Popularity Tracking for Proactive Content Caching with Dynamic Factor Analysis

Sajad Mehrizi*, Anestis Tsakmalis*, Shahram ShahbazPanahi*†, Symeon Chatzinotas* and Björn Ottersten*
*Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg
†University of Ontario Institute of Technology

***Abstract*—The performance of all proactive edge-caching policies is largely influenced by the content popularity prediction algorithm, which in turn relies on having a good understanding of the underlying request process. However, due to the non-deterministic and time-varying nature of popularities, prediction is not a trivial task. In this work, we suggest a probabilistic dynamic factor analysis model to describe content requests for real-world time-varying scenarios. The dynamic factor analyzer is a flexible model to capture common patterns among content requests through temporal stochastic processes, which lay in a low-dimensional latent space. Modeling these common dynamic patterns provides a better accuracy for tracking and predicting the evolution of content popularities. Model learning is performed from the Bayesian aspect which provides a systematic mechanism to incorporate uncertainty for robust prediction. Due to the model complexity, we derive a simple approximation method based on variational Bayes (VB) to infer the model parameters. Finally, we show simulation results where the proposed popularity prediction method outperforms the traditional ones available in the literature on a real-world dataset.**

***Index Terms*—Popularity prediction, Proactive Caching, Dynamic Scenario, Bayesian Learning**

I. Introduction

Due to the emergence of communities with a massive number of users, the demands for content (e.g., video) are explosively growing [1]. This explosive growth is challenging the capability of current network architectures in satisfying users' demands with acceptable quality of service. A promising approach to mitigate this challenge is to offload network traffic by caching *popular contents* at the network edge [2]. The motivation behind caching is that typically the majority of data traffic is caused by only a small number of popular contents, as indicated by the Zipf-law content behavior [3]. Caching these highly popular contents at the edge bypasses the need for fetching these contents from the content provider through the backhaul links for every request. Therefore, this reduces latency and downloading time.

A central problem in caching systems is content placement i.e., selecting which proper contents to cache. Several algorithms that select contents for caching or ejection have been proposed [4]. For example, least recently used (LRU) policy keeps a record of the recent time request for each content, and when there is not enough space it replaces contents with the longest idle time with newly requested ones. Likewise, the least frequently used (LFU) approach ejects the least frequently used contents. These traditional policies and their variants are widely used in practice due to their simplicity. Nevertheless, they disregard patterns in content requests, and therefore, do not have a satisfactory performance. The limitations in cache storage and backhaul bandwidth highlight the need for efficient proactive caching policies with the ability to predict the popularity of contents.

While significant volume of results have been published on edge-caching, there exist only a few for popularity prediction. The authors of [5] model the caching policy as a multi-armed-bandit problem and the content popularity is learned online. In [6], the required training time is derived to obtain an acceptable prediction accuracy and to improve it transfer learning is proposed. The authors of [7] used neural networks for the prediction task. In [8], popularity prediction is modeled as a matrix completion problem where the missing entries are estimated by a matrix factorization technique. A binary logistic classification method was introduced in [9] to classify user interests based on content features. In [10], content features are used to improve the prediction accuracy within the Bayesian framework. The authors of [11] introduce a Bayesian factor analysis model to model the correlations between contents. A main assumption of the aforementioned studies is that the popularity does not change over time or it changes very slowly. Nevertheless, in practice, content popularity is far from stationary and might change even within a few hours e.g. for viral contents.

To tackle the time-varying nature of content popularity, several time-series analysis methods have been used in the literature to model the view count dynamics of videos. As one of the most popular models for time series, auto-regressive moving-average (ARMA) model is utilized in [12], [13] for video popularity prediction. However, ARMA modeling suffers from three important weaknesses. Firstly, this model can only handle continuous data, which is not the case for content request, secondly the assumed model is linear which is quite restrictive. Also, it ignores the correlation among contents. More specifically, in reality, requests for some contents may be highly correlated and exhibit similar patterns. For example, some contents may have the same features or are interesting for the same user community. Modeling the discrete nature of and the correlation among requests can provide much more accuracy, and as a result, a better caching performance.

In this paper, we extend the factor analysis introduced in [11] to a time-varying scenario. More precisely, in factor analysis the correlation is modeled by some hidden random variables which are shared among all contents. In the stationary scenario, the assumption is that these random variables are independently and identically distributed. For the dynamic scenario, we assume that they follow a first order Markov process and their evolutions can explain the evolution of content popularities over time. We then perform model fitting within the Bayesian framework. Since the local caches may not ob-

serve enough requests [14], traditional estimation techniques, e.g. Maximum Likelihood Estimator (MLE), may not provide a satisfactory performance and also suffer from overfitting. On the other hand, Bayesian methods have the capability to mitigate the overfitting problem efficiently. Additionally, a fast and scalable VB technique is developed to learn this Bayesian model. Finally, in the simulation results, we show the accuracy of the VB method and the caching performance on a real-world dataset.

The rest of the paper is organized as follows. In Section II, we describe the system model and present a proactive caching policy at base station (BS) of cellular networks. The probabilistic dynamic factor analyzer and its inference are described in Section III. In Section IV, we present the simulation results and conclude the paper in Section V .

