

Wireless Symbolic Positioning using Support Vector Machines

C. Philipp Schloter
Department of Electrical Engineering
Stanford University
Stanford, CA 94305
pschloter@stanfordalumni.org

Hamid Aghajan
Department of Electrical Engineering
Stanford University
Stanford, CA 94305
aghajan@stanford.edu

ABSTRACT

This paper introduces a novel symbolic positioning system based on wireless access points and Support Vector Machines. The system works both indoors and outdoors and is cost-effective since it can even work with widely deployed 802.11 access points as infrastructure. The system requires minimal setup time, which makes it readily available for real-world applications.

Categories and Subject Descriptors

C.2.1 [Computer-Communication Networks]: Wireless communication – *Wi-Fi, GPS, Location Algorithms*.

General Terms

Algorithms, Design, Experimentation.

Keywords

802.11, Wi-Fi, WiMAX, OFDM, GPS, Location-based Services, LBS, Support Vector Machines, Radio Mapping, RSSI

1. INTRODUCTION

Knowing a user's location enables many new applications, often times called location-based services. These applications can be categorized into four main areas: triggering, information filtering, tracking and assistance services. An example of triggering is location-based advertising – an advertisement gets triggered when a user enters a certain area. Entering a location can also trigger different rates or tariffs for wireless services. Another application is surveillance. Information filtering allows users to receive information relevant to their current location. For example, users can find close-by restaurants, get the quickest route to a local movie theatre or get the current weather conditions based on the location they are at. Tracking allows finding out the location of another entity. Possible applications of tracking include finding people, managing vehicle fleets and tracking goods.

There are many approaches to determining a user's location – some of which are based on Wi-Fi networks [1, 2], and the correlation between signal strength and distance. The ability to

use Wi-Fi-based positioning has significant advantages over other approaches. Satellite-based GPS only works outdoors. Other methods such as infrared (IR) or radio-frequency (RF) beacon tagging overcome this limitation of GPS, but often require the expensive deployment of new infrastructure. Wi-Fi-based positioning builds on an already existing, widely available infrastructure and works indoors as well as outdoors. For a survey of location systems, see [3].

The Global Positioning System (GPS) is one of the most widely know positioning systems. Based on 24 satellites in six orbital planes, GPS can achieve up to 1 to 5 meters precision (using a differential reference or Wide Area Augmentation System). GPS has worldwide coverage and was designed so that there are always at least four satellites visible from any given point on the Earth. However, GPS requires receivers to be in line of sight with the satellites, which leads to GPS' major shortcoming: GPS does not work indoors.

There are several techniques that support indoor positioning and overcome the limitation of GPS. For example, one technique is to place beacons throughout a building (for example IR, ultrasound, or RF beacons). Once a user comes into the proximity of a beacon, the user can determine its location by looking up the beacon's identifier in a database. A more sophisticated approach is to employ multiple badges at a single location, so that a more accurate position can be determined via triangulation. Beacon-based positioning systems usually require significant infrastructure. If there is no beacon in reach, the system does not work. As a result most propriety beacon-based systems are expensive to deploy. Open infrastructures such as cellular GSM or 802.11b get around this cost issue since they are already widely available.

This paper aims to implement a positioning system on 802.11, which allows determining a user's location. The paper focuses on systems that determine a user's location at the device or client itself and not at an external server or access point. Section 2 of the paper discusses the various aspects involved with Wi-Fi-based positioning systems. Section 3 gives an overview of symbolic positioning and Support Vector Machines and the proposed approach. Section 4 describes the implementation of the system, which is followed by section 5 on experimental results and measurements. Section 6 describes a demo of the system. The paper concludes with a discussion of future work and some concluding remarks.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

IWCMC'06, July 3–6, 2006, Vancouver, British Columbia, Canada.
Copyright 2006 ACM 1-59593-306-9/06/0007...\$5.00.

