ARMA based Popularity Prediction for Caching in Content Delivery Networks

Nesrine Ben Hassine, Ruben Milocco, Pascale Minet

ARMA based Popularity Prediction for Caching in Content Delivery Networks

Nesrine Ben Hassine*†, Ruben Milocco‡, Pascale Minet*
*Inria Paris, 2 rue Simone Iff, CS 42112, 75589 Paris Cedex 12, France
Email: nesrine.ben-hassine@inria.fr, pascale.minet@inria.fr
†DAVID, University of Versailles, Versailles, France
‡Universidad Nacional Comahue, Buenos Aires 1400, 8300 Neuquén, Argentina.
Email: ruben.milocco@fain.uncoma.edu.ar

***Abstract*—Content Delivery Networks (CDNs) are faced with an increasing and time varying demand of video contents. Their ability to promptly react to this demand is a success factor. Caching helps, but the question is: which contents to cache? Considering that the most popular contents should be cached, this paper focuses on how to predict the popularity of video contents. With real traces extracted from YouTube, we show that Auto-Regressive and Moving Average (ARMA) models can provide accurate predictions. We propose an original solution combining the predictions of several ARMA models. This solution achieves a better Hit Ratio and a smaller Update Ratio than the classical Least Frequently Used (LFU) caching technique.**

***Index Terms*—Popularity prediction, ARMA, CDN, YouTube.**

I. Introduction

Contents Delivery Networks (CDNs) know an increasing success as shown by their huge number of users. To meet their various requirements, they have to provide a huge number of contents. To maximize the satisfaction degree of users raises a performance problem at the network and server levels. To alleviate this problem, the contents should be located very close to the users. Hence, the solution consists in caching the most popular contents. The CDN response time is then reduced as well as the network traffic.

The simplest solution consists in caching the contents that were the most popular the day before. This solution, called LFU [1], ejects the Least Frequently Used contents from the cache. In this paper, we want to determine whether a predictive approach based on an ARMA (Auto-Regressive and Moving Average) model is able to outperform LFU by improving the caching hit ratio.

In this paper, we apply different prediction methods to caching in CDNs. Unlike in [2], we investigate here prediction methods originated from the statistic field. More precisely, the contributions of this paper are the following:

- Based on a time varying identification of an ARMA model for the solicitation evolution of video contents in a CDN, we predict the next step using the linear predictor. Applying the $ARMA(\rho, q)$ model on the number of solicitations for the w previous days, where w is the size of the observation window, we predict the number of solicitations for the following day.
- Taking into account the constraints specific to the caching application, we show how to perform parametric identification of the ARMA model order adaptively using a sliding horizon of past samples. We then propose to predict the next value using this adaptive $ARMA(\rho, q)$ model based on a sliding window.
- Finally, we identify the conditions for which this ARMA model outperforms LFU for caching by comparing performances using different datasets.

II. Related work

Let us consider a time series y_t that can be estimated using:

- An Auto Regressive (AR) model of order ρ that is a linear regression based on the ρ prior values. We then have $y_t = \varphi_1 y_{t-1} + ... + \varphi_\rho y_{t-\rho}$, where $\varphi_i,\ i = 1..\rho$, are the parameters of the AR model that need to be estimated and p_t is the estimated value of y_t.
- A Moving Average (MA) model of order q that reflects the influence of randomness on the q prior values. We then have $y_t = \epsilon_t + \theta_1 \epsilon_{t-1} + ... + \theta_q \epsilon_{t-q}$, where both the parameters $\theta_i,\ i = 1..q$, and the sequence of ϵ_t need to be estimated for the MA model. The sequence ϵ_t is a zero mean, white sequence normally distributed and of variance σ^2.
- An AutoRegressive Moving Average (ARMA) model, denoted $ARMA(\rho, q)$, that combines the two previous models.

Thus the $ARMA(\rho, q)$ model can be formulated as:

$$y_t = \epsilon_t + \varphi_1 y_{t-1} + ... + \varphi_\rho y_{t-\rho} + \theta_1 \epsilon_{t-1} + ... + \theta_q \epsilon_{t-q} \quad (1)$$

where $\varphi_1, ..., \varphi_\rho$ are the parameters for the AR part, $\theta_1, ..., \theta_q$ are the parameters for the MA part, and ϵ is the white noise. (ρ, q) is the model order of the identified ARMA model. ρ (respectively q) represents the number of observations used to compute the value of y_t according to the AR model (respectively MA model). ρ and q take integer values greater than or equal to 0 and do not necessarily coincide with the parameters of the AR or MA models identified separately for the same time series.

