

Confidence-based multiclass AdaBoost for physical activity monitoring

Attila Reiss, Didier Stricker and Gustaf Hendeby

Linköping University Post Print



N.B.: When citing this work, cite the original article.

Original Publication:

Attila Reiss, Didier Stricker and Gustaf Hendeby, Confidence-based multiclass AdaBoost for physical activity monitoring, 2013, ISWC '13: Proceedings of the 2013 International Symposium on Wearable Computers, 13-20.

<http://dx.doi.org/10.1145/2493988.2494325>

Postprint available at: Linköping University Electronic Press

<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-97524>

Confidence-based Multiclass AdaBoost for Physical Activity Monitoring

Attila Reiss, Didier Stricker
Department of Augmented Vision
German Research Center for Artificial
Intelligence (DFKI)
Kaiserslautern, Germany
firstname.lastname@dfki.de

Gustaf Hendeby
Department of Electrical Engineering
Linköping University
Linköping, Sweden
hendeby@isy.liu.se

ABSTRACT

Physical activity monitoring has recently become an important topic in wearable computing, motivated by *e.g.* health-care applications. However, new benchmark results show that the difficulty of the complex classification problems exceeds the potential of existing classifiers. Therefore, this paper proposes the ConfAdaBoost.M1 algorithm. The proposed algorithm is a variant of the AdaBoost.M1 that incorporates well established ideas for confidence based boosting. The method is compared to the most commonly used boosting methods using benchmark datasets from the UCI machine learning repository and it is also evaluated on an activity recognition and an intensity estimation problem, including a large number of physical activities from the recently released PAMAP2 dataset. The presented results indicate that the proposed ConfAdaBoost.M1 algorithm significantly improves the classification performance on most of the evaluated datasets, especially for larger and more complex classification tasks.

Author Keywords

Physical activity monitoring, activity recognition, boosting, multiclass classification, algorithm, evaluation

ACM Classification Keywords

I.2.1 Artificial Intelligence: Applications and Expert Systems

INTRODUCTION

Recent progress in wearable sensing makes it reasonable to expect individuals to wear different sensors all day. Physical activity monitoring is an application area strongly benefiting from this progress. The goals of activity monitoring systems are among others to tell the type of physical activity an individual performs, and to estimate the duration and intensity of the performed activity. With the information obtained this way, the individual's daily routine can be described. Recognizing basic activities such as sitting, walking, running or cycling is a well researched area, and good recognition performance can be achieved even with just one 3D-accelerometer and simple classifiers [2, 9]. However, since these methods

only consider a limited set of similar activities, they only apply to specific scenarios. Therefore, current research in this area focuses among others on increasing the number of activities to recognize. This can for instance be achieved by introducing new classification techniques.

The use of meta-level classifiers for activity monitoring problems is not as widespread as using different base-level classifiers. However, comparing base-level and meta-level classifiers on different activity recognition tasks shows that meta-level classifiers (such as boosting, bagging, plurality voting, etc.) outperform base-level classifiers [11]. In [12], the authors evaluated the most widely used base-level and meta-level classifiers on a complex activity recognition problem: best performance was achieved with a boosted decision tree classifier. Recently, a new dataset for physical activity monitoring was introduced and made publicly available: the PAMAP2 dataset [13, 14]. Different classification problems have been defined and benchmarked on this dataset, confirming that using a boosted C4.5 decision tree classifier is one of the most promising methods. The boosted decision tree classifier has — apart from good performance results — further benefits: it is a fast classification algorithm with a simple structure, and is therefore easy to implement. These benefits are especially important for activity monitoring applications since they are usually running on mobile, portable systems for everyday usage, thus the available computational power is limited. Previous work showed the feasibility of using boosted decision tree classifier for activity recognition on a mobile platform [15]. Therefore, considering all the above mentioned benefits, this work focuses on using boosting, and in particular boosted decision tree classifiers for physical activity monitoring.

The benchmark results on the PAMAP2 dataset [13, 14] reveal that the difficulty of the more complex tasks exceed the potential of existing classifiers. Therefore, there is a reasonable demand for modifying and improving existing algorithms. This paper proposes a confidence-based extension of the well-known AdaBoost.M1 algorithm, called ConfAdaBoost.M1. It builds on established ideas of existing boosting methods. The main contribution of this work is to show that ConfAdaBoost.M1 significantly improves the results of previous boosting algorithms. The rest of this paper is organized as follows. The next section gives an overview of existing boosting algorithms, highlighting their benefits and drawbacks. Then the new ConfAdaBoost.M1 algorithm is introduced and evaluated: first on various benchmark datasets

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
ISWC'13, September 9–12, 2013, Zurich, Switzerland.
Copyright © 2013 ACM 978-1-4503-2127-3/13/09...\$15.00.
<http://dx.doi.org/10.1145/2493988.2494325>

from the UCI machine learning repository, and second on a complex activity recognition and intensity estimation problem defined on the PAMAP2 dataset. The main motivation for presenting the ConfAdaBoost.M1 algorithm is the better performance it achieves, compared to existing algorithms, on activity monitoring classification tasks.

BOOSTING METHODS: RELATED WORK, CONCEPTS

Boosting is a widely used and very successful technique for solving classification problems. The idea behind boosting is to iteratively learn weak classifiers by manipulating the training dataset, and then combine the weak classifiers into a final strong classifier. Boosting was introduced in the computational learning theory literature in the early and mid 90's [4, 17]. The adaptive boosting algorithm — called AdaBoost [4] — evolved from the early versions of this technique, and became the most commonly used boosting algorithm, from which many versions have been developed.

