Content Popularity Prediction in Fog-RANs: A Bayesian Learning Approach

Yunwei Tao[1], Yanxiang Jiang[1,2], Fu-Chun Zheng[1,2], Mehdi Bennis[3], and Xiaohu You[1]
[1]National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China
[2]School of Electronic and Information Engineering, Harbin Institute of Technology, Shenzhen 518055, China
[3]Centre for Wireless Communications, University of Oulu, Oulu 90014, Finland
E-mail: { tarzan_yw@seu.edu.cn, yxjiang@seu.edu.cn, fzheng@ieee.org, mehdi.bennis@oulu.fi, xhyu@seu.edu.cn }

***Abstract*—In this paper, the content popularity prediction problem in cache-enabled fog radio access networks (F-RANs) is investigated. In order to predict the content popularity with high accuracy and low complexity, we propose a Gaussian process based Poisson regressor to model the content request pattern. Firstly, the relationship between content features and popularity is captured by our developed model. Then, we utilize Bayesian learning to learn the model parameters, which are robust to over-fitting. However, Bayesian methods are usually unable to find a closed-form expression of the posterior distribution. To tackle this issue, we apply a *Stochastic Variance Reduced Gradient* Hamiltonian Monte Carlo (SVRG-HMC) to approximate the posterior distribution. Two types of predictive content popularity are formulated for the requests of existing contents and newly-added contents. Simulation results show that the performance of our proposed policy outperforms the policy based on other Monte Carlo based method.**

***Index Terms*—Fog radio access networks, content popularity prediction, stochastic gradient, content feature, Bayesian learning**

I. Introduction

The rapid increase of smart devices and cellular users pose unprecedented challenges to wireless networks. The limited wireless resources of cellular networks cannot cope with the increasing data traffic pressure. In order to address this issue, fog radio access network (F-RAN) has emerged as a promising solution to avoid congestion caused by repeated transmission of the same content on backhaul links. In F-RANs, fog access points (F-APs) with limited caching capacity and computing resources are deployed at the edge of the network, and popular contents can be cached in them to satisfy users' requests [1]. In view of the caching capacity constraints, the issue of predicting content popularity has attracted more and more attention, which can play an important role in improving caching efficiency [2].

Traditional caching strategies such as least recently used [3] and least frequently used [4] are widely used in wired networks with sufficient storage and computing resources, but are inefficient since their overlook of content popularity. Recently, researchers have shown an increased interest in predicting content popularity. In [5], the authors proposed a user preference model to predict content popularity and the model parameters were learned by online gradient descent (OGD), the method was proved to be superior to traditional caching schemes in [4]. A simplified bidirectional long short-term memory (Bi-LSTM) network based content popularity prediction scheme was proposed in [6], which tracked popularity trend by the number of requests. In [7], federated learning was utilized to obtain the context-aware popularity prediction model. In [8], popular contents were learned through a Gaussian process based Possion regressor model whereby Bayesian learning was applied to optimize the model parameters. However, in F-RANs, it is difficult to obtain adequate data about numerous contents (e.g request numbers) through F-APs with limited storage [9]. These existing works except [7] and [8] fail to predict the popularity of newly-added content with few statistical data. In [8], the algorithm converges slowly, which make the popularity cannot be predicted in time. In [7], the interference among mobile devices in federated learning remains an open issue. In addition, most of the existing works ignore the correlation between content popularity and content feature.

Motivated by the aforementioned discussions, we propose a popularity prediction policy via content request probability model based on content feature. The model parameters are learned by Bayesian learning, a robust method against overfitting caused by lack of statistical data [10]. The training inputs of the prediction model are the features of contents requested by users and the number of requests in different time slots. Owing to the computing resources constraint in F-APs, a modified stochastic gradient based Bayesian learning method is adopted to learn the popularity prediction model. Our proposed policy is able to capture the similarity among contents on the basis of their features. Specifically, our proposed popularity prediction policy enables us to predict the popularity of newly-added contents and reduce the time complexity simultaneously.

The rest of this paper is organized as follows. In Section II, the system model is described. The proposed popularity prediction policy is presented in Section III. Simulation results are shown in Section IV. Final conclusions are drawn in Section V.