Consequently, for an $ARMA(\rho, q)$ model there are $\rho + q$ parameters. The problem consists in finding the best tradeoff between on the one hand, model complexity increased by a high order of ρ and q and on the other hand, model accuracy evaluated by means of the prediction error. A model that has been over parameterized has poor predictive performance. The model requires the determination of both the order and the parameters.

Determining the best value of ρ and q is called model order identification. There are several error criteria used for model identification, they aim at determining the best model that is the model minimizing an error criterion. We can cite the most frequently used that are the Akaike's Information Criterion (AIC), the sample-size corrected AIC ($AICc$), and the Bayesian Information Criteria (BIC). These three criteria apply a log-likelihood function and penalize more complex models having a great number of parameters. More precisely, let $log(L)$ denote the value of the maximized log-likelihood objective function for a model with k parameters fit to n data points, we have:

$$AIC = -2log(L) + 2(\rho + q)$$

$$AICc = AIC + \frac{2(\rho + q)(\rho + q + 1)}{n - \rho - q - 1}$$

$$BIC = -2log(L) + (\rho + q)log(n)$$

$AICc$ is an AIC with a correction for finite sample sizes. It is used when the observation size is small relative to the model dimension, usually $n/(\rho + q) < 40$. For the BIC criterion, the penalty is also function of the sample size. The models providing the smallest values of the selected error criterion are chosen. These indicators will be used to analyze the optimum ratio $(\rho + q, w)$, with w the size of the sliding horizon of past samples, for this type of application by using typical records. It is not for use in real time because it requires to know the a priori Mean Squared Error of the prediction errors which is in fact related with the log(L) in the case of residuals normal distributed.

The learning sample is used to compute the parameters of the model. The estimation of the ARMA model parameters consists in finding the parameters that minimize some error criterion. Usually, an iterative algorithm like Recursive Prediction Error Method (RPEM), [3] is applied that stops when the error is less than a given threshold.

The computational complexity of the parameter estimation algorithm of an $ARMA(\rho, q)$ model was evaluated in [4]:

Property 1: The parameter estimation of an $ARMA(\rho, q)$ model has a computational complexity of $O(m^3 w)$, where w is the length of the time series sample and $m = max(\rho, q + 1)$.

Proof: see [4].

Once the model parameters have been computed, the $ARMA(\rho, q)$ model is able to predict the next value of the time series considered, using Equation 2.

Property 2: Because the prediction is one step, the optimal predictor is the same $ARMA(\rho, q)$ model. The prediction of y_t is given by:

$$p_t = \varphi_1 y_{t-1} + ... + \varphi_\rho y_{t-\rho} + \theta_1 \epsilon_{t-1} + ... + \theta_q \epsilon_{t-q} \quad (2)$$

and the prediction error is ϵ_t.

Proof: Since ϵ_t is white, there is no way to obtain lesser prediction error variance with any other predictor. Thereby the prediction is optimal in the sense on minimum variance.

Notice that in this paper, the prediction of y_t is denoted p_t instead of $\hat{y}_t$ in the literature. The $ARMA(\rho, q)$ model has been successfully applied for prediction as a statistical method in various fields. Focusing on networks, we can list some examples such as:

- Aggregation techniques based on an ARMA model in wireless sensor networks [5], [6]. The difficulties come from the resource limitations of wireless sensor nodes. Processing power, memory capacity and network bandwidth being strongly limited, the model order is upper-bounded by 5 for both ρ and q and the window size is fixed to 20. Each time the difference between the new data and the data predicted by the ARMA model is higher than a given threshold, data stored in the buffer up to the arrival of new data are aggregated and a new ARMA model is found to better fit the new data. This aggregation technique has the advantage of being self-adaptive to the data series. By reducing the number of transmissions done by wireless sensor nodes, this technique provides important gains in terms of network bandwidth and energy of sensor nodes.
- Distance estimation in wireless networks. In [7], an efficient adaptive estimation of the distance between sensors in a mobile network was presented using an $ARMA(1, 0)$ model that adaptively tunes the parameter using the RPEM over the following cost function:

$$V_t = \frac{1}{2} \sum_{i=1}^{t} \lambda^{t-i} \epsilon_i^2, \quad (3)$$

where λ is a scalar in the interval $(0, 1]$ called the *forgetting factor* which performs an exponential windowing over the previous prediction errors. The width of the exponential window depends on the value of λ. If $\lambda < 1$, previous prediction errors contribute only marginally to the criterion function. The window's width is reduced as λ decreases. In the case $\lambda = 1$, all past data are equally weighted. Thus, the value of λ determines the memory of the past data, which is a suitable parameter to take into account time-variant mobility dynamics of the nodes in a mobile network. The proposed method outperfoms several classical approaches reaching error values very close to the Cramer-Rao Lower Bound for both the static and the dynamic cases.