Binary classification

The fundamental idea of the boosting technique can be outlined the following way [4]: Assume that a training dataset of N instances is given: $(\underline{x}_i, y_i) \ i = 1, \dots, N$ (\underline{x}_i is the feature vector, $y_i \in \{-1, +1\}$). The algorithm trains weak learners, $f_t(\underline{x})$, on weighted versions of the training dataset, giving higher weight to currently misclassified instances. This is performed a predefined number of iterations, T . The final classifier is a linear combination of the weak learners from each iteration, weighted according to their error rate on the training dataset. The first version of the AdaBoost algorithm only uses the binary output of the weak learners, and is thus called Discrete AdaBoost. The Real AdaBoost algorithm [5] is a generalization of the original algorithm that use real-valued predictions of the weak learners rather than the $\{-1, +1\}$ output. The weak learners then return a class probability estimate, $p_t(\underline{x})$, in each boosting iteration, from which the classification rule $f_t(\underline{x})$ is derived. The sign of $f_t(\underline{x})$ gives the classification prediction, and $|f_t(\underline{x})|$ gives a measure of how confident the weak learner is in the prediction. Experiments on various datasets from the UCI machine learning repository [3] show that this confidence-based version of AdaBoost outperforms the original Discrete AdaBoost algorithm [5]. However, both the Discrete and Real AdaBoost are limited to binary classification problems.¹

Apart from Discrete and Real AdaBoost, further boosting methods have been developed for the binary classification case the past decade. The Discrete and Real AdaBoost algorithms can be interpreted as sequential estimation procedures for fitting an additive logistic regression model, optimizing an exponential criterion which to second order is equivalent to the binomial log-likelihood criterion [5]. Based on this interpretation, the LogitBoost algorithm was introduced, which optimizes a more standard (the Bernoulli) log-likelihood [5]. Moreover, [5] also presents the Gentle AdaBoost algorithm, a modified version of Real AdaBoost. It uses Newton stepping

¹Due to space limitations only the main concepts of the relevant boosting techniques are given within this paper. For a full algorithmic description please refer to the supporting webpage [16], where the boosting algorithms Real AdaBoost, Real AdaBoost.MH, AdaBoost.M1 and SAMME are formally described.

rather than exact optimization at each boosting iteration. Further variants of binary AdaBoost are Emphasis Boost [6] and Modest AdaBoost [21].

Pseudo-multiclass classification

The first extensions of AdaBoost for multiclass classification problems can be regarded as pseudo-multiclass solutions: they reduce the multiclass problem into multiple two class problems [18, 19]. One of the most common solutions using binary boosting methods for multiclass problems is AdaBoost.MH [19]. It converts a C class problem into that of estimating a two class classifier on a training set C times as large, by adding a new “feature” which is defined by the class labels. Thus the original number of N instances is expanded into NC instances. On this new, augmented dataset a binary AdaBoost method (*e.g.* Discrete or Real AdaBoost) can then be applied. There exist other solutions to reduce the multiclass problem into multiple binary classification problems, *e.g.* the AdaBoost.MO algorithm [18] or the extension of the binary LogitBoost [5]. In [18] experimental results are given comparing a few pseudo-multiclass algorithms on a set of benchmark problems, which show that Real AdaBoost.MH (the extension of the binary Real AdaBoost algorithm for the multiclass case using the AdaBoost.MH technique) performs best amongst these methods.

However, reducing the multiclass classification problem into multiple two class problems has several drawbacks. For instance, as the class label becomes a regular feature in the AdaBoost.MH method, its importance is significantly reduced. AdaBoost.MH is an asymmetric strategy, building separate two class models for each individual class against the pooled complement classes. Pooling classes can produce more complex decision boundaries that are difficult to approximate, while separating class pairs could be relatively simple [5]. Moreover, pseudo-multiclass algorithms might create resource problems by increasing (basically multiplying) *e.g.* training time or memory required, especially for problems with a large number of classes. Therefore, to overcome these drawbacks direct multiclass extensions of the AdaBoost method should be developed and investigated.

Multiclass classification

The first direct multiclass extension of the original AdaBoost algorithm, AdaBoost.M1, was introduced in [4] and is the most widely used multiclass boosting method. It is also the basis of many further variants of multiclass boosting. Similar to the binary AdaBoost methods, it can be used with any weak classifier that has an error rate of less than 0.5. However, this criterion is more restrictive than for binary classification, where an error rate of 0.5 means basically random guessing. The SAMME algorithm [24] overcomes this restriction by adding a constant taking the number of classes (C) into account, this way relaxing the requirement of the weak classifiers to an error rate of less than random guessing $(1 - \frac{1}{C})$. An evaluation on various benchmark datasets from the UCI repository showed that SAMME's performance is comparable with that of the AdaBoost.MH method, or even slightly better. The SAMME.R variation of the SAMME algorithm uses the probability estimates from the weak classifiers. However, SAMME.R showed overall slightly worse performance

Algorithm 1 ConfAdaBoost.M1

Require: Training dataset of N instances: $(\underline{x}_i, y_i) \ i = 1, \dots, N$ (\underline{x}_i : feature vector, $y_i \in [1, \dots, C]$)

New instance to classify: \underline{x}_n

```
1: procedure TRAINING( $(\underline{x}_i, y_i) \ i = 1, \dots, N$ )
2:   Assign equal weight to each training instance:  $w_i = \frac{1}{N}, i = 1, \dots, N$ 
3:   for  $t \leftarrow 1, T$  do
4:     Fit weak learner on the weighted dataset:  $f_t(\underline{x}) \in [1, \dots, C]$ 
5:     Compute the confidence of the prediction that instance  $\underline{x}_i$  belongs to the predicted class:  $p_{ti}, i = 1, \dots, N$ 
6:     Compute error  $e_t$  of model on weighted dataset:  $e_t = \sum_{i: y_i \neq f_t(\underline{x}_i)} p_{ti} w_i$ 
7:     if  $e_t = 0$  or  $e_t \geq 0.5$  then
8:       Delete last  $f_t(\underline{x})$  and terminate model generation.
9:     end if
10:    Compute  $\alpha_t = \frac{1}{2} \log \frac{1-e_t}{e_t}$ 
11:    for  $i \leftarrow 1, N$  do
12:       $w_i \leftarrow w_i e^{\left(\frac{1}{2} - \mathbb{I}(y_i = f_t(\underline{x}_i))\right) p_{ti} \alpha_t}$ 
13:    end for
14:    Normalize the weight of all instances so that  $\sum_i w_i = 1$ 
15:  end for
16: end procedure

17: procedure PREDICTION( $\underline{x}_n$ )
18:   Set zero weight to all classes:  $\mu_j = 0, j = 1, \dots, C$ 
19:   for  $t \leftarrow 1, T$  do
20:     Predict class with current model:  $[c, p_t(\underline{x}_n)] = f_t(\underline{x}_n)$ ,
      where  $p_t(\underline{x}_n)$  is the confidence of the prediction that instance  $\underline{x}_n$  belongs to the predicted class  $c$ 
21:      $\mu_c \leftarrow \mu_c + p_t(\underline{x}_n) \alpha_t$ 
22:   end for
23:   The output class is  $\arg \max_j \mu_j \quad j = 1, \dots, C$ 
24: end procedure
```

