Computationally Tractable Location Estimation on WiFi Enabled Mobile Phones

Damian Kelly[†*], **Ross Behan**[†], **Rudi Villing**[†] **and Seán McLoone**[†]

[†] *Department of Electronic Engineering*
National University of Ireland, Maynooth

[*]E-mail: `damian.kelly@eeng.nuim.ie`

Abstract — **Enriching a mobile device with the ability to detect its location can enable the provision of a range of Location Based Services to its user. Outdoors, the location detection facility is sufficiently provided by GPS, however GPS is not suited to the challenge of non-line-of-sight indoor environments. In these environments smaller scale location estimation techniques must be employed. Due to their ubiquity, WiFi signals are a commonly employed indicator of location; knowledge of the identity and intensity of these signals throughout an environment can allow the estimation of the receiving device's location. This paper outlines work towards the development of efficient, privacy conservative positioning algorithms suitable for deployment on commonly available mobile phones. For a number of algorithms, the frequency of correct location prediction is presented along with the execution time on a real mobile phone.**

Keywords — **Location Estimation, WiFi, Smartphone, Classification**

I Introduction

Location Based Services (LBSs) are enjoying increased adoption due to their convenience and the growing affordability of LBS capable devices. Even Global Positioning System (GPS) receivers are frequently being included in mobile devices, enabling outdoor LBSs. One recent example of such LBSs is Google's new Latitude service [1], which allows people to locate their nearby friends using the device's GPS. However, GPS is unable to obtain reliable position estimates indoors due to GPS's dependence on line-of-sight to estimate distances.

A commonly used technique to determine indoor location is to analyse readings of the available radio-frequency (RF) signals. WiFi (or IEEE 802.11) is one such signal which is commonly utilised due to its high deployment density in commonly inhabited areas such as offices, universities and urban homes. Prior research on WiFi location prediction (or localisation) techniques is targeted towards devices not typically carried by the majority of people such as laptop computers [2], or Personal Digital Assistants (PDAs) [3]. However, the need to carry such specialist devices actually serves to decrease the convenience of LBSs. Due to the ubiquity of WiFi connectivity, the number of WiFi capable mobile phones is steadily increasing. Our WiFi localisation techniques are targeted specifically at commonly available mobile phones, eliminating the need for the user to carry any extra hardware to enjoy these services.

The deployment of WiFi localisation on a mobile phone introduces some restrictions. The first is that the computational power available on a mobile phone is relatively low, hence the computational complexity of the algorithms must be minimised. As will be explained in Section II privacy concerns can be an issue impeding technology adoption. We eliminate these concerns by ensuring all necessary reference data is pre-loaded on the mobile phone. Another restriction of mobile phones is the available memory. The result is that the reference data for an entire environment needs to have a small memory footprint to be able to reside on memory impoverished mobile phones.

In Section II the main techniques and considerations for indoor localisation are described. Then in Section III the details of our chosen deployment platform and test environment are presented.

Section IV continues by describing the algorithms we implemented on a mobile phone and Section V summarises the location prediction accuracy results achieved with these algorithms. Then taking into account the algorithm execution requirements, the optimal choice for the target platform is identified.

II Related Work

The availability of location to a mobile device enables services such as navigation, location sensitive advertising, and friend and resource finding. The most recent example of a friend finding service is Google's Latitude service. This service, which works on a variety of mobile phones uses the phone's GPS receiver to obtain best position accuracy. GPS uses time-of-flight readings to estimate distances from a number of satellites at known locations. Triangulation is used to convert these distances into a position. GPS can usually determine outdoor position within 15m of the true position [4].

When GPS is not present on the mobile phone or when GPS signals are not reliably detectable, such as indoors, cellular network readings can be used. As a result, when indoors, the position error for Google Latitude's service is limited only by the coverage radius of a cell tower. This error can be anything up to 3 km, assuming the position of the cell tower is correct in the database, which is not always the case. Sending information to a server such as cell tower ID to retrieve location estimates also raises privacy concerns. If data indicative of one's location has been sent to a server, there is no way of knowing who can access and interpret this data. In this way it is difficult to ensure one's privacy while using this type of location prediction technique.

