Indoor Positioning System Using Wifi Fingerprint

Dan Li, Le Wang, Shiqi Wu

Stanford University

Abstract

Indoor Positioning System aims at locating objects inside buildings wirelessly, and have huge benefit for indoor location-aware mobile application. To explore this immature system design, we choose UJIndoorLoc database as our data set, use PCA for feature selection, and build prediction models based on decision tree, gradient boosting, kNN and SVM, respectively. Our experiment results indicate that combination of kNN and Gradient Boosting provides high prediction accuracy for Indoor Positioning. kNN shows good performance for large volume of data set with sample size greater 1000, and Gradient Boosting has small cross validation error for small data volume and is robust to missing data.

I. Introduction

Indoor Positioning System (IPS) aims at wirelessly locating objects or people inside buildings based on magnetic sensor network, or other source of data. The major consumer benefit of indoor positioning is the expansion of location-aware mobile computing indoors, such as augmented reality, store navigation, etc. As mobile devices become ubiquitous, contextual awareness for applications has become a priority for developers. Most applications currently rely on GPS, however, and function poorly indoors. Up till now, there is no de facto standards for IPS system design [1]. Due to the proliferation of both wireless local area networks (WLANs) and mobile devices, WiFi-based IPS has become a practical and valid approach for IPS [2][3] that does not require extra facility cost. However, Wifi-based position system as (WPS) accuracy depends on the number of positions that have been entered into the database. The possible signal fluctuations that may occur can increase errors and inaccuracies in the path of the user.

Mike Y. Chen, Timothy Sohn, et al have explored the influence of data size and prediction algorithm on location predicting accuracy, and has proposed that with centroid algorithm, a limited size of data set can provide provide a highly reliable result[4]. Sunkywu Woo, Seongsu Jeong, et al have chosen fingerprint methods for Wifi positioning system[5]. By adapting comparison algorithm and using RFID device as receiver, they achieved locating accuracy of within 5m. William Ching, Rue Jing Teh, et al have conducted similar result using T-mobile G-1 phone, and suggested that the predicting accuracy would be improved with the user contribution, in other words, by constantly increasing the data size[6]. Joaquin Torres-Sospedra, Raul Mntoliu, et al, have proposed UJIndoorLoc database for a common public database for WLAN fingerprint-based indoor localization[7].

Inspired by previous work, we plan to use fingerprint of Web Access Points(WAPs) as features to predict the position of mobile device holder. The fingerprint of WAP we use is the Received Signal Strength Indicator(RSSI). In this project, we locate the floor level of a mobile device using Wifi fingerprint via machine learning methods, and explore the data size, feature dimension, model combination and parameter selection to maintain, if not improve, prediction accuracy, for different test environment.

II. Methods

I. Data preprocessing

We use UJIndoorLoc database [7] for this project. It consists of 520 RSSI fingerprints detected using 16 different phone models by 18 users from 4 to 5 different floors of 3 buildings, as shown in figure 1. For each building, this dataset gives us thousands of RSSI samples generated at various locations inside the building.

Thus, with given roadmap of fingerprints as training set, we could generate a model using machine learning techniques, with which we would be able to predict the floor number of unknown mobile device holder in a certain building.

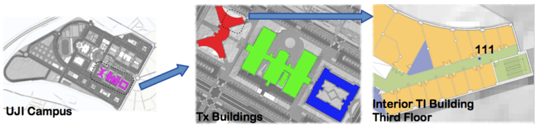


Figure 1: Source [7] Left: map of UJI Riu Sec campus and Tx buildings. Middle: red indicates ESTCE - Tx building. Right: example of a reference point.

I.1 dimensionality reduction

However, the high dimensional feature space with redundant features would hurt computational efficiency. Therefore, we used Principal Component Analysis(PCA) to extract principal features. The energy levels of the first 200 principal components are shown in figure 2 . We found that the top three principal components contain most of the energy, and for the components beyond the first 200, each of their energy levels is less than one. As we can see from figure 3, there is a tradeoff between prediction accuracy and number of features required, which indicates the computation complexity, for the learning task. Depending on the characteristics of each learning algorithm, we choose 5, 50 and 200 top features for the suitable algorithms to compare and explore for the best prediction accuracy.

