LoCo: A Ready-to-Deploy Framework for Efficient Room Localization using Wi-Fi

Jacob T. Biehl, Matthew Cooper, Gerry Filby, Sven Kratz
FX Palo Alto Laboratory, Inc.
3174 Porter Drive
Palo Alto, CA, USA
{biehl, cooper, filby, kratz}@fxpal.com

ABSTRACT
In recent years, there has been an explosion of social and collaborative applications that leverage location to provide users novel and engaging experiences. Current location technologies work well outdoors but fare poorly indoors. In this paper we present LoCo, a new framework that can provide highly accurate room-level location using a supervised classification scheme. We provide experiments that show this technique is orders of magnitude more efficient than current state-of-the-art Wi-Fi localization techniques. Low classification overhead and computational footprint make classification practical and efficient even on mobile devices. Our framework has also been designed to be easily deployed and leveraged by developers to help create a new wave of location-driven applications and services.

Author Keywords
Indoor Location; Ubiquitous Computing; Workplace Movement; Presence; Context.

ACM Classification Keywords
H.3.4. Systems and Software; H.5.3. Group and Organization Interfaces

General Terms
Human Factors; Algorithms; Measurement.

INTRODUCTION
Leveraging physical location to drive meaningful, context specific user experiences has been a hallmark of mobile computing from the very beginning. While GPS and other satellite-based technologies provide pervasive outdoor location, reliable indoor location remains out of reach despite years of research. A primary adoption inhibitor to many proposed solutions is that they require users to wear or carry specialized devices for tracking; e.g. badges that recognize infrared beacons [19] or specialized radio devices.

Ubicomp '14, September 13 - 17 2014, Seattle, WA, USA

ACM 978-1-4503-2968-2/14/09...$15.00.
http://dx.doi.org/10.1145/2632048.2636083

Recently, there has been interest in leveraging existing Wi-Fi infrastructure and mobile devices to determine indoor location. The typical approach to Wi-Fi localization is to use the client's received signal strength indicator (RSSI) of fixed Wi-Fi access points (APs) in a multilateration calculation to predict a point-based location of the device. Due to the inherent error of these calculations, most tracking results have up to several meters of error.

An alternative to multilateration is classification using RSSI fingerprints. The approach trades precision for accuracy, and provides a high confidence prediction of what room the device is in. State-of-the-art Wi-Fi fingerprinting techniques (e.g. [2]) are documented to provide accuracy near 85%. We have designed our location framework, LoCo, to be more practical to deploy and more easily leveraged by location-driven application developers. We use supervised learning to provide high accuracy indoor room classification based on the relative ordering of pairs of APs by RSSI. LoCo enables classification to be performed more frequently and on resource-constrained devices without imposing a heavy burden on users (it's merely an app on their smartphone).

RELATED WORK
Wi-Fi location determination methods can be broadly grouped as either distance-based or probabilistic. Distance-based methods represent the set of RSSI values at a given location by a known (absolute) set of values and minimize a distance measure to assign a test data point to a known location [2, 5, 8, 16]. More recent work has focused on probabilistic approaches, which involve modeling radio propagation statistically in combination with previously defined templates for the signal strengths observed at various locations [10, 22]. Other approaches include localization via genetic programming [3], Bayesian networks [9], and combined mapping of locations with tracking of devices as a semi-supervised learning problem [15].

Other methods have emphasized AP ordering according to RSSI. One system originally from the University of Toronto [18] created a rank vector which was then normalized and used with k-nearest-neighbors (kNN) matching for location estimation. High accuracy matching methods based on kNN typically require the storage of a search data structure that grows with the training set. Also, the application of specialized data structures for efficient kNN search with non-metric distance measures can be difficult [1]. These accelerated versions of kNN also scale poorly with feature dimensionality, limiting expansion of the location system to areas with large numbers of APs.