II. System Model

As shown in Fig. 1, we consider an F-RAN architecture with many F-APs, where the users are served by these F-APs and the cloud server. Let $\mathcal{C} = \{c_1, c_2, \ldots, c_f, \ldots, c_F\}$ denote the content library stored in the cloud server. Let $\boldsymbol{x}_f$ denote the

GLOBECOM 2021 - 2021 IEEE Global Communications Conference | 978-1-7281-8104-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/GLOBECOM46510.2021.9685947

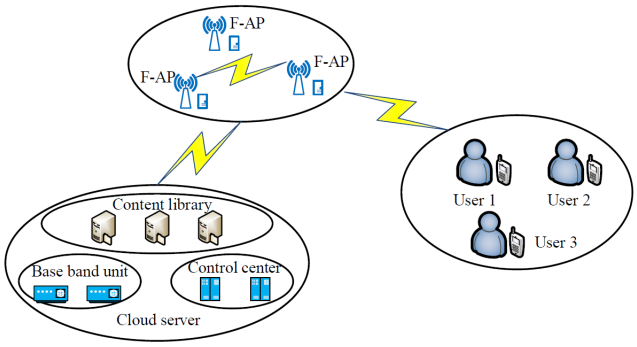

Fig. 1: Illustration of the scenario in F-RAN

feature vector of content c_f to describe the feature of it. Take the movie as an example, it may include genre (e.g., action, comedy, science) and some other features such as release year and publisher. If the content c_f requested by a user is stored in the F-AP associated with it, then a cache hit occurs, where the content can be fetched directly from that F-AP. Otherwise, the content needs to be fetched from the neighboring F-APs or the cloud server [11].

The F-AP can collect information about user requests for content. We define $\boldsymbol{r}_c[n] = \left[r_{c_1}[n], r_{c_2}[n], \ldots, r_{c_f}[n], \ldots, r_{c_F}[n]\right]^T$ as a vector to represent the number of requests for each content in content library during the n-th time slot. By Considering that the number of requests show the degree of preference for movies, we take the number of requests as the popularity for the contents. Assume that N time slots have been observed, let $r^*_{c_f}[N+1]$ and $r_{c_f}[N+1]$ denote the predicted and real number of requests of c_f, respectively. There exists deviation between predicted and real number of requests. Correspondingly, the root mean-square error (RMSE) is utilized to measure the accuracy of the prediction as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{F}\sum_{f=1}^{F}\left|r^*_{c_f}[N+1] - r_{c_f}[N+1]\right|^2}. \quad (1)$$

The cache hit rate is defined as the ratio of the number of cache hits occurring to the total number of requests. Each F-AP is available to the predicted popularity and it will adjust the contents stored based on their popularity. Besides, the F-RAN makes it able to share predicted popularity among F-APs. By caching content with higher popularity, the cache hit rate can be significantly increased.

The objective of this paper is to find a content popularity prediction strategy that maximizes the cache hit rate with low computational complexity while maintaining accuracy.

III. Proposed Popularity Prediction Policy

In this section, we propose a Bayesian learning based content popularity prediction policy which includes probabilistic model construction phase, model learning phase and predicting phase. The proposed policy utilizes the number of requests and content feature, and can predict content popularity accurately with low computational complexity.

A. Policy Description

1) Probabilistic model construction phase: By considering the nonlinear relationships between content feature and content popularity, traditional regression models are not able to capture it. Gaussian process can model nonlinear relationship flexibly and effectively. Therefore, the construction of a probabilistic regression model based on Gaussian process is the first step in our proposed policy. If the feature vectors of contents are similar, the prevalence of the two contents will be close to each other.

2) Model learning phase: In the edge caching structure in F-RAN, F-APs obtain only a few request observations so that overfitting is an important issue to be faced. We propose to learn the probabilistic model via Bayesian Learning since its robustness to overfitting. However, Bayesian learning cannot yield a closed-form expression in most cases. Therefore, Markov Chain Monte Carlo (MCMC) is utilized to approximate the model parameters. Specifically, we apply the stochastic variance reduced gradient based variant of MCMC to content popularity prediction for the first time. First, we obtain the unnormalized posterior approximation of the parameters based on the prior knowledge of them. Second, we compute the gradient of the posterior distribution based on the existing request observations accordingly, and finally we apply a discretization method to approximate the updated parameters via multiple sampling.

3) Predicting phase: After training phase, the approximate model parameters can be obtained. Correspondingly, F-AP enables to predict the popularity of the existing content in the next time slot or the popularity of the newly-added content with lower computational complexity.

B. Probabilistic Model Construction Phase

In this section, we introduce the multilevel probabilistic regression model:

$$\boldsymbol{r}_{c_f}[n]\,|\lambda_f(\boldsymbol{x}_f) \sim \mathrm{Poisson}\left(e^{\lambda_f(\boldsymbol{x}_f)}\right), \forall n = 1, \ldots, N, \quad (2a)$$

$$\lambda_f(\boldsymbol{x}_f)\ \ |g(\boldsymbol{x}_f), \beta_0 \sim \mathcal{N}(g(\boldsymbol{x}_f), \beta_0), \quad (2b)$$

$$g(\boldsymbol{x})\,|\boldsymbol{x}, \beta_1, \ldots,\ \beta_{Q+1} \sim \mathcal{GP}(0, K(\boldsymbol{x}, \boldsymbol{x}')). \quad (2c)$$

We assume that requests for content are independent of each other, hence we can use a Poisson distribution (2a) at the first level of the model to describe the arrival of content requests. The natural parameter $\lambda_f(\boldsymbol{x}_f)$ of Poisson distribution is the arrival rate, which is a function of the Q-dimensional content feature vector $\boldsymbol{x}$. This allow contents with similar features to have a similar number of requests, corresponding to the prior information.

