Estimating Location Using Wi-Fi

Qiang Yang, Sinno Jialin Pan, and Vincent Wenchen Zheng,
Hong Kong University of Science and Technology

Recent advances in pervasive computing and mobile technology have enabled accurate location and activity tracking of users wearing wireless devices indoors, where GPS isn't available. A practical way to do this is by leveraging the Wi-Fi signals that a mobile client receives from various access points. For example, many indoor location estimation techniques use received radio signal strength (RSS) values and radio signal propagation models to track users. Machine learning-based methods have proven among the most accurate.

However, Wi-Fi data is noisy owing to the indoor environment's multipath and shadow fading effects. The data distribution changes constantly as people move and as temperature and humidity change.[1–3] Moreover, it can be expensive to collect and label RSS training data in a large building because it requires a human to walk with a mobile device, collecting RSS values and recording ground locations.[4,5]

Despite intense research in indoor location estimation and activity recognition, the field lacks benchmark data that researchers and practitioners can use to compare their solutions. The 2007 Data Mining Contest (www.ist.unomaha.edu/icdm2007/contest), sponsored by the IEEE International Conference on Data Mining, provided the first realistic public benchmark data for indoor location estimation using RSS that a client device received from Wi-Fi access points. We collected the data sets in a 145.5 m × 37.5 m academic building at the Hong Kong University of Science and Technology. We divided the location into a grid of 247 units, each about 1.5 m × 1.5 m. We focused on discrete classification as well as regression versions of the tasks (we've posted these and the benchmark data set at www.cs.ust.hk/~qyang/ICDMDMC07).

This year's contest focused on two tasks: indoor location estimation and transferring knowledge (learned from training data) for indoor location estimation.

Task 1

In this semisupervised-learning problem, we asked participants to predict a client's location on the basis of RSS values received from Wi-Fi access points. We provided a set of data (RSS values, location label) as training data, with discrete *location labels*, which correspond to different grids. To make the problem more interesting, we also provided some unlabeled data (with only the RSS values) and some partially labeled user traces.

In this task, the training data had 3,196 RSS vectors in both nontrace and trace data; only 787 were labeled. We obtained the test data by collecting the RSS values as we walked around a building that had 43 user traces and a total of 2,180 vectors of RSS values. We asked participants to predict the location label for each RSS vector in the test data.

Task 2

The second task resembled task 1, but we collected the training data at a different time from the test data. For this semisupervised transfer-learning problem, the test data were discrete (that is, they weren't sequential). To aid prediction when the training and test data came from different distributions, we provided some labeled test data that participants could use as benchmarks. For this task, we asked participants to adapt or transfer the learned knowledge from the training data. This was difficult because of the data-distribution changes.

In this task, the training data had 4,361 RSS examples; 715 were labeled. The test data had 3,128 vectors.

Evaluation criterion

We asked participants to submit their predictions for each task for all test data separately. We conducted the evaluation on a test data set and ranked the final results in descending order of their precision values for each task:

precision = (*number of correct predictions*)/(*total number of test data*)

1541-1672/08/$25.00 © 2008 IEEE
Published by the IEEE Computer Society

Table 1. Evaluation of contest submissions.

Task	Max	Min	Median	Average	Std-dev
1	0.8227	0.040	0.6135	0.5877	0.1707
2	0.3223	0.1527	0.2922	0.2677	0.0532

Worldwide, 115 teams registered for our contest. In the end, 21 teams submitted 32 results—15 for task 1 and 17 for task 2. Among the solutions, participants most frequently used k-nearest neighbor methods, decision trees,[6] and semisupervised or transductive learning models.[7] A team from IBM Research, Tokyo, won task 1, obtaining 0.8226 precision. The two runners-up were teams from the University of Tokyo and from Tsinghua University.

A graduate student from HeBei University won task 2. The runners-up were teams from the Chinese Academy of Sciences and from IBM Research, Tokyo. Table 1 evaluates the contest's submissions. Max, min, median, average, and std-dev represent the highest, lowest, median, average, and standard deviation precision values among the submissions.

System science and data mining makes localization through Wi-Fi and sensors feasible. This data mining contest brought many innovative solutions to this challenging and important problem. At the same time, it brought new research issues for the future, including transfer learning and semi-supervised learning.

