

# Classificação de Atividades a partir da Aceleração

José Rolo, Luís Rato

Universidade de Évora

m11170@alunos.uevora.pt, lmr@di.uevora.pt

**Resumo** Um dos tipos de problemas mais populares em Mineração de Dados é a classificação. Através da aplicação de algoritmos, como o *J48*, *k-Nearest Neighbour* ou *Naive Bayes* é efetuada uma tentativa de classificação de várias atividades quotidianas como Andar a Pé, Correr, Andar de Bicicleta ou de Automóvel a partir da aceleração registada pelo *smartphone*. Os dados recolhidos serão tratados de forma a que possam ser utilizados pelos algoritmos. No final será apresentada uma comparação de desempenho entre os mesmos. Dadas as semelhanças com o tipo de problemas tratados, é feito ainda um teste suplementar no campo da deteção de quedas.

**Palavras-Chave:** Mineração de Dados, Naive Bayes, J48, k-Nearest Neighbours, Processamento de Dados

## 1 Introdução

Cada vez mais os dispositivos móveis fazem parte da vida diária de muitas pessoas, sendo usados para inúmeras tarefas, desde leitor de livros digitais a treinador pessoal em determinado tipo de desporto. Para responder a este uso intensivo dos dispositivos móveis, os fabricantes colocam cada vez mais tecnologia nos mesmos, fazendo deles um autêntico canivete suíço no que toca ao número de sensores incluídos. Atualmente a maioria dos *smartphones* possuem um acelerómetro de 3 eixos, que fornece o vetor da aceleração aplicada no dispositivo. Este sensor pode ser utilizado diretamente por muitas aplicações, por exemplo para simular um volante num jogo de corrida, fazendo apenas simples cálculos para inferir a posição do dispositivo.

A classificação de atividades a partir da aceleração é uma área de investigação que tenta inferir automaticamente o tipo de movimentos de um humano a partir do seu vetor de aceleração. No que toca à recolha de dados é imprescindível o uso de uma janela de amostras, pelo facto de as atividades que se pretende classificar serem compostas por movimentos cíclicos, que só podem ser distinguidos ao analisar um conjunto de amostras sequenciais, com uma certa duração [11] [4] [9] [2].

Neste tipo de análise o tratamento inicial dos dados é essencial, pois deste modo são extraídas as características do movimento que são consideradas relevantes para uma determinada atividade. Deste modo relativamente ao pré-processamento dos dados, existem variados tipos de abordagens no que toca à

extração de características a partir de uma janela temporal, que são bastante influenciadas pelo tipo e quantidade de atividades que se pretendem classificar. Existem trabalhos que calculam as correlações entre os atributos [2][9], de modo a distinguir quais são os atributos que se influenciam mutuamente. Outros trabalhos obtêm a média da aceleração e o desvio padrão para cada eixo, calculando também o módulo da aceleração de modo a obter a aceleração aplicada independentemente do eixo, apresentando uma abordagem que calcula algumas grandezas físicas antes das tradicionais medidas probabilísticas [4]. De modo a aumentar a precisão da classificação existem ainda trabalhos que aplicam uma transformada de Fourier, mais concretamente FFT<sup>1</sup>, para extrair características relacionadas com a amplitude de determinado sinal numa certa frequência [7].

No que toca aos algoritmos a escolha é variada, sendo mais populares algoritmos de classificação Bayesiana (*Naive Bayes* principalmente), mas também árvores de decisão e algoritmos denominados como "preguiçosos" [4][9][7].

O trabalho descrito ao longo deste artigo, tem como objetivo principal a classificação de quatro atividades: Correr, Andar a Pé, Andar de Bicicleta e Andar de Carro. Para isso irá basear-se em algumas características de trabalhos relacionados utilizando alguns dos tradicionais algoritmos de classificação comparando de seguida os resultados.

## 2 Dados

Os dados recolhidos para este artigo foram recolhidos através do acelerómetro de um *smartphone* usual de gama média, com sistema operativo Android 4.1. Para isso foi criada uma aplicação móvel específica para o efeito, que recolhe os dados do acelerómetro à máxima velocidade que o hardware permite, 100 amostras por segundo (100Hz). Os dados recolhidos são guardados automaticamente num ficheiro CSV<sup>2</sup> que será depois utilizado como entrada para o pré-processamento dos dados. Na Figura 1 estão representados os dados provenientes de várias atividades ainda sem qualquer processamento, sendo já visível notórias diferenças entre as várias atividades.