II. System model

In this work, we focus on content request prediction in a single region in a cellular network within the service area of a central BS. We assume that the time axis is in divided non-overlapping intervals with equal durations of, for example, a day or a week. At time interval t, $t = 1, 2, ..., T$, each user in the cell requests a content from the library content $\mathcal{C} \triangleq \{c_1, ..., c_M\}$ with M contents, where the size of content c_m is denoted as s_m, for $m = 1, 2, \ldots, M$. The BS is equipped with a cache of C_1 memory units which stores a subset of the content library $\mathcal{C}$. At time t, upon receiving the user requests, those contents that are already stored in the cache, will be directly sent to the requesting users and those contents that are not available in the cache will be fetched from the content provider via a back-haul link.

At the end of time interval T, the task is to predict the requests in the next time slot $\tau = T+1$ given the past request history $\mathcal{D}_T = \{\mathbf{d}_t\}_{t=1}^T$, where $\mathbf{d}_t$ is the $M \times 1$ request vector whose m-th element, denoted as d_{mt}, represents the total number of requests from all users for content c_m during time interval t. After the prediction, suitable contents are stored in the cache depending on the caching policy. At time t, a caching policy is represented by the $M \times 1$ vector $\mathbf{p}_t$ of only zeros and ones. If the m-th element of $\mathbf{p}_t$ is 1, then c_m is stored in the cache, otherwise the cache does not contain c_m.

Here, we design the caching policy by solving the following optimization problem:

$$\max_{\mathbf{p}_\tau} \ \mathbf{p}_\tau^T \hat{\mathbf{d}}_\tau \tag{1a}$$

$$\text{subject to} \ \ \mathbf{p}_\tau^T \mathbf{s} \leq C_1 \tag{1b}$$

$$\mathbf{p}_\tau^T \left(\mathbf{s} \odot (\mathbf{1} - \mathbf{p}_T)\right) \leq C_2 \tag{1c}$$

$$\mathbf{p}_\tau \in \{0,1\}^{M \times 1} \tag{1d}$$

where we define $\mathbf{s} \triangleq [s_1 \ \ s_2 \ \ ... \ \ s_M]^T$, and $\hat{\mathbf{d}}_\tau = E\{\mathbf{d}_\tau | \mathcal{D}_T\}$ is the conditional mean estimate of the request vector at time τ and $\mathbf{p}_\tau$ is the decision variable for updating the cache at the beginning of time interval τ. The objective function is the average cache hit ratio at time slot τ. Constraint (1b) denotes the cache size limitation as $\mathbf{p}_\tau^T \mathbf{s}$ measures number of the memory units required by policy $\mathbf{p}_\tau$. Constraint (1c) ensures the total number of data units required to fetch new contents from the back-haul link does not exceed C_2 and constraint (1d) denotes the space restriction for $\mathbf{p}_\tau$. Solving problem (1) requires the knowledge of $\hat{\mathbf{d}}_\tau$, which is unknown and has to predicted. To obtain $\hat{\mathbf{d}}_\tau$, we specifically focus on Bayesian modeling which consists of three steps. First, by making some assumptions about the hidden patterns that govern the content requests a probabilistic is built. Second, in the inference, the posterior distribution of the hidden patterns given the requests is computed. Third, we compute the posterior predictive distribution to acquire $\hat{\mathbf{d}}_\tau$. These three steps are the subject of the next section.

III. Probabilistic demand model

In this section, we introduce a probabilistic model to accurately capture the evolution of the content requests over time. In practice, the content requests often exhibit similar patterns leading to the belief that they might be driven by common factors. Similar to [11], in order to capture the correlation among the requests for different contents, we assume the content request at time t is a Poisson stochastic process:

$$d_{mt} \sim \text{Pois}\left(e^{\mathbf{w}_m^T \mathbf{x}_t}\right), \quad t = 1, ..., T. \tag{2}$$

From (2), the request rate (the popularity) for content m at time t, r_{mt}, is defined by $e^{\mathbf{w}_m^T \mathbf{x}_t}$, where $\mathbf{x}_t \in \mathbb{R}^{K \times 1}$, with $K \ll M$, is a low-dimensional vector of hidden factors that are common among all contents and $\mathbf{w}_m$ is called the factor loading for content m . The assumption in (2) is that the natural parameter of the Poisson distributed demands for each content is a linear combination of the low dimensional vector $\mathbf{x}_t$ of hidden factors with weights being the element of $\mathbf{w}_m$. We mention the factor loading matrix as $\mathbf{W} \triangleq [\mathbf{w}_1 \ \ \mathbf{w}_2 \ \ ... \ \ \mathbf{w}_M]^T$. The hidden factors which are shared among all contents model their correlations. To model the time-varying nature of the content popularities, we assume that $\mathbf{x}_t$ follows a first order Markov process:

$$\mathbf{x}_t = a\mathbf{x}_{t-1} + \mathbf{e}_t, \quad t = 1, ..., T \tag{3}$$

starting from initial state $\mathbf{x}_0$, where a is a known constant and $\mathbf{e}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$. We assume $|a| < 1$ to ensure the covariance of $\mathbf{x}_t$ is norm-bounded, otherwise for $|a| > 1$, the norm of this matrix becomes unbounded as time progresses [15]. In (2) and (3), time dependencies in the content requests are materialized by allowing $\mathbf{x}_t$ be a correlated with $\mathbf{x}_{t-1}$. Therefore, requests at time t are usually more similar to requests at time $t-1$ than any other previous values. We should also note that (2) and (3) form a state-space model where (2) is the observation model and (3) is the unobserved state model.

