Seeing the Unobservable: Channel Learning for Wireless Communication Networks

Jingchu Liu, Ruichen Deng, Sheng Zhou, and Zhisheng Niu
Tsinghua National Laboratory for Information Science and Technology
Department of Electronic Engineering, Tsinghua University
Beijing 100084, China
Email: {liu-jc12,drc09}@mails.tsinghua.edu.cn, {sheng.zhou,niuzhs}@tsinghua.edu.cn

***Abstract*—Wireless communication networks rely heavily on channel state information (CSI) to make informed decision for signal processing and network operations. However, the traditional CSI acquisition methods is facing many difficulties: pilot-aided channel training consumes a great deal of channel resources and reduces the opportunities for energy saving, while location-aided channel estimation suffers from inaccurate and insufficient location information. In this paper, we propose a novel *channel learning* framework, which can tackle these difficulties by inferring unobservable CSI from the observable one. We formulate this framework theoretically and illustrate a special case in which the learnability of the unobservable CSI can be guaranteed. Possible applications of channel learning are then described, including cell selection in multi-tier networks, device discovery for device-to-device (D2D) communications, as well as end-to-end user association for load balancing. We also propose a neuron-network-based algorithm for the cell selection problem in multi-tier networks. The performance of this algorithm is evaluated using geometry-based stochastic channel model (GSCM). In settings with 5 small cells, the average cell-selection accuracy is 73% - only an 3.9% loss compared with a location-aided algorithm which requires genuine location information.**

I. Introduction

Channel state information (CSI) plays a vital role in wireless communication systems. Both signal processing functions such as pre-coding and network operations such as user association requires CSI. The most prevalent method to obtain CSI is through pilot-aided channel training, in which a known sequence of signals is send between the transmitter and the receiver so that the channel responses can be estimated. Nevertheless, the price is the channel resources and energy required to transmit and process pilots. This drawback renders pilot-aided channel training seriously challenged by the recent developments of wireless networks. Due to the recent surge in mobile traffic, cellular networks are now witnessing rapid densification [1]. One part of this trend is the shrinking cell size and the increasing spatial reuse. The overlapping coverages of multiple access points (APs), which can be from the same tier, multiple heterogeneous tiers, or different radio access technologies (RATs), facilitate alternative user association strategies that are optimized for interference avoidance, load balancing, energy saving, and etc [2]. Nevertheless, pilot-aided channel training requires the network to acquire the CSI between a massive amount of antenna pairs. Moreover, when base station (BS) sleeping is introduced into the network [3], the CSI about sleeping BSs will be missing unless they are frequently turned on to transmit pilots. This greatly reduces the energy saving opportunities. Another part of the network densification trend is contributed by massive MIMO [4]. The large number of antennas at BSs help the network to fully exploit the degrees of freedom (DoF) in spatial domain, making it possible to serve a large number of users simultaneous using the same piece of spectrum resources. However, the contamination of reused pilots in neighboring cells is shown to be the main limiting factor for massive MIMO systems [5], and downlink CSI is difficult and costly to obtain in frequency-division-duplexing (FDD) massive MIMO systems because of the large number of BS antennas and the expense of uplink feedback.

Realizing the drawbacks of traditional pilot-aided channel training, location-aided channel estimation has been proposed to infer the statistical CSI of users with the help of geographical channel database [6]. Some recent work also utilizes the Guassian model to extrapolate channels for locations that are not registered in the database [7]. However, accurate localization is generally expensive and difficult: network-side localization which utilizes the angle-of-arrival (AoA) and time-of-arrival (ToA) information is often inaccurate because of multi-path propagation and small system bandwidth; at the same time, device-side localization which rely on satellite assistance can drain device battery fast and is inapplicable to indoor scenarios. These deficiencies make location-aided methods only applicable for estimating coarse CSI in limited scenarios.

In this paper, we propose a novel *channel learning* framework which can infer unobservable CSI from the observable one. Essentially, this framework is build upon the dependence between channel responses and location information. However, channel learning *does not* explicitly exploit location information like existing location-assisted methods. Instead, this framework regards location information as a hidden factor and only deals with the relationship between observable and unobservable channel responses. Harnessing the power of machine learning models, we can build effective algorithms for approximating the mapping between the observable and unobservable channels. The channel learning framework is very flexible and can be applied into many scenarios, including cell selection in multi-tier networks, device discovery in device-to-device (D2D) communications, as well as end-to-end network operations.