2. 802.11-BASED POSITIONING

While GSM base stations are relatively far apart and their positioning resolution is low, 802.11 access points are usually not as far apart and provide a higher positioning resolution that is sufficient to distinguish even between different rooms in a building. Like GSM, 802.11 works indoors as well as outdoors. 802.11-based positioning is possible since the standard exposes signal strength measurements via the received signal strength indicator (RSSI). Since a change in signal strength is correlated to the distance between two stations, we can estimate locations based on RSSI values from multiple access points.

2.1 RSSI

The following discussion will focus on the 802.11b high rate, direct sequence spread spectrum physical layer (DSSS PHY) specification since it is currently the most widely deployed standard. Access points transmit signal strength information to stations. The RSSI is a measure of the RF energy received by the DSSS PHY. RSSI values of up to 8 bits (256 levels) are supported. The PMD primitives PMD_RSSI is issued to update the RSSI parameters reported to the MAC layer. The primitive follows the following parameters [4]:

Table 1. Overview of the “PMD_RSSI.INDICATE” primitive

Parameter	Associated primitive	Value	Description
RSSI	PMD_RSSI.indicate	0–8 bits of RSSI	RSSI is a measure of RF energy received by the high rate PHY

2.2 Path Loss Model

The basic path loss model tells us that a drop in signal power or signal strength is correlated to distance. For example, if we disregard obstructions, multipath delay spread, scattering, refraction and diffractions effects, and assume all stations to be stationary (i.e. no Doppler effect), we can model the signal loss by [5]:

$$L = \frac{P_r}{P_t} = \left[\frac{4\pi r}{\lambda} \right]^2, \text{ where:}$$

P_r = Power measured at Receiver in Watts (W)

P_t = Power measured at the transmitter in Watts (W)

λ = Wavelength in Meters (m)

r = Receiver to transmitter distance in Meters (m)

2.3 Triangulation

Solving the above equation for r , we can easily determine the distance between receiver and transmitter based on the wavelength (in the case of 802.11b, $\lambda = c/f = 3 \times 10^8 \text{ m/s} / 2.4 \text{ GHz} = 0.125 \text{ m}$) and the measured power at receiver and transmitter. Knowing the distance between multiple transmitters and a

receiver, one can determine the location of the receiver by triangulation.

2.4 Assumptions

The described model clearly makes assumptions (e.g. no obstructions, no multipath delay spread) that clearly do not hold in the real world. Wireless LANs are mostly deployed indoors where there are many obstacles such as walls or furniture. Of course, one can augment the above propagation model by taking different types of materials or multiple building floors into account [6, 7]. However, propagation models quickly become overly complex if one aims at high level of accuracy.

2.5 Radio Mapping

Instead of using a theoretical propagation model, one can use sampling techniques to build a “Radio Map”. One of the first 802.11-based positioning systems to take this approach was Microsoft’s “RADAR system” [8, 9]. RADAR requires users to sample a Radio Map of a location upfront. The Radio Map represents signal strength vectors at absolute positions (x,y,z). Before employing the system, one records a set of positions and measures the according signal strength vectors at these positions. Each component of the signal strength vector represents the signal strength of one of the access points in reach. With this type of calibration, the system can achieve 2-3 meters of accuracy. Sampling Radio Maps can take several hours for a single floor of a medium-size office building. Moreover, Radio Maps need to be regularly updated - for example if an access point gets moved.

2.6 Reducing Setup Time

To reduce the time required for calibration, Kruman and Patt recently proposed an algorithm that interpolates positions as a function of signal strength from relatively coarse calibration data [10]. While the algorithm eliminates the need to sample every location, it does not eliminate the need for a location blueprint and associating position coordinates with signal strength samples. The algorithm does not eliminate the need for calibrating signal strength samples with specific position coordinates. While the number of samples required was reduced, the effort required to set such system up hinders mass-market adoption.

3. APPROACH

The need for Radio Maps limits real-world applicability. In order to increase the usage of 802.11-based positioning, we propose a symbolic positioning system employing Support Vector Machines (SVMs), which is one of the most effective algorithms available for location classification. By trading off complexity with implied physical coordinates, as e.g. in [15], the symbolic approach significantly reduces upfront setup time.