Several localisation solutions have been developed for the indoor scenario. Krumm et al. [5] have developed a localisation system in which each person to be tracked carries a badge actively sending RF packets. Custom receivers throughout the environment deduce the Received Signal Strength Intensity (RSSI) for each packet and send the information to a server for location calculation. This technique obtains an average coordinate location error of 3 m. Along with the custom hardware requirements for this system, there are also privacy issues since the location of each participant is calculated and stored on a server remote from the user.

More readily available, hence less expensive, localisation solutions exist in the form of short range RF communication protocols such as Bluetooth, Zigbee and WiFi. WiFi is an excellent RF communication technology on which to base a localisation system due to the high density of detectable Access Points (APs) in many areas, both outdoor and indoor. Commercially, WiFi has been used to compensate for GPS's shortcomings. For example, Apple's iPhone uses Skyhook's location service when GPS is unavailable. This service can predict location within 50 meters of the true position by searching for and returning the position of the detected WiFi APs from an online database of WiFi locations. Hence, this technique suffers the same privacy concerns as Google's cell tower ID technique.

Work by di Flora and Hermersdorf in [6] uses rule based classification to estimate a user's indoor position using mobile phones similar to our test device. The algorithm presented is computationally tractable on the target devices, however it only resolves location at the building floor and wing level. This resolution is not sufficient for many LBSs such as people or resource finding and navigation. A significant amount of research has been presented on calculating coordinate position from WiFi RSSI readings from laptops (e.g. [2, 7]). Coordinate location estimates typically need to be converted to room-level predictions to be meaningful to a user. Our work develops mobile phone algorithms to estimate room-level location directly from WiFi data, without the computationally excessive need to estimate coordinate position. The next section will present the test hardware and environment adopted for this study.

III Localisation Platform Deployment

There are two main components in this indoor localisation study; the mobile phone device and the environment in which it is deployed.

a) Hardware

The mobile phone adopted for this study is the Nokia E60. This phone is one of a wide range of Nokia phones which come with WiFi connectivity as standard. It has a 220MHz processor, 21MB of free RAM and 64MB of integrated memory. More recent WiFi enabled Nokia devices have superior specifications, which is why we chose the E60; if the algorithms perform well on this lower-end model they are likely to perform similarly or better on newer models. This device uses the Symbian Series 60 operating system and our application was developed using the Python script shell.

The wlantools Python package, developed by Christophe Berger, was used to obtain a scan of the available WiFi APs and the corresponding RSSIs. Symbian mobile phones have a restriction on the maximum rate at which updated WiFi scans can occur. It has been found that the E60 phone can only retrieve an updated WiFi scan every 13 seconds. Newer devices such as the N95 and E51 have been found to allow updated WiFi scans every 9 seconds, leading to higher frequency of position update.

This sampling interval places an upper limit on the time a position calculation can take. To enable the highest rate of position update on any of these devices, a position calculation must complete within 9 seconds. On the other hand, the relatively slow update rate of these devices precludes the use of Bayesian filtering techniques such as Kalman filters or Hidden Markov Models, since a person can walk a large distance in 9 seconds. Particle filters are also impractical, not only because of the high sampling interval but also because of their high computational burden.

b) Test Environment

The test environment was the first two floors of the Electronic Engineering building at the National University of Ireland, Maynooth. The building has an area of 2016 m^2 and consists of student computer labs, electronics research labs and offices. Walls are typically 12cm thick and constructed from concrete blocks. There were 11 APs detectable throughout the 2 floors of the environment and 6 APs were selected for these experiments due to the likelihood of the other APs moving during the study period.

To obtain RF data characteristic of each position within the environment, the environment was divided up into 117 approximately equally spaced positions. In each position the experimenter held the mobile phone in their hand and 3 scans were conducted while the experimenter pointed in each of the directions; North, South, East and West. Fig. 1 illustrates the mean intensity of the signal received for 3 different APs throughout the second floor of the building. Understandably the RSSI attenuates over distance and more severe attenuation can be observed across walls. This is advantageous to our approach; since the different locations are separated by walls, the signals detected in different locations will be significantly different and discriminable.