results than SAMME on different datasets [23]. Another multiclass boosting method is introduced in [7]: GAMBLE is the generalized version of the binary Gentle AdaBoost algorithm. However, GAMBLE fits a regression model rather than a classification model at each boosting iteration, thus requires several additional steps in order to be used for classification tasks (which is the actual focus of this work). First the class labels have to be encoded (*e.g.* with response encoding), then the regression model is fitted which is then used to obtain the weak classifier. Overall, the training time and computational cost is significantly increased compared to AdaBoost models using directly classification models.

CONFADABOOST.M1

Various boosting algorithms exist and were presented in the previous section. However, there are still classification problems where the difficulty of the task exceeds the potential of existing methods, *e.g.*, in the field of physical activity monitoring [13, 14]. Moreover, experiments presented in this paper show a high error rate on the PAMAP2 physical activity monitoring dataset with selected, commonly used boosting algorithms. Therefore, there is a need for further development of boosting techniques to improve the performance on such complex classification tasks.

This section introduces a new variant of the boosting algorithm, called ConfAdaBoost.M1. It is a confidence-based extension of the AdaBoost.M1 algorithm based on a combination of concepts and ideas used in the previously described

boosting methods. First of all it is a direct multiclass classification technique, thus it overcomes the drawbacks of pseudo-multiclass boosting methods. Moreover, it keeps the structure of AdaBoost.M1, thus when already using AdaBoost.M1 in a classification task it can be easily extended to ConfAdaBoost.M1. Furthermore, the new algorithm uses the information about how confident the weak learners are to predict the class of the instances. This approach has been beneficial in both binary (when developing the Real AdaBoost algorithm from Discrete AdaBoost in [5]) and pseudo-multiclass (the improvement of Discrete AdaBoost.MH to Real AdaBoost.MH in [19]) classification. Therefore, this work takes the next step by applying the idea of a confidence-based version of AdaBoost for the direct multiclass classification case. It is worth mentioning that [10] already proposed to modify the prediction step of the AdaBoost.M1 algorithm to allow the voting weights of the weak learners to vary in response to the confidence with which \underline{x}_n (the new instance to be classified) is classified. However, no confidence-based extension of the training part of the AdaBoost.M1 algorithm has previously been proposed.

The ConfAdaBoost.M1 algorithm is shown in Algorithm 1. The main idea of the new algorithm can be described as follows. First of all, after training the weak learner on the weighted dataset (line 4), the confidence of the classification estimation is returned for each instance by this weak learner (line 5). These p_{ti} confidence values are used when computing the error rate of the weak learner (line 6): the more

Table 1. Summary of the benchmark datasets used in the experiments

Dataset	Total	#Instances		#Variables	#Classes
		Training	Testing		
Glass	214	—	—	9	6
Iris	150	—	—	4	3
Vehicle	846	—	—	18	4
Letter	20000	16000	4000	16	26
Pendigits	10992	7494	3498	16	10
Satimage	6435	4435	2000	36	6
Segmentation	2310	210	2100	19	7
Thyroid	7200	3772	3428	21	3
PAMAP2_AR	19863	—	—	137	15
PAMAP2_IJ	24197	—	—	137	3

confident the model is in the misclassification the more that instance’s weight counts in the overall error rate. The factor $\frac{1}{2}$ on line 10 of the algorithm is used to compensate the lower e_t compared to the computed error rate of AdaBoost.M1. The p_{ti} confidence values are also used to recomputing the weights of the instances. The more confident the weak learner is in an instance’s correct classification or misclassification, the more that instance’s weight is reduced or increased, respectively (line 12). The factor $\frac{1}{2}$ on line 12 (determined in an empirical study) is applied in addition compared to the original AdaBoost.M1 algorithm, to compensate that weights are modified in both directions before the renormalization of the weights. In the prediction part of ConfAdaBoost.M1 the only modification compared to the AdaBoost.M1 algorithm is that the confidence of the prediction ($p_t(\underline{x}_n)$) is computed (line 20), and then used to adjust the voting weights of the weak learners. Therefore, the more confident the weak learner is in a new instance’s prediction, the more it counts in the output of the final combined classifier, as proposed by [10].

It should be noticed that the stopping criterion of $e_t \geq 0.5$ of the original AdaBoost.M1 remains the same in the proposed ConfAdaBoost.M1 algorithm (line 7). This means that, similar to AdaBoost.M1, only classifiers achieving a reasonably high accuracy value can be used as weak learners, thus *e.g.* decision stumps are not suitable for multiclass problems. However, the stopping criterion of $e_t \geq 0.5$ is less restrictive in ConfAdaBoost.M1, since the computation of the error rate also uses the p_{ti} confidence values, thus the computed e_t is lower. Therefore, when using the same weak learner, ConfAdaBoost.M1 can perform significantly more boosting iterations before stopping compared to AdaBoost.M1, as shown in the experiments of the next sections.