Acknowledgments

We thank the Hong Kong Research Grants Council (grant 621307) for their support. We thank contest co-chairs Gang Kou from Thomson Corporation and Chris Ding from the University of Texas, Arlington for their support. We also thank the ICDM 2007 conference organizers Yong Shi, Christopher W. Clifton, Naren Ramakrishnan, Osmar Zaiane, and Xindong Wu for their support.

References

1. A. Goldsmith, *Wireless Communications*, Cambridge Univ. Press, 2005.
2. J. Yin, Q. Yang, and L. Ni, "Adaptive Temporal Radio Maps for Indoor Location Estimation," *Proc. 3rd Ann. IEEE Int'l Conf. Pervasive Computing and Communications* (PerCom 05), IEEE CS Press, 2005, pp. 85–94.
3. S.J. Pan et al., "Adaptive Localization in a Dynamic Wi-Fi Environment through Multi-View Learning," *Proc. 22nd AAAI Conf. Artificial Intelligence* (AAAI 07), AAAI Press, 2007, pp. 1108–1113.
4. B. Ferris, D. Fox, and N. Lawrence, "WiFi-SLAM Using Gaussian Process Latent Variable Models," *Proc. 20th Int'l Joint Conf. Artificial Intelligence* (IJCAI 07), 2007, pp. 2480–2485; www.ijcai.org/papers07/Papers/IJCAI07-399.pdf.
5. J.J. Pan et al., "A Manifold Regularization Approach to Calibration Reduction for Sensor-Network Based Tracking," *Proc. 21st Nat'l Conf. Artificial Intelligence* (AAAI 06), AAAI Press, 2006, pp. 988–993.
6. T.M. Mitchell, *Machine Learning,* McGraw-Hill, 1997.
7. O. Chapelle, B. Schölkopf, and A. Zien, *Semi-Supervised Learning*, MIT Press, 2006.

Winner: Task 1

A Semisupervised Approach Using Spatiotemporal Information for Indoor Location Estimation

Hisashi Kashima, Shoko Suzuki, Shohei Hido, Yuta Tsuboi, Toshihiro Takahashi, Tsuyoshi Idé, Rikiya Takahashi, and Akira Tajima, *Tokyo Research Laboratory, IBM Research*

We formulated task 1 as a transductive multiclass classification problem. The whole data set consists of $N = 5{,}333$ instances, where the $\ell = 505$ instances are labeled data and the rest are unlabeled data. The ith data instance is given as $(\mathbf{x}^{(i)}, y^{(i)})$, where $\mathbf{x}^{(i)} \in \mathbb{R}^{101}$ is the vector of the RSS values from the Wi-Fi access points and $y^{(i)} \in \{1, 2, \ldots, 247\}$ is the RSS vector's location label. The unobserved RSS values are set to –100, which is their minimum value.

In addition to the RSS values, we gave some instances trace IDs and observation times, which indicate the trace each instance belongs to and the time it was observed. We denote by $TID^{(i)}$ and $t^{(i)}$ the trace ID and the observation time of the ith instance. For simplicity, we treated the observation times as just the order of observation (integer values), although the original data gave them as real values. This task aimed to predict the unlabeled data's location labels, $y^{(\ell+1)}, \ldots, y^{(N)}$. This is a transduction problem where you can use test inputs in the training.

We employed a multiclass version of the *label propagation* method,[1] a semisupervised-learning approach.[2] Let $f^{(i)}(c) \in [0, 1]$ indicate the probability that the location label of the ith instance is c. For the labeled data ($i \le \ell$), $f^{(i)}(c) = \delta(c = y^{(i)})$ must be satisfied, where $\delta(\cdot)$ is a function that returns 1 if its argument is true and 0 otherwise. The task was to predict $f^{(i)}(y^{(i)})$ for $i > \ell$ and $\forall c$, from which we obtained the prediction $\hat{c}^{(i)}$ for $i > \ell$ as

$$\hat{c}^{(i)} = \arg\max_c f^{(i)}(c)$$

In the label propagation framework, we minimized the sum of discrepancies of the label distributions among neighborhood instances, which we defined as