Like in [8], we use $ARMA(\rho, q)$ models to predict the popularity of video contents. However, the approach used for the so-called frequently-accessed videos is not adaptive as ours. In fact, for a time series, giving the popularity of video contents on 365 days, they use 183 days for parameter identification of the ARMA model and the rest for prediction. In addition, to reduce computation time, the authors use a data transformation based on the singular values decomposition in order to predict the main component of the individual time series using ARMA modeling.

In this paper, we carry out an analysis of different orders of $ARMA(\rho, q)$ models and we select the best ones: those providing the smallest MSE. Notice that MSE refers to the cumulated loss up to time t instead of the instantaneous loss used in [8] because of the high fluctuations of popularity of video contents. In addition, the final purpose of our study is to improve caching using popularity predictions.

III. Theoretical framework

A. Parameter estimation of an ARMA model

Let us assume the following quadratic cost over an horizon of t past samples:

$$V_t = \frac{1}{2}\sum_{i=1}^{t}\epsilon_i^2, \tag{4}$$

where ϵ_t is the prediction error given by

$$\epsilon_t = y_t - p_t, \tag{5}$$

where p_t is the prediction using the $ARMA(\rho, q)$ model with parameters vector $\Phi_t = [\varphi_{1,t} \ldots, \varphi_{\rho,t}, \theta_{1,t}, \ldots, \theta_{q,t}]$ that minimizes V_t defined by Equation 4. Notice that the cost V_t is the same as this defined by Equation 3 but considering $\lambda = 1$.

The parameters vector Φ_t is unknown and it is estimated by using the well-known Recursive Prediction Error Method (RPEM), [3]. To this end, the Gauss-Newton recursive algorithm over the cost function is used. The algorithm and its properties are given by the following theorem:

Theorem: Consider the cost function V_t defined by Equation 4 to be minimized, with respect to the parameter vector Φ_t, by the following Gauss Newton recursion:

$$\epsilon_t = y_t - p_t; \tag{6}$$

$$\varphi_t = -\epsilon_t'; \tag{7}$$

$$M_t = M_{t-1} - \frac{M_{t-1}\varphi_t\varphi_t^T M_{t-1}}{1 + \varphi_t^T M_{t-1}\varphi_t} \tag{8}$$

$$\Phi_t = \Phi_{t-1} + M_t\varphi_t\epsilon_t \tag{9}$$

where t is the iteration step, M_t is a square matrix of dimension $(\rho + q)$; Φ_t and φ_t are column vectors, ϵ_t' is the derivative of ϵ with respect to the parameters in Φ_{t-1}, and T denotes the transpose.

Then, the following holds: Φ_t converges as $k \to \infty$ with probability 1 to one element of the set of minimizers.

$$\left\{\Phi | \sigma'^2 = 0\right\}; \tag{10}$$

where σ'^2 is the derivative of the prediction error variance with respect to Φ.

Proof: See [3].

The initial values are as follows: $t = 1$, M_1 is the identity matrix and ϵ_1 is a vector of zeros. $\Phi(0)$ is obtained by doing the least squared estimation from data. This recursive algorithm can be repeated several times over the observation window where the parameters obtained in the previous stage are used as initial values of the new stage. In the convergence vector optimal parameters are obtained at each step t.

B. Problem statement

Given any video content C, we focus on the time series y_t describing the evolution of its number of solicitations. We want to use an $ARMA(\rho, q)$ model to predict the future values of this time series, according to Equation 2. Each $ARMA(\rho, q)$ model is characterized by its complexity and its accuracy. The choice of the model must find the best trade-off between complexity and accuracy taking into account the constraint of the caching application.

To improve the accuracy of the model, we propose to use an **adaptive** $ARMA(\rho, q)$ **model**: at each time t, the parameters of the $ARMA(\rho, q)$ model are computed on a sliding window of size w. More precisely, at time $t-1$, the $ARMA(\rho, q)$ model predicts the value of the time series at time t according to Equation 2, using the w last observations in the window $[t-w, t-1]$ to compute the parameters ϕ_i and θ_j of the $ARMA(\rho, q)$ model. This principle is illustrated in Figure 1. Then, the sliding window moves one step ahead, starting at time $t-w+1$ and ending at time t, the parameters are computed using the observations in this window, and the prediction for time $t+1$ is given.