EVALUATION ON UCI DATASETS

In this section experiments on various datasets from the UCI repository [3] are presented. These experiments compare the newly introduced ConfAdaBoost.M1 algorithm to the most commonly used existing boosting methods. The first part of this section presents the basic conditions of the experiments, then results are given and discussed.²

Basic conditions

The experiments were performed on 8 datasets from the UCI repository. The selected benchmark datasets include 3 small datasets: *Glass*, *Iris* and *Vehicle*, as well as 5 pre-partitioned larger datasets: *Letter*, *Pendigits*, *Satimage*, *Segmentation* and *Thyroid*. These datasets were selected with the goal to

²To ensure that the experiments are easily reproducible, the supporting webpage [16] provides source code and links to all used datasets.

cover a wide range of scenarios: the size of the datasets ranges from 150 to 20 000 instances, the number of classes ranges from 3 to 26, and the difficulty of the classification problems they define vary a lot as well according to experiments performed in previous work (*cf. e.g.* [19] or [24]). The parameters of the used datasets are summarized in Table 1.

On the selected datasets, the ConfAdaBoost.M1 algorithm is compared to 4 other existing boosting methods. First of all to AdaBoost.M1 to provide the baseline performance of the experiments (since, as many other boosting variants, ConfAdaBoost.M1 is also an extension of AdaBoost.M1). The proposed confidence-based modification of the prediction step of AdaBoost.M1 in [10] is part of the ConfAdaBoost.M1 algorithm. Therefore, it is of interest to compare to this extension (which will be referred to as QuinlanAdaBoost.M1 hereafter) of the original AdaBoost.M1, to investigate whether possible performance improvements come from only the confidence-based prediction step or the confidence-based extension of both the training and prediction steps, as proposed by ConfAdaBoost.M1. The next boosting method used for comparison is SAMME, since according to [24] this direct multiclass extension of AdaBoost.M1 outperforms traditionally used boosting techniques. Finally, the most common pseudo-multiclass classification technique (Real AdaBoost.MH) is used for comparison, since it performs best amongst the pseudo-multiclass methods and is a confidence-based boosting version similar to ConfAdaBoost.M1.

The C4.5 decision tree classifier is used as weak learner in each of the evaluated boosting methods. This classifier is, together with decision stumps, the most commonly used weak learner for boosting, and fulfills the requirement of achieving a reasonably high accuracy on the different classification problems (it has an error rate of significantly less than 0.5 on the various datasets, as shown below by the results). Considering confidence-based versions of AdaBoost, the C4.5 decision tree has another benefit: there is no need to modify the C4.5 algorithm, the confidence values of the weak learners’ predictions can be directly extracted from the trained decision trees. Assume that a C4.5 decision tree is trained as $f_t(\underline{x})$ weak learner in ConfAdaBoost.M1 (Algorithm 1, line 4). The p_{ti} confidence of the prediction that instance \underline{x}_i belongs to the predicted class can be computed as follows, based on [10]. In the trained C4.5 decision tree a single leaf node classifies \underline{x}_i : $c = f_t(\underline{x}_i)$. Let S be the training instances mapped to this leaf, and let S_c be the subset of S belonging to class c . The confidence of the prediction is then:

$$p_{ti} = \frac{\sum_{j \in S_c} w_j}{\sum_{j \in S} w_j}. \quad (1)$$

On the 5 larger, pre-partitioned datasets pruned C4.5 decision trees are used. The level of pruning is defined by 5-fold cross-validation (CV) on the training part of these datasets, for each of the evaluated boosting methods separately. On the 3 smaller datasets (*Glass*, *Iris* and *Vehicle*), non-pruned C4.5 decision trees are used as weak learners. Between 1 and 500 boosting iterations are evaluated for all algorithms and benchmark datasets. All results presented below are averages of multiple test runs. On datasets providing a training and test