Os dados foram sempre recolhidos com o dispositivo no bolso direito das calças e com o ecrã virado para dentro, excetuando os dados recolhidos para a classe *Carro* em que o dispositivo foi colocado numa posição horizontal com o ecrã virado para cima. Este método de recolha é usado devido ao facto de a maioria das atividades serem baseadas no movimento dos membros inferiores. Deste modo esses movimentos são melhor registados usando o *smartphone* no bolso. No caso de andar de carro, isto não se aplica por isso é usada outra posição. Existem trabalhos similares que também usam o *smartphone* para efetuar a recolha dos dados, como por exemplo [4][11].

Cada classe dos dados recolhidos, equivale a uma duração entre 2 a 5 minutos da atividade em questão.

---

<sup>1</sup> *Fast Fourier Transform*

<sup>2</sup> *Comma Separated Values*

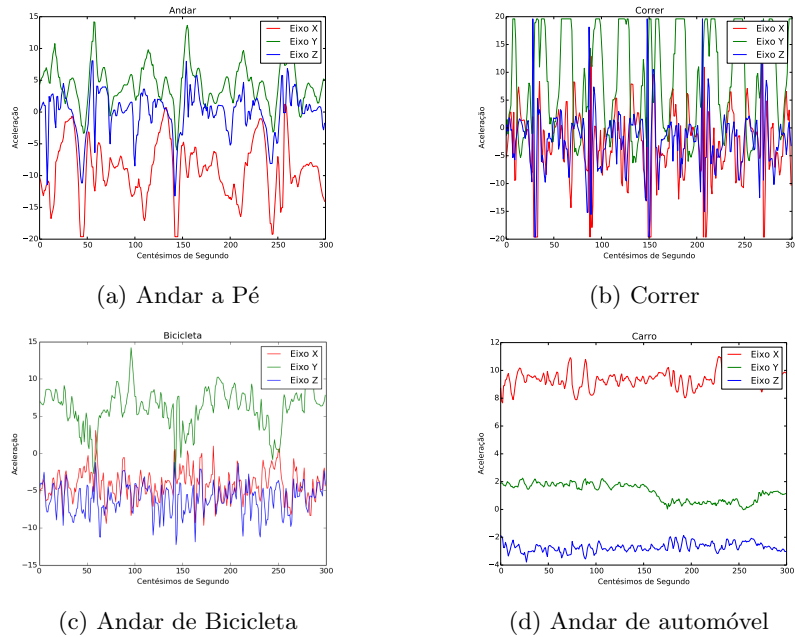


Figura 1: Comparação da aceleração entre as várias actividades.

## 2.1 Pré-Processamento

Os dados provenientes da aplicação Android, encontram-se no formato CSV possuindo apenas a sequência de vetores de aceleração lidos durante um certo período de tempo para uma determinada actividade. Estes dados são de seguida divididos em janelas de 100 amostras, ou seja 1 segundo, dando origem a uma instância do conjunto de dados que sairá da fase de pré-processamento. De salientar que as janelas de dados são sempre disjuntas. O *script* que executa a fase de pré-processamento dos dados foi escrito em `python 2.3`, encontrando-se disponível em <http://goo.gl/Kwb7xi>, bem como o código fonte da aplicação Android criada para o efeito, em <http://goo.gl/oCxqPN>.

A fase de pré-processamento é dividida em duas fases: o cálculo de algumas medidas a partir do vetor da aceleração e a extração de características a partir de cada janela de dados. Na primeira fase são calculadas as seguintes medidas, efetuadas para cada amostra recolhida, usando apenas o vetor de aceleração  $\mathbf{a} = (a_x, a_y, a_z)$ :

– Ângulo  $\phi$ :

$$\phi = \arctan \frac{a_x}{a_y}$$

Ângulo efectuado pelo *smartphone* entre o plano definido pelos eixos  $x$  e  $z$ , e o eixo  $y$ .