3.1 Limitations of Physical Positioning

Usually the physical position coordinates are only an intermediate step in determining a user’s location. First, the signal strength vector is interpreted into position coordinates. Then, in most applications, the physical position is interpreted yet again into a location description or symbol (see Figure 1).

For example, someone looks up a position on a map or building blueprint. Before using such a system, one needs to map signal

strength measurements to physical coordinates, e.g. by using a building blueprint or map. Getting the required map, storing it in the system and capturing the respective signal strength and positioning values can be time-consuming. Moreover, with physical positioning systems, one usually obtains a fixed average resolution, e.g. 2.5 meters. However, this resolution might be highly ineffective in determining whether a user is in location A or B since locations of interest are usually not laid out on an equidistant grid (see Figure 2).

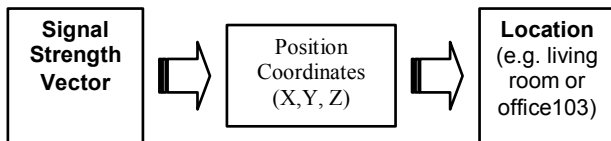


Figure 1. Traditional location determination in Radio Map approach. Based on signal strength values, the system looks up a position in the radio map. Based on the position, the system or user looks up a location.

3.2 Symbolic Positioning

Instead of providing physical location information, the proposed system provides symbolic information [3]. Cutting out the intermediate step of determining the physical position greatly reduces setup time. No location blueprint is required and as little as three samples (approx. two seconds each) can be enough to learn how to distinguish one location from another. Furthermore, symbolic positioning overcomes the resolution limitation of physical positioning.

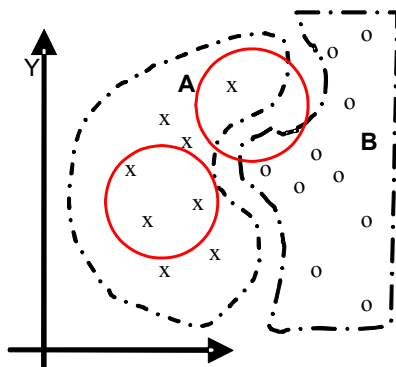


Figure 2. “X” marks positions in location A and “o” marks positions in location B. The circle indicates the fixed resolution of a physical system. While in one case the resolution is high enough to determine the location correctly, in the other case the circle overlaps with the two locations.

The system regards the locations directly in signal strength space and does not translate them into physical position coordinates. It considers every location as a cluster of pre-captured signal strength vectors and predicts the location directly from the current signal strength vector. While the system no longer has the position coordinates available, it provides enough resolution

where needed to determine a location. It predicts location symbols instead of physical coordinates. Not only does this approach overcome the resolution problem, it also significantly reduces the required setup time. The system eliminates the need to acquire a blueprint or map. One no longer needs to go through the process of entering physical position coordinates when conducting measurements. Instead, one quickly types in a symbol or location name and moves the capture device within the target location area for a couple of seconds. Even with only three samples per location the system was able to achieve 99.9% prediction accuracy in distinguishing two locations. With three samples per location and no upfront mapping requirement, one should be able to classify a mid-sized office building in less than 30 minutes.

3.3 Support Vector Machines

In order to learn and predict the signal strength vectors, we employ SVMs as recommended in [11]. SVMs perform binary classification and regression estimation. They minimize the VC (Vapnik Chervonenkis) dimension based on structural risk minimization. A low VC dimension means a low expected error rate and good generalization. To minimize the VC dimension, SVMs map the input space to a high dimensional feature space by a non-linear transformation. In the high dimensional feature space a linear classifier is constructed. This approach works well since the generalization ability of SVMs only depends on the VC dimension and not on the dimension of the input space. Moreover to construct the classifier, SVMs only need to calculate an inner product between two input vectors.

In order to classify some signal strength data vectors, consisting of values from n access points, measured at two different locations, into two according classes, the algorithm will identify a hyperplane that separates this data. The hyperplane will separate the data with maximum distance to the closest data point from both classes, i.e. maximize the margin. This property allows to more accurately classify new incoming data vectors since the separation between the two classes is greater. The vectors closest to the maximum-margin hyperplane are called support vectors.

