# Resource-Aware On-line RFID Localization Using Proximity Data

Christoph Scholz[1], Stephan Doerfel[1],
Martin Atzmueller[1], Andreas Hotho[2], and Gerd Stumme[1]

[1] Knowledge & Data Engineering Group, University of Kassel,
34121 Kassel, Germany
{scholz,doerfel,atzmueller,stumme}@cs.uni-kassel.de
[2] Data Mining and Information Retrieval Group, University of Würzburg,
D-97074 Würzburg, Germany
hotho@informatik.uni-wuerzburg.de

**Abstract.** This paper focuses on resource-aware and cost-effective indoor-localization at room-level using RFID technology. In addition to the tracking information of people wearing active RFID tags, we also include information about their proximity contacts. We present an evaluation using real-world data collected during a conference: We complement state-of-the-art machine learning approaches with strategies utilizing the proximity data in order to improve a core localization technique further.

## 1 Introduction

While approaches for outdoor localization can utilize various existing global sources, e.g., GPS signals, mobile broadcasting signals, or wireless network signatures, methods for indoor localization usually apply special installations (e.g., RFID or Bluetooth readers), or require extensive training and calibration efforts.

In this paper, we propose an approach for indoor localization using active RFID technology: We focus on a cost-effective and resource-aware solution that requires only a small number of RFID readers (Figure 1). Furthermore, our method can be applied to installations, where readers cannot be positioned freely. The latter constraint is encountered often, especially in buildings under monumental protection. Our application context is a conference, where conference participants wear active RFID tags for tracking, for memorizing their contact information, and for the personalization of conference services. Therefore, we present an analysis of data collected in a real-life context, in contrast to scenarios that examine RFID localization in laboratory experiments, e.g., [18][12]. In Section 3.1 we discuss additional challenges, that such an application faces and that are difficult to implement in simulated scenarios. We consider a real-life localization problem at room-level, i. e., the task to determine the room, that a person is in at a given point in time.

Our contribution is three-fold: We present an analysis of the contact and proximity data in order to prove the validity and applicability for the sketched application. Additionally, we evaluate the benefits of several state-of-the-art machine

D. Gunopulos et al. (Eds.): ECML PKDD 2011, Part III, LNAI 6913, pp. 129–144, 2011.

learning techniques for predicting the locations of participants at the room-level. We propose to utilize the (proximity) contacts of participants for improving the predictions of a given core localization algorithm. We evaluate the impact of different strategies considering the top performing machine learning algorithm. The real-world evaluation data was collected at the LWA 2010 conference (of the German Association of Computer Science) in Kassel, Germany[1].

The rest of the paper is structured as follows: Section 2 discusses related work. After that, Section 3 describes the approach for resource-aware localization at room-level using RFID technology. Next, Section 4 features the evaluation of the approach utilizing several machine learning algorithms and different strategies for implementing the proximity contacts. Finally, Section 5 concludes the paper with a summary and interesting directions for future research.

## 2 Related Work

The Global Positioning System (GPS) is the most widely used localization system for outdoor positioning. It is based on a network of 24-30 satellites placed in the orbit. One of the drawbacks of GPS is that it cannot be used for indoor localization, because its signals are blocked by most construction materials. Therefore, the research on indoor localization systems has received great interest during the last decade.

For indoor localization several algorithms have been proposed, usually based on angle of arrival (AoA) [19], time of arrival (ToA) [15] or time difference of arrival (TDoA) [20] methodologies. On the one hand, these methods are highly accurate in estimating the position of an object; on the other hand they consume a lot of energy. Furthermore, they require expensive hardware and an extensive deployment of suitable infrastructure. Another class of localization algorithms estimates the position of a target based on the received signal strength [12]. Most of these approaches use the log-distance path loss model [21] to estimate the distance from the object to at least three reference points. Then, the possible position of the object is calculated using triangulation. The disadvantage of this approach is that propagation effects such as reflection, multi-path-fading or phase-fluctuations limit the precision of positioning.

Scene analysis is another option to estimate the position of an object [4][18][6]. Usually, this technique works in two phases, the off-line learning phase and the online localization phase. In the off-line phase, data about the received signal strengths (RSS) for each point in the localization area is stored in a database to save the localization points. In the online phase two scene analysis techniques of predicting the position exist: $k$-nearest-neighbor (KNN) and probabilistic methods. KNN predicts the position by finding the $k$ closest fingerprints in the database. The estimated location is the (weighted) centroid of the corresponding $k$ locations. The probabilistic model selects the location with the highest probability.

[1] http://www.kde.cs.uni-kassel.de/conf/lwa10/

In our experiments, we consider a different approach using a new generation of cost-effective and resource-aware RFID tags, i.e., tags with a low power consumption. These RFID tags (proximity tags) are developed by the SocioPatterns project[2] and the company Bitmanufaktur[3]; at the time of writing the project will soon become open-source, see the SocioPatterns web site for more information. The technical innovation of the applied tags is their ability to detect the proximity of other tags within a range of up to 1.5 meters. Due to the fact, that the human body blocks RFID signals, face-to-face contacts can then be detected. In this context, one of the first experiments using RFID tags for tracking the position of persons on room basis was conducted by Meriac et al. (cf., [16]) in the Jewish Museum Berlin in 2007. Cattuto et al. [8] added proximity sensing in the SocioPatterns project. Barrat et al. [13] did further experiments.

For several research questions, e.g., for social network analysis, it is rather interesting to combine the movement and contact data of persons. To conduct such analysis we apply active RFID tags that provide data from which we can extract positioning data as well as contact data. The proximity-tags are primarily developed for recognizing face-to-face contacts.