To illustrate how these signals can be used to resolve room-level location the sample space for 3 of the available APs was plotted. Fig. 2 represents the different RSSI values for different APs in different locations. It is possible to discriminate between different locations for a majority of the samples using only 3 APs. Although not easily visualised, further discriminatory power is available when all 6 APs are used. This work employs classifiers to decide which room resulted in a given WiFi scan. The remainder of this paper investigates which classifier is the most suitable for mobile phone implementation in terms of accuracy and resource conservation.

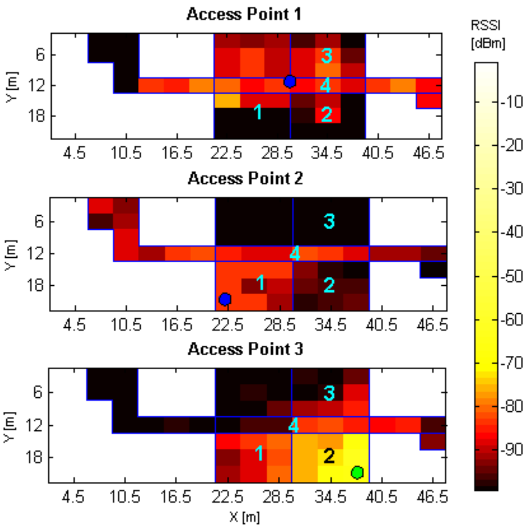

Fig. 1: 2nd floor RSSI profiles for 3 different APs with positions marked by circles. AP 3 is on the second floor. APs 1 and 2 are on the third floor. Areas in which the particular AP was not detected are in black

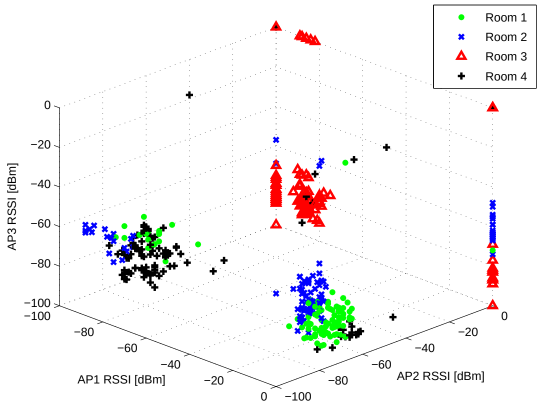

Fig. 2: RSSI Samples from the APs in Fig. 1, in rooms 1-4 indicated in Fig. 1

IV Localisation Algorithms

There are two categories of location estimation algorithms. The first is model based algorithms. These algorithms utilise small amounts of information like the channel attenuation properties and AP locations to estimate location. Triangulation is an example of this type of algorithm. The second category is empirical algorithms. These algorithms compare a test sample with the training dataset, using probabilistic or deterministic methods, to estimate the most likely position. In general the larger and more comprehensive the training dataset, the better the accuracy. Our proposed algorithms have elements of both techniques as will be explained.

a) k-Nearest Neighbour

The k-Nearest Neighbour (KNN) algorithm in its purest form is an entirely empirical algorithm. Every time location is requested by the user or another application, a test sample vector is obtained. This sample vector consists of a set of RSSI values, one for each detected AP. The sample vector is then filtered to remove references to APs which are not present in the training dataset. Next this test sample is compared with the entire training dataset to discover which training dataset samples are the most similar to this sample. Similarity can be measured with a variety of techniques in KNN, in this work the commonly used Euclidean distance measure is employed.

When the list of the k nearest or most similar training samples is populated the label for the test sample is predicted from a majority vote of the labels of the k nearest neighbours. A seminal piece of WiFi location tracking work by Bahl and Padmanabhan [7] proposed approximating a device's coordinate position by the mean of the positions of the k nearest training samples. Mantoro and Johnson [8] proposed using the KNN classifier to predict symbolic location, also based on WiFi technology. That work is similar to ours in that we are interested in room-level location, which can be considered a symbolic location tracking problem. Our work is different in that we must predict location from only one test sample rather than 14 hours of test data.

