Pavel DOHNÁLEK, Petr GAJDOŠ, Pavel MORAVEC, Tomáš PETEREK, Václav SNÁŠEL

VŠB – Technical University of Ostrava

Application and comparison of modified classifiers for human activity recognition

***Abstract**. In this paper, custom modifications of Orthogonal Matching Pursuit and Self Organizing Maps based classification algorithms are used and compared to standard and widely used classification techniques with applications to human activity recognition. Seven algorithms are compared in terms of their accuracy performance. The modifications are described in this paper and shown to perform better than commonly used classifiers. The results indicate that human activities can be successfully and reliably recognized even without data preprocessing.*

***Streszczenie.** W artykule opisano klasyczne i rzadziej używane metody klasyfikacji danych używanych do rozpoznawania aktywności człowieka. Po równano szereg algorytmów oraz zmodyfikowano algorytm OMP w celu usunięcia ograniczeń. (**Przegląd metod klasyfikacji danych używanych do rozpoznawania aktywności człowieka**)*

Keywords: comparison, pattern matching, classification, sparse approximation
Słowa kluczowe: rozpoznawanie I klasyfikacji wzoru

Introduction

Human activity recognition is one of the more recent research topics that recently gained on popularity and focus of both academic and commercial researchers. Since human activity monitoring has a broad range of applications, like homecare systems, prisoner monitoring, physical therapy and rehabilitation, public security, military uses and others, motivation to create a reliable human activity recognition system is great.

Generally, approaches to recognizing activities can be divided into two groups - sensor-based and vision-based. Sensor-based systems use various sensors that are attached to the subject being monitored. Vision-based activity recognition systems, on the other hand, try to eliminate the need for sensors and attempt to recognize subject's behavior from images and video sequences. Both approaches have their challenges arising from their nature. While sensor-based systems require classification algorithms to be as speedy as possible in order to be implemented in low-power wearable devices, accurate and reliable vision-based systems are still a challenge no matter the computation power. This paper focuses on sensor-based systems, one of which was used to create the Physical Activity Monitoring for Aging People (PAMAP2) dataset that this paper elaborates on.

Current approaches in activity recognition

In general, in activity recognition authors attempt to recognize static states (lying, sitting, standing, etc.), dynamic states (walking, running, etc.) and/or transition states (i.e. standing to walking). Data preprocessing to improve the classification accuracy is common [1, 2, 3]. Classification methods currently widely used in the area are based both on classic algorithms like the Classification And Regression Tree (CART) [4, 5] or k-Nearest Neighbor (k-NN) [6, 7] and more advanced techniques like the Adaptive Network based Fuzzy Interference System (ANFIS) [8, 9], Iterative Dichotomiser 3 (ID3) [10] and others.

Classifiers being evaluated

To provide an overview of learning algorithms with application to human activity recognition, five distinct classifiers were tested, including the above mentioned k-NN and CART. Also, the Orthogonal Matching Pursuit (OMP) based classifier as defined in [11] was evaluated against a custom modification that significantly improves the reliability of the recognition. Global-Merged Self Organizing Maps (GM-SOM) [12, 13] were also used. The following subsections provide a brief informal description for each of the classifiers with the exception of OMP which is explained in more detail in its own section.

k-Nearest Neighbors

k-NN is a non-parametric algorithm, meaning that it makes no assumptions about the structure or distribution of the underlying data, thus being suitable for real-world problems that usually do not follow the theoretical models exactly. The method is also considered to be a lazy learning algorithm as it performs little to no training during computation. As a result, the method uses the whole training dataset during classification. k-NN is well known for its simplicity, speed and generally good classification results in applications like bioinformatics [14], image processing [15], audio processing [16] and many others [17, 18].

Classification and Regression Tree

This algorithm classifies a sample according to groups of other samples with similar properties. During training, the training data is continuously divided into smaller subsets (tree nodes). When the divisions are finished, the samples are clustered together according to their properties. Testing samples are then evaluated against certain conditions in each node and propagated throughout the tree. When the sample reaches a leaf node, it is then assigned the class to which the samples in that node belong. In this paper, a binary tree with logical conditions was used. CARTs are still under extensive research and can be used as a standalone classifier [19] or as part of larger algorithmic structures [20].

