WASP: An Enhanced Indoor Locationing Algorithm for a Congested Wi-Fi Environment

Hsiuping Lin, Ying Zhang, Martin Griss, and Ilya Landa

Carnegie Mellon Silicon Valley,
NASA Research Park Building 23, Moffett Field, CA 95035, USA
{tony.lin,joy.zhang,martin.griss,ilya.landa}@sv.cmu.edu

Abstract. Accurate and reliable location information is important to many context-aware mobile applications. While the Global Positioning System (GPS) works quite well outside, it is quite problematic for indoor locationing. In this paper, we introduce WASP, an enhanced indoor locationing algorithm. WASP is based on the Redpin algorithm which matches the received Wi-Fi signal with the signals in the training data and uses the position of the closest training data as the user's current location. However, in a congested Wi-Fi environment the Redpin algorithm gets confused because of the unstable radio signals received from too many APs. WASP addresses this issue by voting the right location from more neighboring training examples, weighting Access Points (AP) based on their correlation with a certain location, and automatic filtering of noisy APs. WASP significantly outperform the-state-of-the-art Redpin algorithm. In addition, this paper also reports our findings on how the size of the training data, the physical size of the room and the number of APs affect the accuracy of indoor locationing.

1 Introduction

Location is crucial information to many context-aware mobile applications. Personal navigation, asset tracking, local information search and friend finder all require accurate and reliable location information from mobile devices. While the Global Positioning System (GPS) works quite well outside, it does not work well inside buildings because GPS signals can not penetrate most buildings. Indoor locationing plays an important role in ubiquitous computing and attracts considerable interest in both industry and research. Some systems are designed specifically for indoor locationing but require special infrastructure [1], [8]. With the growth of Wi-Fi networks due to declining prices, increased ubiquity of devices (laptops, cell phones, and other devices using Wi-Fi) and simplified installation of Wi-Fi access points (AP), indoor locationing using wireless LAN (WLAN) is becoming more promising.

There are two major types of WLAN-based indoor locationing approaches: signal propagation model and fingerprinting. In the signal propagation approach, we have to know physical locations of all APs in advance. The received signal strength (RSS) on the mobile device can then be used to estimate the distance

R. Fuller and X.D. Koutsoukos (Eds.): MELT 2009, LNCS 5801, pp. 183–196, 2009.

from AP to the mobile device. One can use a multi-lateration algorithm to calculate the the physical location of the mobile device. To reflect the real environment, some signal propagation methods even include wall-attenuation and reflections into the model. In general, the signal propagation approach does not always provide satisfying results because of the Wi-Fi signal fluctuation caused by environmental variations [9].

The fingerprinting approach requires a training database of RSS fingerprints and their corresponding locations. The location of the mobile device is determined by the location of similar fingerprints in the database [2] or from the statistical model derived from the training data [4].

There are many location fingerprinting algorithms. The simplest one is based on the K-nearest neighbor algorithm (KNN). It converts fingerprints into vectors and chooses the K historical fingerprints that are most similar to the testing fingerprint. The location of the testing fingerprint is determined by the majority of its K nearest neighbors. Extending the KNN algorithm, [3] measures not only the contribution of RSSes but also the number of common access points and not-common access points. Another fingerprinting approach is to model the distribution of RSSes at various locations and tries to handle the uncertainty and errors of signal strength measurements .

In this paper, we describe the WASP algorithm, an enhanced indoor locationing algorithm for congested Wi-Fi environments. WASP is a fingerprinting approach and it significantly improves the state-of-the-art indoor locationing algorithm in our experiments.

The rest of this paper is organized as follows. Section 2 reports related work. Section 3 introduces the WASP algorithm and other statistical methods evaluated in this paper. Section 4 describes the dataset, our experimental environment and key results and we conclude the paper with discussion and plans for future work in section 5.

2 Related Work

A reliable and stable interior positioning system (IPS) would be of great benefit to many applications. Considerable research has been performed to determine the indoor location of a mobile user or a mobile device. RADAR, developed by Bahl *et al.*, is an IPS based on Wi-Fi technology [1]. It uses signal strength information gathered at multiple receiver locations by the PC based stations to triangulate the user's coordinates. Paschalidis *et al.* presents an approach that allows a wireless sensor network to determine the physical locations of its nodes by partitioning the wireless sensor network into regions and the localization algorithm identifies the region where a given sensor resides [11].

