

# WiFiBoost: a terminal-based method for detection of indoor/outdoor places

Oscar Canovas  
DITEC  
University of Murcia  
Murcia, Spain  
ocanovas@um.es

Pedro E. Lopez-de-Teruel  
DITEC  
University of Murcia  
Murcia, Spain  
pedroe@um.es

Alberto Ruiz  
DIS  
University of Murcia  
Murcia, Spain  
aruiz@um.es

## ABSTRACT

We present a binary classifier named WiFiBoost which makes use of RSSIs from the existing 802.11 access points (AP) in order to infer whether a device is inside a building. Our weak learner is based on the feature of RSSI comparison between pairs of APs. The resulting model is really compact and efficient, and it can be stored in the mobile device. Our classifier also minimizes the negative effect of unpredicted AP failures or changes in the network infrastructure, which will result in a moderate impact on the classification error under those circumstances.

## Categories and Subject Descriptors

H.4 [Information Systems Applications]: Miscellaneous;  
I.2.6 [Artificial Intelligence]: Learning

## Keywords

AdaBoost, binary classification, indoor localization, RSSIs

## 1. INTRODUCTION

Mobile devices are shipped with several sensors that can be used to infer context information. However, a preliminary step for most applications is to infer whether the device is inside a building [2] supported by, for example, a localization service. That would prevent us from activating unnecessary sensors and from performing useless calculations. Many systems simply assume that the indoor/outdoor condition has been known in some way, which barely holds in practice.

Our proposal, called WiFiBoost, is based on AdaBoost [1] (short for Adaptive Boosting) and it is especially designed to be run on mobile devices. It relies only on the received signal strength indicators (RSSIs) obtained from the WiFi beacons of the existing access points (APs). A binary classifier is built using the fingerprinting map obtained during the training phase of the existing indoor localization services. The keystone of the boosting process is the definition of good weak learners, which individually can be weak but their performance must be slightly better than random

guessing. Our learner is mainly based on the feature of relative comparison between measured APs in order to solve the problem of device heterogeneity. Our algorithm supports the inclusion of several weak learners per stage to mitigate the effects derived from APs that cannot be received any longer, due to temporal/permanent failures. We present also some results obtained from tests where different techniques and failure conditions were analysed.

## 2. WIFIBOOST

WiFiBoost takes as input a training set  $(x_1, y_1), \dots, (x_n, y_n)$  where each  $x_i$  belongs to a  $R^m$  space containing RSSI measurements from  $m$  existing access points, for a particular environment. Each label  $y_i$  is in some label set  $Y = (+1, -1)$ , where  $-1$  is used for outdoor and  $+1$  for indoor observations.

The algorithm maintains a distribution or set of weights over the training set. Initially all weights are set equally, but on each round the weights of incorrectly classified examples are increased so that the weak learner is focused on the hard observations. The final hypothesis  $H$  is a weighted majority vote for the  $T$  weak hypothesis, where  $\alpha_t$  is the weight assigned to  $h_t : X \rightarrow \{-1, 0, +1\}$ .

In our case, every training sample is constituted by set of  $m$  RSSIs from the set  $A$  of  $m$  different access points that were observed during the offline phase. Those access points are located both indoor and outdoor, but it is not required to know their locations. To address the problem of device heterogeneity, we define a weak learner which is based on relative comparisons instead of the absolute values of RSSs. This classifier (Alg. 1) is defined taking into account all the possible pairs obtained from  $A$ . The idea behind this classifier is that if we compare all the access points by pairs  $(a_i, a_j)$ , we might find several pairs acting as linear classifiers which can be used to separate outdoor/indoor locations.

---

### Algorithm 1 : WiFiBoost Weak Learner

---

**Access Points:**  $a_i, a_j \in A$  **Sample:**  $x = (r_1..r_m, label)$   
**if**  $(r_i = 0$  and  $r_j = 0)$  **then**  $h(x) = 0$ ;  
**else if**  $(r_i > r_j)$  **then**  $h(x) = 1$ ;  
**else if**  $(r_i < r_j)$  **then**  $h(x) = -1$   
**else**  $h(x) = 0$

---

The number of unique access points in the final AdaBoost classifier tends to be low since this technique selects those weak classifiers that minimize the classification error. That might involve a serious problem during the online phase in case some of the access points composing the final classifier are disconnected or temporally down, resulting in a high

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. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MOBIQUITOUS 2014, December 02-05, London, Great Britain

Copyright © 2014 ICST 978-1-63190-039-6

DOI 10.4108/icst.mobiquitous.2014.258063

error rate during the failure. In order to address this issue, WiFiBoost (Algorithm 2) is able to add a number  $s$  of learners per stage. It is also worth noting that the final classification is an efficient calculation consisting of a weighted majority vote for the weak  $sT$  hypothesis.