$$\sum_{(i,j)} w^{(i,j)} = \sum_c \left(f^{(i)}(c) - f^{(j)}(c)\right)^2$$

where $w^{(i,j)}$ is a constant called *affinity*, indicating the similarity between the ith and the jth instances. The values of $f^{(i)}$ are fixed for $i \le \ell$. It's easy to see that the previous optimization problem's solution satisfies

$$f^{(i)}(c)=\left(\sum_j w^{(i,j)} f^{(j)}(c)\right)\Big/\left(\sum_j w^{(i,j)}\right)$$

for $\forall i > \ell$ and $\forall c$. So, instead of solving the large optimization problem directly, we could iteratively use it to make local updates of the predictions until convergence. We defined the affinity $w^{(i,j)}$ as

$$w^{(i,j)} = \max\left\{w_{\mathbf{x}}^{(i,j)}, w_t^{(i,j)}\right\}$$

which is the maximum of either $w_{\mathbf{x}}^{(i,j)}$ defined by the RSS vectors or $w_t^{(i,j)}$ defined by the trace IDs and observation times. Our definition of the affinity implies that two instances are similar if their RSS vectors are similar or they have consecutive observation times. For the affinity between two RSS vectors $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$, we used a heat-kernel-like function

$$w_{\mathbf{x}}^{(i,j)} = \exp\left(-\left\|\mathbf{x}^{(i)}-\mathbf{x}^{(j)}\right\|_q \Big/ \sigma\right)$$

where σ is a scale parameter, and we set $\sigma = 0.5$ in our submission. Also, $\|\cdot\|_q$ is the q-norm, which we defined as

$$\|\mathbf{x}\|_q = \sum_d |x_d|^q$$

and we set $q = 0.5$ because a q-norm with $q < 1$ prioritizes whether each signal exists rather than the amount it changes. It's robust to drastic changes of each RSS value caused by reflection, interference, and shielding, and at the same time, it's sensitive to multiple RSS value changes. We define the affinity between two trace ID pairs and an observation time as

$$w_t^{(i,j)} = p \cdot \delta\left(TID^{(i)} = TID^{(j)}\right) \cdot \delta\left(\left|t^{(i)} - t^{(j)}\right| = 1\right)$$

where $p \in [0, 1]$ is a constant indicating the affinity of two consecutive observations, and we set $p = 1$ in the result submission.

References

1. X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions," *Proc. 20th Int'l Conf. Machine Learning*, AAAI Press, 2003, pp. 912–919.
2. X. Zhu, *Semi-Supervised Learning Literature Survey*, tech. report 1530, Computer Science Dept., Univ. of Wisconsin-Madison, 2006.

First Runner-Up: Task 1

Simple Algorithm for Location Estimation from Wi-Fi Signal Strength

Yuichi Katori, *University of Tokyo*

Suppose that a_i is an access point index and that d_i is the RSS value of the ith of N collected access points in the test data (traced data). $a_i^{l,k}$, $d_i^{l,k}$, and $N^{l,k}$ are correspondences in a training data (nontraced data) of the kth observation with location label l. Observations exist in each l. Step 1 of the algorithm measures the dissimilarity[1] of the RSS vector between the query point and 247 location labels and then specifies the location label that most resembles the query point. The distance between the query point and l should be

$$D_l = \frac{1}{M^l}\sum_k^{M^l}\left[\begin{array}{l}\alpha\left(N+N^{l,k}\right)+ \\ \sum_i^N \sum_j^{N^{l,k}}\left(\left|a_i - a_j^{l,k}\right| - 2\alpha\right)\delta\left(a_i, a^{l,k}\right)\end{array}\right] \quad (1)$$

where $\delta(x, y) = 1$ if $(x = y)$, 0 otherwise. The first candidate location label L is defined as $L = argmin_l D_l$.