- Ângulo  $\theta$ :  

$$\theta = \arctan \frac{a_z}{\sqrt{a_x^2 + a_y^2}}$$
 Ângulo efectuado pelo *smartphone* entre o plano definido pelos eixos  $x$  e  $y$ , e o eixo  $z$
- Módulo da aceleração  $|\mathbf{a}|$ :  

$$|\mathbf{a}| = \sqrt{a_x^2 + a_y^2 + a_z^2}$$
 O valor da aceleração independentemente do eixo onde ela é aplicada.
- Variação do Módulo da aceleração:  

$$\Delta \mathbf{a}_i = ||\mathbf{a}|_i - |\mathbf{a}|_{i-1}|$$
 O valor absoluto da variação do módulo da aceleração. No primeiro caso, em que  $i = 0$ ,  $\Delta \mathbf{a}_i = |\mathbf{a}|_i$

Na segunda fase do pré-processamento o conjunto de dados existente é dividido em partes de 100 amostras, sendo calculadas as seguintes medidas para cada janela:

- O valor médio, mínimo e máximo para as seguintes medidas:
  - ângulo  $\phi$
  - ângulo  $\theta$
  - $|\mathbf{a}|$
  - $|\Delta \mathbf{a}|$
- $|\Delta v| = |\max(g) - \min(g)|$ , onde  

$$g = \sqrt{TF^2(a_x) + TF^2(a_y) + TF^2(a_z)}$$
, em que  
 $FT(x)$  é a transformada de Fourier de um sinal  $x$  e  $(a_x, a_y, a_z)$  é o vetor da aceleração num instante de tempo.
 O módulo da variação das amplitudes da transformada de Fourier entre 0 e 30Hz. A transformada de Fourier é aplicada ao sinal proveniente de cada eixo  $(a_x, a_y, a_z)$ , sendo depois feito o módulo do vetor resultante para cada frequência, dando origem apenas a um único sinal resultante onde é calculada a medida acima descrita. A transformada de Fourier é calculada usando a implementação FFT da biblioteca `numpy` para `python 2.7` [10].

As medidas calculadas são todas muito simples e diretas, tendo como único objetivo a extração de características relevantes para o tipo de atividades que se pretende classificar. Contudo existe uma medida que assenta diretamente no módulo do vetor das transformadas de Fourier de cada eixo. Esta medida foi escolhida depois de alguma observação dos gráficos das transformadas de Fourier para várias janelas de dados, sendo perceptível a olho nu um aumento da amplitude entre os 0 e os 30Hz quando se trata de atividades como andar ou correr. Devido a este fator ser à partida diferenciador das várias atividades foi escolhido como atributo do conjunto de dados que servirão de entrada para os algoritmos de classificação.

No conjunto de dados que sai do pré-processamento existem 130 instâncias (17%) para as classes *Correr* e *Bicicleta*, 210 instâncias (27%) para a classe *Andar* e 290 instâncias (38%) para a classe *Carro*, perfazendo um total de 760 instâncias.

### 3 Abordagem

A ferramenta utilizada para a aplicação e teste dos vários algoritmos de classificação escolhidos foi o software Weka[12], que possui inúmeros algoritmos, com muitas parametrizações que permitem afiná-los de modo a adaptarem-se o melhor possível ao problema.

#### 3.1 Algoritmos

O tipo de mineração de dados que se impõe para este conjunto de dados é a classificação, por isso foram escolhidos 3 algoritmos distintos: o algoritmo J48<sup>3</sup>, utilizado como comparação em [6],[4], [2] e [9], o algoritmo Naive Bayes (NB) utilizado em [5], [9] e [6], e por fim o algoritmo *k-Nearest Neighbours* (KNN) utilizado em [7], [9] e [6].

Os algoritmos anteriores foram escolhidos devido ao facto de apresentarem um desempenho consideravelmente elevado em trabalhos homólogos, mas também por pertencerem a classes distintas de algoritmos, como os Bayesianos (NB), as árvores de decisão (J48 ou C4.5) ou os algoritmos denominados por preguiçosos<sup>4</sup> (KNN). A escolha de algoritmos de 3 classes distintas poderá indicar uma classe de algoritmos que melhor se adequa a este conjunto de dados.