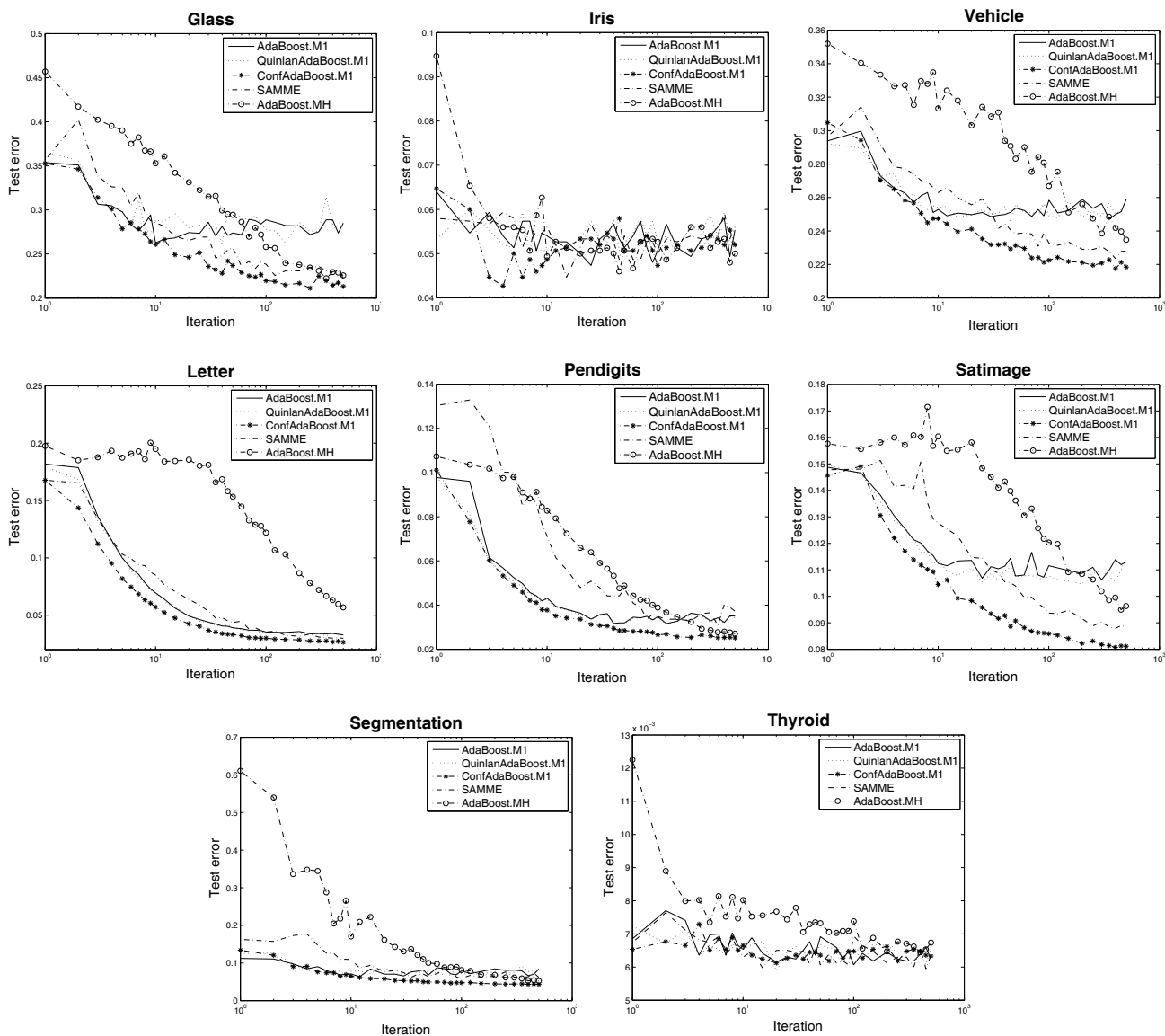


Figure 1. Test error of the 5 evaluated boosting algorithms on the UCI benchmark datasets. The results are averages over 10 test runs.

part training is performed 10 times on the training set, and the trained classifier is then evaluated on the provided test set each time. On datasets without a predefined test part, 10-fold CV is used and performed 10 separate times.

Results and discussion

Figure 1 shows the results on the selected UCI datasets, the test errors are summarized in Table 2. Overall it is clear that the ConfAdaBoost.M1 algorithm performed best in the experiments: on 7 out of 8 datasets there is a noticeable increase in performance compared to existing boosting methods, while on one dataset (Thyroid) ConfAdaBoost.M1 has essentially the same performance as the other algorithms. According to the results of Table 2, the second best boosting algorithm is SAMME, closely followed by AdaBoost.MH, confirming the results of [24]. The original AdaBoost.M1 and its variation QuinlanAdaBoost.M1 performed overall clearly worse, the latter algorithm being slightly but not significantly better.

A statistical significance test (the McNemar test [8] is used to pair-wise compare the predictions of the different methods) indicates that the reduction of the test error rate by ConfAdaBoost.M1 compared to SAMME is significant with p -value 0.01 on the datasets Pendigits and Segmentation, significant with p -value 0.05 on the datasets Letter and Satimage, and that on the remaining datasets no statistical significance was observed. In conclusion, the ConfAdaBoost.M1 algorithm has more potential for improvement the larger the dataset and the more complex the classification problem is. This statement is supported by the results on the PAMAP2 classification tasks in the next section. On the Thyroid dataset on the other hand even AdaBoost.M1 reaches an accuracy of over 99% leaving only a few outlier instances misclassified, thus explaining the minimal (not statistically significant) difference between the results of the 5 algorithms.

One of the main reasons why AdaBoost.M1 and QuinlanAdaBoost.M1 performs significantly worse than the other

Table 2. Comparison of the 5 evaluated boosting algorithms: test error rates [%] on the selected benchmark datasets. The results are averaged over 10 test runs (mean and standard deviation are given), the best performance is shown for each of the methods.

Dataset	AdaBoost.M1	Quinlan-AdaBoost.M1	Conf-AdaBoost.M1	SAMME	AdaBoost.MH
Glass	26.26 ± 1.42	26.17 ± 2.60	21.12 ± 1.22	22.29 ± 1.38	22.24 ± 1.81
Iris	4.73 ± 0.73	5.00 ± 0.85	4.27 ± 0.64	4.47 ± 1.22	4.60 ± 0.80
Vehicle	24.72 ± 1.05	24.52 ± 1.10	21.75 ± 0.44	22.35 ± 1.14	23.48 ± 1.21
Letter	3.28 ± 0.14	3.19 ± 0.15	2.64 ± 0.11	2.99 ± 0.13	5.68 ± 0.39
Pendigits	3.16 ± 0.27	3.14 ± 0.45	2.51 ± 0.11	3.08 ± 0.15	2.70 ± 0.08
Satimage	10.63 ± 0.80	10.47 ± 1.01	8.07 ± 0.15	8.79 ± 0.25	9.50 ± 0.39
Segmentation	6.36 ± 1.03	6.55 ± 1.08	4.31 ± 0.20	5.92 ± 0.79	5.22 ± 0.78
Thyroid	0.61 ± 0.04	0.59 ± 0.05	0.61 ± 0.05	0.60 ± 0.06	0.64 ± 0.08
PAMAP2_AR	29.28 ± 1.40	27.90 ± 1.06	22.22 ± 0.77	27.98 ± 1.34	—
PAMAP2_IE	7.98 ± 1.04	7.73 ± 0.66	5.60 ± 0.31	7.81 ± 0.60	—

methods is that they reach the stopping criterion of $e_t \geq 0.5$ quickly. This can be observed especially on the results of the datasets Glass, Vehicle or Satimage: the test error decreases at the beginning but levels off already at around 10 to 20 boosting iterations, no further improvement can be reached with the increase of the number of boosting rounds. This effect is not observed when using the ConfAdaBoost.M1 algorithm due to the modified computation of the error rate of the weak learners. Another benefit of ConfAdaBoost.M1 over the other methods can be observed *e.g.* on the results of the datasets Vehicle, Letter and Pendigits: the test error even at lower numbers of boosting iterations is the lowest when using ConfAdaBoost.M1. This means that for a particular level of accuracy fewer boosting rounds are necessary with ConfAdaBoost.M1, thus a smaller classifier size is required for the same performance compared to existing boosting algorithms. This quality is especially beneficial when the available computational resources are limited.