---

**Algorithm 2** : WiFiBoost

---

Samples:  $x_1..x_n$   
Desired outputs:  $y_1..y_n, y \in \{-1, 1\}$   
Initial weights:  $w_{1,0}..w_{n,0}$  set to  $\frac{1}{n}$   
Error function:  $E(f(x), y, i) = e^{-y_i f(x_i)}$   
Weak learners:  $h(x) = \{-1, 0, 1\}$   
Number of learners per stage:  $s$   
**for**  $t$  in  $1..T$  **do**  
Sort  $H$  in ascent order of  $\epsilon_t = \sum_i w_{i,t} e^{-y_i h(x_i)}$   
Choose the first  $s \in H$  learners with disjoint APs:  $H_s$   
For every  $h_t \in H_s$  set  $\alpha_t = \frac{1}{2} \ln \frac{1-\epsilon_t}{\epsilon_t}$   
Generate stage classifier  $S_t(x) = \sum_{h_t \in H_s} \alpha_t h_t(x)$   
Update weights  $\forall i w_{i,t+1} = w_{i,t} e^{-\alpha_t y_i S_t(x_i)}$   
Renormalize  $w_{i,t+1}$  such that  $\sum_i w_{i,t+1} = 1$   
Calculate  $\epsilon_{S_t} = \sum_i w_{i,t} e^{-y_i S_t(x_i)}$   
Calculate  $\alpha_{S_t} = \frac{1}{2} \ln \frac{1-\epsilon_{S_t}}{\epsilon_{S_t}}$   
For every  $h_t \in H_s$  set  $\alpha_t = \alpha_{S_t}$   
Add to final classifier  $F_t(x) = F_{t-1}(x) + \sum_{h_t \in H_s} \alpha_t h_t(x)$   
**end for**  
Output the final hypothesis:  $H(x) = \text{sign}(\sum_{t=1}^{sT} \alpha_t h_t(x))$

---

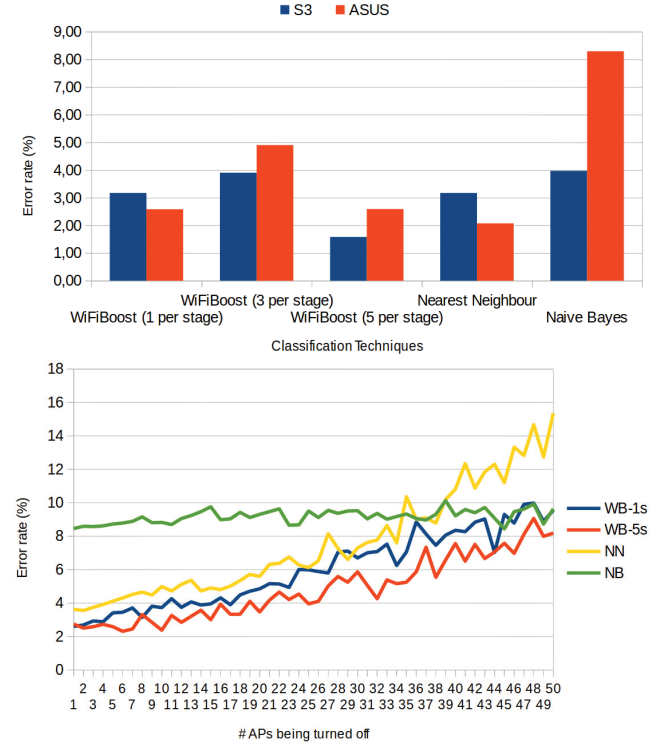
### 3. EXPERIMENTAL RESULTS

We have tested WiFiBoost in several facilities located in our campus but, due to the lack of space, we include information about one building. The training database is composed by 1,735 observations (890 indoor and 845 outdoor). There are 135 different access points and we used different devices which exhibit a completely different behaviour in relation to the RSSIs values and the sampling frequency to test the support for device heterogeneity.

When only one learner is included in every iteration, only 27 weak learners are required and only 32 different APs are selected out of 135. When three and five learners are added per stage we have 49 and 64 different APs in the final classifier respectively. As Figure 1(a) shows, that increment in the number of APs does not affect the average performance of our proposal. In fact, performance is improved selecting 5 different learners per iteration. In order to compare the performance we obtain using WiFiBoost, we made some tests making use of other well-known techniques such as nearest neighbour(NN) and naive Bayes(NB). Figure 1(a) shows that, in the absence of failures, NN obtains the best results but they are quite similar to those obtained by WiFiBoost. However, NN imposes higher requirements in terms of memory storage and execution time in relation to WiFiBoost. Naive Bayes provides higher error rates.

In order to check whether our approach is fault tolerant, we simulated random failures of an increasing number of APs by removing the RSS values in the original test samples. Figure 1(b) shows the average error values using different classification techniques. The WiFiBoost variation using 5

learners per stage always behaves better than the one adding one learner per stage. Therefore, the increment of memory required to store the model is compensated by the lower error rates under those circumstances. It also outperforms the rest of techniques.



**Figure 1:** (a) Performance of WiFiBoost; (b) Error rate in function of the number of APs down

### 4. CONCLUSIONS

WiFiBoost is an efficient and fast binary classifier which combines several weak learners mainly based on the feature of relative comparisons between measured pairs of access points. It is able to mitigate the implications derived from faults caused by APs that cannot be received any longer. We compared our proposal with other well known techniques such as Nearest Neighbour and Naive Bayes, showing that even with a more compact classification model we are able to provide a better performance in terms of error rates.

### 5. ACKNOWLEDGMENTS

This work was supported by the Spanish MINECO, as well as FEDER funds, under grant TIN2012-38341-C04-03

### 6. REFERENCES

- [1] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Computational learning theory*, pages 23–37. Springer, 1995.
- [2] P. Zhou, Y. Zheng, Z. Li, M. Li, and G. Shen. Iodetector: A generic service for indoor outdoor detection. In *Proc. of the ACM Conf. on Embedded Network Sensor Systems*, pages 113–126, 2012.