Most recent research collects RSSes directly on the mobile devices, avoiding the need for extra hardware elements. Li *et al.* compares the trilateration and fingerprinting approaches, including both deterministic methods and probabilistic methods [10]. Brunato *et al.* provide a general comparison of SVM, KNN, Bayesian modeling and multi-layer perceptrons for locationing [4]. Carlotto *et* *al.* evaluate the proximity of two mobile devices by classifying the degree of similarity of the Wi-Fi scanned data using a statistical Gaussian Mixture Model [5]. Correa *et al.* report experiences using an existing Wi-Fi infrastructure without specialized hardware added to support room-level Wi-Fi location tracking by signature matching, as well as the use of a specialized AP controller [6]. Bolliger proposes the Redpin system, a novel approach that does not require an explicit offline phase but allows users to create and manage the location fingerprints collaboratively [3]. In our work, we started from the open source Redpin system and made substantial enhancements for the congested Wi-Fi environment.

3 Indoor Locationing Algorithms

Our location fingerprinting is based on the assumption that a mobile device will experience a different RSS fingerprint at different locations in the building, and that the variation of the fingerprints seen over time in one location does not vary too much[1]. We collect training data using handsets from several locations in our building. Each training point is a tuple $(L,\boldsymbol{t})$ of a location label L and the detected RSSes fingerprint $\boldsymbol{t} = (t_1, t_2, ..., t_N)$ where t_i is the RSS received from AP_i. In this section, we first describe several statistical learning algorithms used in our experiments.

3.1 Naive Bayes Classifier

In Naive Bayes approach, we predict a user's location to be L^* if $P(L^*|\boldsymbol{t})$ is the highest probability of all possible locations:

$$L^* = \arg\max_L P(L|\boldsymbol{t}). \tag{1}$$

By Bayesian theorem, we have

$$L^* = \arg\max_L \frac{P(\boldsymbol{t}|L)P(L)}{P(\boldsymbol{t})} = \arg\max_L P(\boldsymbol{t}|L)P(L) \tag{2}$$

$P(\boldsymbol{t})$ is dropped because it does not depend on L. The conditional probability $P(\boldsymbol{t}|L)$ can be estimated by

$$P(\boldsymbol{t}|L) = P(t_1, \ldots, t_N|L) = P(t_1|L)P(t_2|L, t_1) \ldots (t_N|L, t_1, ...t_{N-1}) \tag{3}$$

With a naive independence assumption that each t_i is conditionally independent of every other t_j for $t_i \neq t_j$, we have

$$P(\boldsymbol{t}|L) = P(t_1|L)P(t_2|L)...P(t_N|L) = \prod_{i=1}^{N} P(t_i|L) \tag{4}$$

[1] Of course, the usefulness of this location-based difference and relatively stable fingerprint depends on the placement of the access points, the shape and construction of the building and the sources of noise and fluctuation.

$P(t_i|L)$ can be derived from the historical fingerprints by maximum likelihood estimation (MLE). Thus, the location L can be derived by

$$L^* = \arg\max_L P(L|\boldsymbol{t}) = \arg\max_L P(L) \prod_{i=1}^{N} P(t_i|L) \tag{5}$$

The problem of the naive Bayes method is that the values of the signal strength are not taken into consideration. In other words, $P(\boldsymbol{t}|L)$ is estimated by counting the frequency where s_i is non-zero at location L and only the existence of a set of APs decides the location. To address this issue, Seshadri *et al.* use Bayesian filtering on a sample set derived by Monte-Carlo sampling to compute the location and orientation estimates [12].

3.2 Support Vector Machine (SVM)

The Support Vector Machine is a useful technique for data classification and some research has applied SVM to the indoor locationing problem [4], [7]. A classification task usually involves training and testing data which consist of many data instances. Each instance in the training set contains one target value (class labels) and several attributes (features). The goal of SVM is to produce a model which predicts the target value of data instances in the testing set when given only the attributes. Although SVM is a powerful classification technique, the fluctuations of signals may cause data instance pollution and affect the accuracy.

3.3 K-Nearest Neighbor (KNN)

The K-nearest neighbor algorithm is a method for classifying objects. Given a training data set with labels, KNN classifies a new data point based on the majority of its k-nearest neighbors. For different applications, different distance functions are defined to quantify the "similarity" between the training and testing points. In the simplest case (K=1), the algorithm finds the single closest match and use that fingerprint's location as prediction.

3.3.1 Distance Function

For a testing fingerprint $\boldsymbol{t}$, the standard KNN algorithm goes through each point $(L, \boldsymbol{s})$ in the training data and calculates the distance between $\boldsymbol{t}$ and $\boldsymbol{s}$. The generalized distance is

$$D_q(\boldsymbol{t}, \boldsymbol{s}) = (\sum_{i=1}^{N} |\boldsymbol{t_i} - \boldsymbol{s_i}|^q)^{\frac{1}{q}} \tag{6}$$

Manhattan distance and Euclidean distance are D_1 and D_2 respectively. The unknown location for $\boldsymbol{t}$ is decided by a majority vote from the K shortest distance fingerprints.