Additionally, the covariance matrix $\mathbf{Q}$ can be absorbed into the factor loading matrix without affecting the popularity evolution. More specifically, the natural parameter of the Poisson process d_{mt}, denoted as y_{mt} is also a stochastic process and can be expressed as $y_{mt} = \mathbf{w}_m^T \left(\sum_{i=1}^{t} a^i \mathbf{e}_i + \mathbf{x}_0\right)$ with $E(y_{mt}) = \mathbf{w}_m^T \mathbf{x}_0$ and $\text{var}(y_{mt}) = \mathbf{w}_m^T \mathbf{Q} \mathbf{w}_m \sum_{i=1}^{t} a^{2i}$. This implies that we can define a new factor loading $\tilde{\mathbf{w}}_m =$

$\mathbf{Q}^{1/2}\mathbf{w}_m$ and a new initial state $\tilde{\mathbf{x}}_0 = \mathbf{Q}^{-1/2}\mathbf{x}_0$ without changing process y_{mt}. Therefore, without loss of generality, we can assume $\mathbf{Q} = \mathbf{I}$.

A. Inference

Here, our goal is to fit the model (2)-(3) to the content requests from a Bayesian perspective. To do so, we set up prior distributions for the unknown parameters, i.e., for the factor loading matrix $\mathbf{W}$ in (2) and the initial value $\mathbf{x}_0$ in (3). For simplicity, the columns of factor loading matrix are assumed to be:

$$\mathbf{w}_m \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}), \quad \forall m = 1, .., M \tag{4}$$

where $\mathbf{\Sigma} = \text{diag}\left(\sigma_1^2, ..., \sigma_{\text{K}}^2\right)$ with σ_k^2 denotes the variance of column k of factor loading matrix $\mathbf{W}$. The value of variance σ_k^2 signifies the importance of the k-th column of $\mathbf{W}$. When σ_k^2 goes to zero, it means that the corresponding column of $\mathbf{W}$ is irrelevant and can be pruned out. Since importance of different columns of $\mathbf{W}$ is unknown, we assign a sparsity promoting prior to σ_k^2. Doing so, the importance of different columns of $\mathbf{W}$ can be automatically determined. This framework is known as automatic relevance determination (ARD) technique [16]. Similar to [17], we use the heavy-tailed half-Cauchy distribution with scale parameter A as a prior for the variance of each column $\sigma_k \sim \mathcal{C}^+(0, A)$. The half-Cauchy distribution can also be represented as a mixture of inverse Gamma's as [18]:

$$\sigma_k^2 \sim \mathcal{IG}(\frac{1}{2}, \theta_k^{-1}), \;\; \theta_k \sim \;\; \mathcal{IG}(\frac{1}{2}, \frac{1}{A^2}). \tag{5}$$

We will use the representation in (5) since it facilitates the mathematical operations. Moreover, we assume that the prior distribution for the initial state $\mathbf{x_0}$ is $p(\mathbf{x}_0) = \mathcal{N}(\mathbf{0}, \mathbf{Q}_0)$. The graphical representation of the Bayesian model is shown in Fig. 1.

The inference problem is then to compute the posterior distribution of all unknown quantities given by the Bayes rule which is

$$\begin{aligned} p(\mathbf{h}|\mathcal{D}_T) \propto p(\mathbf{h}, \mathcal{D}_T) = \\ \prod_{t=1}^{T}\prod_{m=1}^{M} p(d_{m,t}|\mathbf{w}_m, \mathbf{x}_t) p(\mathbf{w}_m) p(\mathbf{x}_t|\mathbf{x}_{t-1}) p(\mathbf{x}_0) \\ \prod_{k=1}^{K} p\left(\sigma_k^2|\theta_k\right) p(\theta_k) \end{aligned} \tag{6}$$

where $\mathbf{h} = \left\{\mathbf{W}, \left\{\sigma_k^2, \theta_k\right\}_{k=1}^{K}, \left\{\mathbf{x}_t\right\}_{t=0}^{T}\right\}$. It is computationally intractable to compute (6). Therefore, we invoke an approximation method in the next subsection.

B. Posterior Approximation

In this subsection, we present a low complexity inference method based on the VB approach [16]. Variational inference performs the inference task as the problem of minimizing the Kullback-Leibler (KL) divergence between an approximate distribution and the true posterior.

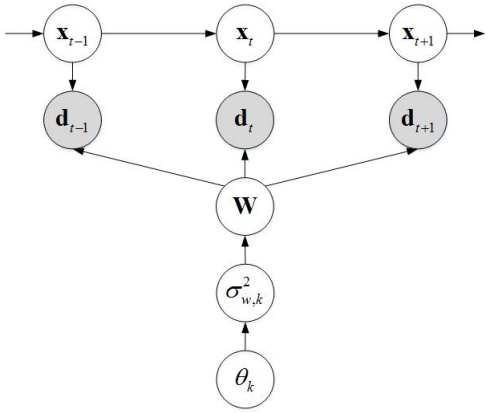

Fig. 1: The graphical representation of the probabilistic time-varying content request model.