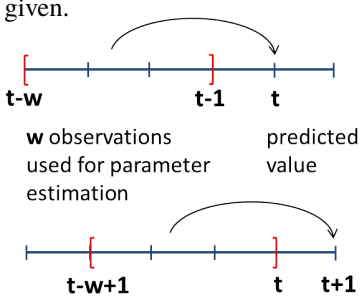


Fig. 1: Principle of the sliding window in the ARMA model.

Notice that the model's order (ρ and q) is fixed, whereas the computation of parameters ϕ_i and θ_j is done at each time k using an iterative parameter estimation algorithm. That is why the $ARMA(\rho, q)$ model is said adaptive.

The prediction error metric is given by the Mean Square Error, MSE:

$$MSE = \frac{\sum_{i=1}^{n}(y_i - p_i)^2}{n},$$

where y_i denotes the real data series considered, p_i is the time series of the predicted value of these data, n is the total number of values in the data series. MSE allows us to compare different models applied on the same video content. If now, we want to compare the performances of a model on two different contents, we need to normalize the error as follows:

$$NMSE = \frac{MSE}{\sum_{i=1}^{n} y_i^2}.$$

The constraints given by the caching application are due to the fact that a CDN is managing millions of contents:

- Since an $ARMA(\rho, q)$ model has $\rho + q$ parameters and these parameters must be computed on all the data values in the window, we prefer small values of ρ and q to minimize the number of parameters to compute, while ensuring an acceptable prediction error. This is the parsimony principle.
- The model order needs to be in accordance with the length of the window: w should be at least equal to $\rho + q$. To this end we apply the AIC criterion. In addition, the length of the window is related to the local stationarity and will be checked on line.
- The complexity of computation and memory requirement per content should be optimized so as not to loose predictability while being kept reasonable.

We adopt the following assumption about the first w observations:

Assumption 1: By convention, the $ARMA(\rho, q)$ model predicts

$$p_t = \begin{cases} 0 & \text{if t=0} \\ y_{t-1} & \text{if 1<t<w-1} \\ \varphi_1 y_{t-1} + ... + \theta_1 \epsilon_{t-1} + ... & \text{otherwise} \end{cases}$$

for any time $t \in [1, w]$, the $ARMA(\rho, q)$ model predicts $p_t = y_{t-1}$ if $t > 1$ and $p_1 = 0$ otherwise.

C. Properties of our adaptive ARMA model

As previously said, we use AIC (or $AICc$ depending on the value of $n/(\rho + q)$) and $NMSE$ to compare our models on a given video content. We can express AIC and $AICc$ as a function of MSE [9]. By replacing $\frac{\sum_{i=1}^{n}(y_i - p_i)^2}{n}$ by MSE, we get:

$$AIC = n \cdot log(MSE) + 2(\rho + q),$$

$$AICc = n \cdot log(MSE) + 2(\rho + q) + \frac{2(\rho + q) \cdot (\rho + q + 1)}{n - \rho - q - 1}.$$

Let us now focus on the smallest window size acceptable for model parameter estimation.

Property 3: For parameter estimation, any $ARMA(\rho, q)$ model accepts the same smallest window size as the $ARMA(q, \rho)$ model.

This can be explained by the fact that in the estimation of the $\rho + q$ parameters, ρ and q play a symmetric role. In addition, we also conjecture the following property that we checked on 15 pairs (ρ, q), with ρ and $q \in [1, 15]$.

Property 4: For any $ARMA(\rho, q)$ model, the smallest window size acceptable for model parameter estimation is given by:

$$w_{min} = 2 * max(\rho, q) + \rho + q. \quad (11)$$

From this property, we can deduce which information criterion, AIC or AICc, should be used to compare two models using a window of minimal size.

IV. Popularity prediction with ARMA

A. Notations

Table I summarizes the notation adopted in this paper.

TABLE I: Notations.

ρ	the order of the AR part of the ARMA model
q	the order of the MA part of the ARMA model
w	the observation window size. The ARMA model parameters are computed using the last w samples
C	the set of video contents
	For each given video content $c \in C$
y_t	the number of solicitations at time t
p_t	the prediction of the number of solicitations for time t
n	the total number of samples

B. Datasets extracted from YouTube

The datasets considered are real traces extracted from the YouTube CDN. We randomly select video contents and extract their traces. For each video content, its trace consists in its number of solicitations for every day since its creation until the trace extraction day.