The rest of the paper is organized as follows. In Section II, we introduce the channel learning framework and discuss its practicability. In Section III, we describe the various applications of the channel learning framework. An algorithm that is based on the neuron network (NN) is then proposed for cell selection learning in multi-tier networks in Section IV. In Section V, We evaluate the accuracy of the proposed algorithm using geometry-based stochastic models and compare it with other algorithms. The paper is concluded in Section VI.

II. Channel Learning Framework

Channel learning is the process of inferring unobservable channel information from the observable one. Assume a user is equipped with a single-antenna device to communicate with BS antennas in set $\mathbb{A}$. However, only the channel to the antennas which is in the observable subset $\mathbb{O} \subset \mathbb{A}$ can be obtained, whereas the channel to the rest of the antennas, which constitutes the unobservable subset $\mathbb{U} \subset \mathbb{A}$, is unknown to either the user or the network. We denote the channel[1] to the observable antennas as $\boldsymbol{h}_o$ and the channel to the unobservable antennas as $\boldsymbol{h}_u$. The user wants to know some metrics of the unobservable channel according to the metric function $\boldsymbol{m} = m(\boldsymbol{h}_u)$. Using these notations, we can formulate channel learning as the task to approximate a function $f(\cdot)$ from the observable channel to the unobservable metrics:

$$\begin{aligned} f\colon \mathbb{C}^{|\mathbb{O}|} &\to \mathbb{M} \\ \boldsymbol{h}_o &\mapsto \boldsymbol{m} \end{aligned} \tag{1}$$

where $\mathbb{C}$ is the set of complex numbers, $|\cdot|$ gives the cardinality of a set, and $\mathbb{M}$ is the co-domain of $m(\cdot)$. As long as function $f(\cdot)$ exist, we can utilize supervised learning algorithms to approximate it. Here we note that the training phase for supervised learning algorithms can be an off-line procedure. So the unobservable metric $\boldsymbol{m}$ in the training set can be easily calculated from the unobservable channel samples, which can be collected through traditional pilot-aided channel estimation during the training phase. Once training is finished, only the observable channel responses are needed to predict the unobservable metric.

The existence of $f(\cdot)$ is not trivial and cannot be guaranteed in the general sense. For example, in a propagation environment where there are no scatterers and just line-of-sight (LoS) propagation, users that lie on symmetric locations with respect to an uniform linear array will have the same array responses. But their channels to the antennas that do not lie on the symmetric axis could be different. If we regard the array response as the observable channel $\boldsymbol{h}_o$ and the channels to other antennas as $\boldsymbol{h}_u$, then one $\boldsymbol{h}_o$ may be mapped to two different values of $\boldsymbol{h}_u$ and $\boldsymbol{m}$. Thus $f(\cdot)$ does not exist. Therefore, we should carefully prove the existence of $f(\cdot)$ in different scenarios. In the next theorem, we provide a special scenario in which the existence of function $f(\cdot)$ can be guaranteed.

Theorem 1. *If the channel between the user and the BS antennas is a function of the user's geographical location $\boldsymbol{x}$, i.e. $\boldsymbol{h}_o = g_o(\boldsymbol{x})$ and $\boldsymbol{h}_u = g_u(\boldsymbol{x})$, and the function from user locations to observable channels $g_o(\cdot)$ is invertible, then there exist a function $f(\cdot)$ as defined in (1).*

Proof: If $g_o(\cdot)$ is invertible and its inverse is $g_o^{-1}(\cdot)$, the user location can be obtained from observable channel as $\boldsymbol{x} = g_o^{-1}(\boldsymbol{h}_o)$, and the unobservable channel can be derived as $\boldsymbol{h}_u = (g_u * g_o^{-1})(\boldsymbol{h}_o)$. Then the unobservable metric can be directly calculated as $\boldsymbol{m} = (m * g_u * g_o^{-1})(\boldsymbol{h}_o) = f(\boldsymbol{h}_o)$. ■

Note even though $g_o(\cdot)$ can be invertible, given the difficulty to obtain accurate location information, we still cannot infer location $\boldsymbol{x}$ explicitly from the observable channel $\boldsymbol{h}_o$.