The SVM considers data points of the form

$$\{(x_1, c_1), (x_2, c_2), \dots, (x_n, c_n)\}$$

where c_i denotes the class point x_i belongs to, either 1 or -1 and x_i is in R^n if we assume for simplicity that set of access points stays the same. This is the training data, which denotes the correct classification for each previously sampled data vector. Eventually the SVM should be able to distinguish newly measured data points by means of the dividing hyperplane, which takes the form

$$w \cdot x - b = 0.$$

In order to maximize the margin, one needs to identify the support vectors and parallel hyperplanes (to the optimal hyperplane) closest to these support vectors in either class. One can write the two parallel hyperplanes as:

$$w \cdot x - b = 1,$$

and $w \cdot x - b = -1.$

Now these hyperplanes should maximize the distance from the dividing hyperplane and have no data points between them. The

distance between the hyperplanes is $2/|w|$, so maximizing this distance is equivalent to minimizing $|w|$. To exclude data points, we need to ensure that:

$$\text{For } \forall_i \quad w \cdot x_i - b \leq -1,$$

$$\text{or} \quad w \cdot x_i - b \geq 1.$$

This can be written as the following constraint:

$$c_i (w \cdot x_i - b) \geq 1 \quad 1 \leq i \leq n$$

The problem is now to minimize $|w|$ subject to above constraint. One of the most common algorithms for solving this problem is called SMO [12]. To learn more about SVMs and the VC dimension, please see [13].

4. IMPLEMENTATION

An 802.11-based positioning system was implemented allowing determination of a user's location. The system captures signal strength values via the WRAPI from different access points. The signal strength measurements are then tagged with location labels or symbols that allow distinguishing between different locations. Then SVMs are employed to classify incoming signal strength information and predict the location label. We developed a short demo to showcase the system, which is described in Section 6.

4.1 Software Architecture

The software architecture is illustrated in Figure 3. The proposed system consists of two main components. The first component – “SS Capture” – consists of code to receive signal strength information from the 802.11 physical layer through the WRAPI driver layer. The second component – “Learning” – contains the code to learn and classify signal strength vectors. This component was implemented on top of *libsvm* as in [14]. The system is multi-threaded so that it can learn new locations, while predicting locations based on existing knowledge.

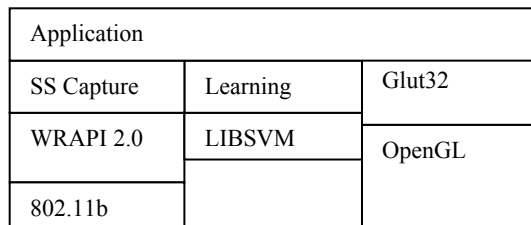


Figure 3. Applications such as the implemented demo are built on two main components – SS Capture, Learning. These in turn build on lower-level APIs and standards. Applications can use additional APIs such as Glut32 and OpenGL for graphical representations.

4.2 RSSI

Several low-level drivers for capturing raw 802.11 signal strength data were tested (see Table II). Different 802.11b cards including IBM and Dell PCMCIA cards as well as an AMD reference design card were employed for this purpose. The PCAP driver does not provide access to PHY layer data on Windows – only on Linux. The AMD reference driver only provides signal strength information from the currently associated access point. WRAPI

2.0 provides signal strength information for all surrounding access points, but only runs on Windows XP. Ultimately, we chose to implement the system on top of the WRAPI 2.0. Note that WRAPI requires the Windows XP SP1 DDK for compilation. The “include directories” from the DDK should be included before other default include directories.

The learning component is implemented on top of *libsvm*. It exposes two core functions to interact with signal strength capturing component:

```
1) void addsample(int v1,int v2, int v3, int v4, int v5,char* symbol);
```

```
2) char* SVMPredict(int v1,int v2, int v3, int v4, int v5);
```

Addsample adds a signal strength vector, associates it with a location symbol and trains the SVM. *SVMPredict* obtains the predicted location symbol based on a new signal strength vector.