More formally, true posterior $p(\mathbf{h}|\mathcal{D}_T)$ is approximated by a family of distributions $q(\mathbf{h})$ that minimize the KL divergence

$$\min_{q(\mathbf{h})} \quad \text{KL}\left(q(\mathbf{h}) \,||\, p(\mathbf{h}|\mathcal{D}_T)\right), \quad s.t: \int q(\mathbf{h})\, d\mathbf{h} = 1 \tag{7}$$

where $\text{KL}\left(q(\mathbf{h}) \,||\, p(\mathbf{h}|\mathcal{D}_T)\right) \triangleq E_{q(\mathbf{h})}\left\{\log\frac{q(\mathbf{h})}{p(\mathbf{h}|\mathcal{D}_T)}\right\}$. One of the most popular forms of VB is the mean field approximation. In this scheme, the assumption is that the variational distribution is factorized as:

$$q(\mathbf{h}) = \prod_i q(\mathbf{h}_i). \tag{8}$$

where $\mathbf{h}_i$ is part of $\mathbf{h}$ with $\bigcup_i \mathbf{h}_i = \mathbf{h}$ and $\bigcap_i \mathbf{h}_i = \emptyset$. With this approximation, it has been shown that problem (7) can be solved by the coordinate decent method where at each update the optimal solution for $q(\mathbf{h}_i)$ is given by [16]:

$$\log q(\mathbf{h}_i) \propto E_{\sim q(\mathbf{h}_i)}\left\{\log p(\mathbf{h}, \mathcal{D}_T)\right\} \tag{9}$$

where the notation $E_{\sim q(\mathbf{h}_i)}\{.\}$ means to take the expectation with respect to all the variables except $\mathbf{h}_i$.

Accordingly, we assume that the approximating family for the posterior in (6) has the following form:

$$p(\mathbf{h}|\mathcal{D}_T) \approx q(\mathbf{h}) = \prod_{m=1}^{M} q(\mathbf{w}_m) \prod_{t=0}^{T} q(\mathbf{x}_t) \prod_{k=1}^{K} q\left(\sigma_k^2\right) q(\theta_k) \tag{10}$$

Later on we will see that, by applying (9), we can derive closed-form densities for σ_k^2, θ_k and $\mathbf{x}_0$ automatically. However, due to non-conjugacy for $\mathbf{W}$ and $\mathbf{x}_{t>0}$ which is resulted from having Poisson and normal distributions, it is not trivial to update them. Therefore, for simplicity, we assume $q(\mathbf{w}_m)$ and $q(\mathbf{x}_{t>0})$ can be approximated by normals as:

$$q(\mathbf{w}_m) \approx \mathcal{N}\left(\boldsymbol{\mu}_{w_m}, \mathbf{\Sigma}_{w_m}\right), q(\mathbf{x}_t) \approx \mathcal{N}\left(\boldsymbol{\mu}_{x_t}, \mathbf{\Sigma}_{x_t}\right). \tag{11}$$

Then, the goal is to find suitable values for the means and the covariances that fit well to $q(\mathbf{w}_m)$ and $q(\mathbf{x}_t)$. In the following, the update rules and a simple approximation method for (11) are given.

- Updating $q\left(\sigma_k^2\right)$

The optimal form for $q\left(\sigma_k^2\right)$ is obtained by averaging over $\mathbf{W}$ and θ_k:

$$\log q\left(\sigma_k^2\right) \propto$$
$$E_{q(\mathbf{w})q(\theta_k)}\{-\frac{M+1}{2}\log\sigma_k^2 - \frac{1}{2}\sum_{m=1}^{M}\frac{w_{mk}^2}{\sigma_k^2} - \frac{1}{\sigma_k^2\theta_k}\} \quad (12)$$

One can show that $q\left(\sigma_k^2\right) = \mathcal{IG}\left(\alpha_{\sigma_k}, \beta_{\sigma_k}\right)$, where

$$\alpha_{\sigma_k} = \frac{M+1}{2}, \beta_{\sigma_k} = \frac{1}{2}\sum_{m=1}^{M} E\left\{w_{mk}^2\right\} + E\{\frac{1}{\theta_k}\} \quad (13)$$

- Updating $q\left(\theta_k\right)$

The optimal form for $q\left(\theta_k\right)$ is given by

$$q\left(\theta_k\right) \propto E_{q\left(\sigma_k^2\right)}\{-\frac{3}{2}\log\sigma_k^2 - \frac{1}{\theta_k\sigma_k^2} - \frac{1}{A^2\theta_k}\} \quad (14)$$

It can be recognized that $q\left(\theta_k\right) = \mathcal{IG}\left(\alpha_{\theta_k}, \beta_{\theta_k}\right)$ where

$$\alpha_{\theta_k} = \frac{1}{2}, \beta_{\theta_k} = \frac{1}{A^2} + E\{\frac{1}{\sigma_k^2}\} \quad (15)$$