In step 2, I defined a confidence level above estimation on the basis of the ratio of the first and second candidates of the dissimilarity $C = D^{(2)}/D^{(1)}$, where $D^{(1)}$ and $D^{(2)}$ are the first and second candidates' dissimilarities. I modified location label candidates in step 1. If the confidence level of query point C was less than some threshold C_1 and good confidence points existed with confidence levels higher than C_2 in m-nearest observations around the query point, I replaced L with the location label of the neighboring good confidence point. I choose the parameters $\alpha = 6$, $C_1 = 20$, $C_2 = 100$, and $m = 2$ so the algorithm would work well with sample test data.

The algorithm correctly predicts location labels with a ratio of 74 percent for sample test data and 69.5 percent for contest test data.

Reference

1. T. Hastie, R. Tibshirani, and J.H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, 2001.

Second Runner-Up: Task 1

Location Estimation Using *k*-Nearest Neighbor in a Line

Yang Qu and Chun Li, *Tsinghua University*

To solve task 1, we used the modified k-nearest neighbor with the shortest path. k-nearest neighbor can classify a large amount of unlabeled data simply and quickly. However, at the position in a cell, the access point vector rotates in different directions, so no actual center access point vector exists for each cell. To handle this problem, instead of considering all neighbors around a point, we considered only neighbors along the shortest path between the current point and the successive point in the trace. In other words, we selected only points in the most likely direction.

To calculate k-nearest neighbor, we defined a 1L-norm distance between points:

$$d(i,j)=\sum_{k=1}^{|AP|}\left|AP_i^k-AP_j^k\right|$$

We then used the classic Floyd algorithm to calculate the shortest path between every point. You could use a more efficient random algorithm to speed up the calculation for the online estimation problem.

Then, we propagated the labeled information to unlabeled points: Let M_i^n be the possibility mass of point *i* labeled cell *n*. For each unlabeled point, we calculated the possibility vector $\{M_i^n\}$ weighted by the distance between two points:

$$M_i^n=\sum\nolimits_{k\in all\ labeled}\frac{M_k^n}{d(i,k)}$$

For three successive points (N_{i-1}, N_i, N_{i+1}) in a trace, we used a shortest path *P* from N_{i-1} to N_{i+1} through N_i to predict the label of N_i. Because we didn't know the density distribution of points, we used the number of hops h_j^i from point *j* in the shortest path *P* to N_i as the distance instead of 1L-norm:

$$M_i^n=\sum\nolimits_{k\in P}\frac{M_k^n}{h(i,k)}$$

The final predicted label for N_i had the highest possibility. Using hops lets us adapt to various point densities. In a very dense area of points, only very near points have a strong effect on the result, while in a sparse area, we would consider farther points. This provides more discriminative power when the point is located on cell edges.

Winner: Task 2

A Minkowski Distance and Nearest-Unlike-Neighbor Distance Method

Xi-Zhao Wang, Feng Guo, and Xianghui Gao, *HeBei University*

Task 2 asked us to predict the location of each collection of RSS values in an indoor environment, received from Wi-Fi access points. On the basis of the problem's particular features, we designed an algorithm using the Minkowski distance and NUN (nearest unlike neighbor) distance.

Key points

First, we gave an offset to adjust the class center while classifying the test data, which we found using

$$DIF=avg\left\{x_k^i\right\}-avg\left\{x_k^j\right\}\left(x_k^i,x_k^j\neq -100\right)$$

where $x^i \in$ *Training Data*, $x^j \in$ *LandMark Data*, and $k = 1 \ldots 100$.

Next, we added the landmark data to the training data to increase the number of labeled instances and to improve the test data's prediction accuracy.

Then, we assigned weights for different class centers C_i while calculating the Minkowski distances. These weights are relevant to the index of the RSS in C_i.

Finally, we modified some of the classification results if

$$\left\|x-x^k\right\|\leq\frac{1}{n}NUN\ Distance\left(x^k\right)$$

The algorithm

Our algorithm consists of seven steps.

- Construct an initial training set containing the training data's labeled data and the landmark data.
- Compute a class center C_i for each location from the initial training set.
- Assign the label to the unlabeled sample in the training data set. If the

  $$\left\|x-C_i\right\|=\min\left\|x-C_k\right\|,\ k=1,\ \ldots,\ 247$$

  the label of the input vector *x* is *i*.
- Calculate a class center C_i for each location from all data in the training set, which is labeled in the previous step. In this step, we can obtain the new class center C_i.
- Adjust the class center and assign the weights. First, let $C_i = C_i + DIF$, then assign the Minkowski distance's weights for each class center.
- Calculate the Minkowski distance between the input vector and each class center C_i. The label for the input vector *x* equals the nearest center.
- Check the classification results using the NUN distance and modify some inconsistent results.