Table 2. Various low-level drivers to access signal strength information

	Win PCAP	PCAP Linux	AMD Reference Driver	WRAPI 2.0
Windows support	X		X	X
SS from current AP		X	X	X
SS from all surrounding APs		X		X

4.3 Maximum Number of Access Points

The system is designed to support any number of access points. A hash table guarantees that incoming signal strength values get associated with the originating access point. Each access point is uniquely identified by its MAC address and gets a unique entry in the hash table.

5. RESULTS

As a first test, the signal strength values from five access points at different locations were recorded without smoothing (see Figure 4). The results were fairly noisy.

Additional tests were implemented to evaluate the effect of smoothing signal strength values over time on accuracy. As more samples are averaged, the variance in accuracy greatly decreases (see Figure 5).

As the number of averaged samples increases, the average accuracy initially goes down and then goes up again to converge to 100%. In typical test scenarios with two locations, the system achieved 99.9% accuracy with 200 samples (Figure 6). The difference from day 1 to day 2 can be explained through a difference in the quality of the training data. Measurements were conducted with 2 locations, 8 runs, 1000 samples per run, and

learning databases with 3 captured samples per location over 2 days.

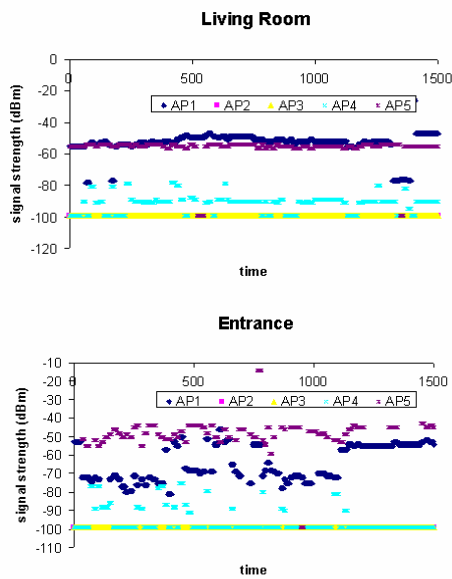


Figure 4. Signal strength at different locations with five access points.

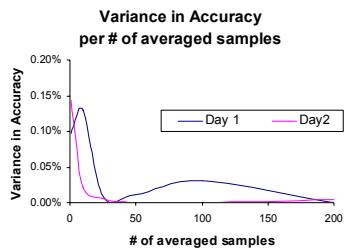


Figure 5. Variance decreases as more samples get averaged.

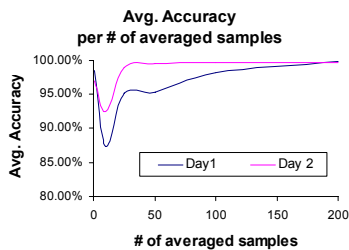


Figure 6. Accuracy increases as more samples get averaged.

6. DEMO

To showcase the system, we implemented a demo program that graphically indicates signal strength values from different access

points, can learn locations and continuously predict locations based on existing knowledge. The demo was developed in C++. For the graphics, we employed the Glut32/OpenGL APIs.

6.1 Starting the Demo

Before starting the system, one must start the *ndisui0* service and stop the Wireless Zero configuration service (WZCSVC). The demo takes several arguments. First, one needs to specify the number of the NICs to use, where the first NIC in the system is numbered 0. The second argument specifies the number of samples to be averaged – the higher this number, the higher the precision, but the slower the system reacts to changes in location. The demo supports additional arguments for testing and logging, which are out of the scope of this paper and are not described here.

6.2 Graphical Representation

The demo is designed for up to 5 access points and up to 3 locations. The signal strength of each access point is shown by 5 colored spheres that move right as the signal strength of the according access point increases (Figure 7).

6.3 Learning

In the beginning, the system needs to learn the different locations. One simply walks to the appropriate location and hits a key. A key can be associated with a particular location – e.g. “the Stanford Quad”. A user could also enter a location description and then associate captured signal strength vectors with this new location. In the demo, three locations are preprogrammed: “the Stanford Quad”, the “Dish” and “San Francisco” (they are activated by hitting the Q, W and R key respectively).