In the context of the presented approach, one additional problem concerns the exact positioning information: Our hardware setting does not provide information (like ToA, TDoA, AoA, RSSI, ...) used for positioning in traditional localization algorithms. Like the work of [16] we use the number of packages each RFID reader received from each RFID tag in a specific time interval (for each signal strength) to determine the users position. Compared to the work of [16] we use a fingerprint technique to estimate the location of the user. In [16] the participant is allocated to the room whose RFID readers received most packages with the weakest signal strength. This approach works fine, but it is based on the fact that at least two readers are placed in each room. Unfortunately, often it is not possible to place the RFID readers at arbitrary positions, e.g., in older buildings, or buildings with monumental protection.

Localization with proximity tags and readers as applied in our hardware setting is challenging for different reasons: First, the number of packages per second sent by the RFID tags is very low. Second, the position of RFID readers can not be chosen freely and the number of readers should be as small as possible. Third, as discussed above, the RFID readers do not offer additional information (like ToA, RSS,...) about the received packages from proximity tags. Fourth, as already described in previous work RFID properties like reflection and multi-path-fading complicate the task of localization. For further reading about RFID we refer to [10,11].

In this paper, we propose a resource-aware approach for indoor localization using proximity tags. To the best of the authors' knowledge, this is the first time that the accuracy of such a localization approach is investigated in a real world application. In contrast, to the presented approach all existing literature studied their approaches under nearly optimal (laboratory) conditions, e.g., [18][12].

[2] `http://www.sociopatterns.org`
[3] `http://www.bitmanufaktur.de`

## 3 Resource-Aware RFID Room-Level-Localization

In the following section, we first outline the resource-aware application scenario using active RFID tags. After that, we describe the application of machine learning for room-level prediction of the tags' location. Next, we summarize the strategies for improving the accuracy of the applied methods by utilizing the proximity contacts between the applied RFID tags.

### 3.1 Resource-Aware RFID Application Scenario

In this paper, we aim at a flexible and resource aware approach for localization using RFID: It should require only a small number of readers, and should further allow the free placement of readers not constrained to single rooms. E. g., a reader might be assigned to several areas, or to larger areas in general. In our experience, such a setting is highly relevant for practical applications, and also needed to be taken into account for our real-world evaluation scenario. Further issues, that have to be overcome in a real-world setting are the interference between tags – if many tags are put into one location and signals transcending room boundaries, i. e., walls or ceilings.

In summary, in a real-life setup the localization problem is much more complicated compared to a simulated environment, using very many readers and resources e. g., [16] . Below, we describe the hardware and system architecture used in our localization experiment.

**Hardware.** For our localization experiment at the poster session of a conference we asked each participant to carry an active RFID tag (see Figure 1). The tags provide localization and proximity detection in a resource-aware and cost-effective way, which conforms to our requirements. Every two seconds each RFID tag sends one package in four different signal strengths (-18dbm, -12dbm, -6dbm, -0dbm) to RFID readers placed at fixed positions in the conference area (see Figure 2). Dependent on the signal strength the range of one package inside a building is up to 25 meters. Each package is 128 bits long, encrypted, and contains information about the tag id from the reporting RFID tag, signal strength and CRC checksum. For more details, we refer to Barrat et al. [5] and [2]. The continuous sending of RFID packages in uniform time-intervals (two seconds) gives us the opportunity of determining the package-loss of an RFID tag at each RFID reader. We use this information to create the characteristic RFID vectors. Here, we note that we do not use the package loss explicitly. Instead we use the number of packages an RFID reader receives from a tag.

One decisive factor, that makes proximity tags interesting for conference scenarios is the possibility to detect other proximity tags within a range of up to 1.5 meters. Since the human body blocks RFID signals, one can detect and analyze face-to-face contacts in this way [5]. In this work, we show that this proximity information helps to improve the localization accuracy. The information about contacts is transmitted in the fourth and strongest signal strength of the tags. Thus, a tag sometimes sends more than one package (every two seconds) in that

strength, because more packages are needed to transport the contact information. Since it is not possible to store information on the tags permanently, every time-dependent information is lost, when a tag is out of the range of all RFID readers.

The RFID readers (see Figure 1) receive RFID signals and forward them to a central server via UDP where the signals are decrypted, analyzed and stored in a database. Because of resource-awareness reasons the RFID readers do not provide information like AoA or RSS of the received packages, which could help to additionally improve the accuracy of the localization results.

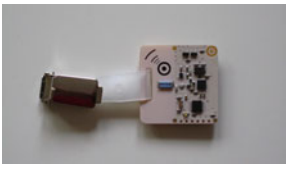

**Fig. 1.** Proximity tag (left) and RFID reader (right)

### 3.2 Machine Learning for Prediction Using RFID Data

As described in Section 3.1, each RFID tag sends one package in four different signal strengths every two seconds. Similar to most fingerprint approaches we assume that the number of packages an RFID reader received is significantly dependent on the position of the sending RFID tag, i. e., when a tag is moved away from the reader, the number of received packages will decrease. Therefore, we can determine sets of characteristic vectors (fingerprints) for each room in the conference area.

**Observation Vector Space.** In a setting with $R$ RFID readers and $P$ proximity tags, each transmitting on $S$ different signal strengths, let $l$ denote the length of a time window and $t$ a point in time. Further, let $V_r^l(p,t) \in \mathbb{N}^s$ $(1 \leq r \leq R,$ $1 \leq p \leq P)$ be an $S$-dimensional vector where the $s$-th entry is the number of packages that RFID reader $r$ received from proximity tag $p$ with signal strength $s$ in the time interval $[t-l,t]$. The vector

$$V^l(p,t) = \big(V_1^l(p,t), V_2^l(p,t), \cdots, V_R^l(p,t)\big) \tag{1}$$

– i. e., the concatenation of the vectors $V_r^l(p,t)$ over all readers $r$, – is called the *package count vector* or *characteristic vector* of the proximity tag $p$ at time $t$. The dimension of vector $V^l(p,t)$ is $S \cdot R$. With the parameter $l$ one can control the influence of older signals. For longer time intervals, the probability rises that packages sent from a previous location influence the vector at the current time point $t$.