First Runner-Up: Task 2

Learning Transfer by Locally Linear Preserving

Zhuo Sun, Juan Qi, Junfa Liu, and Yiqiang Chen, *Chinese Academy of Sciences*

In task 2, we had to estimate locations at time *B* given training data at time *A* and some benchmark data. Our solution has three phases: training, adaptation, and localization.

Training

For time *A*, we used LapRLS (Laplacian Regularized Least Squares Regression), a semisupervised-learning method for classification. This let us label each unlabeled data and update the average RSS for each location.

Adaptation

We transferred knowledge from time *A* to time *B*. Although their

Qiang Yang is a professor in the Department of Computer Science and Engineering at Hong Kong University of Science and Technology. Contact him at qyang@cse.ust.hk; www.cse.ust.hk/~qyang.

Sinno Jialin Pan is a PhD candidate in the Computer Science and Engineering Department at the Hong Kong University of Science and Technology. Contact him at sinnopan@cse.ust.hk; www.cse.ust.hk/~sinnopan.

Vincent Wenchen Zheng is a PhD student in the Computer Science Department at the Hong Kong University of Science and Technology. Contact him at vincentz@cse.ust.hk; www.cse.ust.hk/~vincentz.

Hisashi Kashima is a researcher at the Tokyo Research Laboratory, IBM Research. Contact him at hkashima@jp.ibm.com.

Shoko Suzuki is a researcher at the Tokyo Research Laboratory, IBM Research. Contact her at e30126@jp.ibm.com.

Shohei Hido is a researcher at the Tokyo Research Laboratory, IBM Research. Contact him at hido@jp.ibm.com.

Yuta Tsuboi is a researcher at the Tokyo Research Laboratory, IBM Research. Contact him at yutat@jp.ibm.com.

Toshihiro Takahashi is a researcher at the Tokyo Research Laboratory, IBM Research. Contact him at e30137@jp.ibm.com.

Tsuyoshi Idé is a researcher at the Tokyo Research Laboratory, IBM Research. Contact him at goodidea@jp.ibm.com.

Rikiya Takahashi is a researcher at the Tokyo Research Laboratory, IBM Research. Contact him at rikiya@jp.ibm.com.

Akira Tajima is the manager of the data analytics group at the Tokyo Research Laboratory, IBM Research. Contact him at tajima@jp.ibm.com.

Yuichi Katori is a researcher with the Aihara Complexity Modeling Project, Erato, Japan Science and Technology Agency. Contact him at katori@sat.t.u-tokyo.ac.jp

Yang Qu is a graduate student in the Computer Science and Technology Department at Tsinghua University. Contact him at dcatcher.qu@gmail.com.

Chun Li is a graduate student in the Computer Science and Technology Department at Tsinghua University. Contact him at li_chun@yahoo.cn.

Xi-Zhao Wang is the dean and professor of the Faculty of Mathematics and Computer Science at HeBei University. Contact him at xizhaowang@ieee.org.

Feng Guo is a postgraduate student at HeBei University. Contact him at guofeng314@163.com.

Xianghui Gao is a member of the Machine Learning Center at HeBei University. Contact him at gaoxianghui2006@yahoo.com.cn.

Zhuo Sun is a master's student at the Institute of Computing Technology at the Chinese Academy of Sciences. Contact him at sunzhuo@ict.ac.cn.

Juan Qi is a master's student at the Institute of Computing Technology at the Chinese Academy of Sciences. Contact her at qijuan@ict.ac.cn.

Junfa Liu is a PhD candidate at the Institute of Computing Technology at the Chinese Academy of Sciences. Contact him at liujunfa@ict.ac.cn.