At the second level (2b), in order to measure the difference in the popularity of contents with the same features, we assume that the natural parameters of the first level follow a Gaussian distribution with mean $g(\boldsymbol{x}_f)$ and variance β_0.

At the third level (2c), we apply the Gaussian process aforementioned and assume that $g(\boldsymbol{x}_f)$ is a realization of a Gaussian process with zero mean and covariance matrix of $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$, which captures the relationship between the different features and reflects in the position of the subsequent samples. The covariance matrix $[\boldsymbol{K}]_{i,j} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$. The $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is the kernel function which determines the expressibility and interpretability of Gaussian process, and it uniquely determines the correlation or similarity between different random variables. We apply the squared exponential kernel (SEK) [12] to our policy. SEK is defined as follows:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \beta_1 e^{-\sum_{q=2}^{Q+1} \beta_q \left\| x_i^{q-1} - x_j^{q-1} \right\|^2}, \tag{3}$$

where β_1 is the vertical scale variation and β_{q+1} is the horizontal scale variation on q-th dimension.

C. Model learning phase

1) Bayesian learning in a Nutshell: Bayesian learning is an important branch in machine learning, which is based on Bayesian theory. By assuming that the variables of a model follow a possible distribution then inferring based on this distribution and observed data to make optimal decisions. The most important advantage is that it integrates prior knowledge of model parameters, which makes the model more robust to over-fitting.

Given a set of training examples $\mathcal{D}$, we need to infer the model $\boldsymbol{h}$ that generates these data. We consider the model $\boldsymbol{h}$ to be jointly determined by the objective function and the parameters $\boldsymbol{\omega}$ of the distribution. Since the objective function is usually specified before training, we are thus more interested in $\boldsymbol{\omega}$. In other words, we can infer the model $\boldsymbol{h}$ by estimating the parameters $\boldsymbol{\omega}$, $\boldsymbol{h} = \arg\max \{\mathrm{P}(\boldsymbol{\omega} | \mathcal{D}), \boldsymbol{h} \in \mathcal{H}\}$, $\mathcal{H}$ is the set of all possible models.

Denote the parameters which maximize the posterior as estimate value, the Bayesian rule provides an effective way to calculate maximum probability:

$$P(\boldsymbol{\omega} | \mathcal{D}) = \frac{P(\boldsymbol{\omega}) P(\mathcal{D} | \boldsymbol{\omega})}{P(\mathcal{D})} = \frac{P(\boldsymbol{\omega}) P(\mathcal{D} | \boldsymbol{\omega})}{\int_{\boldsymbol{\omega}} P(\boldsymbol{\omega}) P(\mathcal{D} | \boldsymbol{\omega}) d\boldsymbol{\theta}} \tag{4}$$

where $P(\boldsymbol{\omega})$ is the prior distribution and $P(\boldsymbol{\omega} | \mathcal{D})$ is the posterior distribution based on training data. $P(\mathcal{D})$ is the normalization constant.

2) Model learning: In this section, we use a Bayesian paradigm approach to train the model in (2). We set up the model assuming some prior information about the unknown parameter, then combine the information with the request observations $\mathcal{R} = \{\boldsymbol{r}_c[1], \boldsymbol{r}_c[2], \ldots, \boldsymbol{r}_c[n], \ldots, \boldsymbol{r}_c[N]\}$ collected by F-APs and update the posterior information about the parameter via Bayesian inference. Finally, we infer the estimated value of the model parameters $\{\lambda_f(\boldsymbol{x}_f)\}_{f=1}^{F}$ and $g(\boldsymbol{x})$. Moreover, to simplify the inference, we can integrate out from the model. It can be expressed as follows:

$$[\lambda_1(\boldsymbol{x}_1), \ldots, \lambda_2(\boldsymbol{x}_f), \ldots, \lambda_F(\boldsymbol{x}_F)]^T \sim \mathcal{N}(0, \boldsymbol{K}'). \tag{5}$$

where $\boldsymbol{K}' = \boldsymbol{K} + \beta_0 \boldsymbol{I}$. In our proposed model, it is challenging to determine $\{\beta_q\}_{q=0}^{Q+1}$ due to the lack of prior knowledge. An intuition is to assume that the parameters follow a prior distribution, which is used to describe its uncertainty. Since $\{\beta_q\}_{q=0}^{Q+1}$ representing parameters of the variance and the kernel function so that they must be positive. Correspondingly, Gamma distribution is adopted as prior knowledge:

$$\beta_q \sim \mathrm{Gamma}(A_q, B_q), \ \forall q = 0, ..., Q+1, \tag{6}$$

where A_q and B_q are the shape and scale of Gamma distribution, respectively.