Os resultados obtidos com os algoritmos definidos anteriormente serão apresentados e comparados na secção seguinte.

**J48/C4.5** O algoritmo J48 é um algoritmo de classificação baseado em árvores de decisão construídas a partir dos dados de treino. Para cada conjunto de dados o algoritmo constrói uma árvore de decisão baseada nos atributos que possuem maior ganho de informação (Fórmula 1). Sendo assim a raiz da árvore dividirá o atributo que possui maior ganho de informação e o mesmo para os nós restantes até não conseguir dividir mais ou todos os atributos de um nó pertencerem a uma só classe, nesse caso será uma folha classificada com essa classe. No caso dos atributos serem contínuos é encontrado um limiar de divisão dos dados de modo a maximizar a entropia (Fórmula 3 dos conjuntos depois de divididos).

$$GI(C, x) = H(C) - H(C|x), \text{ onde} \quad (1)$$

$$H(C|a) = \sum_{x \in X, y \in Y} p(x, C) \log_2 \frac{p(x)}{p(x, C)} \text{ e} \quad (2)$$

$$H(C) = \sum_i p(c_i) \log_2 p(c_i) \quad (3)$$

No final é obtida uma árvore de decisão através da qual é possível classificar de forma simples e perceptível por humanos qualquer conjunto de dados [8].

<sup>3</sup> mais conhecido por C4.5, sendo o J48 uma implementação *open-source* do algoritmo C4.5 no software Weka

<sup>4</sup> do Inglês *lazy algorithms*

**Naive Bayes** O algoritmo Naive Bayes assenta num modelo probabilístico que assume à partida duas condições: 1) os atributos são condicionalmente independentes da classe; 2) não existem atributos escondidos que influenciam o processo de predição. Posto isto, toda a aprendizagem é baseada na lei de Bayes (Fórmula 4), considerando  $X$  um vetor de atributos e  $C$  o conjunto das classes. O resultado da classificação será a classe com maior probabilidade dado um conjunto de atributos.

$$p(C|X) = \frac{p(C) \cdot p(X|C)}{p(X)} \quad (4)$$

Quando se trata de dados com domínio contínuo, é assumido que os atributos associados a cada classe estão distribuídos de acordo com uma distribuição Normal. Para isso é necessário calcular a média e a variância para cada atributo contínuo de cada classe. Através destas medidas o valor de  $p(X|C)$  é facilmente calculado a partir da equação da distribuição Gaussiana (Fórmula 5 e 6). Outra abordagem que poderá ser utilizada é a discretização dos dados contínuos, passando estes a estar agrupados em intervalos, sendo que a partir daí já se poderá aplicar diretamente a lei de Bayes (Fórmula 4) [3].

$$P(X = v|C = c) = g(x; \mu_c, \sigma_c), \text{ onde} \quad (5)$$

$$g(x; \mu_c, \sigma_c) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} \quad (6)$$

**k-Nearest Neighbours** O algoritmo *k-Nearest Neighbours* é bem mais simples que os anteriores, por isso é muitas vezes considerado um algoritmo preguiçoso. Este algoritmo classifica um conjunto de dados a partir do modelo de treino diretamente, isto é, não aplica nenhuma transformação ou cálculo aos dados utilizados como treino. Considerando os pontos triangulares e quadrangulares da Figura 2 como os dados de treino de determinado problema, a classificação do ponto verde iria ser ditada pelos  $k$  vizinhos mais próximos. Neste caso iria ser classificado como triangular. O valor de  $k$  é parametrizável sendo que a partir de um certo limiar o erro de classificação irá aumentar substancialmente [1].