KNN is simple to implement and it provides reasonable accuracy. However, one drawback of the standard KNN is that RSSes detected in the same location vary from time to time. The fluctuations likely to cause errors in predicting locations. This can be partially overcome by having multiple fingerprint sets for a given location, taken at different times, assuming that one or other finger print may cover that fluctuation.

3.3.2 Redpin Algorithm - AP Similarity

The Redpin[2] algorithm is a variation of the standard KNN algorithm where the Euclidean distance is augmented with a bonus factor to reward training and testing fingerprints to have common APs and a penalty factor for not-common APs in two fingerprints. Thus, in addition to the signal strength, the number of common access points (NCAP) and the number of not-common access points (NNAP) also contribute to identifying the similarity of two fingerprints. The Redpin algorithm chooses K=1 to decide the best match and works as follows. We define a mapping function $\delta(s)$ as

$$\delta(s) = \begin{cases} 0, \text{ if } s = 0 \\ 1, \text{ if } s \neq 0 \end{cases} \tag{7}$$

NCAP of two fingerprints, $\boldsymbol{t}$ and $\boldsymbol{s}$, can be expressed as

$$\text{NCAP} = \sum_{i=1}^{N} \delta(t_i)\delta(s_i) \tag{8}$$

NNAP of $\boldsymbol{t}$ and $\boldsymbol{s}$ can be expressed as

$$\text{NNAP} = \sum_{i=1}^{N} \delta(t_i) \oplus \delta(s_i) \tag{9}$$

where $\oplus$ represents the exclusive disjunction. The generalized similarity value of $\boldsymbol{t}$ and $\boldsymbol{s}$ is

$$D(\boldsymbol{t}, \boldsymbol{s}) = \alpha \sum_{i=1}^{N} \delta(t_i)\delta(s_i) - \beta \sum_{i=1}^{N} \delta(t_i) \oplus \delta(s_i) + \gamma \Lambda(t_i, s_i) \tag{10}$$

Λ is a heuristic function defined in the Redpin algorithm which calculates the similarity of $\boldsymbol{t}$ and $\boldsymbol{s}$ based on the signal strengths. The factors α and γ are the bonus-weights for the common APs while β is the penalty-weight for the not-common APs. The key idea behind Redpin is using NCAP and NNAP as bonus-penalty adjustments which reduces the impact of signal fluctuations.

3.3.3 Weighted AP Similarity

To further reduce the impact of signal fluctuations, we observe that the visibility of the APs at one location is not always the same because the environmental variations cause significant Wi-Fi signal fluctuations in the same location over time, especially inside a large building with sparse APs. Intuitively, APs with higher visibility at a location L should be weighted more in determining whether

[2] The open source Redpin can be found at `http://www.redpin.org`

a fingerprint is located at L . In this paper, we use the correlation between APs and locations as the weight for each AP. We use the Point-wise Mutual Information (PMI) as the correlation measurement. PMI is defined as

$$I(L; AP) = log\frac{P(L, AP)}{P(L)P(AP)} \tag{11}$$

The higher the $I(L; AP)$ value, the more likely L is associated with AP. From the historical fingerprints in the database, we can calculate the $I(L; AP)$ value of each location L and AP pairs. We normalize the PMI value to be between 0 (least correlated) and 1 (most correlated). PMI values are applied as weighting modifiers to the bonus of each common AP (CAP) and the penalty of each not-common AP (NAP). The weighted similarity value of the measured fingerprint $\boldsymbol{t}$ and a historical fingerprint $\boldsymbol{s}$ located at L is

$$D(\boldsymbol{t}, \boldsymbol{s}) = \alpha \sum_{i=1}^{N} \delta(t_i)\delta(s_i)I(L; AP_i) - \beta \sum_{i=1}^{N} \delta(t_i) \oplus \delta(s_i)I(L; AP_i) + \gamma\Lambda(t_i, s_i) \tag{12}$$

3.3.4 Noise Filter (NF)

Extending the idea of weighting APs based on their visibility at each location, we can filter out some APs from one location if they are irrelevant to this location since not all APs have the same contribution to one location. We treat those APs that occur less than the average frequency as irrelevant APs. The remaining APs of the fingerprints are considered as "relevant APs". The average frequency of APs to a location is calculated as

$$\bar{C}(L, AP) = \frac{1}{N} \sum_{i=1}^{N} C(L, AP_i) \tag{13}$$

where $C(L, AP_i)$ is the frequency of AP_i visible from the location L in the training data. NF is then a mapping function which maps the fingerprint $\boldsymbol{s}$ to $\boldsymbol{s'}$, where

$$\boldsymbol{s'_i} = \begin{cases} 0, \text{ if } C(L, AP_i) < \bar{C}(L, AP) \\ \boldsymbol{s_i}, \text{ if } C(L, AP_i) \geq \bar{C}(L, AP) \end{cases} \tag{14}$$