One issue with the KNN algorithm is that its accuracy is dependent on the value selected for k. Hence, using leave-one-out cross-validation an optimal value of $k = 6$ was chosen for this dataset. Implementing this algorithm on the mobile phone means that all of the training data must be stored on the phone. Depending on the size of the environment and the amount of data obtained in each location this dataset may exceed the phone's memory. In an attempt to reduce the amount of data necessary for this algorithm, a minimal representation of the training data was obtained by replacing the data in each location by a single sample vector: the mean vector of all the data in that location. As will be demonstrated in Section V this Nearest Neighbour Mean (NNM) algorithm reduces the size of the dataset and increases execution speed.

b) Linear Discriminant Maximum Likelihood Classification

Linear Discriminant Analysis (LDA) is a parametric classifier, which means that LDA classifications take place using parameters determined offline from the training data. For optimal classification it is necessary to know the posterior probability of class i, C_i, given the measurement vector x, $P(C_i|x)$. It is difficult to determine this quantity from data. Instead we can obtain $P(x|C_i)$ and use Bayes rule;

$$P(C_i|x) = \frac{P(x|C_i)\pi_i}{\sum_{l=1}^{K} P(C_l|x)\pi_l}, \tag{1}$$

where π_i is the prior probability of class i and K is the total number of classes. The denominator is simply a normalising term which can be ignored. The class density $P(x|C_i)$ can be estimated in a number of ways. In LDA it is approximated by a by a multivariate Gaussian density of the form,

$$P(x|C_i) = \frac{1}{(2\pi)^{\frac{p}{2}}|\mathbf{\Sigma}_i|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_i)^T\mathbf{\Sigma}_i^{-1}(x-\mu_i)}. \tag{2}$$

Using this expression the relative probability of class i can be defined, using the log-likelihood, as,

$$\delta_i(x) = \log\left(\pi_i \frac{1}{(2\pi)^{\frac{p}{2}}|\mathbf{\Sigma}_i|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_i)^T\mathbf{\Sigma}_i^{-1}(x-\mu_i)}\right) \tag{3}$$

For LDA it is assumed that the covariance matrix, $\mathbf{\Sigma}_i$, is equal across all classes, hence $\delta_i(x)$ becomes

$$\delta_i(x) = \log(\pi_i) + x^T\mathbf{\Sigma^{-1}}\mu_i - \frac{1}{2}\mu_i^T\mathbf{\Sigma}^{-1}\mu_i. \tag{4}$$

This function is linear in x, hence the name LDA. Based on $\delta_i(x)$ the class with the highest probability of containing the measurement x can be predicted using the discriminant function,

$$D(x) = \arg\max_i(\delta_i(x)). \tag{5}$$

The parameters of the Gaussian distribution $(\pi_i, \mu_i, \mathbf{\Sigma})$ are estimated from the training data, then classification takes place using these parameters only. When deploying this algorithm on the phone it is not necessary to perform all of the calculations online. The partial probability in equation (4) is calculated for every room when a new signal vector, x, is obtained. However the constant scalar part of this equation, denoted $a_i = \log(\pi_i) - \frac{1}{2}\mu_i^T\mathbf{\Sigma}^{-1}\mu_i$, can be calculated offline. Also, $b_i = \mathbf{\Sigma}^{-1}\mu_i$ can be computed offline giving an nx1 vector, where n is the number of features, which is equivalent to the number of detected APs. Hence, the partial probability for each room can be calculated entirely from the expression;

$$\delta_i(x) = a_i + x^T b_i, \tag{6}$$

which means that only $(n+1)$ numbers are necessary to estimate the probability of each room in the dataset. Because of LDA's efficiency in representing class probabilities, and its use of linear discriminant regions it suffers decreased discriminatory power compared to KNN. A more flexible classifier, Quadratic Discriminant Analysis (QDA), is considered in an attempt to overcome this limitation.

