INFERENCE MACHINES FOR SUPERVISED BLUETOOTH LOCALIZATION

Tatsuya Ishihara[⋆‡] *Kris M. Kitani*[†] *Chieko Asakawa*[⋆†] *Michitaka Hirose*[‡]

[⋆]IBM Research [†]Carnegie Mellon University [‡]The University of Tokyo

ABSTRACT

State space models, such as Kalman filters or Particle filters, have been applied to improve the accuracy of radio-wave-based localization. However, these models can drift radically when assumptions of the models are violated, and they do not have a mechanism to fix errors. Therefore, we propose an approach to apply supervised learning to pedestrian localization, which is based on the *Inference Machines* framework. During training, we collect localization ground truths using computer vision while also collecting Bluetooth signals to train a state space model for localization, which can recover from model drift. During testing, our proposed approach uses only Bluetooth signals. Our experimental results show that our approach can improve the accuracy of Bluetooth-based localization with a small number of training examples. Moreover, our multi-modal supervision can also be used to estimate additional parameters, such as device rotation, from Bluetooth signals that do not have such information.

Index Terms— Inference Machines, State Space Model, Bluetooth Localization, Structure from Motion

1. INTRODUCTION

Localization approaches have been studied extensively [1]. It is becoming a deeply critical technology for applications such as aiding the visually impaired in public places [2, 3]. Highly accurate localization is important to enable them to navigate such places safely. Some research pursued accuracy by using special devices such as ultrasonic [4], ultrawideband (UWB) [5], RFID [6], and Zigbee [7] *etc*. In contrast, most commercial navigation systems are based on GPS, Wi-Fi, and Bluetooth low energy (BLE) devices, which can be used by smartphones, and we seldom see actual systems that have accuracy beyond a few meters [8]. More accurate localization is needed in navigation systems for the visually impaired.

The accuracy of various localization approaches can be improved by applying state space models, such as Kalman filters or Particle filters [9]. However, because these state space models are usually based on assumptions about the physical dynamics models (constant velocity, or acceleration), the models can drift radically when those assumptions are violated. Furthermore, these approaches are not explicitly designed to handle drift and have no mechanism to fix errors.

Therefore, we propose an approach to apply a supervised state space model [10] for pedestrian localization. Our proposed approach can estimate more accurate locations by learning from several user trajectories in the same environment. It is based on the *Inference Machines* framework [11], which estimates approximate inference for graphical models with theoretical guarantee. To the best of our knowledge, this is the first application of supervised state space models to pedestrian localization. Our approach can be applied widely because user trajectories are often available, especially in public spaces, and our approach can be used with generic localization methods.

Our approach is related to radio-wave-based localization, which can be calibrated by crowdsourcing [12, 13]. These approaches train accurate localization models by collecting temporal patterns of radio-wave signals. In contrast, our approach trains a state space model from temporal patterns of estimated locations. With our approach, localization methods or devices are not assumed, and it can be applied widely. In addition, the state space model of our approach can predict rotation even if the BLE RSSI signal does not have rotational information.

To train our state space model, we need the ground truth data of user trajectories. To collect training data, we used the computer-vision-based localization, which is called Structure from Motion(SfM) [14]. SfM creates a 3D model of the environment using images and is much more accurate than radio-wave-based localization in general [15]. We can easily collect training data of the state space model by walking around the navigation area several times. The collected trajectories are also used as fingerprinting data of the radio-wave signal, and we do not require a large workload for a site survey.

Our experimental results revealed our approach improved the accuracy of BLE-based localization compared with that of current approaches. In our experiment, we applied our approach to the basic BLE-based localization, but our approach can be applied to other state of the arts methods, such as the work of Faragher et al. [16], which uses Gaussian process regression for building a continuous map of radio-wave signals [17]. Our approach can also estimate rotation without decreasing localization accuracy, and it works well even in an environment with fewer BLE beacons. The results also show a small amount of data is enough to train the state space model.

2. SUPERVISED STATE SPACE MODEL

2.1. Predictive State Inference Machines (PSIM)

Our approach applies supervised learning of a state space model for pedestrian localization. We apply *Predictive State*

This work was supported by Shimizu Corporation. Kris M. Kitani was supported in part by JST CREST.