We consider the localization problem as a classification task. In the learning phase, we create a set of fingerprints (training data) for each room, and learn a classification model based on these fingerprints. In the online classification (localization phase) we determine the position of a participant from his current

fingerprint, using the classification model. In this paper, we modify four state of the art machine learning methods for that classification task by including proximity contact information and analyze the resulting increase in their accuracy due to these contacts.

**Basic Room Prediction.** In the following, we outline the basic machine learning methods that we applied as a benchmark for predicting locations of the participants, and as initial methods to be complemented with the contact strategies described below. We briefly summarize their basic features, and discuss their application using the RFID data. We refer to the basic localization methods as the LOC-BASIC approach.

*Naive Bayes* (NBAY). While naive Bayes [17] is a rather simple approach, studies comparing classification algorithms have shown that the naive Bayes classifier is often comparable in performance with decision trees, while achieving high accuracy and speed being applied to large databases. Therefore, for the localization naive Bayes is a good candidate due to its learning performance and accuracy.

*K-Nearest Neighbor* (KNN). As a lazy learner, the k-nearest neighbor algorithm [17] is easy to setup and implement, since only a certain set of training data needs to be stored, and a suitable distance (similarity) metric be applied for retrieving a similar case for a given query. Therefore, a scenario that does not allow for long training periods favors a nearest neighbor classifier. The parameter $k$ controls the number of neighbors considered for each prediction.

*Support Vector Machines* (SVM). Support vector machines [9] have become one of the benchmark techniques for machine learning approaches due to their good classification performance for a broad range of applications. Therefore, we also consider support vector machines as our basic learning strategy and benchmark method. In this scientific work we use the $\text{SVM}^{\text{light}}$ C-implementation [3] from Thorsten Joachims. For our experiments described in Section 4 we chose an RBF kernel

$$K_{a,b} = \exp\left(-\gamma||x_a - x_b||^2\right), \tag{2}$$

where $x_a$ and $x_b$ are package count vectors. In Section 4 we analyze the best parameter combination for parameter $\gamma \in \mathbb{R}$ and parameter $j \in \mathbb{R}$. Here, the parameter $j$ is the cost factor, by which training errors on positive examples dominate errors on negative examples[4]. For all other parameters we chose the default values as described in [14].

*Random Forest* (RF). The random forest classifier [7] is an ensemble classification method: It applies a set of unpruned decision trees for classification. It can usually be learned in a cost-effective manner. Therefore, we also selected a random forest method for our set of base learners for the localization approach. In Section 4 we analyze the accuracy for different combinations of the two input parameters *mtry* (denoting the number of predictors sampled for splitting at each node) and *ntree* (the number of trees). For our experiments we use the *R*-implementation of Random Forest [1].

[4] `http://svmlight.joachims.org/`

### 3.3 Advanced Room Prediction Using Contacts

In this section we describe three simple but effective techniques which include contact information for improving the accuracy of the Loc-Basic algorithms. Let $C^w(p,t)$ denote the set of users (proximity tags) that were in contact with user $p$ within a time interval $[t-w,t]$. Hereby, the length $w$ of that interval is independent from the length $l$ of the time interval used in the construction of the characteristic vectors. Assume, that we want to predict the position of user $p$ at time $t$.

**Mean-Approach.** As input for the Loc-Basic algorithm the following vector is used:

$$V^l_{mean}(p,t) = \frac{V^l(p,t) + \sum_{q \in C^w(p,t)} V^l(q,t)}{1 + |C^w(p,t)|} \,. \tag{3}$$

Thus, the new characteristic vector $V^l_{mean}(p,t)$ of user $p$ is the average over all package count vectors of the contacts of user $p$ and of user $p$ himself.

**Max-Approach.** Let $(v_p^1, \cdots, v_p^{SR})$ be the component representation of $V^l(p,t)$. As input for the Loc-Basic algorithm the vector

$$V^l_{max}(p,t) = \left( \max_{q \in C^w(p,t) \cup \{p\}} \left\{ v_q^1 \right\}, \cdots, \max_{q \in C^w(p,t) \cup \{p\}} \left\{ v_q^{S \cdot R} \right\} \right) \tag{4}$$

is used. $V^l_{max}(p,t)$ is the component-wise maximum of the characteristic vectors of user $p$ and his contacts.

**Vote-Approach.** This approach consists of two phases. At first (preliminary) positions for user $p$ and all his contact users are predicted using Loc-Basic. Then, the final prediction of $p$'s position is established by a majority vote among all these Loc-Basic predictions.

## 4 Evaluation

Below, we first discuss the applied data before we describe the evaluation setting in detail. After that, we present the results of our experiments, and conclude with a comprehensive discussion.

### 4.1 Datasets

We utilized real-world data collected at the LWA 2010 conference in Kassel, covering the locations of tracked participants and contacts between these. In order to obtain a diverse and interesting set of observations we focused on the two hour poster session, since during that time many participants had gathered in 5 adjacent rooms. This provides us a challenging scenario for our methods. To ensure that each point in the conference area was covered, we placed 6 RFID readers at adequate positions in the conference area (see Figure 2).