c) Quadratic Discriminant Maximum Likelihood Classification

LDA has linear discriminant hyperplanes, which results in reduced discriminatory power for closely intermingled classes. Hence it is necessary to consider a more flexible classifier, QDA. There are two ways to obtain non-linear discriminant hyperplanes for QDA. The first is to translate the inputs to a higher dimensional space using a polynomial, then perform LDA in this higher dimensional space. Another method is to use Gaussians with differing covariance matrices to represent each class. By permitting the covariance matrices in equation (3) to differ across classes, the simplifications in equation (4) do not occur. Instead we obtain a slightly more complicated quadratic expression,

$$\delta_i(x) = \log(\pi_i) - \frac{1}{2}\log|\mathbf{\Sigma}_i| - \frac{1}{2}(x-\mu_i)^T\mathbf{\Sigma}_i^{-1}(x-\mu_i). \tag{7}$$

This method is preferred to the polynomial technique since it does not require the optimal selection of polynomial order. It has been shown that the classification flexibility is similar for both polynomial and Gaussian QDA [9]. Without the assumption of a common covariance matrix, the simplifications of LDA cannot occur. The calculation of $\log(\pi_i) - \frac{1}{2}\log(|\mathbf{\Sigma}_i|)$ can occur offline to give a constant. However, $\frac{1}{2}(x-\mu_i)^T\mathbf{\Sigma}_i^{-1}(x-\mu_i)$ must be calculated online when a new sample is obtained. This not only increases the algorithm execution time and reduces battery life over LDA but increases the amount of data which must be stored. Now an nxn covariance matrix and an nx1 mean vector must be stored for every location.

KNN can be thought of as an empirical approach to estimating position, whereas LDA and QDA can be thought of as model based approaches. LDA and QDA are probabilistic models derived entirely from empirical data. Previous work implemented on PDAs ([3]) estimates the likelihood of reading a given RSSI from a given AP in a given location using histograms. Then a naive independence assumption allows the approximation of the joint probability by the product of the individual distributions. We avoid this assumption by using Gaussian distributions to efficiently estimate the joint distribution across several features. Now that the algorithms have been developed, they can be applied to data obtained from the environment.

V Real Environment Tests

Due to the considerable time involved in obtaining data throughout a test environment as large as ours, localisation accuracy is determined using leave-one-out cross validation on the illustrative dataset presented in Section III. Instead of the typical localisation accuracy metric of mean error distance we employ a classification success metric. However, overall accuracy is not the best measure of success since large rooms with a large amount of test samples will bias the overall accuracy. Instead we use the unweighted average of the accuracies of each individual room as an unbiased error metric. The accuracy of each room is calculated to be the ratio of correct predictions for a given room to the number of test samples in that room.

This unbiased accuracy can easily be calculated as the average of the diagonal terms in the confusion matrix. We propose that this error metric is more relevant to real world applications than error distance because error distances do not indicate containment of predictions within the correct room. For example a large error distance in a large room may not be as incorrect as a large error in a small room. Conversely, a small distance error near a wall may translate to an incorrect room prediction; an effect not highlighted in other WiFi localisation work.

Table 1 highlights the findings of this study in terms of algorithm accuracy and other classification costs such as algorithm execution time, phone dataset size and phone battery life. Out of all the algorithms KNN appears to have highest accuracy. For this to occur it requires the largest dataset, hence longest execution time. The maximum likelihood classifiers LDA and QDA do not provide increased accuracy, although they do allow more efficient representation of the environment data and faster localisation time. Unexpectedly, the NNM algorithm outperforms LDA and QDA in terms of accuracy while still providing an incredibly minimal representation of the entire environment dataset.