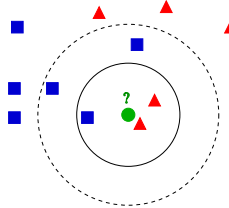


Figura 2: Algoritmo *k-Nearest Neighbours*

### 3.2 Avaliação de Desempenho

O desempenho dos algoritmos no conjunto de dados escolhido será analisado tendo em conta a sua taxa de acerto e o seu custo de acordo com uma matriz de custo. A taxa de acerto possui limitações, principalmente quando o número de instâncias por classe não tem a mesma ordem de grandeza. Deste modo para evitar uma análise errada baseada apenas na taxa de acerto foi elaborada uma matriz de custo que serve para atribuir penalizações a classificações com menos sentido, por exemplo a classificação de andar a pé como andar de carro. A matriz de custo está definida na tabela 1. A avaliação de desempenho será feita tendo em conta as duas medidas apresentadas anteriormente. O custo de classificar uma atividade  $i$  como  $j$  está definido na Fórmula 7, sendo a distância definida segundo a Figura 3.

$$Custo(i, j) = \begin{cases} distancia(i, j) & i \neq j \\ 0 & i = j \end{cases} \quad (7)$$

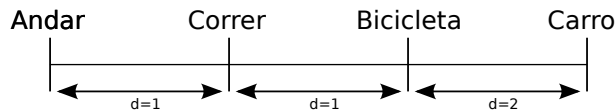


Figura 3: Diagrama representativo do custo considerado.

	Classificado como			
	Correr	Bicicleta	Andar	Carro
Correr	0	1	1	3
Bicicleta	1	0	2	2
Andar	1	2	0	4
Carro	3	2	4	0

Tabela 1: Matriz de custo

A matriz de custo (tabela 1) foi feita de modo a penalizar mais os erros nas atividades que possuem uma menor semelhança com a classe correta e menos em atividades com tipos de movimentos semelhantes. Por exemplo, para a primeira linha, foi considerado que o erro de classificar correr como andar de bicicleta tem um custo menor do que classificar como andar a pé, que por sua vez também possui um custo menor do que andar de carro, por serem atividades bastante distintas.

## 4 Resultados

De modo a garantir a correção da classificação dos dados, foram efetuados testes com 2 versões do conjunto de dados. A primeira usando todos os atributos definidos na secção do Pré-Processamento e a segunda eliminando os atributos relacionados com os ângulos (média, mínimo e máximo de  $\theta$  e  $\phi$ ). Esta divisão foi feita para garantir que a posição de recolha dos dados, que é diferente quando se trata da classe *Carro*, não define o tipo de atividade recolhida.

No que toca às parametrizações dos algoritmos, no J48 são usadas as definições padrão do Weka, no Naive Bayes existe uma opção que discretiza automaticamente os dados, mas esta não foi escolhida porque poderia enviesar os resultados de acordo com as partições feitas durante a discretização, no KNN é usado apenas o vizinho mais próximo para a classificação, ou seja  $k = 1$ . Em todos os testes é usada uma validação cruzada com 10 subconjuntos, ou seja o conjunto de dados é partido em 10 partes iguais, sendo depois uma dessas partes usada para teste e as restantes nove para treino, sendo de seguida o conjunto de treino e testes de novo alterado, construindo de novo o modelo.

Tal como foi descrito na secção anterior as medidas de desempenho usadas serão a taxa de acerto e o custo de acordo com a matriz de custo definida na Tabela 1. Esta análise irá ser efetuada paralelamente para os dois conjuntos de dados, com os atributos todos (A) e sem os atributos relacionados com ângulos (B).

A matriz de confusão para os 3 algoritmos escolhidos está definida na Tabela 2, para o conjunto de dados A e na Tabela 3 para o conjunto de dados B.

De acordo com os resultados apresentados na Tabela 4, o algoritmo KNN foi o que obteve melhores resultados tanto no conjunto de dados A como no B, considerando a taxa de acerto e o custo resultante da aplicação da matriz de custo da Tabela 1.

Em relação ao conjunto de dados A, onde se encontram todos os atributos calculados no pré-processamento são apresentadas taxas de acerto na ordem do 98%, contudo ao ser verificado o valor do custo verifica-se que o algoritmo KNN possui um custo bastante inferior ao dos restantes em cerca de 64%.

No conjunto de dados B, onde foi retirada toda a informação relativa aos ângulos, todos os algoritmos apresentam resultados na mesma ordem de grandeza, contudo o algoritmo KNN é aquele que consegue melhor desempenho, apesar de não ser muita a diferença com os restantes.