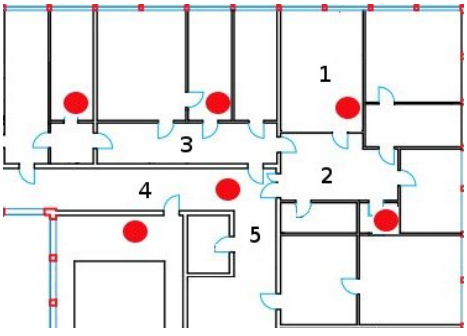


**Fig. 2.** Conference Area: the numbered rooms were used by participants during the poster-session, the circles mark the positions of RFID readers

We consider two kinds of tags: *user tags* and *object tags*: A user tag is a proximity tag worn by a participant during the conference. With an object tag we denote a proximity tag fixed to an unmovable object. In total, we fixed 46 object tags to several posters, tables and seats. Depending on its size we put between two and thirteen object tags in each room. The *training data* contains the first 1500 characteristic vectors collected with the object tags for each room of the conference area. Obtaining the training set took about 25 minutes.

*Ground truth*: In summary, 46 people took part in our localization experiment during the poster session. We collected their tag data over a duration of two hours. To evaluate the accuracy of our predictions we needed to determine the positions of the participants, for which we applied the object tags. Since the tags detect other proximity tags only within a range of up to 1.5 meters, whenever a contact between the tag of a participant tag and an object tag was recorded, we could infer that this participant was in the same room as the object tag (ground truth). In the experiments, we predicted the rooms for those vectors where the precise location could be verified with the ground truth data.

### 4.2 Setting

In all experiments, the target is to maximize the overall localization accuracy. The setup of the experiments contains a variety of parameters such as tuning parameters of the algorithms, parameters that control the vector space of our observations or parameters to control the set of contacts for each user at a specific time. Several of these parameters are data set dependent. Due to the nature of our setting as a social get-together most users did not switch locations very often. It is therefore possible and useful to choose large intervals to construct the observation vector space, in our experiments we chose $l = 10, 30$ and 50 seconds. However, other contexts might demand more frequent changes of the locations. In such cases fingerprints should be collected only over rather short time intervals.

Since the fourth and strongest signal strength of the tags transmits in irregular intervals (in contrast to the other signal strengths), we considered vectors

including or excluding the fourth signal strength. All in all, we obtained six different datasets, in the following referred to as $F_{10-3}$ through $F_{50-4}$ where e. g., the vectors of $F_{50-4}$ are collected over $l = 50$ seconds and constructed with all four signal strengths of each tag. Depending on the length of the time window $l$ and the number of used signals, the size of training data is shown in Table 1. To include the contact information we used the mean, max and vote approach. The first parameter to choose is the length $w$ of the time window over which we collect contacts. We experimented with five time windows: $w \in \{2, 5, 10, 20, 30\}$ (in seconds). A second parameter $d$ is the *degree of transitive closure*, that is added to the contact set. Contact information for one user at a specific time can be sparse. In such cases it may be of help to "add more contacts" based on the rationale, that contacts between users $u_1$ and $u_2$ and between $u_2$ and $u_3$ might indicate a contact between users $u_1$ and $u_3$. This procedure of adding such (transitive) contacts can be iterated and $d$ is the count of these iterations. Since for $d = 7$ no new contacts were produced we investigated the values $d = 0, 1, \ldots, 6$.

**Table 1.** Size of the ground truth dataset for different time window lengths $l$ and numbers of signals.

| $F_{10-3}$ | $F_{10-4}$ | $F_{30-3}$ | $F_{30-4}$ | $F_{50-3}$ | $F_{50-4}$ |
|---|---|---|---|---|---|
| 135208 | 137126 | 137454 | 137579 | 137570 | 137586 |

To prevent combinatorial explosion, we structured our experiments into two parts, described below: In the first part, we applied each of our four LOC-BASIC algorithms with different parameter settings to each of the six datasets. In the second part we additionally considered contact data to increase the localization quality for those parameter settings, that performed best in part one. Additionally, we conducted several experiments exploring variations of the size of the training set.

### 4.3 Results and Discussion - Part 1: Machine Learning Baseline

Table 2 presents the results of the first phase showing the best parameter combinations for each dataset and algorithm together with the achieved overall accuracies. We ran KNN with values for $k$ from 5 through 200 in steps of 5. For RF we tried all combinations of $mtry = 1, \ldots, 20$ and forest sizes $ntree$ of 25 through 500 in steps of 25. SVM was run with combinations of $j = 1, \ldots, 20$ and $\gamma \in \{2, 0, -2, -4, \ldots, -18, -20\}$. Finally, the NBAY does not depend on a parameter. An immediate observation is, that NBAY was always outperformed by any of the other algorithms. This is not surprising as the basic assumption of NBAY is the complete independence of the entries in each observation. Such independence can not be claimed for our datasets. If a reader receives, e. g., packages from a tag in its lowest signal strength, then it is much more likely that the reader will also receive packages in a higher strength from that tag. However, since we are interested in observing the boost that contact information can have on the results of a given classifier, we experimented with NBAY rather than with more complex Bayes approaches taking dependencies into account.