4 Experiment

We test and compare different indoor locationing algorithms in a two-floor campus building with a congested Wi-Fi environment. The WLAN in this building is composed of 16 APs, including seven 3-COM APs, six Motorola APs and three external APs. The fingerprints are collected from the second floor, an area of 60mx15m with a 15mx12m lounge. The floor plans and the locations of APs are shown in Fig. 1.

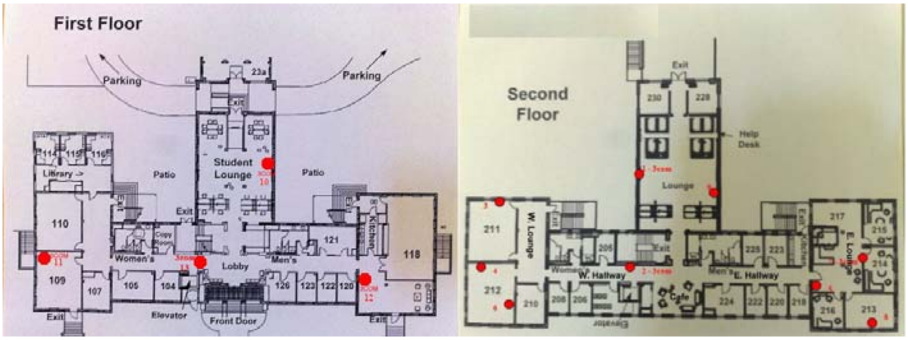

Fig. 1. The floor plans. (The dots are the locations of APs).

4.1 Data Collecting

We selected nine public rooms from the second floor and collected 1,002 unique fingerprints[3] from these locations over a period of seven days using Nokia N95 smart phones. For each room, we collected at least 100 fingerprints to ensure that every location has enough training data (Table 1). Although the size of the rooms is different, instead of measuring the mobile users by physical distance, we find that room-level location information is useful enough for most applications. Therefore, the following experiments are based on room-level location detection.

Table 1. The fingerprint distribution for each room

Rm211	Rm212	W. Lounge	W. Hallway	Cafe	Lounge	E. Hallway	E. Lounge	Rm213
150	100	125	101	101	100	100	124	101

4.2 Experiment Setup and Evaluation

We use stratified 10-fold cross-validation to evaluate the accuracies of different indoor locationing algorithms. To measure the confidence interval of the accuracy, we use the repeated random sub-sampling validation where we repeat the process, randomly choosing 90% of all fingerprints as training data and the remaining 10% as testing data for 100 times. The algorithms we evaluate include:

– Naive Bayes Classifier (NBC)
– Support Vector Machine (SVM)
  We use LIBSVM[4] to infer the locations of the measured fingerprints [13].
– K-Nearest Neighbor (KNN)
  We choose K=5 in our KNN implementation.

[3] A text file containing all fingerprints can be found at `http://mlt.sv.cmu.edu/WASP/data.csv`

[4] LIBSVM software is available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

- Redpin
  We use the original Redpin algorithm with $\alpha = 1$, $\beta = 0.4$, $\gamma = 0.2$ and K=1. We also extend the original Redpin by choosing K=5 and name this variation "Redpin5".
- Weighted AP Similarity Positioning (WASP)
  We apply NF to the historical fingerprints and extend the Redpin algorithm by choosing K=5 and adding PMI to weight different APs.

4.3 Result

The accuracies of different algorithms are shown in Table 2. NBC has the lowest accuracy because NBC only calculates the existence of a particular set of APs without considering the signal strengths. When two rooms are quite close to each other, the detected fingerprints are too similar to accurately discriminate two separate rooms. SVM and KNN have similar accuracy because they both use the signal strength information to separate fingerprints from different locations. The Redpin algorithm has better performance than KNN because it reduces the signal fluctuations by using NCAP and NNAP as bonus-penalty adjustments. The WASP algorithm we propose in this paper outperforms the original Redpin by 9% and the 95% confidence interval of the improvement is [0%, 17%], which is statistically significant.

