

# Smartphone-Based Recognition of Human Activities and Postural Transitions

Bruno M. Dobrovolski<sup>1</sup>, Eduardo A. Schmoller<sup>1</sup>

<sup>1</sup>Departamento Acadêmico de Informática – Universidade Tecnológica Federal do Paraná  
Pato Branco – PR – Brasil

{brunod,schmoller}@alunos.utfpr.edu.br

## 1. Introdução

Exemplo artigo: <https://www.eleu.ucl.ac.be/Proceedings/esann/esannpdf/es2013-11.pdf>

Este trabalho apresentará resultados da aplicação de métodos de aprendizado de máquinas na predição de movimentos humanos. Foram implementados os seguintes métodos de aprendizado: KNN, Perceptron e SVM.

Os dados para base para treinamento e teste foram obtidos de [Dheeru and Karra Taniskidou 2017]. Essas informações são referente a sensores de um dispositivo móvel Galaxy Samsung S2. O *smartphone* foi mantido junto ao corpo dos voluntários em atividades de: Levantar, sentar, deitar, caminhar, subir escadas e descer escadas. Os testes foram executados com 30 voluntários e os dados coletados através dos sensores do smartphone, acelerômetro de 3 eixos e giroscópio de 3 eixos.

Os dados obtidos dos sensores foram pré-processados com a aplicação de filtros para remoção de ruídos, amostrados em intervalos constantes de tempo e normalizados. Também foram segmentados separando cada um dos 3 eixos e calculando várias informações estatísticas (totalmente descritas em citeDua:2017), como média, desvio padrão, máximo, mínimo, entre outras, totalizando 561 características. Cada conjunto de dados que representa os movimentos recebeu uma identificação. O conjunto de características e de identificações (*label*) são entradas para os algoritmos de classificação.

Como forma de validação do processo de aprendizado, o conjunto de dados de entrada foi dividido em dois subconjuntos denominados conjunto de treino e conjunto de testes. O conjunto de treino é utilizado no processo de treinamento, após o processo de treinamento o conjunto de testes é classificado e os resultados são comparados com os resultados esperados.

Ainda, os algoritmos foram alterados através da redução de dimensionalidade com o uso do método PCA. Como esse método não entrega um resultado ótimo, foi necessário variar o número de componentes para encontrar a melhor combinação.

Como métrica para a determinação da qualidade do processo de aprendizados foram utilizadas a acurácia da predição e utilização do *kappa score*.

## 2. Revisão

Esta seção apresenta os métodos aplicados para a classificação dos movimentos.

## 2.1. Perceptron

[http://ml.informatik.uni-freiburg.de/former/\\_media/documents/teaching/ss09/ml/perceptrons.pdf](http://ml.informatik.uni-freiburg.de/former/_media/documents/teaching/ss09/ml/perceptrons.pdf)

Classificador linear

## 2.2. KNN

KNN algoritmo simples que realiza o aprendizado de todas as entradas e classifica as novas entradas com base na similaridade dos casos já aprendidos. Ao realizar a classificação de uma nova entrada é retornado  $k$  vizinhos mais próximos, classificação se dá pela maioria dos vizinhos.

## 2.3. SVM

SVM é um classificador definindo a separação entre as classes através de um hiperplano, a entrada são dados com *labels* e a saída é um hiperplano ótimo para categorização de novas entradas. Em duas dimensões o hiperplano é representado através de uma reta que divide o plano em duas partes.

## 2.4. PCA

[http://www.iro.umontreal.ca/~pift6080/H09/documents/papers/pca\\_tutorial.pdf](http://www.iro.umontreal.ca/~pift6080/H09/documents/papers/pca_tutorial.pdf)

<http://www.stat.cmu.edu/~cshalizi/350/lectures/10/lecture-10.pdf>

Redutor de dimensionalidade selecionando as principais componentes, entre as formas de encontrar as componentes principais é buscar as projeções que maximizem a variância.

## 2.5. Testes Estatísticos

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.558.7319&rep=rep1&type=pdf>

<https://ai.vub.ac.be/sites/default/files/4.%20Hypothesis%20Testing.pdf>

<https://www.cl.cam.ac.uk/teaching/1617/MLRD/slides/slides4.pdf>

[http://www.public.asu.edu/~huanliu/dmml\\_presentation/T-test.pdf](http://www.public.asu.edu/~huanliu/dmml_presentation/T-test.pdf)

### 2.5.1. Validação Cruzada

<https://www.analyticsvidhya.com/blog/2018/05/improve-model-performance-cross-validation-in-python-r/>

<https://towardsdatascience.com/cross-validation-in-machine-learning-729>

<https://www.kaggle.com/dansbecker/cross-validation>

[https://www.researchgate.net/profile/Ron\\_Kohavi/publication/2352264\\_A\\_Study\\_of\\_Cross-Validation\\_and\\_](https://www.researchgate.net/profile/Ron_Kohavi/publication/2352264_A_Study_of_Cross-Validation_and_)

Bootstrap\_for\_Accuracy\_Estimation\_and\_Model\_Selection/  
links/02e7e51bcc14c5e91c000000.pdf  
[http://scikit-learn.org/stable/modules/model\\_evaluation.html](http://scikit-learn.org/stable/modules/model_evaluation.html)

### 2.5.2. Matriz de Confusão

[http://www2.cs.uregina.ca/~hamilton/courses/831/notes/confusion\\_matrix/confusion\\_matrix.html](http://www2.cs.uregina.ca/~hamilton/courses/831/notes/confusion_matrix/confusion_matrix.html)

### 2.5.3. Kappa Score

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3900052/>  
<http://www.pmean.com/definitions/kappa.htm>  
<http://www.statisticshowto.com/cohens-kappa-statistic/>  
<https://academic.oup.com/ptj/article/85/3/257/2805022>

## 3. Resultados

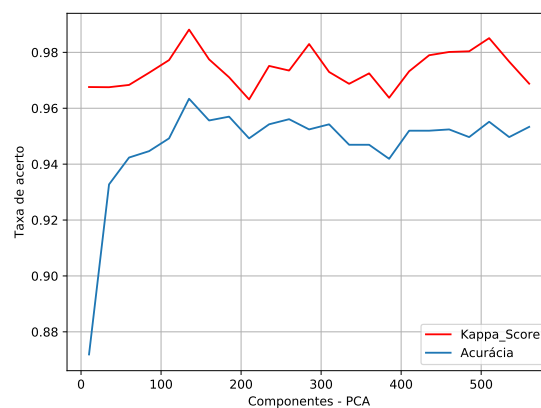


Figura 1. Acurácia e Kappa score KNN

## 4. Conclusões

...

## Referências

[Dheeru and Karra Taniskidou 2017] Dheeru, D. and Karra Taniskidou, E. (2017). {UCI} machine learning repository.

**Tabela 1. Resultados**

Método	Acurracia	Kappa Score
KNN		
Perceptron		
SVM		
KNN + PCA		
Perceptron + PCA		
SVM + PCA		