- Updating $q\left(\mathbf{w}_m\right)$

The optimal form for $q\left(\mathbf{w}_m\right)$ is given by

$$\log q\left(\mathbf{w}_m\right) \propto \sum_{t=1}^{T} E_{q(\mathbf{x}_t)}\left\{d_{mt}\mathbf{w}_m^T\mathbf{x}_t - e^{\mathbf{w}_m^T\mathbf{x}_t}\right\} - \frac{\mathbf{w}_m^T E_{q(\mathbf{\Sigma})}\left\{\mathbf{\Sigma}^{-1}\right\}\mathbf{w}_m}{2} \quad (16)$$

which can be simplified as

$$\log q\left(\mathbf{w}_m\right) \propto \sum_{t=1}^{T} d_{mt}\mathbf{w}_m^T\boldsymbol{\mu}_{x_t} - e^{\mathbf{w}_m^T\boldsymbol{\mu}_{x_t} + \frac{1}{2}\mathbf{w}_m^T\mathbf{\Sigma}_{x_t}\mathbf{w}_m} - \frac{\mathbf{w}_m^T E_{q(\mathbf{\Sigma})}\left\{\mathbf{\Sigma}^{-1}\right\}\mathbf{w}_m}{2} \quad (17)$$

Due to the non-conjugacy which results from the Poisson and normal distributions, there is no closed form expression for (17). As an approximation for (17), we use a quadratic function around its maximum value with the help of the second order Taylor approximation i.e.

$$\log\left(q\left(\mathbf{w}_m\right)\right) \approx \log\left(q\left(\mathbf{w}_m^*\right)\right) + \frac{1}{2}\left(\mathbf{w}_m^* - \mathbf{w}_m\right)^T\nabla^2\log\left(q\left(\mathbf{w}_m^*\right)\right)\left(\mathbf{w}_m^* - \mathbf{w}_m\right) \quad (18)$$

where $\mathbf{w}_m^*$ is the maximum of (17). Hence, the mean and the covariance of $\mathbf{w}_m$ are given by $\boldsymbol{\mu}_{w_m} = \mathbf{w}_m^*$ and $\mathbf{\Sigma}_{w_m} = -\left[\nabla^2\log\left(q\left(\mathbf{w}_m^*\right)\right)\right]^{-1}$. We also note that (17) is a strongly concave function therefore the Hessian matrix is always positive definite and invertible.

- Updating $q\left(\mathbf{x}_t\right)$

The update rule is derived similarly to the one of $q\left(\mathbf{w}_m\right)$, where the optimal form:

$$\log q\left(\mathbf{x}_t\right) \propto \sum_{m=1}^{M} d_{mt}\boldsymbol{\mu}_{w_m}^T\mathbf{x}_t - e^{\boldsymbol{\mu}_{wm}^T\mathbf{x}_t + \frac{1}{2}\mathbf{x}_t^T\mathbf{\Sigma}_{w_m}\mathbf{x}_t} - \frac{2\mathbf{x}_t^T\mathbf{x}_t - 2a\mathbf{x}_t^T(\boldsymbol{\mu}_{x_{t-1}} + \boldsymbol{\mu}_{x_{t+1}})}{2} \quad (19)$$

is approximated by a normal distribution with mean $\boldsymbol{\mu}_{x_t} = \mathbf{x}_t^*$ and covariance $\mathbf{\Sigma}_{x_t} = -\left[\nabla^2\log\left(q\left(\mathbf{x}_t^*\right)\right)\right]^{-1}$ where $\mathbf{x}_t^*$ is the maximum of (19).

- Updating $q\left(\mathbf{x}_0\right)$

The optimal form for $\mathbf{x}_0$ is easily obtained by $q\left(\mathbf{x}_0\right) = \mathcal{N}\left(\boldsymbol{\mu}_{x_0}, \mathbf{\Sigma}_{x_0}\right)$ where:

$$\boldsymbol{\mu}_{x_0} = \mathbf{\Sigma}_{x_0}\boldsymbol{\mu}_{x1}, \mathbf{\Sigma}_{x_0} = \left(\mathbf{I} + \mathbf{Q}_0^{-1}\right)^{-1}. \quad (20)$$

To compute the required expectations in (13) (15) and (16), we note from (13), (15) and (11) that they are given by

$$E\{\frac{1}{\theta_k}\} = \frac{\alpha_{\theta_k}}{\beta_{\theta_k}}, E\{\frac{1}{\sigma_k^2}\} = \frac{\alpha_{\sigma_k}}{\beta_{\sigma_k}} \quad (21)$$
$$E\left\{w_{mk}^2\right\} = \mu_{mk}^2 + \left[\mathbf{\Sigma}_{w_m}\right]_{kk} \quad (22)$$
$$E_{q(\mathbf{\Sigma})}\left\{\mathbf{\Sigma}^{-1}\right\} = \mathrm{diag}(\frac{\alpha_{\sigma_k}}{\beta_{\sigma_k}}, ..., \frac{\alpha_{\sigma_k}}{\beta_{\sigma_k}}) \quad (23)$$

To implement the coordinate descent method, we initialize the parameters of the variational distributions and iteratively updating them, according to their derivations as in the above, until convergence. Moreover, in order to compute the maximum values of (17) and (19), since they are strongly concave functions, we use Newton's method to speed up the convergence rate.