6.4 Prediction

Once the locations are learned, the picture in the background gets constantly updated based on the currently observed signal strength vector – the demo program smooths the signal by averaging multiple samples. For example in Figure 7, the system predicts that the user is at the Stanford Quad and shows a picture of the Quad. New locations can be learned anytime, while the program continues to predict the location based on the existing knowledge. This way the knowledge base can incrementally grow over time.

7. CONCLUSION AND FUTURE WORK

A symbolic positioning system based on 802.11b and SVMs was introduced and implemented. The system works both indoors and outdoors and is cost-efficient since it only requires widely available 802.11b access points as infrastructure. The system requires minimal setup time compared to other systems, which makes it readily available for real-world applications.

In the case of very large service areas, additional simulation tools might be able to estimate parts of the radio-map up-front and then get refreshed via real-world measurements as the system gets used, further decreasing initial setup time.

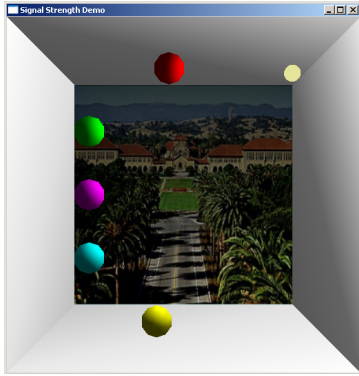


Figure 7. In the demo, the five colored spheres illustrate the signal strength for the five access points – spheres more to the right indicate higher values.

8. REFERENCES

- [1] IEEE Std. 802-11.1997, *IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*. Approved 26 June 1997.
- [2] Adam Harder, Lanlan Song, and Yu Wang, *Towards an Indoor Location System Using RF Signal Strength* in IEEE 802.11 Networks, Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05), 2005.
- [3] Hightower, Jeffrey and Borriello, Gaetano. Location Systems for Ubiquitous Computing, *IEEE Computer* 34(8). pp. 57 – 66, 2001.
- [4] IEEE 802.11b-1999 Supplement to 802.11-1999, *Wireless LAN MAC and PHY specifications: Higher speed Physical Layer (PHY) extension in the 2.4 GHz band*.
- [5] Smith, Albert A. Jr. *Radio Frequency Principles and Applications*. New York: IEEE Press, pp 43-46, 1998.
- [6] H. Hashemi, *The Indoor Radio Propagation Channel*, Proceedings of the IEEE, Vol. 81, No. 7, pages 943-968 July 1993.
- [7] Pahlavan, Kaveh and Krishnamurthy, Prashant. *Principles of Wireless Networks*, pp 417-448, Prentice Hall PTR, 2002.
- [8] Paramvir Bahl and Venkata N. Padmanabhan. *RADAR: An in-building RF-based user location and tracking system*. In IEEE INFOCOM 2000, pages 775–784, March 2000.
- [9] Bahl, P., V.N. Padmanabhan, and A. Balachandran, *Enhancements to the RADAR User Location and Tracking System*, Microsoft Research: Redmond, WA, 2000.
- [10] John Krumm and John Platt, *Minimizing Calibration Effort for an Indoor 802.11 Device Location Measurement System*, November 13, 2003, Technical Report, MSR-TR-2003-82, Microsoft Research, Redmond, WA, 2003.
- [11] Roberto Battiti, Mauro Brunato, Allesandro Villani, *Statistical Learning Theory for Location Fingerprinting in Wireless LANs*, Technical report DIT-02-0086 Università di Trento, October 2002.
- [12] J. Platt. *Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines*, Microsoft Research Technical Report MSR-TR-98-14, 1998.
- [13] V. N. Vapnik. *The Nature of Statistical Learning Theory*, Springer Verlag, 1995.
- [14] Chih-Chung Chang and Chih-Jen Lin, *LIBSVM: a library for support vector machines*, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
- [15] Guolin Sun and Wei Guo, *Robust Mobile Geo-Location Algorithm Based on LS-SVM*, IEEE Transactions on vehicular technology, Vol. 54, No. 3, May 2005.