Table 2. Accuracy of each algorithm

	NBC	SVM	KNN	Redpin	Redpin5	**WASP**
Accuracy	61%	80%	79%	81%	86%	**87%**
Confidence interval(95%)	54%-68%	75%-86%	71%-85%	76%-88%	80%-92%	**86%-96%**

Since KNN, Redpin and WASP are all instance-based learning algorithms, we compare their accuracy using different numbers of nearest neighbors (K). The result is shown in Table 3. Increasing the number of nearest neighbors leads to higher accuracy. However, we do not see any major improvement after K reaches 5. Redpin+PMI consistently improves over the original Redpin by around 1%. Though not obvious, the correlation between locations and APs does contribute to the accuracy. More research on alternative statistical methods for the correlation is planned for the future.

To see if NF can successfully reduce the impact of signal noise in the congested Wi-Fi environment for all algorithms, we apply NF to the training dataset and

Table 3. Accuracy of KNN, Redpin and WASP (K from 1 to 10)

	K=1	K=2	K=3	K=4	**K=5**	K=6	K=7	K=8	K=9	K=10
KNN	78%	78%	79%	80%	**79%**	79%	78%	79%	78%	78%
Redpin	**81%**	81%	83%	85%	**86%**	86%	85%	86%	86%	85%
Redpin(PMI)	82%	82%	84%	85%	**87%**	87%	87%	87%	86%	86%
WASP	88%	88%	90%	90%	**90%**	90%	90%	91%	90%	90%

Table 4. Accuracy of each algorithm without and with NF

Accuracy	NBC	SVM	KNN	Redpin	WASP
W/O NF	61%	80%	79%	81%	87%
With NF	**64%**	**86%**	**88%**	**88%**	**90%**

Table 5. The maximum number of original APs and relevant APs from each location

	Rm 211	Rm 212	W. Lounge	W. Hallway	Cafe	Lounge	E. Hallway	E. Lounge	Rm 213
Original APs	11	12	14	13	15	12	14	14	6
Relevant APs	**6**	**7**	**7**	**8**	**5**	**5**	**9**	**5**	**6**
Invisible APs	5	4	2	3	1	4	2	2	10

run the same experiment. The result is shown in Table 4. All algorithms benefit from NF and the accuracies are improved by 3% to 9%. To better understand the relevant APs, we list the maximum number of the original APs, the relevant APs and the invisible APs in the fingerprints of each location in Table 5 and the visibility of APs from each location in Table 6.

Table 6. The visibility of APs from each room (The relevant APs of each room are bold)

	Rm 211	Rm 212	W Lounge	W Hallway	Cafe	Lounge	E Hallway	E Lounge	Rm 213
AP1	0%	1%	5%	26%	52%	**61%**	**52%**	3%	0%
AP2	16%	1%	**42%**	**69%**	**99%**	**100%**	**79%**	18%	0%
AP3	0%	0%	1%	34%	**95%**	**100%**	**50%**	2%	0%
AP4	**65%**	**98%**	**99%**	**100%**	**99%**	**95%**	**100%**	**84%**	**61%**
AP5	8%	0%	1%	0%	0%	18%	4%	12%	0%
AP6	**43%**	**62%**	**82%**	**71%**	**83%**	**44%**	**66%**	15%	0%
AP7	**57%**	**94%**	**89%**	**77%**	**78%**	8%	**62%**	**44%**	**38%**
AP8	1%	23%	13%	46%	39%	1%	**49%**	**85%**	**96%**
AP9	5%	**39%**	34%	**60%**	58%	6%	**62%**	**94%**	**81%**
AP10	**66%**	**98%**	**100%**	**87%**	36%	10%	34%	2%	0%
AP11	**65%**	**97%**	**99%**	**85%**	37%	0%	40%	15%	0%
AP12	1%	6%	2%	18%	23%	0%	17%	28%	**34%**
AP13	0%	1%	1%	28%	26%	0%	**78%**	**97%**	**98%**
AP14	**65%**	**95%**	**97%**	**78%**	10%	0%	19%	1%	0%
AP15	0%	0%	0%	0%	1%	2%	0%	0%	0%
AP16	0%	0%	0%	0%	8%	3%	0%	0%	0%

4.4 Granularity of Rooms

In addition to the overall accuracy, we also want to know the room-level accuracy. The room-level accuracy is shown in Fig. 2. Surprisingly even though NBC has overall the worst accuracy, it has the highest accuracy in Room213. The most