To test the effect of each algorithm on battery life the battery was fully charged and the algorithm scanned and predicted location once every 13 seconds on the E60. The amount of time for the battery to completely drain when using each algorithm was recorded. The NNM has the best battery life due to its quick algorithm execution time. However the battery life is approximately similar for all algorithms. Hence, the difference in battery life has little impact on algorithm selec-

Table 1: Comparison of localisation algorithms

	Acc [%]	Execution Time[s]	Dataset Size[kB]	Battery Life[hrs.]
KNN	62	0.383	53.4	10.19
NNM	58	0.0047	0.522	10.41
LDA	53	0.0157	1.92	10.38
QDA	56	0.0351	5.65	10.35

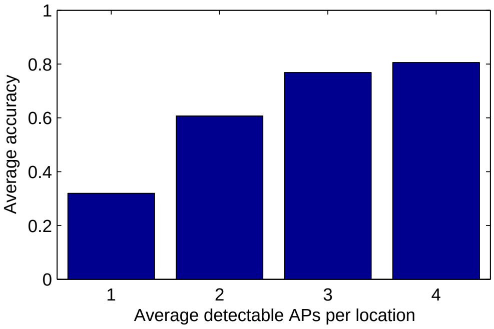

Fig. 3: Average accuracy as a function of average detectable APs per location for the KNN algorithm

tion.

Even though the most accurate algorithm, KNN, has the longest execution time it is still well within the upper limit of 13 seconds imposed by the phone's sampling rate. When higher amounts of training samples are obtained per location or for larger environments the NNM algorithm will allow for the most efficient localisation in terms of execution time and memory usage. However, even our highest accuracy not particularly high. To understand the accuracy of the KNN algorithm the average amount of APs available in each location was considered. The rooms were grouped according to the average number of APs detectable in that room. Then the average accuracy per group of rooms was calculated. Fig. 3 indicates how average accuracy varies as a function of the number of APs detectable in a given location. This indicates that it is possible to get 80% accuracy in locations where 4 APs are detectable. But since only 21% of the rooms in this test environment have an average of 4 detectable APs, our test environment does not have WiFi infrastructure deployment sufficiently dense to allow reliable location prediction.

VI Conclusions and Future Work

This paper has presented work towards indoor WiFi localisation algorithms which can perform all calculations on and source all data from the phone itself. Even though not possible at all locations, it has been shown that higher accuracy predictions are possible with a higher amount of detectable APs. It has been found that the best classification accuracy is possible in this environment using a simple brute force approach, KNN, and that KNN can outperform maximum probability classifiers even when it is trained on absolutely minimal data, namely one mean sample per location.

The main reason the probabilistic classifiers don't perform as well as KNN is that they assume that each class is unimodal Gaussian, which is not the case in light of Fig. 2. Future work will attempt to improve classification accuracy using more sophisticated density estimation techniques, such as Gaussian Mixture Models. Another issue for further investigation is the suitability of training data obtained from one phone for localisation on a different phone with different hardware.

References

[1] Google Latitude [online]. Available: `http://www.google.com/latitude/`.

[2] T. Roos, Petri M., H. Tirri, P. Misikangas, and J. Sievanen. "A Probabilistic Approach to WLAN User Location Estimation". *International Journal of Wireless Information Networks*, 9(3), July 2002.

[3] M.A. Youssef, A. Agrawala, and A.U. Shankar. "WLAN Location Determination via Clustering and Probability Distributions". In *PERCOM '03*, page 143, Washington, DC, USA, 2003. IEEE Computer Society.

[4] A. Kupper. *Location Based Sevices - Fundamentals and Operation.* John Wiley & Sons, Ltd, 2005.

[5] J. Krumm, L. Williams, and G. Smith. "SmartMoveX on a Graph - An Inexpensive Active Badge Tracker". In *UbiComp '02*, pages 299–307, London, UK, 2002. Springer-Verlag.

[6] C. di Flora and M. Hermersdorf. "A practical implementation of indoor location-based services using simple WiFi positioning". *Journal of Location Based Services*, 2(2):87 – 111, June 2008.

[7] P. Bahl and V.N. Padmanabhan. "RADAR: an in-building RF-based user location and tracking system". In *INFOCOM'00*, volume 2, pages 775–784 vol.2, 2000.

[8] T. Mantoro and C.W. Johnson. "nk-nearest neighbor algorithm for estimation of symbolic user location in pervasive computing environments". In *WoWMoM'05*, pages 472–474, 2005.

[9] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning.* Springer-Verlag, 2001.