The applicability of the above theorem depends on two assumptions: 1) channel response is a function of user location; and 2) observable channel function $g_o(\cdot)$ is invertible. Next we discuss the practicability of these assumptions:

Discussion 1 (Location Determines Channel Responses). Channel responses in wireless fading channels can be decomposed into three factors: path loss, large-scale fading, and small-scale fading. Among them, path loss and large-scale fading is often considered to be static over time. Therefore, these two factors can be considered an unknown but deterministic function of user location. Small-scale fading is caused by the interference among multi-path components coming from ambient scattering. The relative location between the user, the scatterers, and the BS antennas should suffice to determine small-scale fading. As a consequence, small-scale fading will be a deterministic function of user location as long as the scatterers remain in the same position. Even if some of the scatterers do move around, the small-scale fading will only change negligibly over time if the scattering power from these scatters is small. In such cases, channel responses is still approximately a function of the user location.

Discussion 2 (Invertible Observable Channel Function). The invertibility of the observable channel function $g_o(\cdot)$ can be guaranteed by the utilization of large number of antennas. In a rich-scattering environment with unlimited number of observable antennas, it is known that the channel responses from different users are asymptotically orthogonal [8]. Thus, we know the observable channel responses from different users are asymptotically different:

$$g_o(\boldsymbol{x}_1) \neq g_o(\boldsymbol{x}_2), \text{ if } \boldsymbol{x}_1 \neq \boldsymbol{x}_2. \tag{2}$$

Therefore, we can say that $g_o(\cdot)$ is an one-to-one mapping from user location to the observable channel responses, and $g_o(\cdot)$ is invertible. Even if there are only limited number of scatterers and observable antennas, the invertibility of $g_o(\cdot)$ can also observed. Fig. 1 illustrates the relationship between

[1] Although the direction of the link, i.e. uplink or downlink, does not change the channel learning framework, some configurations are more preferable due to practical consideration. For example, it is more convenient that the observable channels are all uplink channels so that the network can directly use them to infer unobservable channels, instead of having to collect feedback from user equipment.

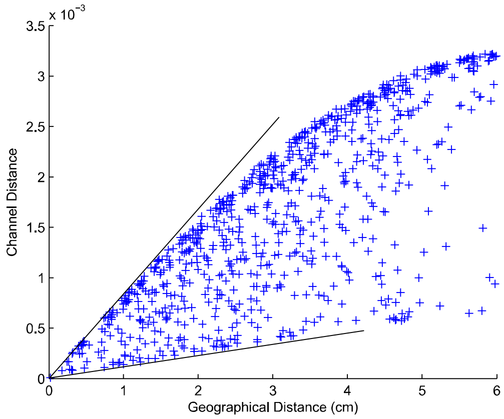

Fig. 1. Channel distance v.s. geographical distance between two users. GSCM simulation using 20 scatterers, Rician factor $K = 10$ dB, uniform linear array with 100 units.

channel distances and users' geographical distances. As can be seen, the lower and upper bounds for channel distances stretch from the origin and diverge as we increase the distance between users. This also implies the relationship as described in (2) and results in a invertible $g_o(\cdot)$.

III. Applications of Channel Learning

Channel learning is a flexible framework with a wide variety of potential applications. It is especially valuable for scenarios where it is costly or difficult to obtain channel information. In this section, we discuss some of these applications and describe how the framework is applied.

A. *Cell Selection Learning in Multi-tier Networks*

Consider the problem of cell selection in multi-tier networks. The network needs to know the user's channel to a large number of small cells in order to select the best cell. This usually requires the network to sacrifice scarce channel resources for channel sounding. And users must scan for potential small cells, which is extremely energy- and time-consuming. Moreover, frequent channel sounding reduces the opportunity for BS sleeping in networks with dynamic sleeping operations, which greatly limits the potential for energy saving.