EVALUATION ON THE PAMAP2 DATASET

The PAMAP2 dataset is a physical activity monitoring dataset created and released recently [13, 14], and is included in the UCI machine learning repository as well. The dataset was recorded from 18 physical activities performed by 9 subjects, wearing 3 inertial measurement units (IMU) and a heart rate monitor. Each of the subjects followed a predefined data collection protocol of 12 activities (lie, sit, stand, walk, run, cycle, Nordic walk, iron, vacuum clean, rope jump, ascend and descend stairs), and optionally performed a few other activities (watch TV, computer work, drive car, fold laundry, clean house, play soccer). Therefore, the PAMAP2 dataset not only includes basic physical activities and postures, but also a wide range of everyday, household and fitness activities. A more detailed description of the dataset can be found in [13]. In this section first an activity recognition and an intensity estimation classification problem is defined on the PAMAP2 dataset. These classification tasks are described in detail, highlighting also the differences to the UCI benchmark datasets of the previous section and pointing out the special challenge these problems pose. Using the defined classification tasks different boosting methods are evaluated and compared to the proposed ConfAdaBoost.M1 algorithm.

Definition of the classification problems

The benchmark of [13, 14] defined 4 different classification problems on the PAMAP2 dataset. One of these problems

— called *All activity recognition task* — uses the 12 activities of the data collection protocol, defining 12 classes corresponding to the activities. This classification task is extended in this paper with 3 additional activities from the optional activity list: fold laundry, clean house and play soccer.³ This activity recognition task of 15 different activity classes will be referred to as the ‘PAMAP2_AR’ task throughout this work. Moreover, an intensity estimation classification task is defined on the PAMAP2 dataset: using all 18 activities, the goal is to distinguish activities of light, moderate and vigorous effort (referred to as ‘PAMAP2_IE’ task). The ground truth for this rough intensity estimation task is based on the metabolic equivalent (MET) of the different physical activities, provided by [1]. Therefore, the 3 intensity classes are defined as follows: lie, sit, stand, drive car, iron, fold laundry, clean house, watch TV and computer work are regarded as activities of light effort (< 3.0 METs); walk, cycle, descend stairs, vacuum clean and Nordic walk as activities of moderate effort (3.0-6.0 METs); run, ascend stairs, rope jump and play soccer as activities of vigorous effort (> 6.0 METs).

Contrary to the 8 UCI benchmark datasets used for the experiments in the previous section, the PAMAP2 dataset does not directly provide a feature vector with each of the instances, but only provides raw sensory data from the 3 IMUs and the heart rate monitor. Therefore, the raw signal data needs to be processed first in order to be used by classification algorithms. A data processing chain is applied on the raw sensory data including preprocessing, segmentation and feature extraction steps (these data processing steps are further described in [13]). In total, 137 features are extracted: 133 features from IMU acceleration data (such as mean, standard deviation, energy, entropy, correlation, etc.) and 4 features from heart rate data. These extracted features serve as input to the classification step, in which different boosting algorithms are evaluated.⁴ The main parameters of the PAMAP2 classification tasks are summarized in Table 1. It is clear that, compared to the other datasets of Table 1, the classification problems defined on the PAMAP2 dataset are significantly

³The remaining 3 activities from the dataset are discarded for the following reasons: *drive car* contains data from only one subject, while *watch TV* and *computer work* are not considered due to their high resemblance to the *sit* class.

⁴The feature matrices of the PAMAP2_AR and PAMAP2_IE tasks are provided at the supporting webpage [16].

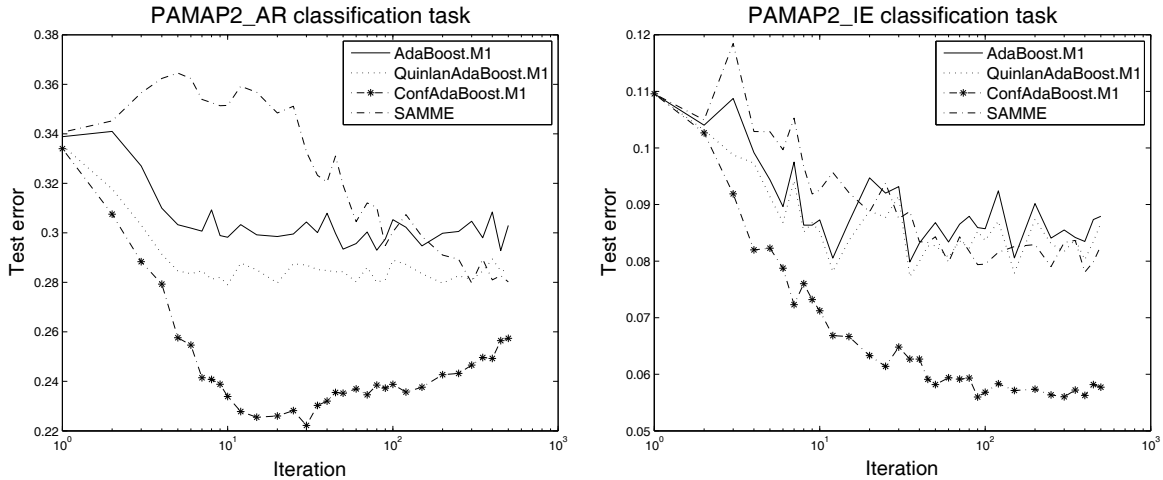


Figure 2. Test error of the 4 evaluated boosting algorithms on the PAMAP2 classification tasks. The results are averages over 10 test runs.

more complex, considering the number of instances and especially the number of variables. To get a first impression about the difficulty of these tasks, experiments with a C4.5 decision tree classifier are performed: 65.79% is reached on the PAMAP2_AR and 88.98% on the PAMAP2_IE task, averaged over 10 test runs. This result serves as baseline performance, showing that improvement is required and to be expected while applying different boosting methods.

The experiments presented below in this section compare the ConfAdaBoost.M1 algorithm to the boosting methods AdaBoost.M1, QuinlanAdaBoost.M1 and SAMME.⁵ Similar to the previous section, the C4.5 decision tree classifier is used for each of the boosting algorithms as weak learner. An important difference in the realization of the experiments in this section is the applied evaluation technique. As discussed in [15], a subject independent validation technique simulates best the goals of systems and applications using physical activity recognition. Therefore, leave-one-subject-out (LOSO) 9-fold cross-validation is used in this section, while evaluating each method from 1 up to 500 boosting iterations.