Inference Machines (PSIM) [10], which is based on the dynamical system representation called *Predictive State Representation* (PSR) [18].

The latent state of a dynamical system at time t is defined as $s_t \in \mathbf{R}^m$, and observation is defined as $x_t \in \mathbf{R}^n$. All past observation until t are denoted as $h_{1:t}$ or more compactly as h_t. PSIM estimates the latent states of current and future k steps at t, and we define this output as $f_t = [s_t, \ldots, s_{t+k-1}] \in \mathbf{R}^{km}$. The output f_t is estimated from the vector concatenating current and future k steps of observations $\phi(f_t) = [x_t, \ldots, x_{t+k-1}] \in \mathbf{R}^{kn}$, which is called a feature function. PSIM maintains a dynamical system state as a belief over this feature function. The state at t is defined as $E[\phi(f_t)|h_{t-1}] = \int_{f_t} \phi(f_t) P(f_t|h_{t-1}) df_t$, which represents the conditional expectation of feature function with respect to all past observations. The state is called the *predictive state*.

PSIM learns the following function F, which can estimate the current predictive state from that of a previous time and the current observation.

$$E\left[\phi\left(f_{t+1}\right)|h_t\right] = F\left(E\left[\phi(f_t)|h_{t-1}\right], x_t\right) \tag{1}$$

This representation allows us to obtain F by applying a generic regression model over all training data.

As shown in Eq. 1, the predictive state of a previous time will affect the future predictive state. This dependency violates the i.i.d. assumption and will degrade the performance of the regression model. PSIM utilizes the *Inference Machines* framework and learns F by dataset aggregation (DAgger) [19] to overcome this issue. DAgger repeats the process of adding predicted states to the training data by using the learned model and training the new model with the aggregated dataset. Aggregated data will contain errors that will likely appear during testing, and repeating the process of data aggregation and retraining will improve the model.

2.2. PSIM for Pedestrian Localization

We apply PSIM for pedestrian localization, which is a flexible framework for multi-modal fusion. We can obtain observation variable x_t by using any kind of localization algorithm. In our study, we focused on BLE-based localization, but an observation variable can be generalized to many other modalities. We use the basic k-Nearest-Neighbor(k-NN) algorithm for estimating the position from BLE RSSI signals [3], but other approaches can also be used.

Our approach can learn the state space model by using Algorithm 1, and we represent $m_t = E\left[\phi\left(f_t\right)|h_t\right]$ for simple notation. To train a state space model via PSIM, we need training trajectories τ_i that have ground truth positions and estimated positions from the BLE signals. We collect τ_i by using SfM, which is generally more accurate than radio-wave-based localization because it matches numerous visual features with a globally optimized 3D map; therefore, we use it as ground truth data. By recording videos with the BLE RSSI signals, we can associate the signals with the position estimated using SfM. The data are collected by walking the navigation path several times. To record the videos and BLE RSSI

Algorithm 1 Learning Pedestrian Localization Model

Input: trajectories $\tau_i, 1 \leq i \leq M$, number of hypothesis generation N

Output: best hypothesis $\hat{F}$ from $F_n, 1 \leq n \leq N$

1: Train k-NN for BLE localization from all trajectories τ_i
2: Apply Kalman filter for all trajectories τ_i and add estimated position and velocity to τ_i
3: Select validation data M_v and training data M_t randomly
4: Initialize training dataset $D_0 \Leftarrow \emptyset$ and train F_0 from M_t
5: Initialize the start point $\hat{m}_1 = \frac{1}{M}\sum_{i=1}^{M} \phi\left(f_1\right)$
6: **for** $n = 0$ to N **do**
7:     Use F_n for all trajectories M_t and predict states
8:     Create new dataset D'_n from predicted state and observation for F_n, and aggregate dataset $D_{n+1} = D_n + D'_n$
9:     Train new hypothesis F_{n+1} from D_{n+1}
10:     Calculate loss function d_{n+1} for M_v by F_n
11:     **if** $d_{n+1} > d_n$ **then**
12:         Discard aggregated dataset $D'_n \Leftarrow \emptyset, D_{n+1} = D_n$
13:     **end if**
14: **end for**
15: Select best hypothesis from $F_n, 1 \leq n \leq N$

signals at the same time, we use commodity smartphones. These data are also used as fingerprint data of the BLE RSSI signals for training k-NN. This process is much easier than the conventional fingerprinting process, which manually measures the ground truth positions of many interest points and then records the RSSI signal of each interest point.