In this paper, we consider two sets: The first set S_1 contains 30 video contents randomly chosen. It is used to determine the best $ARMA$ parameters which are the model order (ρ, q) and the window size w. This reduces the number of the experts $ARMA$ to be evaluated on the second set S_2. The random choice of video contents can be justified by the high fluctuation of the popularity of the video contents. Each video could be put in the cache at some point during its lifetime. The second set S_2 contains 60 contents chosen to evaluate the caching strategy. These contents have close popularities with different fluctuations. This allows us to highlight the dynamics of the cache (insertion, eviction).

Each video content has its own profile of popularity evolution. The popularity is evaluated by its number of solicitations. Figure 2 depicts five different profiles of video contents among those belonging to the first set.

C. Selection of the best $ARMA(\rho, q)$ models on the first set

The purpose of this section is to determine the $ARMA(\rho, q)$ models providing the most accurate predictions on the set S_1 considered. The accuracy of predictions is evaluated by their MSE at the end of the simulation. Since we want to compare caching based on ARMA prediction and LFU in terms of MSE, we start by computing the value of MSE for LFU. We recall that LFU predicts for day t the same number of solicitations as for day $t - 1$. The values of MSE for 5 video contents among the 30 contents of the set S_1 are given in Table II.

TABLE II: MSE error for LFU.

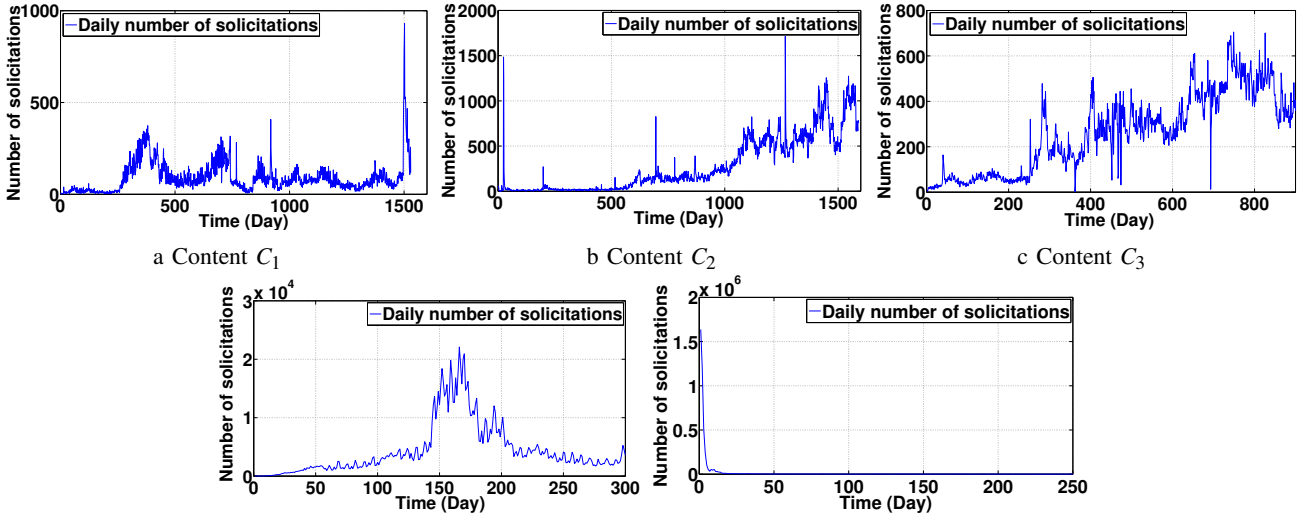
Content	MSE
C_1	1.28E+03
C_2	6.24E+03
C_3	2.84E+03
C_4	1.77E+06
C_5	1.33E+10

In the second series of experiment, we study the impact of the window size on the prediction error for a given model. For each profile tested, we make vary $\rho = q$ from 1 to 15. For any pair (ρ, q) we compute MSE, NMSE, AIC and AICc of the model for different window sizes. We recall that it is useless to select ARMA models providing a MSE error higher than this computed for LFU. Results obtained for each content C_1 to C_5 are depicted in Figure 3.

Results show that for values of $\rho \leq 15$, small values of w minimize the MSE error for any $ARMA(\rho, \rho)$ model applied to the five profiles C_1 to C_5. More precisely, a window size of $w = w_{min}$ or $w = 2w_{min}$ minimizes the MSE. In addition, there exists small values of ρ giving predictions with an MSE error close to the smallest one.