Global Merged Self Organizing Maps

The following text is focused on the description of our method first introduced in [12] which leads to results similar to the classic Self Organizing Maps (SOM). Named Global-Merged SOM, it divides computation into independent parts, similarly to parallel SOM [13], which are then merged to obtain the expected result. Following steps describe the whole process of GM-SOM:

1. Input set split – the set of input vectors is divided into a pre-defined number of parts. The precision of GM-SOM increases with the increasing number of parts, although this has its own disadvantages related to larger set of vectors in the final phase of the computation process. The number of parts can be usually determined from the number of input vectors. Generally, $k >> N * p$, where k is the number of input vectors, N is the number of neurons and p is the number of parts.

2. Computation in individual parts – classic SOM is applied on each part (we will label it PSOM in further text).

All PSOMs start with the same setting (the first distribution of weight vectors, number of neurons, etc.). Such division speeds up the parallel computation of PSOMs when using GPU. Also, the number of epochs can be lower than the number of epochs for processing the input set by a single SOM. This is represented by a factor *f*.

3. **Merging the individual parts** – weight vectors which were computed for each part (and correspond to neurons with at least one hit) are used as input vectors in the final phase of GM-SOM. A merged SOM with the same setting is computed and output weight vectors make the final result of the GM-SOM method.

Table 1. Recognition accuracy for each of the classifiers

	Training set size (%)				
Classifier	**10**	**20**	**30**	**40**	**50**
OMP2	98.27	99.14	99.56	99.43	99.60
3-NN	97.51	98.72	99.31	99.29	99.54
CART	97.87	98.49	99.15	99.11	99.37
OMP	95.85	97.56	98.29	98.39	98.77
GM-SOM	87.23	88.09	90.35	89.67	91,86

Individual parts are fully independent on each other and they update different PSOMs. Also, different SOM algorithms can be applied on PSOM of a given part, which makes the algorithm more variable. Since different settings of PSOMs can be used, a denser neuron network can be used in case of the input set is larger.

Orthogonal Matching Pursuit

Well described in [24], OMP is an iterative sparse approximation algorithm that reduces data into a given number of sparse coefficients and thus can be considered a dimensionality reduction algorithm. Given an overcomplete dictionary of observations, for each observation to be classified OMP picks a number of the best fitting observations from the dictionary and uses them to compute the sparse coefficients. Those are then checked against the dictionary itself for similarity and classified.

The dictionary can be represented as an $m \times n$ real-valued matrix $\mathbf{A}$, where m is the length of an observation and n is the number of observations in the dictionary (training observations). The iterative nature of the algorithm allows for sparse coefficient number to be chosen in advance. It stands to reason to limit the number of sparse coefficients s such that $s \leq m$, although the number can be truly limited only by the number of training observations, n.

Originally, the classifier proposed in [11] requires $n = t \times c$, where t is the number of training observations for a given class and c is the number of classes. This means the algorithm requires the training set to contain the same number of training observations for each class. It is also necessary to keep the observations of a given class grouped together. Therefore, the training matrix has the form of $\mathbf{A} = [\mathbf{a}_{11}, \mathbf{a}_{21}, \ldots, \mathbf{a}_{t1}, \mathbf{a}_{12}, \ldots, \mathbf{a}_{tc}]$, where $\mathbf{a}_{ij}$, $i = 1..t, j = 1..c$ is the i-th training observation of class j and length m.

The proposed modification changes the meaning of t and the resulting number of observations in the dictionary. Here,

$$\mathbf{n} = \| \mathbf{t} \|_1 = \sum_{x=1}^{c} \mathbf{t}_x, \tag{1}$$

where $\mathbf{t}$ is the c-dimensional vector consisting of numbers of training observations for a given class. By this, the limitation imposed on the number of training samples in the original classification approach is lifted. The sparse coefficients are obtained by finding the sparse solution to the equation

$$\mathbf{y} = \mathbf{A}\mathbf{s}, \tag{2}$$

where $\mathbf{y} \in \mathbb{R}^m$ is the query vector, $\mathbf{A} \in \mathbb{R}^{m \times n}$ is the training matrix and $\mathbf{s} \in \mathbb{R}^n$ is the sparse coefficient vector. The stopping criterion in the implementation is reaching the sparse coefficient vector with the desired number of non-zero values.