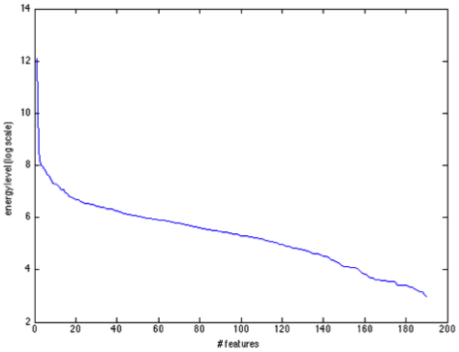
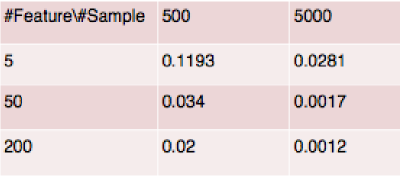


Figure 2: Energy levels of components in PCA



#Feature\#Sample	500	5000
5	0.1193	0.0281
50	0.034	0.0017
200	0.02	0.0012

Figure 3: feature size vs. error for kNN

I.2 sample size reduction

The original dataset has more than 5000 samples for each building, which would require a lot of efforts for data collection before building a model for another building in real life scenario. Therefore, we randomly select a subset of the original sample space to explore the effects of data size to the accuracy of the model. Data sizes we explored here in various experiments consists of 100, 200, 500, 1000, 2000 and 5000 samples.

II. Model Selection

We implement four classification methods in machine learning, including k-Nearest Neighbor (kNN), decision tree, Gradient Boosting and Support Vector Machine (SVM). All the four methods are applied with 10-fold cross validation to avoid overfitting.

II.1 kNN

kNN seems to be a good candidate for classification of this sort. It is due to the fact that kNN tries to make the classification by calculating the distance between features, while the intensity of various RSSI signals depends on the physical distance between Wifi source and mobile phones. In this case, closeness in feature space is a good indication of closeness in physical space.

II.2 SVM

We apply multi class SVM to determine the decision boundary between classes. However, the results are worse than decision tree or KNN. A potential reason of SVM failure is because of irrelevant variables with high dimensional dataset. High prediction accuracy can hardly be achieved even we reduce the feature dimension from 520 to 200. To solve this problem, we further explore the effort of dimension reduction using PCA.

II.3 Decision Tree

Decision tree is then implemented, which has the advantages of fast training process, easy interpretation and resistance to many irrelevant variables. But decision tree has the disadvantage of inaccuracy compared with kNN, even cross validation is used. In decision tree, two criteria are applied to prune the tree. One is cross validation and the other is one stand error. Surrogate splits are used in construction of the optimal tree as a missing value strategy, which encourages variables within highly correlated sets.

II.4 Gradient Boosting

To maintain most advantages of trees while dramatically improve accuracy, bagging algorithm could be a good choice. Here gradient boosting is chosen to improve the accuracy. Also, to handle missing data, we use surrogates to distribute instances. Best number of iterations (number of trees) are identified using cross validation and the depth for each simple tree is set to be four. This parameter could be further studied get a better accuracy.

III. Results & Discussion

We tested our models using data from three buildings separately. As mentioned before, at each building, we performed a PCA on the feature space to reduce its dimension, and randomly selected samples to perform a ten-fold cross validation on the four classification models. Both of the number of reduce dimension and the number of samples in the sample space are tunable for each model to achieve the least error. Each set of parameters was performed for five rounds and the error is averaged among those rounds to reduce noise.

I. kNN

We first explore the number of nearest neighbors we need to consider for classifying a testing set. As shown in figure 4, the error of classification increases with increasing of k. Then we look into the influence from the number of samples and the number of top principal features in kNN (k=1) classification. The error reduces dramatically with increase of sample size, but not much improvement is seen from adding more principal features.

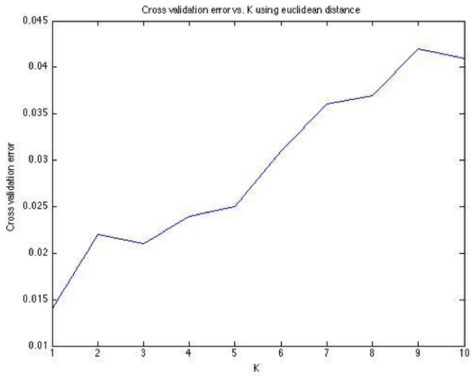


Figure 4: errors vs K for kNN methods using Euclidean distance

II. SVM

SVM does not perform well for this problem. Here we explore both linear kernel and third order polynomial kernel, and decided to use polynomial kernel for better accuracy. From figure 5, we can see the descending trends of SVM error with increasing of sample size, and data with 150 principal features perform better than data with 50 principal features.