De acordo com os resultados apresentados na Tabela 4, o algoritmo KNN foi o que obteve melhores resultados tanto no conjunto de dados A como no B, considerando a taxa de acerto e o custo resultante da aplicação da matriz de custo da Tabela 1.

Em relação ao conjunto de dados A, onde se encontram todos os atributos calculados no pré-processamento são apresentadas taxas de acerto na ordem do 98%, contudo ao ser verificado o valor do custo verifica-se que o algoritmo KNN possui um custo bastante inferior ao dos restantes em cerca de 64%.

No conjunto de dados B, onde foi retirada toda a informação relativa aos ângulos, todos os algoritmos apresentam resultados na mesma ordem de gran-



	Classificado como											
	Correr			Bicicleta			Andar			Carro		
	J48	NB	KNN	J48	NB	KNN	J48	NB	KNN	J48	NB	KNN
<b>Correr</b>	127	126	128	1	2	1	1	0	1	1	2	0
<b>Bicicleta</b>	2	1	2	126	127	127	1	0	0	1	2	1
<b>Andar</b>	1	1	0	1	2	1	206	207	209	2	1	0
<b>Carro</b>	0	0	0	0	0	0	0	0	0	290	290	290

Tabela 2: Matriz de confusão(A) dos algoritmos J48, *Naive Bayes* e *k-Nearest Neighbours*.

	Classificado como											
	Correr			Bicicleta			Andar			Carro		
	J48	NB	KNN	J48	NB	KNN	J48	NB	KNN	J48	NB	KNN
<b>Correr</b>	124	124	123	3	2	3	3	4	4	0	0	0
<b>Bicicleta</b>	1	6	2	106	112	111	1	2	3	22	10	14
<b>Andar</b>	3	4	1	3	2	1	204	204	207	0	0	1
<b>Carro</b>	0	0	0	18	32	20	0	0	1	272	258	269

Tabela 3: Matriz de confusão(B) dos algoritmos J48, *Naive Bayes* e *k-Nearest Neighbours*.

deza, contudo o algoritmo KNN é aquele que consegue melhor desempenho, apesar de não ser muita a diferença com os restantes.

Mesmo considerando o conjunto de dados com pior desempenho (B), as taxas de acerto estão na ordem dos 92%, com um custo a rondar os 100, que apesar de ser bastante superior ao conjunto A é natural, pois são cometidos erros de classificação mais "graves" neste conjunto de dados.

	J48		NB		KNN	
	Tx Acerto	Custo	Tx Acerto	Custo	Tx Acerto	Custo
Conjunto A	98.5526 %	22	98.6842 %	21	99.2105 %	8
Conjunto B	92.8947 %	98	91.8421 %	108	93.4211 %	94

Tabela 4: Medidas de Desempenho.

#### 4.1 Teste Suplementar

Devido à sua enorme utilidade prática surgiu a hipótese da deteção de quedas. Esta deteção é em tudo semelhante às atividades anteriores, excluindo apenas o facto de ser uma atividade pontual, em vez de cíclica como andar ou correr. Deste modo foram extraídas exatamente as mesmas características das atividades classificadas anteriormente e descritas na secção 2, sendo alterado apenas o tamanho da janela usada para a extração de características, passando a ser de 3 segundos (300 amostras). Esta alteração ao tamanho da janela é necessária devido ao facto de ser necessário captar toda a distorção no sinal da aceleração durante a queda até os sinais estabilizarem. A Figura 4 mostra o gráfico da aceleração onde é possível ver a queda a partir dos 300 centésimos de segundo. Foram apenas registados dados provenientes de 10 quedas, dado que é apenas um teste para saber se as medidas usadas até agora se aplicam também neste domínio.

Após a aplicação dos algoritmos usados anteriormente são obtidas as matrizes de confusão apresentadas na Tabela 5. Neste caso, devido ao facto de as classes possuírem um número de amostras substancialmente diferente, a taxa de acerto não é a medida mais correta para analisar o desempenho dos classificadores, sendo a Medida-F a medida que faz mais sentido por ter em conta os Verdadeiros Positivos (VP), os Falsos Negativos (FN) e os Falsos Positivos (FP). Os valores da Medida-F encontram-se na Tabela 6.