C. Prediction

Once the model is fit, the goal is to predict the requests at future time slot $\tau = T+1$. This can be achieved by computing the posterior predictive distribution:

$$p\left(\mathbf{d}_\tau|D_T\right) = \int\int\prod_{m=1}^{M} p\left(d_{m\tau}|\mathbf{w}_m, \mathbf{x}_\tau\right) p\left(\mathbf{x}_\tau|\mathbf{x}_T\right) p\left(\mathbf{W}, \mathbf{x}_T|D_T\right) d\mathbf{W}d\mathbf{x}_T d\mathbf{x}_\tau \quad (24)$$

where $p\left(\mathbf{W}, \mathbf{x}_T|\mathcal{D}_T\right)$ is the marginal posterior of $\mathbf{W}$ and $\mathbf{x}_T$ which was approximated by $\prod_{m=1}^{M} q\left(\mathbf{w}_m\right) q\left(\mathbf{x}_T\right)$ in (10). Nevertheless, (24) is a multidimensional integral and difficult to compute. However, using the mean square error criterion as a risk function, the optimal point estimate is given by the conditional expectation of $\hat{\mathbf{d}}_\tau = E\left\{\mathbf{d}_\tau|\mathcal{D}_T\right\}$. To compute it, we apply the law of total probability for expectation and obtain:

$$\hat{d}_{m\tau} = \frac{e^{\frac{-1}{2}\mu_{w_m}\mathbf{\Sigma}_{w_m}^{-1}\mu_{w_m}}}{\sqrt{\left(2\pi\right)^K\left|\mathbf{\Sigma}_{w_m}\right|}} \int e^{\frac{-1}{2}\mathbf{w}_m^T\left(\mathbf{\Sigma}_{w_m}^{-1} - \mathbf{\Sigma}_{x_\tau}\right)\mathbf{w}_m + \mathbf{w}_m^T\left(\mu_{x_\tau} + \mathbf{\Sigma}_{w_m}^{-1}\mu_{w_m}\right)} d\mathbf{w}_m \quad (25)$$

where

$$\boldsymbol{\mu}_{x_\tau} = a\boldsymbol{\mu}_{x_T}, \mathbf{\Sigma}_{x_\tau} = a^2\mathbf{\Sigma}_{x_T} + \mathbf{I}. \quad (26)$$

It can be seen that (25) is a multidimensional Gaussian integral [19] and can be computed as $\hat{d}_{m\tau} = \frac{1}{\left|\mathbf{I} - \mathbf{\Sigma}_{w_m}\mathbf{\Sigma}_{x_\tau}\right|^{1/2}} e^{g_{m\tau}}$ where

$$g_{m\tau} = \frac{-1}{2}\boldsymbol{\mu}_{w_m}\mathbf{\Sigma}_{w_m}^{-1}\boldsymbol{\mu}_{w_m} + \frac{1}{2}\left(\boldsymbol{\mu}_{x_\tau} + \mathbf{\Sigma}_{w_m}^{-1}\boldsymbol{\mu}_{w_m}\right)^T\left(\mathbf{\Sigma}_{w_m}^{-1} - \mathbf{\Sigma}_{x_\tau}\right)^{-1}\left(\boldsymbol{\mu}_{x_\tau} + \mathbf{\Sigma}_{w_m}^{-1}\boldsymbol{\mu}_{w_m}\right).$$

This conditional mean prediction can be used for the caching policy defined in (1).

IV. Simulation Results

In this section, we test the performance of the proposed probabilistic dynamic model on synthetic and real-world data. The experiments on synthetic data are performed to show the accuracy of the VB method used to approximate the model parameters. For the real-world dataset, we investigate the caching performance of the model since parameter estimation accuracy cannot be assessed. The parameters A, $\mathbf{Q}_0$ and a are set to 1, $10\mathbf{I}$ and 0.99 respectively and they are common for all simulations.

A. Synthetic data

To generate the synthetic data, model (2)-(3) is used where the number of contents M is set to 100. Moreover, $\mathbf{Q}$ is chosen to be diagonal, $q\mathbf{I}$, and the number of latent dimensions K is 10, where relevant and irrelevant dimensions, K_r and K_{ir} respectively, have to sum to K. Simulation results are presented for different q values and K_r-K_{ir} configurations. The factor loadings of the relevant dimensions are independently drawn from a normal distribution with zero mean and variance 0.05 and of the irrelevant dimensions are set to zero. Additionally, the initial values for the hidden factors are generated from $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, 10\mathbf{I})$. In total, 20 synthetic datasets are utilized.

Fig. 2 shows the model parameter estimation accuracy versus the number of time slots, i.e. the number of observed request vectors, for different state noise variances q. The accuracy is measured by the error metric $\sqrt{\frac{\sum_{t=1}^{T}\|\mathbf{r}_t-\hat{\mathbf{r}}_t\|^2}{\sum_{t=1}^{T}\|\mathbf{r}_t\|^2}}$, where $\hat{\mathbf{r}}_t$ is the inferred mean of requests at time i.e. $\hat{r}_{mt} = E\left\{e^{\mathbf{w}_m^T\mathbf{x}_t}|\mathcal{D}_T\right\}$ and $r_{mt} = e^{\mathbf{w}_m^T\mathbf{x}_t}$ is the true value. In this simulation part, we assume that $K_r = 2$. This figure shows that as T increases the accuracy gets better and that as the noise variance increases the estimation error increases as well.