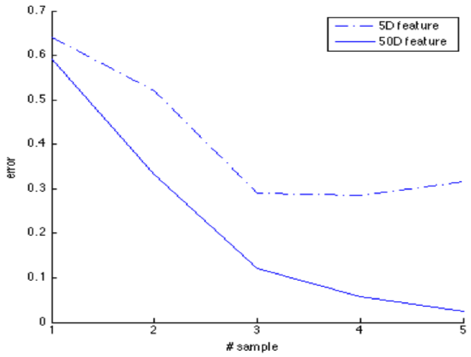


Figure 5: SVM error vs. sample size. Dashed line denotes result using 50 features; solid line denotes result using 500 features

We did not test the experiment with 5000 samples because SVM tends to perform better for small dataset given enough margin data. Therefore, experiment in large feature and sample space will increase the difficulty of convergence for the optimization problem.

III. Decision Tree

We implement the decision tree in R. First of all a full tree is grown with complexity parameter to be 0. The we utilize two criteria to prune the tree, the first criterion to find an optimal complexity parameter is to choose the cp with minimum cross validation error. One standard error rule is used as the alternative criterion. Note from figure 8, the total number of splits to minimize cv error is 28, and the total splits according to the one standard error rule is twenty. Cross validation criterion has the advantage of smallest expected prediction error resulting from the smallest bias. While it shows the disadvantage of more complex tree structure and relatively higher instability, or higher variance. For one standard error criterion, we obtain a simpler tree structure with relatively low variance. And it is proved to be able to screen out noise in the data. The disadvantage of one standard error is larger bias since it has less split than cv error minimized tree.

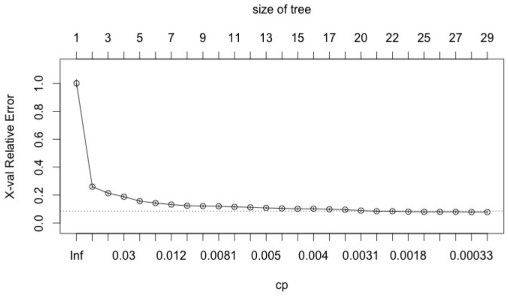


Figure 6: prune the tree with cross validation

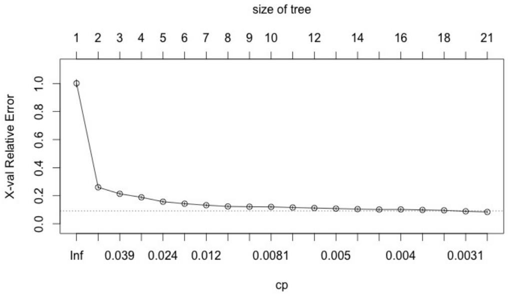
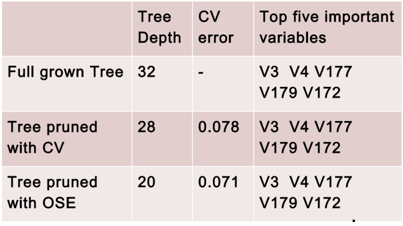


Figure 7: prune the tree with one standard error



	Tree Depth	CV error	Top five important variables
Full grown Tree	32	-	V3 V4 V177 V179 V172
Tree pruned with CV	28	0.078	V3 V4 V177 V179 V172
Tree pruned with OSE	20	0.071	V3 V4 V177 V179 V172

Figure 8: Statistics of Decision Tree

Surrogate splits are used when the predictor used to determine the split is missing. When considering a predictor for a split, we use only the observations for which the predictor is not missing. Then we form a list of surrogate predictors and split points. When sending observations down the tree either in the training phase or during prediction, we use the surrogate splits in order, if the primary splitting predictor is missing.

IV. Gradient Boosting

A gradient boosting (GB) model is fitted based on the training data. With cross validation, optimal number of iteration is determined to be 189. Figure 9 shows the misclassification error risk versus number of iterations. The misclassification error is calculated to be 0.05 for test set. Figure 10 shows the relative importance of each variables and figure 11 shows the partial dependency of the most important variable V3. From this figure we could see that the floor number is strongly dependent on variable V3, which indicates that V3 comes from a strong signal source. Figure 12 shows the overall error rate for each floor. It is found that GB has very low error rate, which is 0 for floor 1, 0.08 for floor 2, 0.06 for floor 3 and 0 for floor 4.