Classification

To classify the query signal vector, a strategy of computing the residual value from the difference between the query vector and its sparse representation converted into the vector space of the training matrix vectors is employed. This is performed for each class resulting in c residuals. The classification is then based on the minimum residual. Formally, the classification problem can be stated as follows:

$$\arg\min r_k(\mathbf{y}) = \| \mathbf{y} - \mathbf{A}\mathbf{s}_k \|_2 \tag{3}$$

Here, $\mathbf{s}_k$ is an n-dimensional vector with non-zero elements located only on indices corresponding to the kth class in the training matrix, hence the need for the training observations of a given class to be grouped together in the matrix. The algorithm could be described with the following steps:

- Set the iteration variable i to 1
- Replace all sparse coefficients not belonging to class i with zeros
- Multiply the training matrix by the modified vector $\mathbf{s}$
- Compute the ℓ^2-norm of the resulting vector
- Increase i by 1 and repeat for all classes
- Output the class whose ℓ^2-norm is the lowest

Computing the residuals is generally not computationally expensive and can be performed in real time, depending on the size of the training matrices. Only very large training matrices can slow the process down significantly.

Experiments

The following section describes the dataset used to evaluate the performance of the classifiers as well as the process of the evaluation and its results.

The PAMAP2 dataset

The PAMAP2 dataset contains data of nine healthy human subjects, each subject wearing three inertial measurement units (IMUs) by Trivisio, Germany and a heart rate monitor. Each of the three IMUs measures temperature and 3D data from accelerometer, gyroscope and magnetometer. The data is sampled at 100 Hz and transmitted to PC via a 2.4 GHz wireless network. Subjects wore one IMU on the dominant wrist, one on the dominant ankle and one on the chest. Detailed information on the dataset can be found in [25] and [26].

The methods have been tested on all 9 test subjects in the dataset, labeled in the dataset as *subject101* through *subject109*. Data of all these subjects consists of 2872532 measurements, each containing 54 values. The description of the values is available in the dataset documentation. Some values can be missing, indicated with a NaN (Not a Number) value. Every NaN value was replaced with a zero.

The activities performed are *lying*, *sitting*, *standing*, *walking*, *running*, *cycling*, *Nordic walking*, *ascending stairs*, *descending stairs*, *vacuum cleaning*, *ironing* and *rope jumping*. Transition activities were discarded. Since some measurements contain only NaN values, these measurements were discarded as well. In total, 1942746 activity measurements were used. From each measurement, irrelevant values, that is the orientation of each IMU and timestamp, were removed. Since most heart-rate values were NaNs due to different operating frequency, they were also removed. As a result, each measurement contains 39 values. These values were not preprocessed any further, the classifiers were tested on raw sensor data as provided by the PAMAP2 dataset.

Table 2. Classification accuracy with regards to individual actions (30% training set)

	Classifier				
Action	OMP2	kNN	CART	OMP	GM-SOM
lying	100	100	99.80	100	93.21
sitting	100	99.83	98.90	99.66	89.84
standing	99.75	99.75	99.63	99.63	89.17
walking	99.27	99.02	99.27	95.36	84.95
running	99.86	98.56	99.0	99.71	86.84
cycling	99.78	100	99.57	98.92	89.64
N. walk.	98.54	97.97	99.92	95.54	83.43
asc.stair	98.81	99.15	97.46	97.63	84.11
des.stairs	99.56	99.12	96.70	96.70	83.62
vacuum	100	99.90	98.89	99.20	84.28
ironing	99.57	99.46	99.46	99.14	87.47
rope	100	99.32	99.55	100	79.54

Experimental settings

The execution of some of the algorithms can be customized through execution parameters which, for these experiments, were set according to the best empirical speed/accuracy ratio. The k-NN algorithm's k parameter was set to 3. The number of sparse coefficients s computed by OMP was 10. For CART, default MATLAB settings were used. GM-SOM was set to compute an 8 × 8 network, 100 epochs and 10 parts. All of the algorithms were implemented in the latest version of MATLAB. The entire dataset was divided into a training and a testing set. Experiments were performed on 5 different settings where the training set was a 10%, 20%, 30%, 40% or 50% portion of the dataset.

Given the significant time complexity of some of the classifiers, finishing the experiments could take up to several hours, greatly depending on the size of the training set. For this reason, the number of observations used in the testing phase of a run was limited to 10000. As all compared classifiers are purely deterministic, it was sufficient to run the experiment with each setting only once.

Results

The classification accuracies given as percentual success rates are shown in Table 1 where the classifiers are sorted according to their success in the descending order. It can be seen that in almost every case the modified version of the OMP classifier (OMP2) is superior to the other classifiers, the exception being the 10% training set where CART performs better. For training set sizes of 30% and higher, OMP2 becomes very closely followed by k-NN which is, in turn, only slightly better than CART. The original OMP, while providing satisfactory results, was at the bottom of the table along with GM-SOM. At 10% training set, the difference between OMP and OMP2 was the most significant at 2.42%. When the training set size was set to 50%, all methods with the exception of GM-SOM provided very accurate recognitions.