**Table 2.** For each algorithm and dataset the best parameters settings and the resulting total accuracy in %. * The same accuracy was achieved with $\gamma = -12$.

| base | | $F_{10-3}$ | $F_{10-4}$ | $F_{30-3}$ | $F_{30-4}$ | $F_{50-3}$ | $F_{50-4}$ |
|---|---|---|---|---|---|---|---|
| kNN | $k$ | 50 | 165 | 125 | 185 | 180 | 200 |
| | acc | 71.96 | 73.58 | 74.36 | 79.33 | 73.26 | 79.80 |
| RF | $mtry$ | 1 | 1 | 1 | 2 | 1 | 4 |
| | $ntree$ | 475 | 375 | 400 | 350 | 275 | 200 |
| | acc | 77.44 | 78.03 | 83.66 | 84.53 | 84.18 | 84.78 |
| SVM | $j$ | 1 | 1 | 7 | 1 | 13 | 1 |
| | $\gamma$ | $-14$ | $-14$ | $-10^*$ | $-18$ | $-10$ | $-20$ |
| | acc | 78.05 | 77.95 | 82.55 | 84.15 | 82.53 | 84.84 |
| nBay | acc | 33.42 | 38.97 | 51.14 | 56.96 | 56.57 | 61.97 |

The results of the other three algorithms are between 71.96% and 84.78%. Taking into account the room layout and the hardware constraints due to our resource-aware approach, these results can be considered acceptable. As can be expected, in all cases the more sophisticated algorithms RF and SVM had higher scores than the simple kNN. Including the fourth signal strength into the datasets yielded better results than ignoring it – with one exception (SVM, $F_{10-4}$) where the two results differ, however only by 0.1%. Furthermore, the datasets where the package vectors are collected over 30 or 50 seconds yield better scores compared to the ones where only 10 seconds are considered.

A closer look at the influence of the algorithms parameters is presented (exemplary) in the diagrams of Figure 3. For higher values of $k$ the accuracy of kNN rises, up to a certain level. After a (dataset dependent) threshold the accuracy almost stabilizes at that level. While the choice of the forest size *ntree* for RF did not influence the result much, the choice of the *mtry* parameter is of importance. In general, with lower values (1 through 4) the results were significantly better than for other choices. The curves of SVM fluctuate strongly on the datasets $F_{10-3}$ and $F_{10-4}$ and yield more stable curves for the others. In general, better results where achieved using very small values for the parameter $j$ – in the cases where the best score was obtained with $j = 7$ or $j = 13$, the scores using $j = 1$ were not significantly lower. In all cases, results were better using lower values for $\gamma$ such as $-20$.

### 4.4 Results and Discussion - Part 2: Utilizing Contact Information

In the second phase of the experiments, we employed the best parameters determined in phase one (Table 2) and included contact information to boost the localization accuracy. Tables 3, 4 and 5 present for each dataset and algorithm the best choice of the two parameters $w$ and $d$ and the achieved accuracy. Furthermore, for each method the lowest accuracy that was achieved with any combination of the two parameters is given. In the tables, bold numbers mark the accuracies of those methods, that performed best for the given algorithm and dataset. Italic numbers indicate accuracies, that are below the according baseline of phase one.

A first encouraging observation is, that in all experiments with the mean or max approach, the methods had a strictly positive influence on the accuracy.

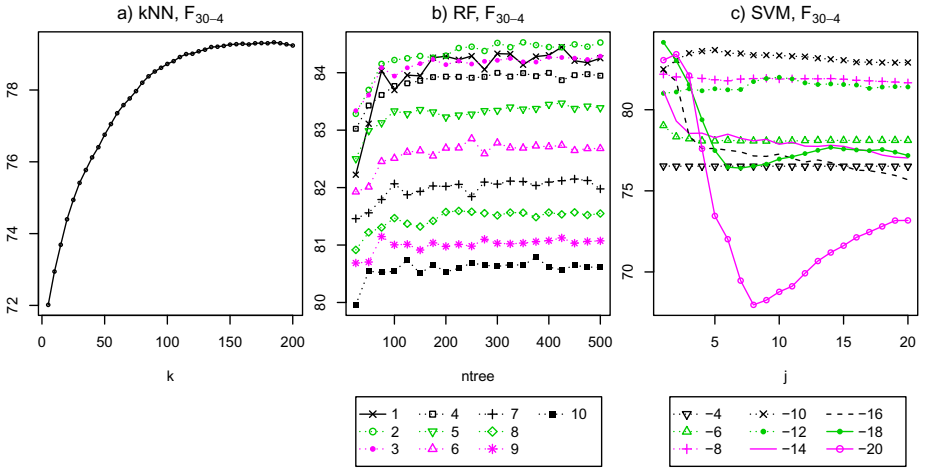


**Fig. 3.** Exemplary for $F_{30-4}$, the diagrams, showing the accuracies in % a) vs. $k$ (kNN), b) vs. *ntree* for different values of *mtry* (RF) and c) vs. $j$ for different values of $\gamma$ (SVM). In c) the graphs for $\gamma = 2$, $\gamma = 0$ and $\gamma = -2$ were left out for the sake of legibility. All three are constant with accuracies 14.81%, 15.94% and 48.74%.

**Table 3.** For each algorithm with max aggregation and each dataset the best choices of $w$ and $d$ with the according accuracy in % and the minimum accuracy achieved with max aggregation.

| max | | $F_{10-3}$ | $F_{10-4}$ | $F_{30-3}$ | $F_{30-4}$ | $F_{50-3}$ | $F_{50-4}$ |
|---|---|---|---|---|---|---|---|
| kNN | $w$ | 30 | 20 | 30 | 30 | 30 | 30 |
| | $d$ | 4 | 4 | 1 | 2 | 1 | 2 |
| | top ACC | **80.26** | **80.28** | **83.39** | 85.40 | **82.78** | 85.53 |
| | min ACC | 78.01 | 78.75 | 80.99 | 84.04 | 80.25 | 84.1 |
| RF | $w$ | 20 | 20 | 20 | 20 | 20 | 5 |
| | $d$ | 3 | 2 | 6 | 1 | 6 | 1 |
| | top ACC | **84.94** | **85.49** | **89.59** | **89.96** | **88.94** | 87.53 |
| | min ACC | 83.25 | 83.99 | 88.31 | 88.92 | 87.95 | 86.73 |
| SVM | $w$ | 20 | 20 | 20 | 20 | 20 | 20 |
| | $d$ | 2 | 2 | 5 | 1 | 2 | 1 |
| | top ACC | **84.43** | **85.46** | 88.16 | 89.14 | 88.65 | 88.72 |
| | min ACC | 82.89 | 83.73 | 87.09 | 88.33 | 87.42 | 88.06 |
| nBay | $w$ | 30 | 30 | 30 | 30 | 30 | 30 |
| | $d$ | 3 | 4 | 2 | 6 | 1 | 1 |
| | top ACC | **50.60** | **56.81** | **65.80** | **71.95** | 69.55 | **76.57** |
| | min ACC | 44.40 | 50.13 | 61.30 | 66.70 | 66.77 | 72.50 |