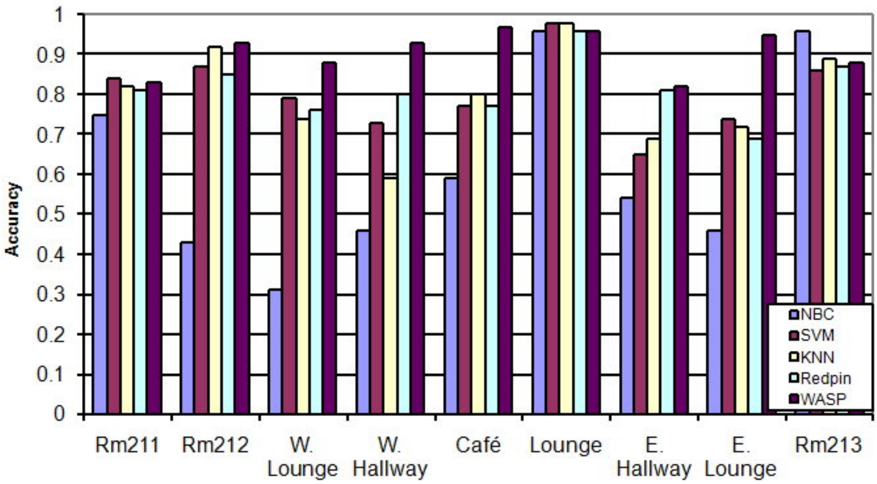

Fig. 2. Room-level accuracy

plausible explanation is that there are ten invisible APs in Room213 which is the highest value among all rooms (see Tables 5, 6). This makes the fingerprint in Room213 the most distinguishable AP set from which NBC can identify its location easily. Another interesting finding is that all algorithms have very high accuracy for the Lounge. Our hypothesis is that the Lounge is the largest room so the estimate error is less significant.

To prove this hypothesis, we create a virtual floor plan by combining adjacent rooms into a larger virtual room. For example, we merge Room211, Room212 into one virtual room (Room 211-212) and a testing fingerprint from Room211 is treated the same as Room212. The accuracy of each virtual room is shown in Fig. 3. For finer-grained locations, the WASP algorithm is the most accurate.

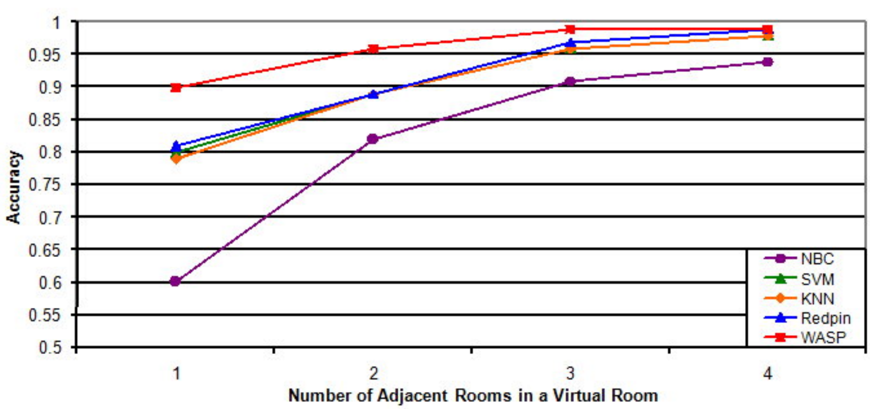

Fig. 3. Accuracy of each virtual room

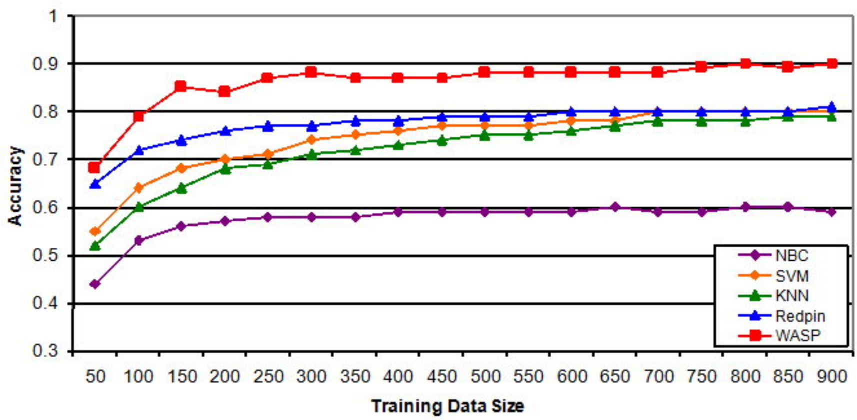

Fig. 4. Accuracy of different training data size

When one adjacent room is added, the accuracy of all algorithms can be enhanced to over 80%. When four adjacent rooms are combined, the accuracy can even be better than 90%. Therefore all algorithms are more accurate when estimating a coarse-grained location.