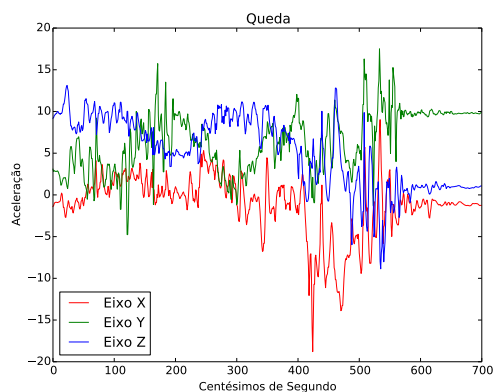


Figura 4: Gráfico dos dados correspondentes a uma queda.

Analisando os dados presentes na tabela 5 e 6 é possível indicar que as medidas usadas anteriormente para a extração de características são indicadas também para efetuar uma boa deteção de quedas, mesmo utilizando uma amostra reduzida de dados de treino para as quedas.

	Classificado como							Medida-F		
	Queda			Andar				J48	NB	KNN
	J48	NB	KNN	J48	NB	KNN				
<b>Queda</b>	10	10	10	0	0	0	0.952	0.952	1	
<b>Andar</b>	1	1	0	69	69	7	0.993	0.993	1	

Tabela 5: Matriz de confusão.

Tabela 6: Medida-F

## 5 Conclusões e Trabalho Futuro

Através do trabalho descrito nas secções anteriores é demonstrada a possibilidade de classificar com eficácia diferentes tipos de atividades, apenas com recurso ao acelerómetro. Contudo é de salientar que os resultados obtidos são baseados em dados recolhidos autonomamente, usando apenas o sensor incluído no *smartphone*, sendo que para uma análise mais consistente e detalhada seria necessário proceder de um modo mais rigoroso na recolha de dados. Todos os testes de classificação efetuados, assim como o teste suplementar no domínio da deteção de quedas revelaram-se um sucesso.

Como trabalho futuro seria interessante a procura de novas características relevantes, possíveis de extrair da aceleração, assim como a implementação prática de um dos classificadores apresentados, permitindo assim a identificação de uma atividade quase em tempo-real.

## Referências

1. Aha, D.W., Kibler, D., Albert, M.K.: Instance-Based learning algorithms. Kluwer Academic Publishers (1991)
2. Chernbumroong, S., Atkins, A.S., Yu, H.: Activity classification using a single wrist-worn accelerometer. 5th IEEE International Conference on Software, Knowledge, Information Management and Applications (September 2011)
3. John, G.H., Langley, P.: Estimating continuous distributions in bayesian classifiers. Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence (1995)
4. Kwapisz, J.R., Gary M. Weiss, Moore, S.A.: Activity recognition using cell phone accelerometers. SIGKDD Explorations 12 (December 2010)
5. Long, X., Yin, B., Aarts, R.M.: Single-accelerometer-based daily physical activity classification. 31st Annual International Conference of the IEEE EMBS (2009)
6. Maurer, U., Smailagic, A., Siewiorek, D.P., Deisher, M.: Activity recognition and monitoring using multiple sensors on different body positions. Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks (2006)
7. Preece, S.J., Goulermas, J.Y., Kenney, L.P., Howard, D.: A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data. IEEE Transactions on Biomedical Engineering 56(3) (March 2009)
8. Quinlan, R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers (1993)

9. Ravi, N., Dandekar, N., Mysore, P., Littman, M.L.: Activity recognition from accelerometer data. 17th Innovative Applications of Artificial Intelligence Conference (July 2005)
10. SciPy Developers: Numpy. [www.numpy.org](http://www.numpy.org), Acedido em: 16-2-2015
11. Sun, L., Zhang, D., Li, B., Guo, B., Li, S.: Activity recognition on an accelerometer embedded mobile phone with varying positions and orientations. UIC'10 Proceedings of the 7th International Conference on Ubiquitous Intelligence and Computing (2010)
12. University of Waikato: Weka 3: Data Mining Software in Java. [www.cs.waikato.ac.nz/ml/weka](http://www.cs.waikato.ac.nz/ml/weka), Acedido em: 16-2-2015