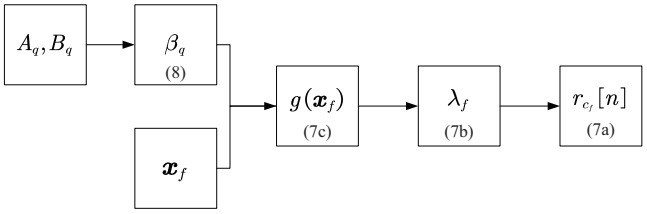

Fig. 2: The mapping relationship of our proposed model

Fig. 2 shows the mapping relationship between the parameters at each level of the probabilistic model. By the Bayes rules, we can derive the posterior distribution of the unknown parameters of the model:

$$p(\boldsymbol{\lambda}, \boldsymbol{\beta} | \mathcal{R}) = \frac{p(\mathcal{R} | \boldsymbol{\lambda}) p(\boldsymbol{\lambda} | \boldsymbol{\beta}) \prod_{q=0}^{Q+1} p(\beta_q)}{H}, \tag{7}$$

where $\boldsymbol{\beta}$ and $\boldsymbol{\lambda}$ denote the set $\{\beta_q\}_{q=0}^{Q+1}$ and $\{\lambda_f(\boldsymbol{x}_f)\}_{f=1}^{F}$, respectively. $p(\boldsymbol{\lambda}, \boldsymbol{\beta} | \mathcal{R})$ is the posterior distribution based on the request observations and prior knowledge and H is a normalization constant. However, complex integration operations are required to yield the normalization constant H, it is impractical to get a closed-form expression for the posterior distribution. Therefore, we need an efficient sampling method that is capable to sample from an unnormalized posterior distribution and average multiple samples to approximate the exact posterior distribution.

The Hamiltonian Monte Carlo (HMC) is a popular method that gets S samples $\{\boldsymbol{\xi}_s\}_{s=1}^{S}$ from a U-dimensional distribution $p(\boldsymbol{\xi})$. HMC uses Hamiltonian dynamics to construct Markov chains and introduces a U-dimensional auxiliary momentum variable $\boldsymbol{\theta}$. HMC can only handle unconstrained variables. However, $\{\beta_q\}_{q=0}^{Q+1}$ must be positive so that we use an exponential transformation $\rho_q = \log(\beta_q)$ to make ρ_q unconstrained. We define $\boldsymbol{\tau} = \left[\boldsymbol{\lambda}^T, \rho_0, \cdots, \rho_{Q+1}\right]^T \in R^{(F+Q+2)\times 1}$ as the set of model parameters $\boldsymbol{r}_{c_f} = \left[r_{c_f}[1], r_{c_f}[2], \cdots, r_{c_f}[N]\right]^T$ as the observation of content c_f in N time slots. Based on the proposed probabilistic model and prior knowledge, we can derive the negative log

of posterior distribution:

$$\begin{aligned}\phi(\boldsymbol{\tau}) &= -\log p\left(\boldsymbol{\lambda}, \boldsymbol{\beta} \left| \boldsymbol{r}_{c_f} \right.\right) \\ &= \sum_{f=1}^{F}\sum_{n=1}^{N} -r_{c_f}[n]\lambda_f + e^{\lambda_f} + \frac{1}{2}\log\det\left(\boldsymbol{K}'\right) \\ &+ \frac{1}{2}\boldsymbol{\lambda}^T\boldsymbol{K}'^{-1}\boldsymbol{\lambda} + \sum_{q=0}^{Q+1} -A_q\rho_q + B_q e^{\rho_q}. \end{aligned} \tag{8}$$

The gradient of (8) can be computed as follows:

$$\frac{\partial\phi(\boldsymbol{\tau})}{\partial\lambda_f} = \sum_{n=1}^{N} -r_{c_f}[n] + Ne^{\lambda_f} + \left[\boldsymbol{K}'^{-1}\boldsymbol{\lambda}\right]_f, \tag{9}$$

$$\begin{aligned}\frac{\partial\phi(\boldsymbol{\tau})}{\partial\rho_q} &= \frac{1}{2}\mathrm{tr}\left(\boldsymbol{K}'^{-1}\frac{\partial\boldsymbol{K}'}{\partial\rho_{\mathrm{q}}}\right) - \frac{1}{2}\boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{K}'^{-1}\frac{\partial\boldsymbol{K}'}{\partial\rho_{\mathrm{q}}}\boldsymbol{K}'^{-1}\boldsymbol{\lambda} \\ &- A_q + B_q e^{\rho_q}. \end{aligned} \tag{10}$$

