Laying |
| Walking | 430 | 630 | 190 | 0 | 0 | 0 |
| Walking Upstairs | 190 | 620 | 170 | 0 | 0 | 0 |
| Walking Downstairs | 260 | 110 | 650 | 0 | 0 | 0 |
| Sitting | 0 | 0 | 0 | 820 | 440 | 100 |
| Standing | 0 | 0 | 0 | 840 | 640 | 0 |
| Laying | 0 | 30 | 0 | 10 | 130 | 1100 |

Accuracy=0.579

**LDA + Minimun Euclidean Distance**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Walking | Walking Upstairs | Walking Downstairs | Sitting | Standing | Laying |
| Walking | 490 | 350 | 260 | 0 | 0 | 0 |
| Walking Upstairs | 150 | 840 | 150 | 0 | 0 | 0 |
| Walking Downstairs | 70 | 140 | 630 | 0 | 0 | 0 |
| Sitting | 0 | 10 | 0 | 160 | 910 | 220 |
| Standing | 0 | 0 | 0 | 320 | 950 | 190 |
| Laying | 0 | 30 | 0 | 350 | 900 | 240 |

Accuracy=0.450

**Kruskal Wallis + SVM**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Walking | Walking Upstairs | Walking Downstairs | Sitting | Standing | Laying |
| Walking | 7 | 81 | 0 | 428 | 122 | 482 |
| Walking Upstairs | 0 | 175 | 0 | 320 | 5 | 690 |
| Walking Downstairs | 67 | 147 | 21 | 337 | 100 | 268 |
| Sitting | 0 | 17 | 0 | 580 | 204 | 389 |
| Standing | 0 | 11 | 0 | 704 | 250 | 455 |
| Laying | 0 | 69 | 0 | 669 | 150 | 612 |

Accuracy=0.224

**Kruskal Wallis + Naive Bayes**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Walking | Walking Upstairs | Walking Downstairs | Sitting | Standing | Laying |
| Walking | 910 | 200 | 90 | 0 | 0 | 0 |
| Walking Upstairs | 220 | 740 | 80 | 0 | 0 | 0 |
| Walking Downstairs | 100 | 180 | 670 | 0 | 0 | 0 |
| Sitting | 0 | 0 | 0 | 90 | 1140 | 10 |
| Standing | 10 | 0 | 0 | 60 | 1510 | 40 |
| Laying | 0 | 0 | 0 | 420 | 780 | 110 |

Accuracy=0.548

**Kruskal Wallis + KNN**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Walking | Walking Upstairs | Walking Downstairs | Sitting | Standing | Laying |
| Walking | 1015 | 70 | 85 | 0 | 0 | 0 |
| Walking Upstairs | 211 | 846 | 93 | 0 | 0 | 0 |
| Walking Downstairs | 66 | 127 | 927 | 0 | 0 | 0 |
| Sitting | 0 | 0 | 0 | 278 | 558 | 524 |
| Standing | 0 | 0 | 0 | 310 | 792 | 278 |
| Laying | 0 | 20 | 0 | 174 | 175 | 811 |

Accuracy=0.634

**PCA + FIsher**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Walking | Walking Upstairs | Walking Downstairs | Sitting | Standing | Laying |
| Walking | 470 | 290 | 560 | 0 | 0 | 0 |
| Walking Upstairs | 50 | 950 | 170 | 0 | 0 | 0 |
| Walking Downstairs | 100 | 150 | 770 | 0 | 0 | 0 |
| Sitting | 0 | 30 | 0 | 280 | 900 | 90 |
| Standing | 0 | 60 | 0 | 50 | 1230 | 0 |
| Laying | 0 | 0 | 0 | 0 | 0 | 1210 |

Accuracy=0.667

**PCA + SVM**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Walking | Walking Upstairs | Walking Downstairs | Sitting | Standing | Laying |
| Walking | 25 | 0 | 95 | 96 | 1054 | 0 |
| Walking Upstairs | 0 | 9 | 0 | 83 | 828 | 0 |
| Walking Downstairs | 4 | 0 | 134 | 93 | 729 | 0 |
| Sitting | 770 | 11 | 0 | 327 | 152 | 20 |
| Standing | 523 | 4 | 0 | 798 | 135 | 0 |
| Laying | 135 | 11 | 0 | 1 | 19 | 1304 |

Accuracy=0.263

**PCA + Naive Bayes**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Walking | Walking Upstairs | Walking Downstairs | Sitting | Standing | Laying |
| Walking | 880 | 120 | 130 | 0 | 0 | 0 |
| Walking Upstairs | 120 | 800 | 100 | 0 | 0 | 0 |
| Walking Downstairs | 340 | 220 | 510 | 0 | 0 | 0 |
| Sitting | 0 | 10 | 0 | 610 | 720 | 40 |
| Standing | 0 | 10 | 0 | 200 | 1200 | 0 |
| Laying | 0 | 10 | 0 | 20 | 0 | 1320 |