One advantage of applying PSIM is that we can learn a state space model with high dimensional input and output variables. To learn an accurate state space model, we use the position and velocity estimated using Kalman filter as input variables (Step 2 of Algorithm 1). At t, we have a position estimated using the K-NN of the BLE RSSI signal p_t and a position and velocity estimated using Kalman Filter $\hat{p}_t$ and $\hat{v}_t$, respectively. The input variables for our state space model are $x_t = [p_t, \hat{p}_t, \hat{v}_t] \in \mathbf{R}^9$. We can obtain 3 dimensional positions by using SfM, and x_t is a 9 dimensional vector. By adding Kalman filter prediction as input variables, PSIM will be trained to fix the errors that are caused by the gap between the dynamical system assumed by Kalman filter and a true dynamical system. Here, we applied Kalman filter, but we can also use other prediction models to add input variables.

For the output variables, we can define $s_t = [\bar{p_t}] \in \mathbf{R}^3$ when we estimate only the position by using our approach. Here, $\bar{p_t}$ is the groundtruth position. If we estimate both position and orientation, we can define $s_t = [\bar{p_t}, \bar{q_t}] \in \mathbf{R}^7$, where $\bar{q_t}$ is the groundtruth quaternion. By defining the loss function as the Euclidean distance between estimated f_t and ground truth data over all training trajectories, hypothesis F_n is estimated using a generic regression model.

Our goal was to estimate the current position by using a learned state space model. We use only s_t in the predicted f_t as the estimated position at each t. Note that we use computer-vision localization for collecting training data, but our approach only needs the BLE RSSI signals for the lo-

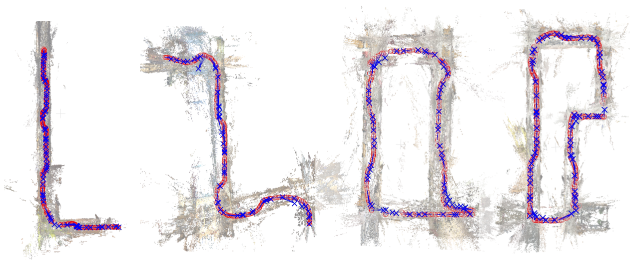

Fig. 1. Example trajectories overlaid on 3D model. Circles are ground truth positions, crosses are positions estimated using k-NN of BLE signals. L1 ($22\mathrm{m} \times 54\mathrm{m}$, 36 beacons), S1 ($25\mathrm{m} \times 30\mathrm{m}$, 76 beacons), O1 ($18\mathrm{m} \times 20\mathrm{m}$, 44 beacons), O2 ($27\mathrm{m} \times 10\mathrm{m}$, 59 beacons)

calization phase, which can be utilized by various navigation systems. It can also be used for other localization approaches as well as BLE based localization.

3. EXPERIMENTAL RESULTS

3.1. CMU BLE Dataset

To evaluate the localization accuracy of our approach, we created a new database called the CMU BLE Dataset. This database includes both videos and BLE RSSI signals for four different indoor locations. Each location has 10 sequences of trajectories for testing localization accuracy. The videos were recorded at 20 fps, and the BLE RSSI signals were recorded at 1Hz. We used iPhone 6 to record the data set.

To create 3D models and ground truth trajectories with SfM, we used the Human Scale Localization Platform (HULOP) [20], which is an open source implementation based on OpenMVG [21][14]. The 3D models are created using videos that are different from those used for testing localization. Fig. 1 shows example trajectories of each dataset. The circles are ground truth positions estimated using SfM, and the crosses are estimated positions using k-NN of the BLE RSSI signals. Each example is overlaid on 3D models used for generating ground truth positions by using SfM. As shown in these examples, we tested various types of routes. In all locations, Kontakt.io Smart Beacons [22] were installed about every 4–6 meters based on a previous study for balancing localization accuracy and deployment cost of BLE beacons [3].