The disadvantage of HMC is that it has to traverse the whole dataset for gradient calculation, which is obviously impractical in large-scale settings. In particular, the caching system needs to update the cache contents in time so that the popularity prediction is very sensitive to computational cost. Therefore, HMC is not affordable to be applied in most scenarios. To address this problem, one intuition is to use stochastic gradients instead of computing the full gradient in HMC, which is called Stochastic Gradient Hamiltonian Monte Carlo (SGHMC):

$$\nabla\widetilde{\phi}(\boldsymbol{\tau}) = \frac{|\widetilde{r}|}{|r|}\sum_{\boldsymbol{\tau}\in\widetilde{r}}\nabla\phi(\boldsymbol{\tau}), \widetilde{r} \subset r. \tag{11}$$

The stochastic gradient introduces noise to the model. According to the central limit theorem, the noisy gradient can be approximated as

$$\nabla\tilde{\phi}(\boldsymbol{\tau}) \approx \nabla\phi(\boldsymbol{\tau}) + \mathcal{N}(0, V(\boldsymbol{\tau})), \tag{12}$$

where $V(\boldsymbol{\tau})$ is the covariance of the noise introduced from the stochastic gradient. It is related to the dimension of the parameters and sample size. The noise will decrease with increasing the size of batches. However, simply replacing $\nabla\phi(\boldsymbol{\tau})$ by unprocessed stochastic gradients $\nabla\tilde{\phi}(\boldsymbol{\tau})$ is likely to cause divergence and changing the properties of the posterior distribution, In [13], there is an alternative to overcome this problem:

$$\nabla\tilde{\phi}(\boldsymbol{\tau}) \approx \nabla\phi(\boldsymbol{\tau}) + \mathcal{N}(0, 2(\boldsymbol{C}-\boldsymbol{B})\varepsilon), \tag{13}$$

where $\boldsymbol{C}$ is the friction term, $\boldsymbol{B}$ is noise model and ε is step-size. Although this method makes the model converge faster, the matrices $\boldsymbol{C}$ and $\boldsymbol{B}$ are difficult to set without sufficient prior information. There is necessity to tune several times to obtain the appropriate parameters.

Thus, in order to reduce the variance due to stochastic gradients and also to ensure the performance of the model with little prior knowledge, the Stochastic Variance Reduced Gradient (SVRG) method was used to improve the HMC as follows:

$$\boldsymbol{\theta}_{t+1} = (1-Dh)\boldsymbol{\theta}_t - h\tilde{\nabla}_t + \sqrt{2Dh}\cdot\boldsymbol{\eta}_t, \tag{14}$$

$$\boldsymbol{\xi}_{t+1} = \boldsymbol{\xi}_t + h\boldsymbol{\theta}_{t+1}. \tag{15}$$

Algorithm 1 SVRG-HMC sampling method

parameters $S, L, b, h > 0, Dh < 1, D \geqslant 1$
initialize $\boldsymbol{\theta}_0 = \boldsymbol{\xi}_0 = \mathbf{0}$
for $s = 0, 1, \ldots, S/L - 1$ **do**
  compute $g = \sum_{i=1}^{n}\nabla\phi_i(\boldsymbol{\xi}_{sl})$
  for $l = 0, 1, \ldots, L-1$ **do**
    uniformly sample an index subset $I \subseteq \mathcal{R}, |I| = b$
    $\tilde{\nabla}_{sL+l} = -\nabla\log p\left(\boldsymbol{\xi}_{sL+l}\right) +$
    $\frac{F}{b}\sum_{i\in I}\left(\nabla\phi_i\left(\boldsymbol{\xi}_{sL+l}\right) - \nabla\phi_i\left(\boldsymbol{\xi}_{sL}\right)\right) + g$
    $\boldsymbol{\theta}_{sL+l+1} = (1-Dh)\boldsymbol{\theta}_{sL+l} - h\tilde{\nabla}_{sL+l} + \sqrt{2Dh}\cdot$
    $\boldsymbol{\eta}_{sL+l}$
    $\boldsymbol{\xi}_{sL+l+1} = \boldsymbol{\xi}_{sL+l} + h\boldsymbol{\theta}_{sL+l+1}$
  end for
  $\boldsymbol{\tau}_s = \boldsymbol{\xi}_{sL+L}$
end for
Return $\{\boldsymbol{\tau}_s\}_{1\leqslant s\leqslant S}$;

The procedure of SVRG-HMC is outlined in Alg. 1. S is the sampling rounds, b is the size of minibatch, L is the number of discretization steps, h is step-size and D is a constant.