Results and discussion

The averaged results of the 10 test runs on the PAMAP2 classification tasks are shown in Figure 2, the test error rates of the 4 evaluated boosting methods are included in Table 2. Compared to the baseline accuracy of the decision tree classifier, all boosting methods significantly improve the performance. The ConfAdaBoost.M1 algorithm clearly outperforms the other methods: *e.g.* on the PAMAP2_AR task, compared to the performance of the second best SAMME algorithm a reduction of the test error rate by nearly 20% can be observed. This reduction of the test error rate is statistically significant with a p -value smaller than 0.001.

Similar to the results of Figure 1, the algorithms AdaBoost.M1 and QuinlanAdaBoost.M1 reach the stopping criterion at lower boosting iteration numbers. However, contrary

⁵AdaBoost.MH is not considered here due to the unfeasible training time it would require, given the complexity of the classification tasks and that the actual size of the training set is a multiple of that of the other algorithms.

to the results of the previous section, QuinlanAdaBoost.M1 performs significantly better here, confirming that it is even worth to apply the confidence-based modification to only the prediction step of the original AdaBoost.M1 algorithm. However, compared to QuinlanAdaBoost.M1, ConfAdaBoost.M1 reduces the test error rate by 20%, thus the major part of the performance improvement achieved by ConfAdaBoost.M1 comes from the confidence-based extension of both the training and prediction step of the original AdaBoost.M1 algorithm, as also confirmed by the results on the 8 other UCI datasets. Therefore, ConfAdaBoost.M1 is clearly a significant improvement over QuinlanAdaBoost.M1. ConfAdaBoost.M1 also adopts one of the beneficial characteristics of boosting: it rarely overfits a classification problem. The only result indicating overfitting is on the PAMAP2_AR task, the reasons need further investigation. Nevertheless, even with higher numbers of boosting iterations, the performance on the PAMAP2_AR task of ConfAdaBoost.M1 is significantly better than that of the other evaluated boosting methods.

To better understand the results of this section, the confusion matrix of the best performing classifier (ConfAdaBoost.M1 with 30 boosting iterations) on the PAMAP2_AR task is presented in Table 3.⁶ The numbering of the activities in the table corresponds to the activity IDs as given in the PAMAP2 dataset. The results are averaged over 10 test runs, the overall accuracy is 77.78%. The confusion matrix shows that some activities are recognized with high accuracy, *e.g.* lie, walk or even distinguishing between ascend and descend stairs. Misclassifications in Table 3 have several reasons. For example, the over 5% confusion between sit and stand can be explained with the positioning of the sensors: an IMU on the thigh would be needed for a reliable differentiation of these postures. Moreover, ironing has a similar characteristics from the used set of sensors' point of view, especially compared

⁶Recently new error metrics were introduced for continuous activity recognition, *e.g.* insertion, merge, overflow, etc [20, 22]. However, contrary to activity recognition in *e.g.* home or industrial settings, for physical activity monitoring the frame by frame metrics (precision, recall, F-measure and accuracy: all derivable from the confusion matrix) are sufficient, as discussed in [13].

Table 3. Confusion matrix of the PAMAP2_AR classification task using the ConfAdaBoost.M1 classifier and 30 boosting iterations. The table shows how different annotated activities are classified in [%].

Annotated activity	Recognized activity																							
	1	2	3	4	5	6	7	12	13	16	17	18	19	20	24									
1 lie	97.1	1.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.9	0.0	0.0								
2 sit	2.0	84.8	5.4	0.0	0.0	0.5	0.0	0.0	0.0	0.1	4.1	0.6	2.5	0.0	0.0									
3 stand	0.0	6.0	83.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	7.4	0.9	2.4	0.0	0.0									
4 walk	0.0	0.0	0.0	92.2	0.0	0.0	0.5	6.8	0.0	0.0	0.0	0.0	0.0	0.4	0.0									
5 run	0.0	0.0	0.0	0.0	89.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	10.1	0.0									
6 cycle	0.0	0.0	0.0	1.1	0.0	91.7	0.4	0.5	0.0	1.3	0.1	0.0	4.9	0.0	0.0									
7 Nordic walk	0.0	0.0	0.0	2.7	0.0	0.0	89.1	1.1	0.1	0.0	0.0	0.0	0.1	7.0	0.0									
12 asc. stairs	0.0	0.0	0.0	6.4	0.0	0.2	0.3	87.2	2.6	0.7	0.0	0.0	0.3	2.5	0.0									
13 desc. stairs	0.0	0.0	0.0	0.1	0.1	0.0	0.2	6.7	94.8	0.0	0.0	0.0	0.2	0.7	0.7									
16 vacuum clean	0.0	0.0	0.1	0.0	0.0	1.1	0.0	0.3	0.4	73.5	1.3	0.3	23.1	0.0	0.0									
17 iron	0.0	2.6	0.8	0.0	0.0	0.0	0.0	0.0	0.0	1.2	77.7	5.0	12.7	0.0	0.0									
18 fold laundry	0.0	1.1	1.5	0.0	0.0	0.1	0.0	0.0	0.0	8.9	61.1	11.1	16.2	0.0	0.0									
19 clean house	0.5	0.6	3.4	0.0	0.0	1.7	0.0	1.2	0.7	21.4	18.1	5.0	47.4	0.0	0.0									
20 play soccer	0.0	0.0	0.0	5.1	27.6	1.4	2.8	7.3	20.6	1.7	0.0	0.0	0.1	20.7	12.8									
24 rope jump	0.0	0.0	0.0	0.0	32.0	0.0	0.0	1.3	0.1	0.0	0.0	0.0	0.0	7.8	58.8									

to talking and gesticulating during standing. Another example of overlapping activity characteristics comes from the introduction of playing soccer into this classification problem. Playing soccer is a composite activity, and it is for instance not trivial to distinguish running with a ball from just running. The significant confusion between the different household activities (vacuum clean, iron, fold laundry and clean house — the latter mainly consisting of dusting shelves) indicates that they can not be reliably distinguished with the given set of sensors. However, arguably, the main reason for the misclassifications in Table 3 is the diversity in how subjects perform physical activities. Therefore, to further increase the accuracy of physical activity recognition, personalization approaches should be introduced and investigated.