We evaluated localization accuracy by using the leave-one-out method. Each test case selected by the leave-one-out process was repeatedly trained and tested 10 times to evaluate our approach because it randomly selects a validation set during the DAgger step. For each test, 1 trajectory is randomly selected from training data as a validation set. We assumed most navigation systems do not know the start point, and the initial position of each test data was estimated by using k-NN for the first BLE signal.

	L1	S1	O1	O2
RFR[23]	1.04	1.87	1.26	0.98
MLP	1.11	2.59	0.78	1.00
RLR	0.63	1.47	0.77	**0.60**
RFF[24]	**0.63**	**1.16**	**0.66**	0.61

Table 1. Comparison of different learning models. RFR: Random Forest Regression, MLP: Multi Layer Perceptron, RLR: Ridge Linear Regression, RFF: Ridge Linear Regression with Random Fourier Features (Number of random features for RFF was 100, number of hypothesis generations was 5, number of predicted future steps was 3)

	L1	S1	O1	O2
PSIM	0.61	1.13	0.82	0.59
PSIMKF	**0.53**	**1.05**	**0.66**	**0.45**

Table 2. Comparison of using input variables estimated using Kalamn filter: PSIM (only position estimated using k-NN of BLE RSSI signals was used as input variables), PSIMKF (position and velocity estimated using Kalman filter were also added as input variables). Number of hypothesis generations was 10, number of predicted future steps was 5, other parameters were the same as those in Table 1

3.2. Comparison of different PSIM models

Our approach can use different regression models and different input and output variables. We evaluated the accuracy of our proposed approach for different settings.

We first compared four different regression models, Random Forest Regression (RFR) [23], Multi Layer Perceptron (MLP), Ridge Linear Regression (RLR), and Ridge Linear Regression with Random Fourier Features (RFF) [24]. Table 1 lists the results, which were evaluated by median error in meters for all tested trajectories. RFF learned non-linear functions efficiently and showed the best performance. We now discuss the results of RFF.

We also compared how much the input variables created using Kalman filter helped improve the localization accuracy (Step 2 of Algorithm 1). Table 2 lists the results. This additional information improved the localization accuracy. We now discuss the results of using additional input variables created using Kalamn filter.

We also compared a different number of DAgger steps (N in Algorithm 1). Fig. 2 shows the results. In general, increasing the number of DAgger steps improved accuracy, and about 5 DAgger steps were sufficient to improve accuracy.

3.3. Comparison with baseline models

We compared our approach with existing state space models. Table 3 lists the results of Kalman filter, Particle filter, and our proposed approach. For both baselines, we used 6 dimensional observed states (3 dimensional positions and 3 dimensional velocities) with a constant acceleration model. For Particle filter, we used 500 particles.

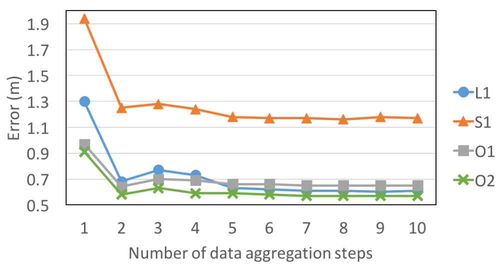

Fig. 2. Comparison of different numbers of data aggregation steps. (Other parameters were the same as those in PSIMKF in Table 2

	L1	S1	O1	O2
KF	0.73	1.16	0.80	0.76
PF	0.88	1.19	0.84	0.82
Ours	0.54	1.06	**0.54**	**0.47**
Ours +Rot	**0.53** (6.6)	**1.03** (16.1)	0.58 (20.2)	0.48 (23.4)

Table 3. Comparison with baseline models: KF: Kalman Filter, PF: Particle Filter, Ours: Our approach. Number of hypothesis generations was 10. Other parameters were same as those in Fig. 2, Ours+Rot: Our approach using rotation information as output variables. Other parameters were same as those for "Ours". Numbers in parentheses are median rotational errors in degrees.