According to the Alg. 1, we can collect enough samples to approximate the moment functions of posterior distribution. As the number of sampling rounds increases, the collected samples will be closer to the true distribution that generates.

D. Prediction Phase

Here, we need to demonstrate prediction for two scenarios. Assume that N time slots have been observed. The first one is to predict the numbers of request for the contents in library, where we can derive the distribution of new requests:

$$p\left(\boldsymbol{r}_c[N+1]\left|\mathcal{R}\right.\right) = \int p\left(\boldsymbol{r}_c[N+1]\left|\boldsymbol{\lambda}\right.\right)p\left(\boldsymbol{\lambda}\left|\mathcal{R}\right.\right)\mathrm{d}\boldsymbol{\lambda}. \tag{16}$$

According to our proposed model, $p\left(\boldsymbol{r}_c[N+1]\left|\boldsymbol{\lambda}\right.\right)$ is a Poisson distribution and $p\left(\boldsymbol{\lambda}\left|\mathcal{R}\right.\right)$ is a marginal likelihood function of $\boldsymbol{\lambda}$ based on request observations $\mathcal{R}$. Nevertheless, it is often intractable to derive the integral operation. Instead, we make a point prediction rather than estimating the entire posterior distribution. We define a quadratic loss function to measure the loss from the predicted and actual values. By minimizing the loss function, we can obtain the mean of the predictive distribution and approximate it as follows:

$$E\left\{\boldsymbol{r}_c[N+1]\left|\mathcal{R}\right.\right\} \approx \frac{1}{S}\sum_{s=1}^{S} e^{\boldsymbol{\lambda}^{(s)}}. \tag{17}$$

In addition to predicting the requests of existing contents, we also need to predict the request of newly-added content.

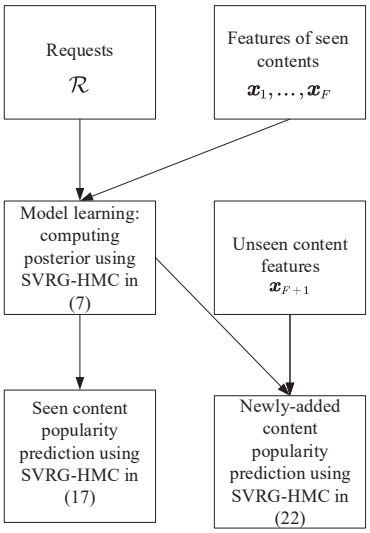

Fig. 3: Procedure of predicting content popularity

The posterior distribution in the second scenario is defined as follows:

$$p\left(\lambda_{F+1}\left|\boldsymbol{x}_{F+1}\right.\right) = \int p\left(\lambda_{F+1}\left|\boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{x}_{F+1}\right.\right) p\left(\boldsymbol{\lambda}, \boldsymbol{\beta}\left|\mathcal{R}\right.\right) \mathrm{d}\boldsymbol{\beta}\mathrm{d}\boldsymbol{\lambda}, \tag{18}$$

where $\boldsymbol{x}_{F+1}$ represents the vector of the newly-added content. To compute $p\left(\lambda_{F+1}\left|\boldsymbol{x}_{F+1}\right.\right)$, we note that the joint distribution of $p\left(\lambda_1, \lambda_2, \ldots, \lambda_{F+1}\right)$ is a Normal distribution with zero mean and covariance matrix:

$$\begin{bmatrix} \boldsymbol{K}' & \boldsymbol{k}' \\ \boldsymbol{k}'^T & K\left(\boldsymbol{x}_{F+1}, \boldsymbol{x}_{F+1}\right) \end{bmatrix}, \tag{19}$$

where $\boldsymbol{k}' = \left[K\left(\boldsymbol{x}_1, \boldsymbol{x}_{F+1}\right), \ldots, K\left(\boldsymbol{x}_F, \boldsymbol{x}_{F+1}\right)\right]^T$, According to the property of the Gaussian distribution, the conditional distribution $p\left(\lambda_{F+1}\left|\boldsymbol{\lambda}, \boldsymbol{x}_{F+1}, \boldsymbol{\beta}\right.\right)$ is a Normal distribution with mean and variance:

$$\lambda_{F+1} = \boldsymbol{k}'^T \boldsymbol{K}'^{-1} \boldsymbol{\lambda}, \tag{20}$$

$$\sigma_{F+1} = K\left(\boldsymbol{x}_{F+1}, \boldsymbol{x}_{F+1}\right) - \boldsymbol{k}'^T \boldsymbol{K}'^{-1} \boldsymbol{k}'. \tag{21}$$

Similarly, the point estimation of the request rate for newly-added content can be approximated as:

$$E\left(r_{F+1}\left[N+1\right]\left|\boldsymbol{x}_{F+1}\right.\right) \approx \frac{1}{S} \sum_{s=1}^{S} e^{\lambda_{F+1}^{(S)} + \frac{1}{2}\sigma_{F+1}^{(S)}}. \tag{22}$$

According to the above descriptions, the procedure of our popularity prediction policy can be summarized as Fig. 3. In the model learning phase, the model parameters are learned based on the number of requests and the feature of contents which is stored in the cloud server (we refer it as seen contents). In the prediction phase, the proposed policy enables to predict the popularities of both seen and newly-added contents.