Now, we investigate the learning accuracy for detecting the relevant hidden state dimensions, K_r. In this simulation part, we set $q = 0.5$. Fig. 3 shows the average number of the inferred relevant dimensions, $\hat{K}_r$, over all datasets for $K_r = 2$ and $K_r = 5$. This figure shows that as T increases the proposed learning method converges to the true K_r values. Furthermore, we note that when T is small usually the number of inferred dimensions is less than that the true ones. This means that for small number of observations only a few factors will be enough to explain the request pattern.

For the same simulation parameter configuration and only for 1 dataset, we depict in Fig. 4 the trajectories of the true and the estimated popularities for two different contents along with their observed requests. It can be seen that our proposed VB method can accurately estimate the true underlying popularity trajectory.

B. Real data

In this subsection, we examine the effect of the prediction accuracy of the suggested dynamic model on the caching policy defined in (1). We consider the MovieLens 20M as a real-world dataset example [20]. From this, we choose ratings over 6 years, from 1997 to 2002, where we select the most popular $M = 5000$ movies. Similar to [11] and [5], a movie's rate is considered as one request for this movie. Since the dataset does not contain the movie sizes, the elements of $\mathbf{s}$ are uniformly randomly generated within $(0, 100)$. Throughout the simulations, cache capacity C_1 and back-haul constraint constant C_2 are chosen as percentages of the sum of content sizes. Furthermore, the length of the time slots is considered as two weeks.

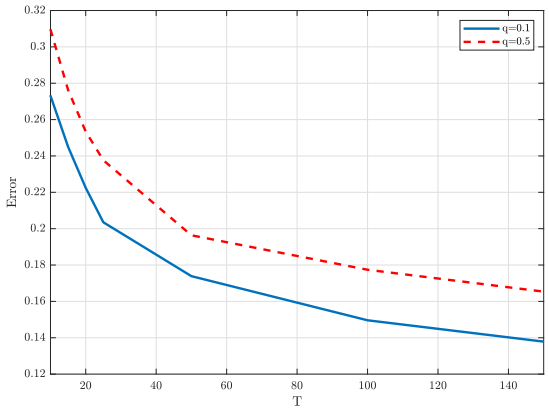

Fig. 2: Parameter estimation error vs number of observations

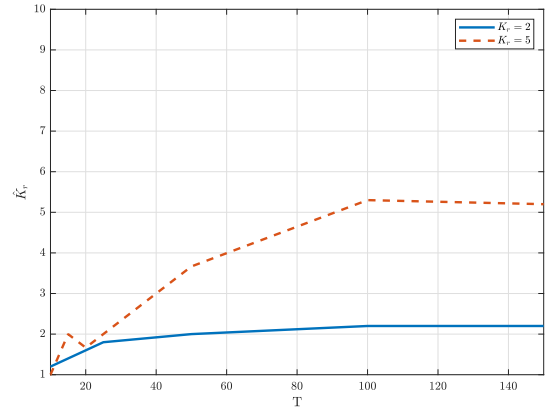

Fig. 3: Average number of the inferred $\hat{K}_r$

The following popularity estimation methods are compared:

- MLE-All: the popularity of a content is estimated by

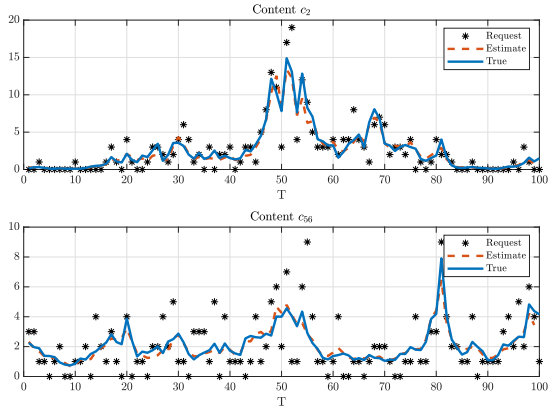

Fig. 4: Requests, true and estimated popularity trajectories for 2 contents

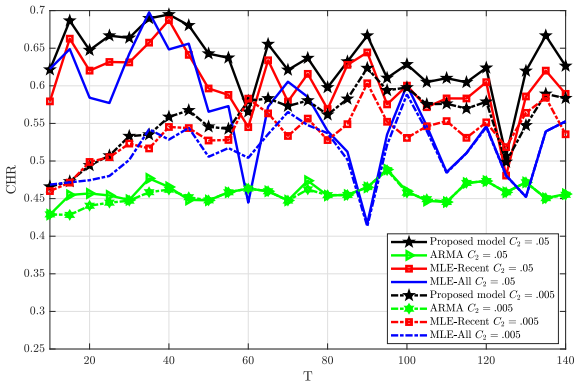

Fig. 5: CHR over time for all estimation methods

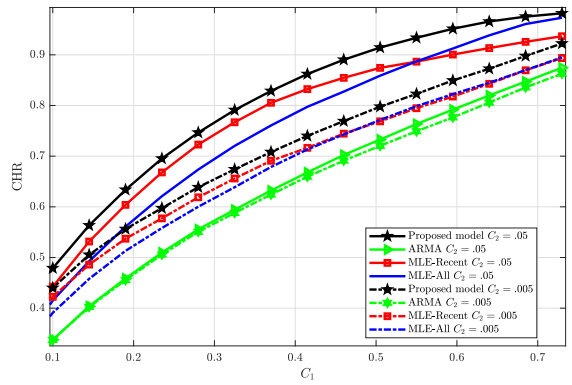

Fig. 6: The average CHR over time vs cache capacity C_1

taking the average of all the historical observations.