We compare LFU and $ARMA(\rho, q)$ predictions. Extracted traces show that, for all the contents tested except content C_5, there exists at least one $ARMA(\rho, q)$ model with $\rho \geq 1$ and $q > 0$ that outperforms LFU in terms of accuracy as depicted in Figure 4. Notice that C_5 has a very specific profile (see Figure 2d): the number of solicitations has a very strong peak the first day, and a strong decrease the second day, followed by a very small number of solicitations during the next days. It is impossible to predict the peak of solicitations without any additional information (e.g. occurrence of a social event).

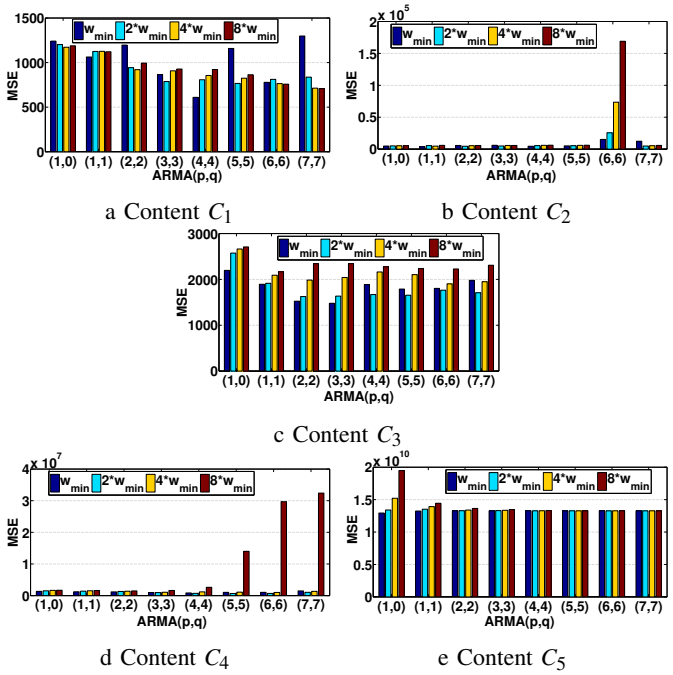
With the results obtained up to now, we want to select the $ARMA(\rho, q)$ models that provide the most accurate predic-



a Content C_1 b Content C_2 c Content C_3

d Content C_4 e Content C_5

Fig. 2: Popularity profiles of five video contents selected in S_1.

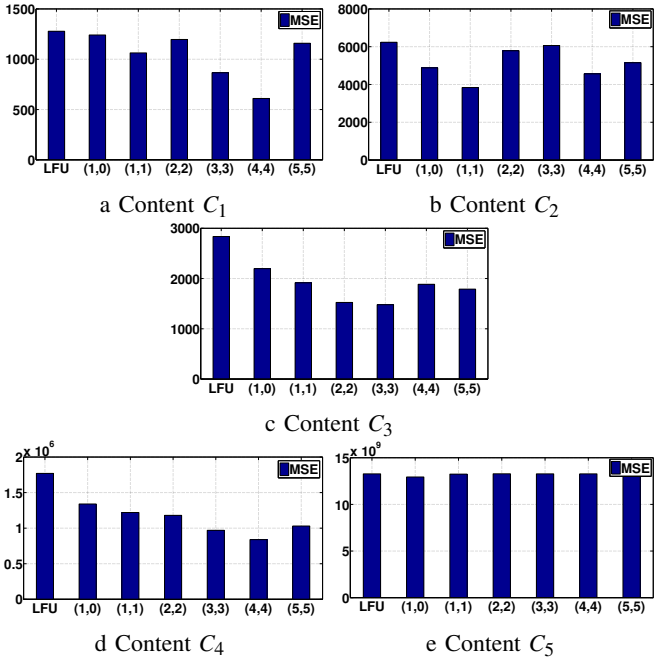


a Content C_1 b Content C_2

c Content C_3

d Content C_4 e Content C_5

Fig. 3: Impact of ρ, q and w on the MSE error of ARMA models applied to the five video contents selected.



a Content C_1 b Content C_2

c Content C_3

d Content C_4 e Content C_5

Fig. 4: Orders vs errors.

tions on the first set considered. We select the model orders verifying the following two conditions:

- the window size computed according to Equation 11 is less than or equal to 20, to meet the parsimony principle.
- the error provided by the $ARMA(\rho, q)$ model, evaluated by its MSE, is less than the error provided by LFU.

Hence, for each $ARMA(\rho, q)$ model, we compute the number of times where it has provided a MSE smaller than the one obtained by LFU. Results are depicted in Table III and the last column denotes the percentage over the video contents forming S_1.