The channel learning framework can be applied here to solve these problems. A possible configuration is to place large antenna arrays at large cells and use them as the observable antennas to infer the channel to the overlapping small cells. The users intending to communication send sounding signals to the observable array at the large cell. Based on the array response $\boldsymbol{h}_o$, the network infers the best small cell for each user. Here the unobservable channel $\boldsymbol{h}_u$ is the channel between the user and small cells, and the unobservable metric function $m(\cdot)$ gives the index of the best small cell. In this case, the channel resources consumed for channel sounding is only proportional to the number of active users; the users just need to send sounding information and wait for the networks cell-selection suggestion; and the sleeping small cells that are not selected can continue to sleep.

B. *Device Discovery Learning in D2D Communications*

The discovery of proximate peer devices is a central problem in D2D communications. However, the traditional solutions either require devices to transmit and receive sounding signals, which greatly reduces battery life, or demand the network to provide location information, which is often inaccurate [9].

The channel learning framework provides a new angle to tackle this problem. Specifically, we consider the case of network-assisted device discovery learning using massive BS antenna array. Assume we want to decide the proximity of two single-antenna devices at locations $\boldsymbol{x}_1 \in \mathbb{R}^3$ and $\boldsymbol{x}_2 \in \mathbb{R}^3$, respectively. Imagine a "virtual" device whose location is the composite of two actual device locations $\boldsymbol{x} = \{\boldsymbol{x}_1, \boldsymbol{x}_2\} \in \mathbb{R}^6$. Then we can define the "virtual" observable channel $\boldsymbol{h}_o$ as the composite of the array responses from these two devices, the unobservable channel $\boldsymbol{h}_u$ to be the channel between these two devices, and the metric $m(\cdot)$ to be the probability that $\boldsymbol{h}_u$ is sufficiently good for direct communications.

Using the device discovery learning described above, the network does not need to estimate the actual device locations, which are inaccurate anyway. And the peering decisions can be determined in a centralized manner by the network, instead of in a distributed manner by the individual devices. This could greatly increase the spectrum and energy efficiency for both devices and the network.

C. *Extended Applications*

The channel learning framework can also be extended to predict channel metrics for users with multiple antennas, which is becoming a prevalent for smart devices such as mobile phones. In such a case, the metric function $m(\cdot)$ can be defined to reflect certian properties of the channel matrix, e.g. matrix rank or distribution of eigen-values. As discussed before, the feasibility of this extension is guaranteed as long as the assumptions in Theorem 1 is met.

Another potential extension for the channel learning framework is for making end-to-end network operations decisions which involve channel information. Take user association as an example. To make optimal user association decisions, the network needs to consider not only the channel quality between the users and the BS antennas so as to guarantee the quality of service (QoS), but also other factors such as the current load of each BS for load balancing . Extension in this aspect only requires a change in the metric function. Again consider user association for load balancing. To extend the channel learning framework, we allow the unobservable metric function to take in the BS loads $\boldsymbol{l}$ as an observable parameter, i.e. $\boldsymbol{m} = m(\boldsymbol{h}_u, \boldsymbol{l})$. In this way, the overall function $f(\cdot)$ is also extended to have two sets of arguments, i.e. $\boldsymbol{m} = f(\boldsymbol{h}_o, \boldsymbol{l})$. Following the derivation in in Theorem 1, the existence of function $f(\cdot)$ can still be guaranteed if the two assumption are met.

IV. A NN-based Algorithm

In order to learn the mapping function $f(\cdot)$, we can employ supervised learning algorithms from the field of machine learning. Concretely, we first obtain a training set which contains some observations of $\boldsymbol{h}_o$ and the corresponding $\boldsymbol{m}$. Then we use these training data to choose a candidate function that best approximates function $f(\cdot)$. For illustration, we propose an NN-based algorithm for cell selection learning in multi-tier network. This algorithm combines the standard approaches in NN with special processing procedures that are tailored to the cell selection problem.

A. Extracting Angular-Domain Information

The first step of our algorithm is to extract angular-domain information from the raw observable channel response $\boldsymbol{h}_o$. We do this because it is hard for machine learning algorithms to automatically decode the complex relationship between spatial-domain channel response and user locations. In contrast, angular-domain information if more directly related with location information. So we have to manually convert the channel response into angular domain. Since the antennas of a large cell form a uniform linear array, we use the Fourier transform to obtain the angular-domain observable array response. After that, we extract the magnitude of the angular-domain response and omit the phase information.