4.5 Impact of Training Data Size

Fingerprint approach requires labeled data collection in advance. It is not a trivia task to collect hundreds or thousands of data points. To see the precision of the five algorithms for different training size, we run an ablation study by increasing the training data size from 50 to 900 and evaluate the accuracy on the same testing data (Fig. 4). We choose the training data to ensure that each room has enough coverage. When the data size is 50, the accuracies of all algorithms are not very good. When the data size is 150, the WASP algorithm can give over 80% of the total accuracy, which is better than other alternatives. However, when the number of historical fingerprints is increased, the accuracy improvement is less apparent. One plausible reason is that while more fingerprints provide more matching samples, they also provide more polluted data which confuses the algorithms and reducing the accuracy. Since collecting training data with labels requires non-trivial human efforts, this result shows that even a small amount of training data (e.g. 150) can already provide reasonable indoor locationing accuracy.

4.6 Number of APs

When irrelevant APs are removed for a location, we observe that the indoor locationing accuracy improves. Currently NF chooses these APs occurring more than the average frequency as "relevant APs". We want to see how many relevant APs for one room are needed for acceptable accuracy. We sort the APs based on their

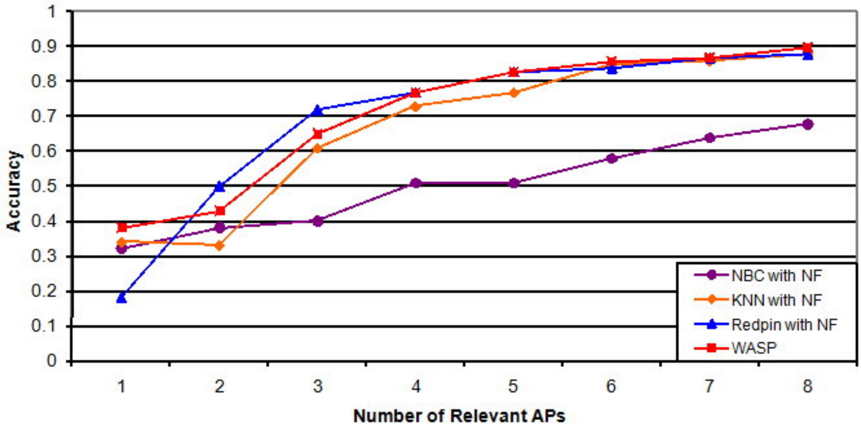

Fig. 5. Locationing accuracy based on different number of relevant APs

relevance for each room. We first choose the most relevant APs and estimate the accuracy by adding the APs one by one. In this experiment we do not include SVM because the probabilities of different training models calculated by LibSVM were not correlated. The accuracies of different numbers of APs for the other four algorithms are shown in Fig. 5. With small numbers of APs (from two to four), the Redpin algorithm has much better accuracy than KNN and WASP. One plausible explanation is that the bonus-penalty adjustment of Redpin makes the discrepancy of two fingerprints more obvious but the PMI cannot provide any additional benefit because the filtered APs are already highly relevant to the locations.

To understand how many APs are needed for reasonable indoor locationing accuracy, we sort APs according their visibility from the highest frequency to the lowest. For each run, we increase the number of APs from 1 to 16 to calculate the

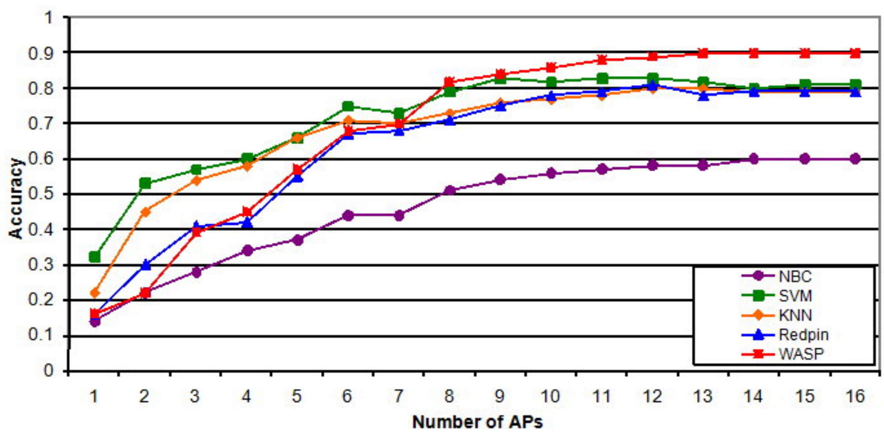

Fig. 6. Accuracy of a different number of APs

accuracy. The result is shown in Fig. 6. We see that SVM has better accuracy when the number of APs is fewer than ten because the primary APs can provide enough information for the SVM classification. However, when the number is more than ten, the SVM classification is affected by fingerprint pollution and the result is worse. On the other hand, the Redpin algorithm performs better than WASP when the number is fewer than three. However, when the number is more than seven, WASP outperforms Redpin by around 10%. Another interesting observation is that the improvement in accuracy slows after nine APs. It seems that these less visible APs do not provide essential information for location detection. Instead, they may even cause confusion when matching fingerprints and lower the accuracy.