TABLE III: Percentage of contents in the set S_1 for which the $ARMA(\rho, q)$ models outperform LFU.

(ρ, q)	w_{min}	Ratio of least MSE (%)
(1, 0)	3	90
(1, 1)	4	93.3
(3, 3)	12	90
(4, 4)	16	100
(5, 5)	20	86.6

D. $ARMA(\rho, q)$ Predictions on the second set

The best $ARMA(\rho, q)$ models being selected on the first set S_1 according to Table III. We now apply these experts on the second set S_2 consisting of 60 video contents. Note that an expert is a logical entity that computes and predicts the future number of requests for each video content. As a first series of tests, we compare their predictions in terms of MSE to those provided by:

- the LFU expert that predicts for time t the number of solicitations at time $t-1$, that is $p_t = y_{t-1}$,
- and the Basic expert [11] that predicts for time t the number of solicitations at time $t-1$ plus the same increase or decrease as at time $t-1$. In other words, it predicts $p_t = y_{t-1} + (y_{t-1} - y_{t-2}) = 2y_{t-1} - y_{t-2}$.

Let $R_E = (r_1, ..., r_c, ..., r_n)$ the vector of rewards of expert E where n is the number of video contents in the set S_2 ,

$$r_c = \begin{cases} 1 & \text{if } E \text{ is the expert that provides} \\ & \text{the least MSE for content } c \\ 0 & \text{otherwise} \end{cases}$$

For each expert evaluated, we define two ranks on the set of contents considered:

- the daily rank of an expert E is the sum of its vector of rewards R_E at time t.

$$dr_E(t) = \sum_{c=1}^{n} r_c$$

- the cumulated rank of an expert E is the sum of the daily rank of this expert on all the days considered in the simulation (here 100 days).

$$cr_E = \sum_{t=1}^{100} dr(t)$$

Results concerning the set S_2 are depicted in Figures 5 and 6 for the daily rank and the cumulated rank, respectively. For the daily rank, we observe that as long as the number of observations is smaller than its window size, any $ARMA(\rho, q)$ model predicts exactly the same value as LFU, as expected because of Assumption 1. We notice that $ARMA(1, 1)$ clearly outperforms all the other experts considered, including LFU and Basic. It is followed by $ARMA(3, 3)$, then $ARMA(4, 4)$ and finally $ARMA(5, 5)$. All these $ARMA(\rho, q)$ models outperform LFU, which itself outperforms $ARMA(1, 0)$ and Basic.

For the cumulated rank depicted in Figure 6, we have the same conclusions: $ARMA(1, 1)$ predicts better than any other ARMA model. In addition, $ARMA(3, 3)$, $ARMA(4, 4)$ and $ARMA(5, 5)$ are better than LFU. $ARMA(1, 0)$ is the only ARMA model that performs worse than LFU and Basic. Hence, the popularity prediction of video contents is improved using $ARMA(\rho, q)$ models with $\rho > 1$ and $q > 0$.

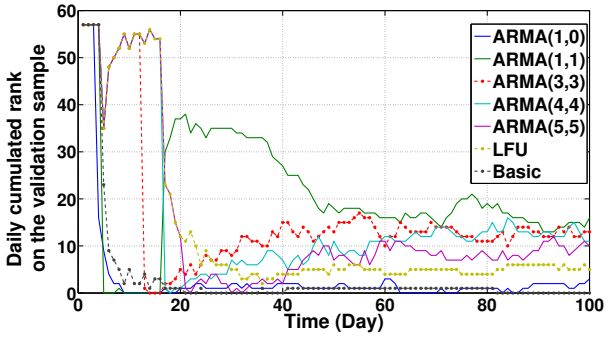


Fig. 5: Daily cumulated rank on the validation set.

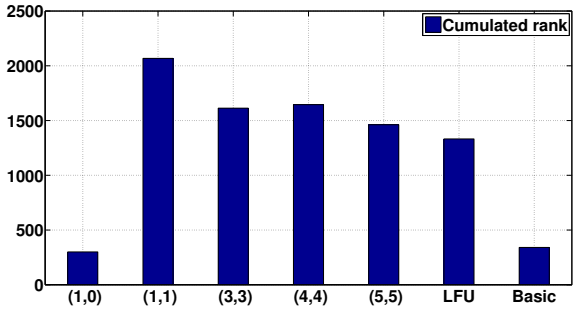


Fig. 6: Total cumulated rank on the validation set.