B. Forming Quantized Input Feature

The second step is to translate the angular-domain magnitude into the feature space. It is well known that NN cannot handle input features that have large dynamic range. This is because the activation function of NN is the Sigmoid function, which saturates at extremely large or small input values. However, the dynamic range of the channel responses may span several orders of magnitude since users may lie in a wide geographical area. To tackle this problem, we take the logarithm of the angular-domain magnitude and then use the Lloyds algorithm to quantize it. After the logarithmic processing and quantization, the dynamic range of the input can be greatly reduced. The resulting input feature vector is formed by the quantization codebook index (normalized to $[-1, 1]$) of the original angular-domain magnitude vector.

C. Output Encoding

The natural form for the output of cell selection learning is an integer index indicating the best small cell. However, due to the same reason as input encoding, the NN cannot well approximate a function with integer-valued output. For this reason, we employ the probabilistic output coding scheme. Suppose we need to choose from K small cells, then the output is encoded as a length K vector, with each entry stands for the probability that the corresponding small cell is the best cell. In this paper, we user a hard coding scheme for training sets. We set the entry that correspond to the best small cell to value 1, and other entires to value 0. However other soft-coding schemes that have multiple non-zero entries are also allowed in our algorithm. For example in the SoftMax encoding, each entry of the output vector can be set as the ratio between the channel magnitude of the corresponding antenna and the sum of the channel magnitudes of all the unobservable antennas.

D. Cost Function and Training

The cost function we use in the training phase is the sum of two parts. The first part is the average *Kullback cross-entropy* [10] between the current NN outputs and the desired training outputs. Since the output can be interpreted as probability distribution, cross-entropy cost can be naturally applied. Also, cross-entropy cost function is known as a well formed cost function that provides better local minima than other cost functions such as squared error [11]. The second part of our cost function is the regularization term. Assume the NN has N parameters $\{\theta_n, n = 1, 2, \cdots, N-1, N\}$, the regularization term is defined as

$$\lambda \sum_{n=1}^{N} \theta_n^2,$$

where $\lambda \geq 0$ is the regularization parameter. Intuitively, proper regularization guarantees the “sparsity” of the parameters, which in turn avoids over-fitting. We employ the *back-propagation* algorithm to derive the partial gradient of the cost function with respect to the network parameters $\{\theta_n, n = 1, 2, \cdots, N-1, N\}$ and use *conjugate gradient* algorithm to minimize the cost function. Note that other optimization algorithms can also be used to minimize the cost function.

E. Cell Selection Prediction

After the NN has been trained, we can use it to select the best small cell. We take the array response from a user and preprocess the array response to form the quantized input feature. Then we feed the input feature into the trained NN. The NN will output a selection probability for each of the output vector entry. We select the small cell with the largest selection probability. Note the entries of the output vector may not (and often don’t) sum to probability one due to approximation error. This should not cause any problem in our application since only the relative magnitude of the entries matters. However, if the output probability distribution is to be used later, we will need to compensate the approximation error by means such as re-normalization.

V. Simulation Results

In this section, we evaluate the performance of the proposed learning algorithm in cell selection problems. We first introduce the simulation settings and then show the accuracy of the proposed algorithm.

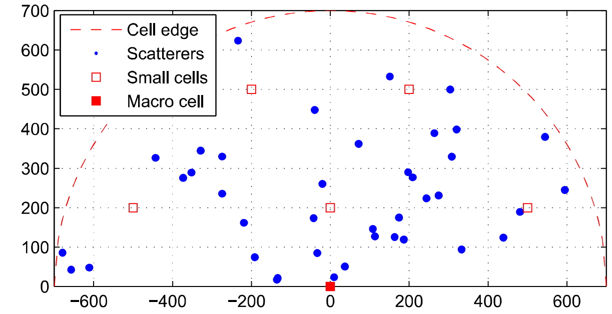

Fig. 2. Network layout for simulation. The macro cell is the red filled square located at the origin, red hollow squares represent small cells, and blue dots represent scatterers.