5 Conclusion and Future Work

In this paper, we propose WASP, an enhanced indoor locationing algorithm for a congested Wi-Fi environment. Our approach takes signal strengths, AP visibility and statistical fingerprint history into consideration to enhance the Redpin algorithm in a congested Wi-Fi environment. This approach obtains the best accuracy and also works well even with the small training data set in the experiments. Even though WASP may not work well with a small number of APs, most office buildings and homes are covered by more than three APs and the fluctuations and congested signals are likely to be more serious in a real world than in the laboratory. We believe WASP can provide an overall satisfying indoor locationing prediction.

In this paper, we only chose nine public rooms on the second floors. We plan to extend the collection to more private and wall-bounded rooms over two floors. Multiple RF fingerprints[5], such as Bluetooth, might also improve the accuracy. In addition, we will explore the use of accelerometer data to determine if a user is moving or not and thereby enable time-averaging or tracking to improve accuracy. Another interesting issue is to study the optimal number of APs and their positions in the building for the best indoor locationing accuracy. Finally, we plan to apply the WASP algorithm to several mobile health and mobile professional applications. We will also design more incentive and intuitive ways to collect the fingerprints through users' collaboration.

Acknowledgements

This research was supported by grants from Nokia Research Center and by CyLab at Carnegie Mellon under grant DAAD19-02-1-0389 from the Army Research Office. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of ARO, CMU, or the U.S. Government or any of its agencies. We want to thank the initial assistance of Philip Bolliger

[5] The original Redpin used Bluetooth and GSM cell-ID to augment Wi-Fi.

to get us started using Redpin. We also acknowledge the efforts of Joshua Correa and Ed Katz, who explored an earlier set of approaches to Wi-Fi based location in our building, to Patricia Collins who reviewed multiple versions of the paper.

References

1. Bahl, P., Padmanabhan, V.: Radar: an in-building rf-based user location and tracking system. In: IEEE INFOCOM 2000, vol. 2, pp. 775–784 (2000)
2. Barcelo, F., Evennou, F., de Nardis, L., Tome, P.: Advances in indoor location. In: LIAISON - ISHTAR Workshop (Septemper 2006)
3. Bolliger, P.: Redpin - adaptive, zero-configuration indoor localization through user collaboration. In: ACM International Workshop, March 2008, pp. 55–60 (2008)
4. Brunato, M., Battiti, R.: Statistical learning theory for location fingerprinting in wireless lans. Computer Networks and ISDN Systems 47, 825–845 (2005)
5. Carlotto, A., Parodi, M., Bonamico, C., Lavagetto, F., Valla, M.: Proximity classification for mobile devices using wi-fi environment similarity. In: ACM International Workshop, pp. 43–48 (2008)
6. Correa, J., Katz, E., Collins, P., Griss, M.: Room-level wi-fi location tracking. CyLab Mobility Research Center technical report MRC-TR-2008-02 (November 2008)
7. Fan, R., Chen, P., Lin, C.: Working set selection using second order information for training support vector machines. Journal of Machine Learning Research 6, 1889–1918 (2005)
8. Hightower, J., Borriello, G.: Location systems for ubiquitous computing. IEEE Computer 34, 57–66 (2001)
9. Ho, W., Smailagic, A., Siewiorek, D., Faloutsos, C.: An adaptive two-phase approach to wifi location sensing. IEEE International Conference 5, 456 (2006)
10. Li, B., Salter, J., Dempster, A., Rizos, C.: Indoor positioning techniques based on wireless lan. In: IEEE International Conference, p. 113 (March 2006)
11. Paschalidis, I., Lai, W., Ray, S.: Statistical Location Detection. In: Mao, G., Fidan, B. (eds.) Localization Algorithms and Strategies for Wireless Sensor Networks: Monitoring and Surveillance Techniques for Target Tracking, IGI Global (2009)
12. Seshadri, V., Zaruba, G., Huber, M.: A bayesian sampling approach to in-door localization of wireless devices using received signal strength indication. In: IEEE International Conference, March 2005, pp. 75–84 (2005)
13. Wu, C., Fu, L., Lian, F.: Wlan location determination in e-home via support vector classification. In: IEEE International Conference, vol. 2, pp. 1026–1031 (2004)