Supervised methods for location estimation are less prevalent than matching approaches. Support vector machines and naive Bayes classifiers compared poorly with the RedPin/kNN approach in [8]. The work in [14] describes a method that combines generative modeling using EM to help address missing data and discriminative modeling with SVMs for localization. More broadly, there has been work that performs trilateration through time of flight and/or angle of arrival (e.g. [12, 4, 21]). While sophisticated, these approaches require modification and/or special software to run on the Wi-Fi base stations. While gains have been made in addressing multipath issues in dense environments like office buildings (e.g. [17]), current techniques have not been shown to be robust in real-world applications.

LoCo uses boosting [6] to build a set of room-specific classifiers. The boosting training procedure greedily selects RSSI features that discriminate each room from the others. The approach has very low computational requirements at classification time, providing substantial complexity reductions relative to matching (i.e. kNN) methods.

THE LOCO FRAMEWORK

The LoCo Framework consists of three main components: a deployment of Wi-Fi Access Points, a client service running on a smartphone, and a classification engine.

LoCo leverages the received signal strength indicator (RSSI) of the BSSID beacons coming from in-range 802.11 Wi-Fi access points (APs). These values are easily accessible by most most smartphone operating systems. Wi-Fi signals travel relatively far, providing data coverage for multiple users and offices. As a result, there can be significant distance, and objects, between the AP and the receiving device that can introduce variable attenuation of the signal.

A scanning service needs to run on the user's smartphone to collect Wi-Fi signal measurements. We have initially implemented this service using Google's Android OS. We leverage the internal API for retrieving a list of in-range Wi-Fi APs and the RSSI from those APs' BSSID advertisements. The client is implemented as an Android service. With fewer than 30 lines of code, an app developer can create a dependence on LoCo and make queries to the service for the device's location. Centralizing in a service not only makes LoCo easily available to other apps, the model also ensures that only one instance of the scanning routine is running at a given time. This eliminates duplicate scanning that can reduce battery performance. When an application subscribes to LoCo, it can also specify how often the framework should determine its location.

The LoCo framework can process the scan data in two ways. The classifier, along with a trained definition library can be loaded on to the smartphone allowing classification to be performed on-device. As we describe below, our classification technique has low computational complexity allowing it to easily run on smartphone devices without consuming a significant amount of energy. Alternately, the scan results can be sent as a JSON structure to a cloud-based classification server.

We use an ensemble learning method known as boosting [6] to achieve high accuracy and efficiency for room level classification. Boosting offloads the bulk of computation to an *offline* training procedure, and retains efficiency at classification time. While boosting for binary classification has been extensively analyzed, multi-class boosting (such as the room identification problem here) remains an area of active research [11, 7]. We use a "one versus all" formulation such that the estimated room is simply the maximum score over the set of per-room classifiers, that each distinguish one room from the rest. For each room, we construct a binary classifier that estimates the probability that the RSSI vector S was observed in that room:

$$F_{\text{room}}(S) = \sum_{m} w_m h_m(S) \quad . \tag{1}$$

Each per-room classifier linearly combines "weak learners", h_m according to scalar weights w_m. The weak learners are decision stumps that compare one feature value to a threshold θ_m:

$$h_m(S) = \begin{cases} 1 & X_m \geq \theta_m \\ 0 & \text{otherwise} \end{cases} \quad . \tag{2}$$

To assemble a rich pool of weak learners to form the per-room classifiers, we propose *margin features*. For an environment with B total APs, we compute the set of unique pairwise differences (margins) between the RSSI vectors' elements, $\{X_m = S(a_m) - S(b_m)\}$, for $a_m, b_m \in \{1, \cdots, B\}$. The set of margin features has size $0.5 \cdot B \cdot (B-1)$. Intuitively, these features express coarse, localized order information of the pairs of APs, but allow more refined thresholding for classification. Missing RSSI values for specific APs in the training set are set to a nominal minimum value, R_{min}, to incorporate the cue that specific APs are not visible at specific locations.