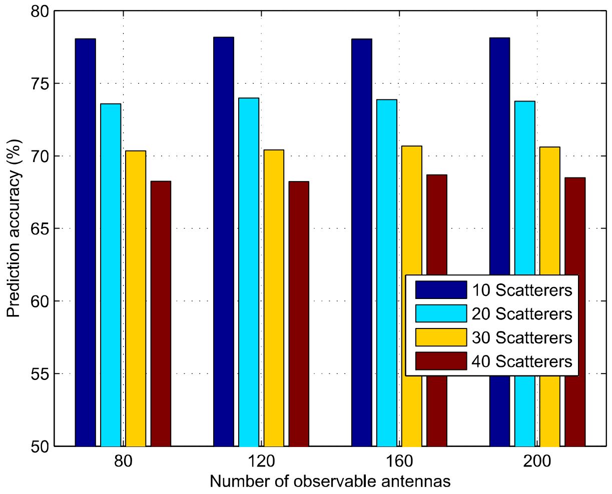

Fig. 3. Prediction accuracy v.s. different number of antennas and scatterers.

A. Simulation Settings

We use the GSCM [12] to generate fading channels in our simulation. The basic procedure is to firstly randomly place scatterers, calculate the complex channel gains of multi-path components, and then sum them up to get the final fading channel response. For simplicity, we only consider LoS propagation and single scattering.

The network layout is shown in Fig. 2. We consider a single macro cell and 5 small cells. The coverage area of the macro cell is a circle with a radius of 700m. To avoid mirroring ambiguity, we only consider the upper half of the circle. Scatterers are randomly dropped in the coverage area of the macro cell following uniform distribution. The macro cell is equipped with an uniform linear array with half-lambda antenna spacing. The Rician factor is 10 dB. And noise is ignored in the simulation.

In every simulation run, we randomly generate scatters and 2000 user locations following uniform distribution, then randomly choose half of the samples for training and the rest half for testing. The prediction accuracy is defined as the percentage of the correct predictions on the test set. The final accuracy is the average accuracy over 50 simulation runs.

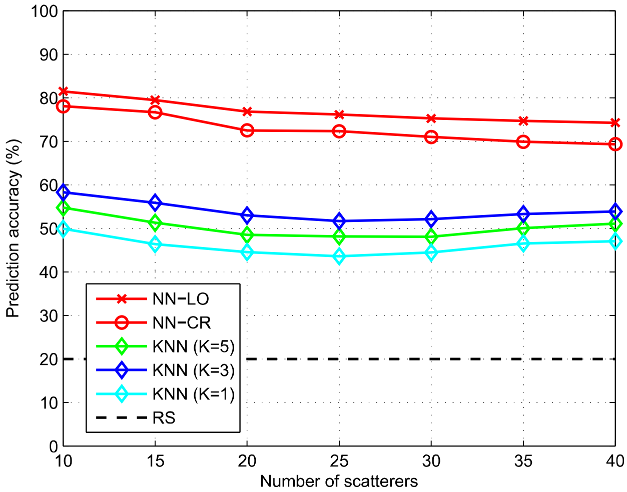

Fig. 4. Prediction accuracy of different algorithms

B. Simulation results

Fig. 3 shows the prediction accuracy of the proposed algorithm with different number of observable antennas and scatterers. We can see that the accuracy slightly increases with the number of observable antennas. This is due to the improvements in angular-domain resolution with more antenna units in the array. Meanwhile, the prediction accuracy decreases with the number of scatterers. This is because the scattering environment becomes more complex when there are more scatterers, making the target function $f(\cdot)$ less smooth and harder to approximate. To solve this problem, deeper machine learning models such as deep NN can be used.

We also compare our algorithm with other prediction algorithms and the results are shown in Fig. 4. As can bee seen, the most naive *random selection* (RS) method provides the baseline accuracy, which is $1/5 = 20\%$ for all scatterer configurations. The simple *K-nearest-neighbor* (KNN) algorithm, which outputs the dominant choice among the k nearest neighbors in the channel space, increases the accuracy to about 50%. There are around 4% variation for different choices of K. In comparison, the accuracy of the proposed NN-based channel learning (NN-CR) algorithm is around 73%, which is far more better than RS and KNN. We also investigate a NN-based algorithm using genuine user location as inputs (NN-LO). Although genuine user location is difficult to obtain in real networks, the performance of this algorithm can be seen as an upper-bound for the performance of NN-based channel learning. We can see that there is an 3.9% gap between the accuracies of NN-CR and NN-LO. This gap is essentially due to the information loss induced by inverting a not-perfectly-invertible observable channel function.