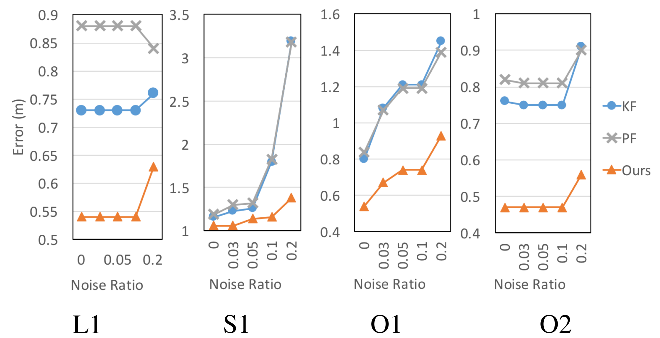

Fig. 3. Comparison of different noise ratios for BLE signals. Other parameters were same as those for "Ours" in Table 3.

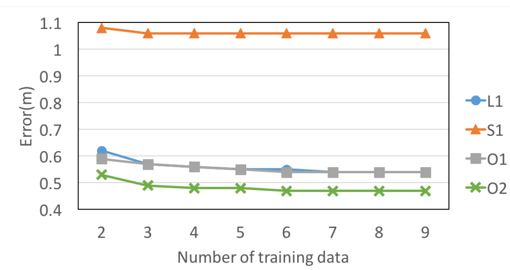

Fig. 4. Comparison of different amounts of training data. Other parameters were same as those for "Ours" in Table 3.

"Ours" in Table 3 shows the results of our approach when we define the output states as a 3 dimensional position vector. The results show that our approach consistently outperformed the other baseline models for all locations, and it improved the accuracy at most about 0.3m

Although the BLE RSSI signal does not have directional information, our approach can also learn a state space model that has rotation as output states. "Ours+Rot" in Table 3 shows the results of our approach when we define output states as a 7 dimensional vector that has position and quaternion. Our approach can estimate rough orientation without decreasing localization accuracy even if we do not use devices with directional information.

In actual navigation systems, some BLE beacons often go out of order due to battery exhaustion or device failure. We evaluated robustness in such a situation by randomly selecting BLE beacons and ignoring the signals of these beacons. The selected BLE beacons were ignored both during the training phase and the testing phase in our approach. Each test case selected by the leave-one-out process was repeatedly tested 10 times for all approaches because the ignored BLE beacons were randomly selected. Fig. 3 shows the results. The horizontal axis is the noise ratio: the ratio of the number of randomly selected BLE beacons to the number of all BLE beacons. The vertical axis is the median error in meters. When increasing the noise, our approach was not affected as much as the baseline approaches were. Our approach improved the accuracy at most about 1.9m for the noise ratio 0.2. The results also show that our approach will work well even in an environment where we cannot install enough BLE beacons because of the constraints in the environment or deployment cost.

3.4. Comparison of different amounts of training data

To evaluate how easy our approach is to deploy for navigation systems, we evaluated how many training trajectories are necessary for training. In this experiment, we used all training trajectories as fingerprinting data and changed only the amount of training data for learning the state space model (τ_i in Algorithm 1). Fig. 4 shows the results. To achieve better accuracy than the baseline models, only two training data were enough for all datasets.

4. CONCLUSION

We proposed an approach to apply a supervised state space model to radio-wave based localization. In many navigation applications, the history of walking data is often available and can be used to train a model in a supervised manner. Our experimental results showed that the proposed approach outperformed the current state space models, and it works well even in an environment with fewer BLE beacons. The accuracy was better even if the size of the training data was small. Our approach is not limited to specific localization methods or devices and can be utilized by various navigation systems that require more accuracy and robustness against errors.

5. REFERENCES

[1] Jiang Xiao, Zimu Zhou, Youwen Yi, and Lionel M. Ni, "A survey on wireless indoor localization from the device perspective," *ACM Comput. Surv.*, vol. 49, no. 2, pp. 25:1–25:31, 2016.

[2] Navid Fallah, Ilias Apostolopoulos, Kostas Bekris, and Eelke Folmer, "The user as a sensor: navigating users with visual impairments in indoor spaces using tactile landmarks," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2012, pp. 425–432.