The margin features form the input to boosted classifier construction. We employ a multiclass extension of the discrete adaboost classifier described in [23] (SAMME - algorithm 2 in reference). The training procedure identifies a location-specific set of weak learners of the form of (2) that best discriminates that location from all others. The weak learners and their corresponding weights

data set	# scans	# rooms	# APs	method	accuracy	time (s)
set I	693	39	90	RedPin/ [2]	0.947	0.651
				LoCo	0.944	0.00553
set II	1181	55	159	RedPin/ [2]	0.913	3.054
				LoCo	0.943	0.00562

Table 1. Classification accuracy and (wall) timing results for the two data sets. Accuracy is computed using 9-fold cross validation. Test timing is averaged per RSSI scan in seconds.

are learned in a greedy stagewise process. A margin feature and threshold are selected at each iteration to minimize the weighted error of the training set with the currently constructed classifier. After the new weak learner is added to (1), the training sample weights w_m are updated to reduce overall misclassification, and the process continues. The output of training F_{room} is the specification of the set of weak learners, h_m and weights w_m of (1). Each weak learner is determined by two hardware IDs a_m, b_m (MAC addresses of specific Wi-Fi APs) and a threshold, θ_m.

Given a scan of RSSI measurements S_{test} for location estimation, each of the per-room classifiers is applied. We compute only the required set of RSSI differences that were selected in the classifier training procedure. These differences are compared to the thresholds (θ_m) determined during training and then combined linearly as in (1). The final room estimate is determined by comparing the scores $\{F_{room}(S_{test})\}$ and selecting the maximum. Note that the bulk of the computation occurs in the offline training procedure. The test complexity depends only on the *selected* number of weak learners. Thus, unlike matching methods, classification complexity is decoupled from the training set size.

EXPERIMENTS

To validate our system, we conducted two sets of experiments using data collected from a single floor of our office building. The RSSI values were recorded by a variety of Android mobile devices that included Samsung Galaxy SIIIs, LG Nexus 4s, and LG Nexus 5s.

[8] presented an empirical evaluation of multiple approaches to room level indoor localization in which variants of the RedPin system [2] achieved superior accuracy. We thus use RedPin to represent state of the art performance. We implemented their basic matching approach using kNN classification with $k = 5$.

For the boosting classification system described here, we run 90 iterations of classifier training offline to select the most informative subset of margin features for discriminating each location. Throughout, we report accuracy and timing results averaged over 9 fold cross validation. Classification timing results here are computed using a PC with a 2.8GHz AMD processor. We report results using the same protocol for two sets of collected scan data.

The first data set consists of 693 scans collected in 39 locations in which RSSI values are observed from a total of 90 unique Wi-Fi BSSIDs. The locations consisted of small, single person offices (approximately 9.5 square meters) and medium sized meeting rooms (approximately 20 square meters). The large number of Wi-Fi devices is the result of two conditions. First, the seven APs that we used are commercial Cisco APs that can service multiple networks and provide access across multiple frequencies (mainly channels on 2.4GHz and 5GHz). These APs produced 42 unique BSSIDs. Second, because our office shares a building with other companies and has close neighbors, the remaining unique BSSIDs are from these organizations' APs. The data set contains an average of 22.2 scans per location (s.d. = 6.26).

The second data set is larger in scale in terms of the number of locations, 55, and the number of Wi-Fi BSSIDs, 159. This data set consists of 1181 total RSSI scans, with an average of 19.05 scans per location (s.d. = 5.21).

Table 1 summarizes our experimental results. The rightmost columns show the classification accuracy and the average time required to classify one test scan. The top two rows compare RedPin/ [2] and boosting/LoCo for the smaller data set. They perform at comparable accuracy, but LoCo is substantially more efficient (e.g. 0.00553s vs. 0.651s). Results in the bottom two rows for the larger data set show LoCo achieves superior accuracy relative to RedPin. Furthermore, while the classification complexity of RedPin increases with the scale of the data set, the complexity of LoCo is unchanged and, as a result, runs in orders of magnitude less time.

These results validate our framework for location classification. First, the accuracy of the boosted classifier shows a high level of performance. Additionally, the boosted classifiers offload the bulk of the computational complexity to an offline training process. Inspection of Table 1 reveals the poor scalability of kNN as the size of the data set and the number of locations increases.