- MLE-Recent: the popularity is the request frequency during the last time slot.
- ARMA: requests are modeled as $d_{mt} = \sum_{i=1}^{I} \varphi_i d_{m,t-i} + \sum_{j=1}^{J} \alpha_j n_{m,t-j}$ for $t = 1, ..., T$ and $m = 1, ..., M$, where I and J specify the order of the model, φ_i and α_j are the parameters, and $n_{m,t}$ is white noise error term. We set $I = J = 7$ which is also used in [12]. For its implementation, we used the econometrics MATLAB toolbox.

Fig. 5 illustrates the CHR variation over time for all prediction approaches and for different back-haul costs. For these simulations, we set $C_1 = 0.2$. In this figure, we can see that the our dynamic model outperforms other baselines especially when C_2 is large. The figure shows that as C_2 decreases the CHR decreases as well. This is expected, since as C_2 decreases the cache has more limitation to update its contents. Hence, some contents whose popularities are outdated will be kept in the cache which results in CHR reduction. In addition, we can see that the ARMA performs poorly with respect to the other approaches.

In Fig. 6, we show the average over time CHR versus the cache capacity. We see that our model performs better compared to the other methods for all C_1 values. Moreover, it can be observed that, for a fixed C_2, as C_1 increases the CHR also increases. Therefore, in order to compensate the CHR reduction due to the back-haul limitation one way is to increase the cache capacity.

V. Conclusion

In this paper, we introduced a probabilistic model for content popularity prediction in time-varying scenarios. The model can capture the correlations among contents and therefore can improve the prediction accuracy. Further, we provided a fully Bayesian approach for accurate prediction under uncertainty. To learn the model, we developed a VB based approximation method which is scalable for large datasets. In the simulation results, we showed that the VB approximation can learn the model parameters efficiently. We also investigated how the prediction accuracy affects the caching performance on a real-world dataset and we showed that our model outperforms the state-of-the-art approaches.

References

[1] Cisco, "Global mobile data traffic forecast update, 2016–2021," *White Paper*, 2017.

[2] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, "Femtocaching: Wireless content delivery through distributed caching helpers," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 8402–8413, Dec. 2013.

[3] L. Breslau, Pei Cao, Li Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: evidence and implications," in *Proc. IEEE INFOCOM*, March 1999, pp. 126–134.

[4] S. Podlipnig and L. Böszörmenyi, "A survey of web cache replacement strategies," *ACM Comput. Surv.*, vol. 35, no. 4, pp. 374–398, 2003.

[5] S. Müller, O. Atan, M. van der Schaar, and A. Klein, "Context-aware proactive content caching with service differentiation in wireless networks," *IEEE Trans. Wireless Commun*, vol. 16, no. 2, pp. 1024–1036, 2017.

[6] B. Bharath, K. Nagananda, and H. V. Poor, "A learning-based approach to caching in heterogenous small cell networks," *IEEE Trans. Commun.*, vol. 64, no. 4, pp. 1674–1686, 2016.

[7] 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, May 2017.

[8] E. Bastug, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5G wireless networks," *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 82–89, Aug. 2014.

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

[10] S. Mehrizi, A. Tsakmalis, S. Chatzinotas, and B. Ottersten, "A feature-based Bayesian method for content popularity prediction in edge-caching networks," in *Poc. IEEE Wireless Commun. Netw. Conf. (WCNC), to be published*.

[11] ——, "Content popularity estimation in edge-caching networks from Bayesian inference perspective," in *Poc. IEEE Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2019, pp. 1–6.

[12] G. Grsun, M. Crovella, and I. Matta, "Describing and forecasting video access patterns," in *Proc. IEEE INFOCOM*, April 2011, pp. 16–20.

[13] N. B. Hassine, R. Milocco, and P. Minet, "ARMA based popularity prediction for caching in content delivery networks," in *Proc. IEEE Wireless Days*, March 2017, pp. 113–120.

[14] G. Paschos, E. Bastug, I. Land, G. Caire, and M. Debbah, "Wireless caching: Technical misconceptions and business barriers," *IEEE Commun. Mag.*, vol. 54, no. 8, pp. 16–22, 2016.

[15] A. H. Sayed, *Fundamentals of adaptive filtering*. John Wiley & Sons, 2003.

[16] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[17] A. Gelman *et al.*, "Prior distributions for variance parameters in hierarchical models," *Bayesian analysis*, vol. 1, no. 3, pp. 515–534, 2006.

[18] M. P. Wand, J. T. Ormerod, S. A. Padoan, R. Frühwirth *et al.*, "Mean field variational bayes for elaborate distributions," *Bayesian Analysis*, vol. 6, no. 4, pp. 847–900, 2011.

[19] K. B. Petersen, M. S. Pedersen *et al.*, "The matrix cookbook," *Technical University of Denmark*, vol. 7, no. 15, p. 510, 2008.

[20] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 5, no. 4, p. 19, 2016.