IV. Simulation Results

To evaluate the performance of our proposed popularity prediction scheme, we consider movie content as an example and perform numerical simulations based on the data extracted from the MovieLens 100K Dataset [14]. The data set consists of 100,000 ratings (1-5) from 943 users on 1682 movies. Each data set contains the information about the movies include a rating, categories and a timestamp. We take the ratings as the number of requests for the movies. We perform simulations on the RMSE versus the number of observed time slots. Finally, we compare their convergence time.

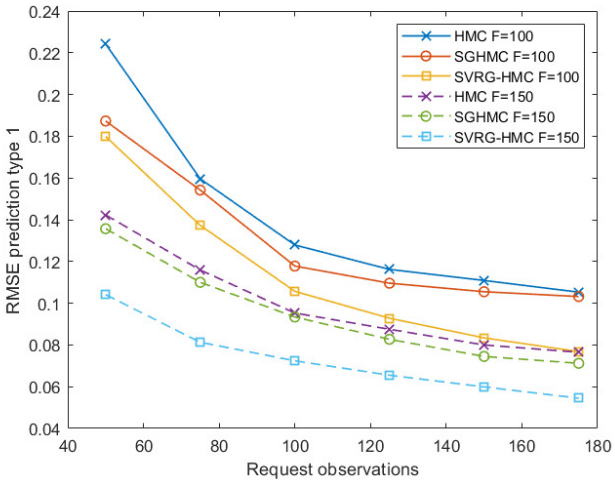

Fig. 4: RMSE prediction type 1 versus request observations

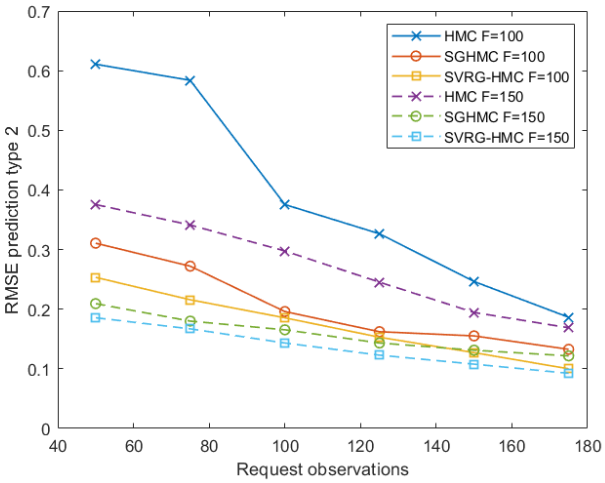

Fig. 5: RMSE prediction type 2 versus request observations

In Fig. 4, we show the RMSE of our proposed popularity prediction policy and the HMC based policy at prediction type 1 in (18). It can be observed that the RMSE of both the proposed policy and the other HMC based policies decreases as the number of requested observations increases. The reason is that the Gaussian process in the probabilistic model allows

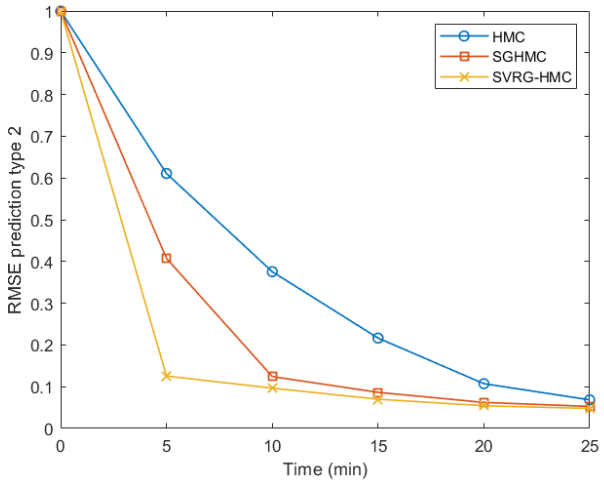

Fig. 6: RMSE versus running time

for a better insight into the relationship between the number of requests and the content features by obtaining more observation samples. It can also be observed that the RMSE of the proposed policy is significantly smaller than that of the policy based on the HMC or SGHMC sampling method. Besides, we change F, the number of contents in library to 150, the RMSE of the prediction is reduced when the contents in the library increase.