Only voting performed in some cases worse than the baseline, mainly for nBay. For nBay we attribute this to the fact, that the voting scheme is a probabilistic method. Since nBay itself has only a very low accuracy, it is likely that among the votes many are in fact false predictions. Thus, the probability of a wrong classification even rises.

With one exception the best results were always achieved using max or mean aggregation. Here, including the contact information yielded significant boosts

**Table 4.** For each algorithm with mean aggregation and each dataset the best choices of $w$ and $d$ with the according accuracy in % and the minimum accuracy achieved with mean aggregation.

| mean | | $F_{10-3}$ | $F_{10-4}$ | $F_{30-3}$ | $F_{30-4}$ | $F_{50-3}$ | $F_{50-4}$ |
|---|---|---|---|---|---|---|---|
| kNN | $w$ | 10 | 20 | 30 | 20 | 30 | 30 |
| | $d$ | 1 | 1 | 1 | 2 | 2 | 2 |
| | top ACC | 75.79 | 79.55 | 80.68 | **85.62** | 79.63 | **86.24** |
| | min ACC | 75.19 | 78.52 | 79.35 | 84.30 | 78.24 | 84.73 |
| RF | $w$ | 10 | 10 | 20 | 20 | 30 | 30 |
| | $d$ | 2 | 3 | 6 | 3 | 4 | 5 |
| | top ACC | 79.51 | 80.65 | 88.04 | 88.33 | 88.29 | 88.83 |
| | min ACC | 78.00 | 79.31 | 86.57 | 87.29 | 86.83 | 87.57 |
| SVM | $w$ | 20 | 20 | 30 | 30 | 30 | 20 |
| | $d$ | 2 | 3 | 2 | 2 | 2 | 2 |
| | top ACC | 83.96 | 85.01 | **88.43** | **89.49** | **88.89** | **89.39** |
| | min ACC | 82.56 | 83.39 | 86.91 | 88.33 | 87.26 | 88.32 |
| nBay | $w$ | 30 | 30 | 30 | 30 | 30 | 30 |
| | $d$ | 5 | 5 | 2 | 2 | 2 | 2 |
| | top ACC | 49.61 | 52.62 | 65.22 | 68.88 | **71.26** | 75.54 |
| | min ACC | 43.93 | 48.20 | 60.23 | 64.31 | 66.26 | 70.59 |

**Table 5.** For each algorithm with voting aggregation and each dataset the best choices of $w$ and $d$ with the according accuracy in % and the minimum accuracy achieved with voting aggregation. * The same accuracy was achieved with $d = 4$

| vote | | $F_{10-3}$ | $F_{10-4}$ | $F_{30-3}$ | $F_{30-4}$ | $F_{50-3}$ | $F_{50-4}$ |
|---|---|---|---|---|---|---|---|
| kNN | $w$ | 20 | 30 | 20 | 20 | 20 | 20 |
| | $d$ | 2 | 3 | 2* | 3 | 3 | 3 |
| | top ACC | 77.39 | 79.71 | 80.02 | 85.17 | 78.63 | 85.94 |
| | min ACC | 76.11 | 78.35 | 78.82 | 83.80 | 77.69 | 84.49 |
| RF | $w$ | 10 | 10 | 20 | 10 | 20 | 20 |
| | $d$ | 5 | 1 | 1 | 2 | 2 | 3 |
| | top ACC | 81.32 | 81.67 | 86.77 | 87.55 | 87.81 | **88.97** |
| | min ACC | 80.75 | 81.14 | 86.27 | 87.08 | 87.03 | 87.88 |
| SVM | $w$ | 20 | 20 | 5 | 20 | 2 | 30 |
| | $d$ | 4 | 4 | 0 | 2 | 0 | 2 |
| | top ACC | 81.35 | 81.09 | 83.22 | 86.75 | 82.76 | 87.46 |
| | min ACC | 80.05 | 79.51 | *81.29* | 85.55 | *80.86* | 86.17 |
| nBay | $w$ | 2 | 2 | 2 | 2 | 2 | 2 |
| | $d$ | 0 | 0 | 0 | 0 | 0 | 0 |
| | top ACC | *33.05* | *38.70* | *50.90* | *56.53* | *56.10* | *61.31* |
| | min ACC | *31.48* | *37.54* | *50.39* | *55.81* | *55.22* | *60.66* |

**Table 6.** For each algorithm (in its best performing aggregation parametrization) the fraction of data for which contact information is available (in %) and a comparison of prediction accuracy of the algorithms without contacts and those using the best performing method of contact data aggregation.

| | kNN | RF | SVM | nBay |
|---|---|---|---|---|
| strategy | $F_{50-3}$ | $F_{10-3}$ | $F_{10-4}$ | $F_{10-4}$ |
| contact fraction | 69.23 | 65.01 | 64.91 | 69.3 |
| contact base ACC | 74.33 | 76.83 | 77.78 | 33.05 |
| contact best ACC | 88.09 | 88.18 | 89.34 | 58.80 |
| boost | 13.76 | 11.34 | 11.57 | 25.74 |