While the experiments only show the performance of LoCo in a single deployment, it is important to point out that there was nothing extraordinary about our experimental configuration. The office space we used is built out of common building materials used in most office buildings in the United States. We also implemented a prior approach [2] and it performed similarly in our space compared to prior published experiments. We plan to conduct further evaluations of LoCo to understand its performance in spaces other than office buildings, for example in homes, apartments, stores, and industrial environments.

DISCUSSION

While our work is not the first to investigate leveraging radio signals to determine device location, we do believe we are the first to demonstrate the efficacy of a boosted classifier that achieves the level of accuracy and performance demonstrated by our experiments. We believe this works provides a significant step forward in allowing indoor location to be performed efficiently, even on resource constrained mobile devices.

We have leveraged LoCo to enhance an existing office presence tool [20], and we are now planning a study to understand how improved location information translates to meaningful changes in communication behavior. More broadly, we believe efficient and accurate room level location can open many new application opportunities. From the mundane, (e.g. an application that could detected the presence of multiple people in a meeting room and automatically reserve the room), to those that could be used for safety (e.g. an application that can inform building security personnel if employees are present in the building after hours) and in emergencies (e.g. a combined mobile/cloud application that could provide the last known location of building occupants at times of fire or earthquake). Since LoCo only requires the user to install an application on his or her smartphone, their burden and overhead is low, in contrast to approaches such as Sociometric Badges [13]. A user simply must carry a device she likely already keeps with her during day-to-day activities. Thus, we believe LoCo can help drive improvements in office communication tools and technologies.

One limitation of LoCo is that a set of training data must be collected per location. This process is simple but involves taking a device into a room and having it perform several scans that it submits to the classification engine with an attached ground truth label. As environments change (e.g. a Wi-Fi AP is moved or replaced) a new survey is likely necessary. One approach to address this, as in [2], is to crowdsource users in providing ground truth. Another approach, which we are actively investigating, is to use a small robot to perform the data collection.

Our framework is not designed to determine the precise location within a room a device is located. In large rooms, this could limit the use of LoCo to classify activities of individuals or groups, or to actuate changes or events within the physical space. Achieving these goals will involve fusing LoCo's location information with other sensor sources. Finally, it is important to point out that, as a consequence of building a framework to run on common devices, we have made dependencies on platforms and APIs that could change with any new OS release.

CONCLUSION AND FUTURE WORK

Location is playing an ever increasing role in mobile computing. Many of the most popular applications and services used today build on knowing the user's current location. For many of these applications the accuracy of current technologies is adequate. However, we believe the next generation of mobile applications and services will need a next generation location technology—one that provides accurate, reliable location information for indoor environments. In this paper, we presented the LoCo framework. We discussed how it has been designed and developed to provide application and service developers easy access to indoor location information. We demonstrated an evaluation of the system and explained its performance and capabilities compared to existing state-of-the-art techniques. Further, we performed this evaluation using hardware and software that is commercially available today.

Our work to date serves as inspiration for future activities. First, we intend to deploy LoCo in different environments to understand its performance characteristics more broadly. This effort will also allow us to better investigate how well LoCo scales to larger configurations, for instance within a multistory high-rise in a densely urban environment. Second, we intend to leverage LoCo to prototype new and novel applications that use LoCo's accurate location information to provide interesting and compelling experiences for users in office and retail settings. Finally we will work to make LoCo available to others in the research community.

REFERENCES

1. Andoni, A., and Indyk, P. Efficient algorithms for substring near neighbor problem. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, SODA '06, ACM (New York, NY, USA, 2006), 1203–1212.
2. Bolliger, P. Redpin - adaptive, zero-configuration indoor localization through user collaboration. In *Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments*, MELT '08, ACM (New York, NY, USA, 2008), 55–60.
3. Chintalapudi, K., Iyer, A. P., and Padmanabhan, V. Indoor localization without the pain. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, MobiCom '10, ACM (New York, NY, USA, 2010), 173–184.
4. Cong, L., and Zhuang, W. Hybrid tdoa/aoa mobile user location for wideband cdma cellular systems. *Wireless Communications, IEEE Transactions on 1*, 3 (Jul 2002), 439–447.
5. Dempster, A., Li, B., and Quader, I. Errors in deterministic wireless fingerprinting systems for localization. In *ISWPC 2008* (2008), 111–115.
6. Hastie, T., Tibshirani, R., and Friedman, J. *The Elements of Statistical Learning*. Springer-Verlag, 2001.
7. Li, P. Robust logitboost and adaptive base class (abc) logitboost. In *UAI 2010, Proceedings of the*