In Fig. 5, we show the RMSE of our proposed popularity prediction policy and other HMC based policies at prediction type 2 in (22). Similar to Fig. 4, the prediction performance of every method will be enhanced when F or N is increased. When N increases to a sufficient value, continuing increasing N has limited improvement to the prediction accuracy. The reason is that the Bayesian paradigm is robust to overfitting and can still perform well with less training data. Also it can be observed that SVRG-HMC outperforms HMC and SGHMC for either $F = 100$ or $F = 150$.

In Fig. 6, we show the RMSE of both method versus the running time. In other words, we can decide the running time by controlling the number of samples, and the more the number of samples we get, the closer the obtained samples are to the true posterior distribution. Meanwhile, it can be observed that SVRG-HMC takes less time to converge to the same accuracy, and reaches a higher accuracy in a shorter time. The reason is that SVRG-HMC utilize variance reduction to accelerate convergence and the application of stochastic gradient decreases the computational afford.

V. Conclusions

In this paper, we have proposed a popularity prediction policy based on Bayesian learning in F-RANs. By using the number of requests and content feature, our proposed policy enables prediction of content popularity with lower computational complexity and high accuracy even for newly-added content. The reason is that the relation between content feature and popularity is considered. Specifically, SVRG-HMC is more efficient than HMC method, which makes our proposed policy more pratical. Simulation results have shown that our proposed SVRG-HMC based policy outperforms the HMC and simple SGHMC based policy in terms of content popularity prediction.

Acknowledgment

This work was supported in part by the Natural Science Foundation of China under grant 61971129, the Natural Science Foundation of Jiangsu Province under grant BK20181264, the Shenzhen Science and Technology Program under Grant KQTD20190929172545139 and JCYJ20180306171815699, and the National Major Research and Development Program of China under Grant 2020YFB1805005.

References

[1] Y. Liu, F. R. Yu, X. Li, H. Ji, and V. C. M. Leung, "Distributed resource allocation and computation offloading in fog and cloud networks with non-orthogonal multiple access," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 12, pp. 12 137–12 151, 2018.

[2] S. M. S. Tanzil, W. Hoiles, and V. Krishnamurthy, "Adaptive scheme for caching youtube content in a cellular network: Machine learning approach," *IEEE Access*, vol. 5, pp. 5870–5881, 2017.

[3] H. Ahlehagh and S. Dey, "Video caching in radio access network: Impact on delay and capacity," in *2012 IEEE Wireless Communications and Networking Conference (WCNC)*, 2012, pp. 2276–2281.

[4] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C. M. Leung, "Cache in the air: Exploiting content caching and delivery techniques for 5G systems," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 131–139, 2014.

[5] Y. Jiang, M. Ma, M. Bennis, F. Zheng, and X. You, "User preference learning-based edge caching for fog radio access network," *IEEE Transactions on Communications*, vol. 67, no. 2, pp. 1268–1283, 2019.

[6] Y. Jiang, H. Feng, F. C. Zheng, D. Niyato, and X. You, "Deep learning-based edge caching in fog radio access networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 12, pp. 8442–8454, 2020.

[7] Y. Wu, Y. Jiang, M. Bennis, F. Zheng, X. Gao, and X. You, "Content popularity prediction in fog radio access networks: A federated learning based approach," in *2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.

[8] S. Mehrizi, A. Tsakmalis, S. Chatzinotas, and B. Ottersten, "A feature-based bayesian method for content popularity prediction in edge-caching networks," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, 2019, pp. 1–6.

[9] A. Sadeghi, F. Sheikholeslami, and G. B. Giannakis, "Optimal and scalable caching for 5G using reinforcement learning of space-time popularities," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 180–190, 2018.

[10] A. Asheralieva and D. Niyato, "Bayesian reinforcement learning and bayesian deep learning for blockchains with mobile edge computing," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 1, pp. 319–335, 2021.

[11] Y. Jiang, X. Chen, F. C. Zheng, D. Niyato, and X. You, "Brain storm optimization-based edge caching in fog radio access networks," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 2, pp. 1807–1820, 2021.

[12] C. E. Rasmussen, "Gaussian processes in machine learning," in *Advanced lectures on machine learning*. Springer, 2004, pp. 63–71.

[13] T. Chen, E. B. Fox, and C. Guestrin, "Stochastic gradient hamiltonian monte carlo," in *2014 International Conference on Machine Learning (ICML)*, 2014, pp. 1–6.

[14] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *ACM Trans. Interactive Intell. Syst.*, vol. 5, no. 4, pp. 1–19, 2015.