Yiqiang Chen is an associate professor at the Institute of Computing Technology and vice director general of the ICT's Pervasive Computing Research Centre, secretary general of the Mobile Audio Video Industry Alliance, and director of the Mobile Multimedia Group, National Audio Video Industry Alliance. Contact him at yqchen@ict.ac.cn.

distributions differ, they share a common manifold structure. We proposed Locally Linear Preserving, an approach similar to Locally Linear Embedding. We represented each data point by its neighbors, and the representation remains constant through time. At time A, we reconstructed each data point from its neighbors. Suppose that n locations exist. We measured reconstruction error by the loss function

$$\varepsilon(W) = \sum_{i=1}^{n} \left| \vec{X}_i - \sum_j W_{ij} \vec{X}_j \right| \quad (1)$$

We found the optimal weights W_{ij} by solving a least-squares problem. We mapped each data point $\vec{X}_i$ at time A to $\vec{Y}_i$ at time B by minimizing the cost function

$$\phi(Y) = \sum_{i=1}^{n} \left| \vec{Y}_i - \sum_j W_{ij} \vec{Y}_j \right| \quad (2)$$

Here, weights W_{ij} stayed the same at time A, and we aimed to optimize $\vec{Y}_i$. Some $\vec{Y}_i$ is the given benchmark data. We used linear programming to find the solution. Using $\vec{Y}_i$ as the average RSS for time B, we applied LapRLS again to train a new classifier C.

Localization

We used C to estimate locations at time B.

Acknowledgments

We received grants 60473045 and 04213533 from the National Natural Science Foundation of China and support from the HeBei Top-100 Scientists Plan. We also thank Hanxue Hao, Sheng Xing, and Qian Li for their contribution.

Second Runner-Up: Task 2

A Dimensionality Reduction Approach to Indoor Location Estimation

Shoko Suzuki, Yuta Tsuboi, Hisashi Kashima, Shohei Hido, Toshihiro Takahashi, Tsuyoshi Idé, Rikiya Takahashi, and Akira Tajima, *Tokyo Research Laboratory, IBM Research*

We formulated task 2 as a transductive multiclass classification problem. The data takes the form $(\mathbf{x}^{(i)}, y^{(i)})$, where $\mathbf{x}^{(i)} \in \mathbb{R}^{101}$ represents a vector of the RSS values (we filled the missing values with −100), and $y^{(i)}$ is the location label. We used all the data except the unlabeled data in the source domain to avoid excessive influence from the source data distribution while using all the labeled data.

Our approach has two steps. First, we used a Laplacian eigenmap, a nonlinear dimension reduction technique.[1] Then we used a nearest-neighbor classifier to predict the labels for the target domain's unlabeled data.

The Laplacian eigenmap gave the coordinates for the data's intrinsic structure using the matrix $L = D - W$. W is a heat-kernel form using the 2-norm, with each element

$$w^{(i,j)} = \exp\left(-\left\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\right\|^2 \Big/ (c\sigma)^2\right)$$

where σ is the standard deviation and c is a tuning parameter. D is diagonal with

$$d^{(i,i)} = \sum_{\ell} w^{(i,\ell)}$$

We solved the generalized eigenvalue problem $L\mathbf{y} = \lambda D\mathbf{y}$ and got the $k + 1$ smallest eigenvalues $\lambda_0, \ldots, \lambda_k$; the set of eigenvectors $\mathbf{y}_0, \ldots, \mathbf{y}_k$. $\lambda_0 = 0$ is trivial. We regarded $\mathbf{y}_1, \ldots, \mathbf{y}_k$ as the intrinsic structure's k-dim coordinates. We rescaled them to get new coordinates

$$\mathbf{z}_i = \frac{1}{\lambda_i}\mathbf{y}_i$$

to reflect the original data distribution's scale.

In the second step we applied supervised classification algorithms by using all the labeled data as training data to obtain the predictions. We used a nearest-neighbor classifier with the 2-norm in the new feature space $\mathbf{z}_1, \ldots, \mathbf{z}_k$.

We set the parameter c to 25 and k to 20 after a tenfold cross-validation using the labeled data in the target domain. ■

Reference

1. M. Belkin and P. Niyogi, "Laplacian Eigenmaps for Dimensionality Reduction and Data Representation," *Neural Computation*, vol. 15, no. 16, 2003, pp. 1373–1396.