Twenty-Sixth Conference on Uncertainty in Artificial Intelligence (2010), 302–311.

8. Lin, H., Zhang, Y., Griss, M., and Landa, I. Enhanced indoor locationing in a congested wi-fi environment. Tech. Rep. MRC-TR-2009-04, Carnegie Mellon Silicon Valley, 2009.

9. Madigan, D., Elnahrawy, E., Martin, R. P., Ju, W.-H., Krishnan, P., and Krishnakumar, A. S. Bayesian indoor positioning systems. In *Proceedings of the 24th joint conference of the IEEE Computer and Communication Societies (INFOCOM 2005)* (2005).

10. Martin, E., Vinyals, O., Friedland, G., and Bajcsy, R. Precise indoor localization using smart phones. In *Proceedings of the international conference on Multimedia*, MM '10, ACM (New York, NY, USA, 2010), 787–790.

11. Mukherjee, I., and Schapire, R. A theory of multiclass boosting. In *Advances in Neural Information Processing Systems 23*, J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, Eds. 2010, 1714–1722.

12. Niculescu, D., and Nath, B. Ad hoc positioning system (aps) using aoa. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 3 (March 2003), 1734–1743 vol.3.

13. Olguín, D., Waber, B. N., Kim, T., Mohan, A., Ara, K., and Pentland, A. Sensible organizations: technology and methodology for automatically measuring organizational behavior. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: 39*, 1 (Feb. 2009), 43–55.

14. Ouyang, R. W., Wong, A. K.-S., Lea, C.-T., and Chiang, M. Indoor location estimation with reduced calibration exploiting unlabeled data via hybrid generative/discriminative learning. *IEEE Transactions on Mobile Computing 99* (2011).

15. Pan, J. J., Pan, S. J., Yin, J., Ni, L. M., and Yang, Q. Tracking mobile users in wireless networks via semi-supervised colocalization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2012), 587–600.

16. Roxin, A., Gaber, J., Wack, M., and Nait-Sidi-Moh, A. Survey of wireless geolocation techniques. In *IEEE Globecom* (2007).

17. Sen, S., Lee, J., Kim, K.-H., and Congdon, P. Avoiding multipath to revive inbuilding wifi localization. In *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '13, ACM (New York, NY, USA, 2013), 249–262.

18. Silva, J. Wips: wifi indoor positioning system. `http://mobile-accessibility.idrc.ocad.ca/projects/wips`.

19. Want, R., Hopper, A., Falcão, V., and Gibbons, J. The active badge location system. *ACM Transactions on Information Systems (TOIS) 10*, 1 (1992), 91–102.

20. Wiese, J., Biehl, J. T., Turner, T., van Melle, W., and Girgensohn, A. Beyond 'yesterday's tomorrow': towards the design of awareness technologies for the contemporary worker. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '11, ACM (New York, NY, USA, 2011), 455–464.

21. Xiong, J., and Jamieson, K. Towards fine-grained radio-based indoor location. In *Proceedings of the Twelfth Workshop on Mobile Computing Systems & Applications*, HotMobile '12, ACM (New York, NY, USA, 2012), 13:1–13:6.

22. Youssef, M., and Agrawala, A. The horus wlan location determination system. In *Proceedings of the 3rd international conference on Mobile systems, applications, and services(MobiSys '05)* (2005), 205–218.

23. Zhu, J., Zou, H., Rosset, S., and Hastie, T. Multi-class adaboost. *Statistics and Its Interface 2* (2009), 349–360.