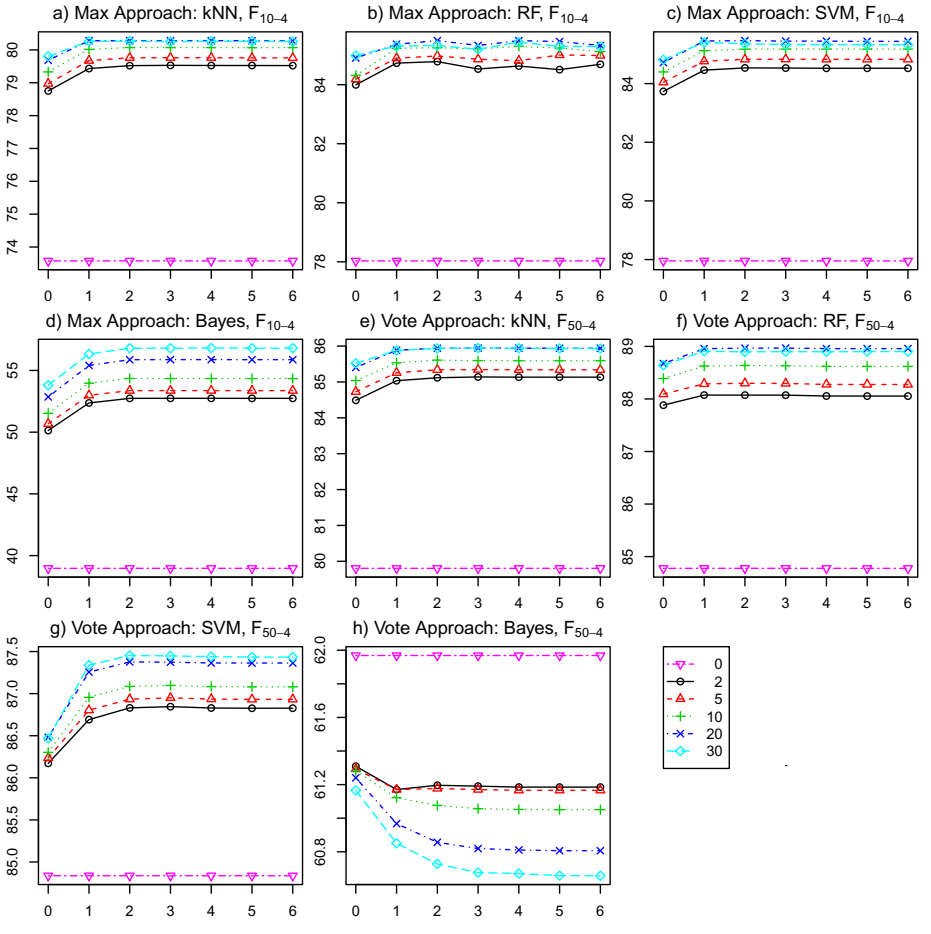


**Fig. 4.** a) through d) present the accuracies in % of all experiments with the max approach on $F_{10-4}$, e) through h) those of the experiments with the vote approach on $F_{50-4}$. For several choices of $w$, the accuracy is plotted vs. the degree of transitivity $d$.

in overall accuracy: up to an additional 9.52% for kNN ($F_{50-3}$), 7.5% for RF ($F_{10-3}$), 7.51% for SVM ($F_{10-4}$) and 17.84% for nBay ($F_{10-4}$). These results are clear evidence, that the contact information can support the localization approach significantly. Even stronger evidence for that presents Table 6. This table shows for the above mentioned four settings the fraction of test data where contact information (depending on the parameters $d$ and $w$, chosen as in Table 3) is available (contact fraction). Further, given are the prediction accuracies on only that fraction of the dataset of both, the Loc-Basic algorithms (contact base acc) and the best contact boosted algorithms (contact best acc). Boost denotes the additional gain of accuracy due to the inclusion of the contact data. Here, the scores of kNN, RF and SVM profit with more than 11% while the

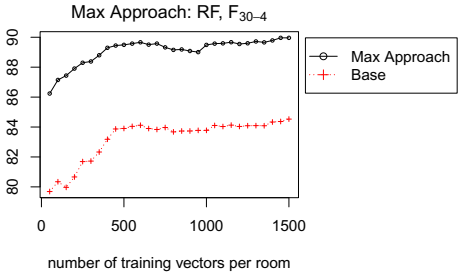


**Fig. 5.** The accuracy in % vs. the number of training samples per room

accuracy of nBay increases by more than 25%. Our best performing algorithm with respect to the complete test set (RF with max aggregation using $F_{30-4}$) yields a prediction accuracy of 92.69% if applied to that part dataset for that contact information is available.

Next, we investigated the influence of the parameters $d$ and $w$. As can be seen in the Tables 3, 4, 5 the fluctuation of the accuracy for different parameter combinations was rather low, often less than 1%. Figure 4 displays exemplary for each algorithm the results of the max approach for the $F_{10-4}$ dataset and of the vote approach for the $F_{50-4}$ dataset. The behavior of the accuracy using the mean approach was generally similar to that of the max approach. The parameter $d$ usually had only a small influence. In most experiments only the difference between $d = 0$ and $d = 1$ was significant. For the values $1, \ldots, 6$ the accuracy stayed almost constant. Variations of the $w$ parameter also caused similar behavior throughout the experiments. In those, where the contacts had positive influence on the accuracy, the choices $w = 20$ or $w = 30$ delivered the best results, while $w = 10$ usually was better than $w = 5$ or $w = 2$.