While GM-SOM provided the worst recognition accuracy, it was the fastest classifiers in terms of classifying test vectors, once again proving that speed and accuracy are often two sides of a single scale. While the OMP or CART experiments took minutes to hours to classify the test vectors, GM-SOM is capable of making the classification within seconds. The most time consuming part of GM-SOM is training the classifier.

How accuracy is dependent on the training set can be seen in Figure 1. The experiments show that the increase of training set size does indeed benefit the recognition accuracy. Since the experiments show that 30% can provide very satisfactory results, it is reasonable to consider this training set size a good compromise between speed and accuracy. For this reason, Table 2 elaborates on the results for this training set size. It shows the percentual success rates for each classifier with regards to each of the actions to be recognized. *Lying* and *sitting* came out as activities fairly easy to recognize with any of the evaluated classifiers while distinguishing *Nordic walking* was a task a bit more difficult to the classifiers. This is to be expected due to the similarity of *Nordic walking* to simple *walking*. Still, the classifiers performed very well in recognizing every activity when OMP2's worst result was misclassifying only 1.46% of total *Nordic walking* observations. k-NN managed to drop below 99% only in a single activity. CART provided fairly consistent and satisfactory results, although 2.86% deficiency in *descending stairs* against OMP2 becomes noteworthy.

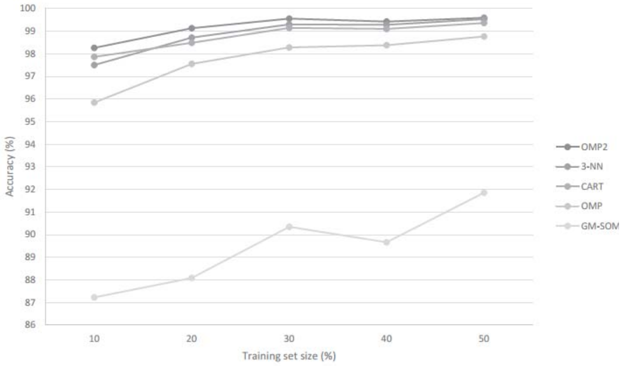

Fig.1. Accuracy dependency on the size of the training set

Conclusion

This paper evaluated several classification techniques and presented their success rates in human activity recognition without any prior preprocessing. Given the sensor technology that was used to create the PAMAP2 dataset, it was shown that activities performed in the database can be recognized reliably and with very high precision. In terms of recognition accuracy, the presented modification of the OMP classifier was shown to perform the best, however the precision comes at the price of significant time complexity. The fastest of the algorithms in the classification phase was GM-SOM, but its recognition accuracy is slightly lower, albeit still reasonably high enough for practical use. From the speed/accuracy ratio perspective, k-NN seems to be the most reasonable choice as its accuracy performance is superseded by OMP2 only closely, but k-NN has a significant edge in computation times. For this reason, the main focus of future work in this area should be making the classifiers more efficient or finding a suitable preprocessing technique that would enable high-speed classifiers to provide comparable results.

The article has been elaborated in the framework of the IT4Innovations Centre of Excellence project, reg. no. CZ.1.05/1.1.00/02.0070 funded by Structural Funds of the European Union and state budget of the Czech Republic. The work is partially supported by Grant of SGS No. SP2013/70, VŠB - Technical University of Ostrava, Czech Republic. This work was also supported by the Bio-Inspired Methods: research, development and knowledge transfer project, reg. no. CZ.1.07/2.3.00/20.0073 funded by Operational Programme Education for Competitiveness, co-financed by ESF and state budget of the Czech Republic.

REFERENCES

[1] Khan, A.M., Young-Koo Lee, Lee, S.Y., Tae-Seong Kim, A Triaxial Accelerometer-Based Physical-

Activity Recognition via Augmented-Signal Features and a Hierarchical Recognizer, *Information Technology in Biomedicine, IEEE Transactions on* , 14(5), 1166-1172, Sept. 2010

[2] Eunju Kim, Helal, S., Cook, D., Human Activity Recognition and Pattern Discovery, *Pervasive Computing, IEEE* , 9(1), 48-53, Jan. 2010

[3] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, Jorge L. Reyes-Ortiz, Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. *Proceedings of the 4th international conference on Ambient Assisted Living and Home Care* (IWAAL'12), José Bravo, Ramón Hervás, and Marcela Rodríguez (Eds.). Springer-Verlag, Berlin, Heidelberg, 216-223, 2012