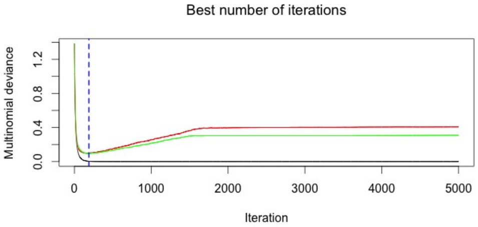


Figure 9: misclassification error risk versus number of iterations for gradient boosting methods

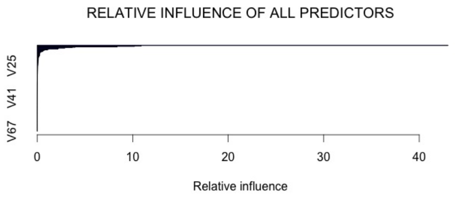


Figure 10: relative influence of all predictors

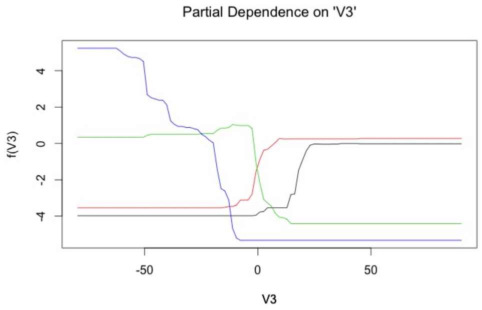


Figure 11: partial dependence on V3

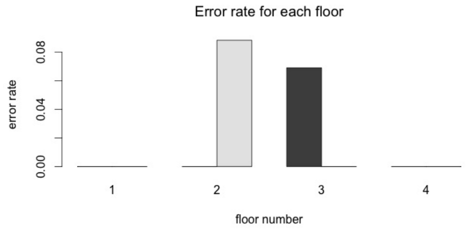


Figure 12: error rate for each floor

We also investigated the performance of GB on small data set. Here a small data volume of only 100 randomly selected samples are used to fit the model, and we got a cross validation error of 0.03.

IV. Conclusion

As demonstrated in this paper, the simplest kNN model gives good accuracy, given a relative small feature space and reasonable large data space. However, SVM performs poorly on this classification algorithm. Although one decision tree does not give satisfying result, bagging of multiple trees through gradient boosting could highly increase the prediction accuracy. To acquire high accuracy, while maintaining the capacity for predicting both small and large data set, we suggest the combination of kNN and gradient boosting for the indoor positioning system.

V. Future Work

Since Gradient boosting is robust of missing value, the effect of missing value for current kNN model is to be investigated. Also, beyond the current models, tracking of moving user, type of phones and minimum number of Wifi sources required for accurate positioning will be explored in the future work.

References

[1] Zhou, Junyi Shi, Jing. RFID localization algorithms and applications: a review. *Journal of Intelligent Manufacturing*, 20:, 695–707, 2009.

[2] Ferris, Brian Fox, Dieter Lawrence, Neil D. WiFi-SLAM Using Gaussian Process Latent Variable Models. *IJCAI*, 7:, 2480–2485, 2007.

[3] Marques, Nelson Meneses, Filipe Moreira, Adriano. Combining similarity functions and majority rules for multi-building, multi-floor, WiFi positioning. *IEEE Xplore*, 2012.

[4] Chen, Mike Y Sohn, Timothy Chmelev, Dmitri Haehnel, Dirk Hightower, Jeffrey Hughes, Jeff LaMarca, Anthony Potter, Fred Smith, Ian Varshavsky, Alex/ Practical metropolitan-scale positioning for gsm phones. *UbiComp 2006: Ubiquitous Computing*, 225–242, 2006.

[5] Woo, Sunkyu Jeong, Seongsu Mok, Esmond Xia, Linyuan Choi, Changsu Pyeon, Muwook Heo, Joon. Application of WiFi-based indoor positioning system for labor tracking at construction sites: A case study in Guangzhou MTR. *Automation in Construction*, 20:, 3–13, 2011.

[6] Ching, William Teh, Rue Jing Li, Binghao Rizos, Chris. Uniwide WiFi based positioning system. *Technology and Society (ISTAS), 2010 IEEE International Symposium on*, 180–189, 2010.

[7] Torres-Sospedra, Joaquın Montoliu, Raúl Martınez-Usó, Adolfo Avariento, Joan P Arnau, Tomás J Benedito-Bordonau, Mauri Huerta, Joaquın. UJIIndoorLoc: A New Multi-building and Multi-floor Database for WLAN Fingerprint-based Indoor Localization Problems.