Furthermore, we analyzed the influence of the training set size. We applied the method from our previous experiments that performed best (RF with $ntree = 350$ , $mtry = 2$, $w = 20$ and $d = 1$ using the max approach on $F_{30-4}$) to classify with models based on differently sized training sets. Figure 5 shows the resulting accuracies compared to those of the according Loc-Basic method. Up to 450 samples per room, increasing the training size increases the accuracy. Afterwards the accuracy increases only little or decreases in some cases. The distance between the curves (the boost due to the contacts) is almost constant, only for very small training set sizes it is slightly larger.

## 5 Conclusions

In this paper, we have presented an approach for cost-effective and resource-aware localization at room level using RFID-tags. We evaluated several state-of-the-art machine-learning algorithms in this context, complemented by novel techniques for improving these using the RFID (proximity) contacts. The results

of the experiments yielded several reasonable values for the applicable parameters. For the simpler algorithms, they could also have been learned in a short preceding training phase, which demonstrates the broad applicability of the approach in the sketched resource-aware setting.

In the presented experiments we always considered training data collected by the object tags. In future work, we aim to analyze the accuracy of the proposed approach using the user tags in more detail. An extended analysis concerns using all available and also no contact information at all, respectively, when we consider the user tags for obtaining the training data. Furthermore, we plan to focus on optimizing the applied parameter combinations, i.e., number of readers, number of packages per second, etc., in order to increase the accuracy further. Testing our algorithms in WiFi and GPS based localization settings is also another interesting option for future work.

**Acknowledgements.** This work has been supported by the VENUS research cluster at the interdisciplinary Research Center for Information System Design (ITeG) at Kassel University. We utilized active RFID technology which was developed within the SocioPatterns project, whose generous support we kindly acknowledge. We also wish to thank Milosch Meriac from Bitmanufaktur in Berlin for helpful discussions regarding the RFID localization algorithm. Our particular thanks go the SocioPatterns team, especially to Ciro Cattuto, who enabled access to the Sociopatterns technology, and who supported us with valuable information concerning the setup of the RFID technology.

## References

1. CRAN - Package randomForest, `http://cran.r-project.org/web/packages/randomForest/index.html`
2. OpenBeacon Active RFID Project, `http://www.openbeacon.org`
3. SVM-Light Support Vector Machine, `http://svmlight.joachims.org/`
4. Bahl, P., Padmanabhan, V.N.: RADAR: An In-Building RF-Based User Location and Tracking System. In: INFOCOM, pp. 775–784 (2000)
5. Barrat, A., Cattuto, C., Colizza, V., Pinton, J.F., den Broeck, W.V., Vespignani, A.: High Resolution Dynamical Mapping of Social Interactions with Active RFID. CoRR abs/0811.4170 (2008)
6. Bekkali, A., Sanson, H., Matsumoto, M.: RFID Indoor Positioning Based on Probabilistic RFID Map and Kalman Filtering. In: WiMob, p. 21 (2007)
7. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
8. Cattuto, C., den Broeck, W.V., Barrat, A., Colizza, V., Pinton, J.F., Vespignani, A.: Dynamics of Person-to-Person Interactions from Distributed RFID Sensor Networks. PLoS ONE 5(7) (July 2010)
9. Cortes, C., Vapnik, V.: Support-Vector Networks. Machine Learning 20(3), 273–297 (1995)
10. Finkenzeller, K.: RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification, 2nd edn. John Wiley & Sons, Inc., New York (2003)
11. Glover, B., Bhatt, H.: RFID Essentials (Theory in Practice (O'Reilly)). O'Reilly Media, Inc., Sebastopol (2006)

12. Hightower, J., Vakili, C., Borriello, G., Want, R.: Design and Calibration of the SpotON Ad-Hoc Location Sensing System. Tech. rep. (2001)
13. Isella, L., Stehlé, J., Barrat, A., Cattuto, C., Pinton, J.F., den Broeck, W.V.: What's in a Crowd? Analysis of Face-to-Face Behavioral Networks. CoRR 1006.1260 (2010)
14. Joachims, T.: Making large-Scale SVM Learning Practical. In: Schölkopf, B., Burges, C., Smola, A. (eds.) Advances in Kernel Methods - Support Vector Learning, ch. 11, pp. 169–184. MIT Press, Cambridge (1999)
15. Li, X., Pahlavan, K.: Super-Resolution TOA Estimation with Diversity for Indoor Geolocation. IEEE Transactions on Wireless Communications 3(1), 224–234 (2004)
16. Meriac, M., Fiedler, A., Hohendorf, A., Reinhardt, J., Starostik, M., Mohnke, J.: Localization Techniques for a Mobile Museum Information System. In: Proceedings of WCI (Wireless Communication and Information) (2007)
17. Mitchell, T.: Machine Learning (Mcgraw-Hill International Edit), 1st edn. McGraw-Hill Education, ISE Editions (October 1997)
18. Ni, L.M., Liu, Y., Lau, Y.C., Patil, A.P.: LANDMARC: Indoor Location Sensing Using Active RFID. Wireless Networks 10(6), 701–710 (2004)
19. Niculescu, D., Badrinath, B.R.: Ad Hoc Positioning System (APS) Using AOA. In: INFOCOM (2003)
20. Priyantha, N.B., Chakraborty, A., Balakrishnan, H.: The Cricket Location-Support System. In: MOBICOM, pp. 32–43 (2000)
21. Rappaport, T.: Wireless Communications: Principles and Practice, 2nd edn. Prentice Hall PTR, Upper Saddle River (2001)