[4] Loh W.Y., Classification and regression trees, 14–23, 2011

[5] Ripley B., Classification and regression trees, *R package version*, 2005

[6] Jatoba L.C., Grossmann U., Kunze C., Ottenbacher J., Stork W., Context-aware mobile health monitoring: Evaluation of different pattern recognition methods for classification of physical activity, *Engineering in Medicine and Biology Society*, 30th Annual International Conference of the IEEE, 5250 –5253, 2008

[7] Maurer U., Smailagic A., Siewiorek D.P., Deisher M., Activity recognition and monitoring using multiple sensors on different body positions, *Wearable and Implantable Body Sensor Networks*, 2006. BSN 2006. International Workshop on, IEEE, 2006

[8] Jang J.S.R., Sun C.T., Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1997

[9] Ziaeefard M., Ebrahimnezhad H., Human action recognition by RANSAC based salient features of skeleton history image using ANFIS, *Machine Vision and Image Processing (MVIP)*, 2010 6th Iranian, 1–5, 2010

[10] D.Lavanya, Rani D.K., Performance Evaluation of Decision Tree Classifiers on Medical Datasets, *International Journal of Computer Applications*, 26(4), 1–4, 2011

[11] Wright J., Yang A.Y., Ganesh A., Sastry S.S., Ma Y., Robust Face Recognition via Sparse Representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31, 210–227, 2009

[12] Gajdos P and Moravec P., Two-step modified SOM for parallel calculation, in DATESO, ser. CEUR Workshop Proceedings, J. Pokorný, V. Snásel, and K. Richta, Eds., vol. 567. CEUR-WS.org, 2010, pp. 13–21.

[13] Wei-gang, L., A study of parallel self-organizing map. In: Proceedings of the International Joint Conference on Neural Networks. (1999)

[14] Horton P., Nakai K., Better Prediction of Protein Cellular Localization Sites with the it k Nearest Neighbors Classifier, *Ismb*, 5, 147–152, 1997

[15] Haralick R.M., Shanmugam K., Dinstein I.H., Textural features for image classification, *Systems, Man and Cybernetics*, IEEE Transactions on, (6), 610–621, 1973

[16] Fujinaga I., MacMillan K., Realtime recognition of orchestral instruments, *Proceedings of the international computer music conference*, 141, 2000

[17] Levner I., Feature selection and nearest centroid classification for protein mass spectrometry., *BMC bioinformatics*, 6, 2005

[18] Dabney A.R., Classification of microarrays to nearest centroids, *Bioinformatics*, 21, 4148–4154, 2005

[19] Hess K.R., Abbruzzese M.C., Lenzi R., Raber M.N., Abbruzzese J.L., Classification and regression tree analysis of 1000 consecutive patients with unknown primary carcinoma, *Clinical Cancer Research*, 5(11), 3403–3410, 1999

[20] Liaw A., Wiener M., Classification and Regression by randomForest, *R news*, 2(3), 18–22, 2002

[21] Li M., Yuan B., 2D-LDA: A statistical linear discriminant analysis for image matrix, *Pattern Recognition Letters*, 26(5), 527–532, 2005

[22] Haeb-Umbach R., Ney H., Linear discriminant analysis for improved large vocabulary continuous speech recognition, *Acoustics, Speech, and Signal Processing*, 1992. ICASSP-92., 1992 IEEE International Conference on, 1, 13–16, IEEE, 1992

[23] Zhang M., Identification of protein coding regions in the human genome by quadratic discriminant analysis, *Proceedings of the National Academy of Sciences*, 94(2), 565–568, 1997

[24] Blumensath T., Davies M.E., On the Difference Between Orthogonal Matching Pursuit and Orthogonal Least Squares, 2007

[25] Reiss A., Stricker D., Creating and Benchmarking a New Dataset for Physical Activity Monitoring, *The 5th Workshop on Affect and Behaviour Related Assistance (ABRA)*, 2012

[26] Reiss A., Stricker D., Introducing a New Benchmarked Dataset for Activity Monitoring, *The 16th IEEE International Symposium on Wearable Computers (ISWC),* 2012

***Authors**: Pavel Dohnálek, Petr Gajdoš, Václav Snášel, IT4Innovations, Centre of Excellence and Department of Computer Science, Faculty of Electrical Engineering and Computer Science, Tomáš Peterek, IT4Innovations, Centre of Excellence, VŠB - Technical University of Ostrava, 17. listopadu 15, 708 33, Ostrava, E-mail: {pavel.dohnalek, petr.gajdos}@vsb.cz*