Accuracy=0.723

**PCA + KNN**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Walking | Walking Upstairs | Walking Downstairs | Sitting | Standing | Laying |
| Walking | 887 | 251 | 172 | 0 | 0 | 0 |
| Walking Upstairs | 136 | 915 | 139 | 0 | 0 | 0 |
| Walking Downstairs | 238 | 319 | 463 | 0 | 0 | 0 |
| Sitting | 0 | 1 | 0 | 668 | 511 | 70 |
| Standing | 0 | 0 | 0 | 261 | 1099 | 0 |
| Laying | 0 | 0 | 0 | 23 | 0 | 1207 |

Accuracy=0.712

**PCA + Híbrido**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Walking | Walking Upstairs | Walking Downstairs | Sitting | Standing | Laying |
| Walking | 793 | 148 | 199 | 0 | 0 | 0 |
| Walking Upstairs | 110 | 771 | 89 | 0 | 0 | 0 |
| Walking Downstairs | 285 | 174 | 481 | 0 | 0 | 0 |
| Sitting | 0 | 10 | 0 | 780 | 610 | 30 |
| Standing | 0 | 20 | 0 | 173 | 1207 | 0 |
| Laying | 0 | 10 | 0 | 30 | 0 | 1440 |

Accuracy=0.744

**PCA + Naive Bayes (binary+multiclass)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Walking | Walking Upstairs | Walking Downstairs | Sitting | Standing | Laying |
| Walking |  |  |  |  |  |  |
| Walking Upstairs |  |  |  |  |  |  |
| Walking Downstairs |  |  |  |  |  |  |
| Sitting |  |  |  |  |  |  |
| Standing |  |  |  |  |  |  |
| Laying |  |  |  |  |  |  |

Accuracy=0.

**Cenário com 6 classes (Conjunto de Teste)**

Após a verificação dos resultados obtidos aplicando os classificadores ao conjunto de treino foi possível escolher a combinação o classificador baseado na combinação PCA + Minimun Euclidean Distance e proceder à sua aplicação ao conjunto de teste e assim verificar a sua *performance,* uma vez que foi o que apresentou melhores resultados na fase de treino*.*

Na segunda fase do projeto foi possível encontrar combinações de classificadores e métodos de seleção de *features* com um desempenho superior às testadas anteriormente, nomeadamente

**PCA + Minimun Euclidean Distance**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Walking | Walking Upstairs | Walking Downstairs | Sitting | Standing | Laying |
| Walking | 340 | 41 | 115 | 0 | 0 | 0 |
| Walking Upstairs | 51 | 392 | 28 | 0 | 0 | 0 |
| Walking Downstairs | 149 | 56 | 215 | 0 | 0 | 0 |
| Sitting | 0 | 6 | 0 | 259 | 223 | 3 |
| Standing | 0 | 9 | 0 | 140 | 383 | 0 |
| Laying | 0 | 2 | 0 | 15 | 0 | 520 |

Accuracy=0.716

**Discussão**

Depois de aplicadas as diferentes combinações entre as técnicas de redução de dimensionalidade e os diferentes tipos de classificadores, seguidas da validação cruzada, foi possível verificar que para o caso binário os que apresentaram uma maior capacidade de generalização foram as combinações PCA+Fisher LDA, PCA + Minimun Mahalanobis Distance e Kruskal Wallis + Minimun Euclidean Distance. No caso *multiclass,* o classificador que se destacou foi o que foi desenvolvido com base no PCA para redução da dimensionalidade e na distância euclidiana mínima para classificação dos dados.

De acordo com os resultados na fase de treino foram aplicados os classificadores desenvolvidos aos dados para teste. No cenário binário, os resultados foram muito satisfatórios uma vez que a *accuracy* observada nos primeiros três testes foi aproximadamente igual a 1, sendo que no último teste foi alcançada uma performance perfeita. Relativamente ao cenário onde se pretende fazer a distinção entre as seis classes, a *accuracy* obtida foi de 70%, o que permite verificar que existe uma relação entre as *features* existentes e o estado da pessoa (a caminhar, sentada, em pé,…), desta forma os resultados poderão ser melhorados aplicando classificadores mais complexos na fase posterior do trabalho.

Na segunda fase do trabalho, foi possível encontrar mais um classificador com desempenho perfeito para o cenário binário (*LDA+SVM*), tanto na fase de treino como na fase de teste, o que permite afirmar que o parte do objetivo foi atingido,

No que diz respeito ao cenário com várias classes, foram encontrados classificadores com um desempenho superior relativamente aos mais simples que foram testados anteriormente.