[3] Dragan Ahmetovic, Cole Gleason, Chengxiong Ruan, Kris Kitani, Hironobu Takagi, and Chieko Asakawa, "Navcog: A navigational cognitive assistant for the blind," in *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services*. ACM, 2016, pp. 90–99.

[4] Patrick Lazik, Niranjini Rajagopal, Oliver Shih, Bruno Sinopoli, and Anthony Rowe, "Alps: A bluetooth and ultrasound platform for mapping and localization," in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2015, pp. 73–84.

[5] Benjamin Kempke, Pat Pannuto, and Prabal Dutta, "Polypoint: Guiding indoor quadrotors with ultra-wideband localization," in *Proceedings of the 2nd International Workshop on Hot Topics in Wireless*. ACM, 2015, pp. 16–20.

[6] Jue Wang and Dina Katabi, "Dude, where's my card?: Rfid positioning that works with multipath and non-line of sight," *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp. 51–62, 2013.

[7] Yuhang Gao, Jianwei Niu, Ruogu Zhou, and Guoliang Xing, "Zifind: Exploiting cross-technology interference signatures for energy-efficient indoor localization," in *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013, pp. 2940–2948.

[8] Suining He and S-H Gary Chan, "Wi-fi fingerprint-based indoor positioning: Recent advances and comparisons," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 466–490, 2016.

[9] Sebastian Thrun, Wolfram Burgard, and Dieter Fox, *Probabilistic robotics*, MIT press, 2005.

[10] Wen Sun, Arun Venkatraman, Byron Boots, and J. Andrew Bagnell, "Learning to filter with predictive state inference machines," in *Proceedings of the 33rd International Conference on Machine Learning*, 2016.

[11] Stephane Ross, Daniel Munoz, Martial Hebert, and J Andrew Bagnell, "Learning message-passing inference machines for structured prediction," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 2737–2744.

[12] Guobin Shen, Zhuo Chen, Peichao Zhang, Thomas Moscibroda, and Yongguang Zhang, "Walkie-markie: indoor pathway mapping made easy," in *Proceedings of the 10th USENIX conference on Networked Systems Design and Implementation*. USENIX Association, 2013, pp. 85–98.

[13] Yungeun Kim, Hyojeong Shin, Yohan Chon, and Hojung Cha, "Smartphone-based wi-fi tracking system exploiting the rss peak to overcome the rss variance problem," *Pervasive and Mobile Computing*, vol. 9, no. 3, pp. 406–420, 2013.

[14] Pierre Moulon, Pascal Monasse, and Renaud Marlet, "Global fusion of relative motions for robust, accurate and scalable structure from motion," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 3248–3255.

[15] Sylvie Treuillet and Eric Royer, "Outdoor/indoor vision-based localization for blind pedestrian navigation assistance," *Int. J. of Image and Graphics*, vol. 10, no. 04, pp. 481–496, 2010.

[16] Ramsey Faragher and Robert Harle, "Location fingerprinting with bluetooth low energy beacons," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 11, pp. 2418–2428, 2015.

[17] Brian Ferris, Dieter Fox, and Neil Lawrence, "Wifi-slam using gaussian process latent variable models," in *Proceedings of the 20th international joint conference on Artifical intelligence*. Morgan Kaufmann Publishers Inc., 2007, pp. 2480–2485.

[18] Michael L Littman, Richard S Sutton, and Satinder P Singh, "Predictive representations of state.," in *NIPS*, 2001, vol. 14, pp. 1555–1561.

[19] Stéphane Ross, Geoffrey J Gordon, and Drew Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *International Conference on Artificial Intelligence and Statistics*, 2011, pp. 627–635.

[20] "Human-scale localization platform (hulop).," https://github.com/hulop/SfMLocalization.

[21] Pierre Moulon, Pascal Monasse, Renaud Marlet, and Others, "Openmvg. an open multiple view geometry library.," https://github.com/openMVG/openMVG.

[22] "Kontakt.io," https://kontakt.io/.

[23] Leo Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[24] Ali Rahimi and Benjamin Recht, "Random features for large-scale kernel machines," in *Advances in neural information processing systems*, 2007, pp. 1177–1184.