CONCLUSION

This paper presented a confidence-based extension of the well-known AdaBoost.M1 algorithm, called ConfAdaBoost.M1. The new algorithm builds on established ideas of existing boosting methods, combining some of their benefits. The ConfAdaBoost.M1 algorithm has been evaluated on various benchmark datasets, comparing it to the most commonly used boosting techniques. ConfAdaBoost.M1 performed significantly best among these algorithms, especially on the larger and more complex activity monitoring problems: on the PAMAP2_AR task the test error rate was reduced by nearly 20% compared to the second best performing classifier. Therefore, the main motivation of proposing this new boosting variant — namely to overcome some of the challenges defined by recent benchmark results in physical activity monitoring — was achieved successfully.

The main concepts of the ConfAdaBoost.M1 algorithm are clear and comprehensible, but a theoretical interpretation of the algorithm and explanation of its success remains for future work. Moreover, it is also planned to slightly modify ConfAdaBoost.M1 to loosen the stopping criterion of $e_t \geq 0.5$, thus allowing the usage of “weak” weak learners (such as decision stumps). However, boosting decision trees proved to be very successful in the experiments presented in this work, and will remain (due to its many benefits discussed in this paper) one of the most widely used classifiers especially in the field of physical activity monitoring.

Acknowledgements

This work has been performed within the project Activity-Plus, funded by the “Stiftung Rheinland-Pfalz für Innovation”

under contract number 961 - 386261/1028, and the European project AlterEgo under contract number 600610.

REFERENCES

- Ainsworth, B. E., Haskell, W. L., Whitt, M. C., Irwin, M. L., Swartz, a. M., Strath, S. J., O'Brien, W. L., Bassett, D. R., Schmitz, K. H., Emplainscourt, P. O., Jacobs, D. R., and Leon, a. S. Compendium of physical activities: an update of activity codes and MET intensities. *Medicine and science in sports and exercise* 32, 9 (2000), 498–504.
- Ermes, M., Pärkkä, J., and Cluitmans, L. Advancing from offline to online activity recognition with wearable sensors. In *Proc. 30th Annual International IEEE EMBS Conference* (2008), 4451–4454.
- Frank, A., and Asuncion, A. UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences, 2010.
- Freund, Y., and Schapire, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences* 55 (1997), 119–139.
- Friedman, J., Hastie, T., and Tibshirani, R. Additive Logistic Regression: A Statistical View of Boosting. *Annals of Statistics* 28 (2000), 337–407.
- Gómez-Verdejo, V., Ortega-Moral, M., Arenas-García, J., and Figueiras-Vidal, A. R. Boosting by weighting critical and erroneous samples. *Neurocomputing* 69, 7-9 (2006), 679–685.
- Huang, J., Ertekin, S., Song, Y., Zha, H., and Giles, C. L. Efficient Multiclass Boosting Classification with Active Learning. In *SIAM International Conference on Data Mining (SDM)* (2007).
- Kuncheva, L. I. Combining Pattern Classifiers: Methods and Algorithms. *Wiley-Interscience* (2004).
- Long, X., Yin, B., and Aarts, R. M. Single-accelerometer based daily physical activity classification. In *Proc. 31st Annual International IEEE EMBS Conference* (2009), 6107–6110.
- Quinlan, J. R. Bagging, boosting and C4.5. In *Proc. AAAI* (1996), 725–730.
- Ravi, N., Dandekar, N., Mysore, P., and Littman, M. Activity recognition from accelerometer data. In *Proc. IAAI* (2005), 1541–1546.
- Reiss, A., and Stricker, D. Introducing a Modular Activity Monitoring System. In *Proc. 33rd Annual International IEEE EMBS Conference* (2011), 5621–5624.
- Reiss, A., and Stricker, D. Creating and Benchmarking a New Dataset for Physical Activity Monitoring. In *Proc. 5th Workshop on Affect and Behaviour Related Assistance (ABRA)* (2012).
- Reiss, A., and Stricker, D. Introducing a New Benchmarked Dataset for Activity Monitoring. In *Proc. ISWC* (2012).
- Reiss, A., Weber, M., and Stricker, D. Exploring and Extending the Boundaries of Physical Activity Recognition. In *Proc. IEEE SMC Workshop on Robust Machine Learning Techniques for Human Activity Recognition* (2011), 46–50.
- Reiss, A. Confidence-based Multiclass AdaBoost for Physical Activity Monitoring. Supporting Webpage (2013-05-27). www.sites.google.com/site/iswc2013confadaboost/.
- Schapire, R. E. The strength of weak learnability. *Machine Learning* 5, 2 (1990), 197–227.
- Schapire, R. E. Using output codes to boost multiclass learning problems. In *Proc. ICML* (1997), 313–321.
- Schapire, R. E., and Singer, Y. Improved boosting algorithms using confidence-rated predictions. *Machine learning* 37, 3 (1999), 297–336.
- van Kasteren, T., Alemdar, H., and Ersoy, C. Effective Performance Metrics for Evaluating Activity Recognition Methods. In *Proc. ARCS* (2011).
- Vezhnevets, A., and Vezhnevets, V. Modest AdaBoost - Teaching AdaBoost to Generalize Better. In *Proc. Graphicon* (2005).
- Ward, J. A., Lukowicz, P., and Gellersen, H. W. Performance Metrics for Activity Recognition. *ACM Transactions on Intelligent Systems and Technology* 2, 1 (2011).
- Zhu, J., Rosset, S., Zou, H., and Hastie, T. Multi-class Adaboost. *Technical Report 430, Department of Statistics, University of Michigan* (2005).
- Zhu, J., Zou, H., Rosset, S., and Hastie, T. Multi-class Adaboost. *Statistics and Its Interface* 2 (2009), 349–360.