VI. Conclusion

In this paper, we propose the novel channel learning framework to infer unobservable CSI in wireless communication networks using observable one. We give conditions under which this framework is applicable and discuss the practicability of these conditions. We also discuss the potential applications of this framework, including cell selection in multi-tier networks, device discovery for D2D communications, and end-to-end user association for load balancing. For the case of cell selection, we also propose a NN-based supervised learning algorithm. The performance of this algorithm is evaluated using GSCM and compared with some other algorithms. Results show that the proposed algorithm can achieve around 73% accuracy in settings with 5 small cells, which is much better than RS and KNN and only 3.9% less than the NN-based algorithm that utilizes genuine user geographical information.

Acknowledgment

This work is sponsored in part by the National Basic Research Program of China (973 Program: No. 2012CB316001), the National Science Foundation of China (NSFC) under grant No. 61201191, No. 61321061, No. 61401250, and No. 61461136004, and Hitachi Ltd.

References

[1] J. Andrews, S. Buzzi, W. Choi, S. Hanly, A. Lozano, A. Soong, and J. Zhang, "What will 5G be?" *Selected Areas in Communications, IEEE Journal on*, vol. 32, no. 6, pp. 1065–1082, June 2014.

[2] S. Zhou, Z. Niu, P. Yang, and S. Tanabe, "CHORUS: A framework for scalable collaboration in heterogeneous networks with cognitive synergy," *IEEE Wireless Communications*, vol. 20, no. 4, pp. 133–139, August 2013.

[3] Z. Niu, S. Zhou, S. Zhou, X. Zhong, and J. Wang, "Energy efficiency and resource optimized hyper-cellular mobile communication system architecture and its technical challenges," *Science China Information Science*, vol. 42, no. 10, pp. 1191–1203, 2012, (In Chinese).

[4] E. Larsson, O. Edfors, F. Tufvesson, and T. Marzetta, "Massive MIMO for next generation wireless systems," *Communications Magazine, IEEE*, vol. 52, no. 2, pp. 186–195, February 2014.

[5] J. Jose, A. Ashikhmin, T. Marzetta, and S. Vishwanath, "Pilot contamination and precoding in multi-cell tdd systems," *Wireless Communications, IEEE Transactions on*, vol. 10, no. 8, pp. 2640–2651, August 2011.

[6] D. Slock, "Location-aided wireless communications," in *Communications Control and Signal Processing (ISCCSP), 2012 5th International Symposium on*, May 2012, pp. 1–6.

[7] R. Di Taranto, S. Muppirisetty, R. Raulefs, D. Slock, T. Svensson, and H. Wymeersch, "Location-aware communications for 5G networks: How location information can improve scalability, latency, and robustness of 5G," vol. 31, no. 6, Nov 2014, pp. 102–112.

[8] T. Marzetta, "Noncooperative cellular wireless with unlimited numbers of base station antennas," *Wireless Communications, IEEE Transactions on*, vol. 9, no. 11, pp. 3590–3600, November 2010.

[9] G. Fodor, E. Dahlman, G. Mildh, S. Parkvall, N. Reider, G. Miklos, and Z. Turanyi, "Design aspects of network assisted device-to-device communications," *Communications Magazine, IEEE*, vol. 50, no. 3, pp. 170–177, 2012.

[10] P.-T. de Boer, D. Kroese, S. Mannor, and R. Rubinstein, "A tutorial on the cross-entropy method," *Annals of Operations Research*, vol. 134, no. 1, pp. 19–67, 2005. [Online]. Available: http://dx.doi.org/10.1007/s10479-005-5724-z

[11] P. Golik, P. Doetsch, and H. Ney, "Cross-entropy vs. squared error training: a theoretical and experimental comparison." in *INTERSPEECH*, 2013, pp. 1756–1760.

[12] A. Molisch, "A generic model for MIMO wireless propagation channels in macro- and microcells," *Signal Processing, IEEE Transactions on*, vol. 52, no. 1, pp. 61–71, Jan 2004.