V. Caching based on ARMA prediction

A. The caching problem

Having defined our $ARMA(\rho, q)$ models, we use their popularity predictions for video content caching. Each day, these $ARMA(\rho, q)$ models predict the popularity of all video contents. The contents having the highest popularity predictions are inserted in the cache until the cache is full.

To evaluate the performance of this caching strategy based on ARMA predictions, we use the two metrics the most frequently adopted:

- the Hit Ratio that is defined as the percentage of requests related to a content already in the cache.
- the Update Ratio that is defined as the percentage of contents in the cache that are replaced by more popular contents.

In the CDN context, the best caching technique maximizes the Hit Ratio as the first criterion and minimizes the Update ratio as the second criterion. The performance of any caching technique strongly depends on the cache size and on the knowledge about requests. It has been proved in [10] that the Optimal caching strategy is an offline strategy knowing in advance the future solicitations. When the cache is full, this strategy evicts the content whose request is the furthest in the future. However, in the traces extracted from YouTube, the only knowledge we have is the number of solicitations of each content per day, this strategy is extrapolated to take into account the number of solicitations of each content per day instead of the arrival times of these solicitations. This new version is denoted $Max\ hit\ ratio$.

Figure 7 depicts the evolution of the hit ratio over time for the different experts evaluated. If the accuracy of the $ARMA(\rho, q)$ predictions is usually very good, we observe some times where these predictions are of poor accuracy for times 55 and 80, where the Hit Ratio obtained is very low. Figure 5 has shown that $ARMA(1, 1)$ is the model that minimizes most frequently the MSE over the validation set.

However, even if it gives almost the best hit ratio in most days, it crashes significantly at time 80.

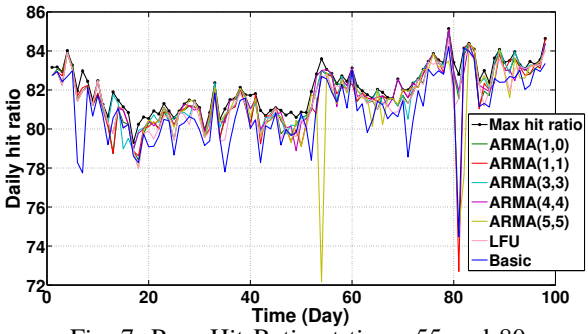


Fig. 7: Poor Hit Ratio at times 55 and 80.

In Figure 8, we depict the hit ratio evolution of $ARMA(1,1)$, $ARMA(3,3)$, LFU and the Max hit ratio. According to Figure 6, $ARMA(1,1)$ minimizes the MSE most frequently than any other evaluated expert. We take a look at its daily hit ratio evolution and compare it with LFU. We observe that in most cases $ARMA(1,1)$ provides a better hit ratio than LFU, except around time 80. This is corroborated by the computation of the average hit ratio: $ARMA(1,1)$ obtains 81.57 whereas LFU gets 81.5. Note that for the set we are evaluating, the average of the best hit ratio we could achieve is 82.14. $ARMA(1,1)$ enables an improvement of 1‰ compared with LFU.

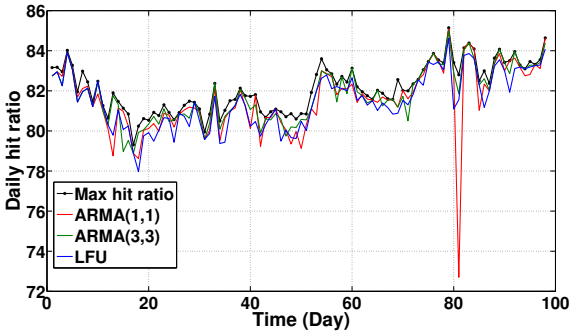


Fig. 8: Zoom on best experts.

Despite this low percentage, this improvement is considered very important in the current context of work given the fact that we are dealing with millions of content having millions of requests. One more aspect to be mentioned here is the behavior of $ARMA(3,3)$. In Figure 6, $ARMA(3,3)$ is not as good as $ARMA(1,1)$. However, in terms of hit ratio, $ARMA(3,3)$ competes $ARMA(1,1)$ and even has the best average hit ratio of about 81.79 compared to the set of the evaluated experts.

While $ARMA(1,1)$ provides most frequently the most accurate prediction, $ARMA(3,3)$ gets the best hit ratio. This proves the fact that it is not only the prediction accuracy that matters. Hence, the challenge is to properly identify the subset of contents that will be solicited more often the next day, and not their exact number of solicitations.

B. Forecasters definition

We saw in Section IV-C that there is no $ARMA(\rho,q)$ model that provides the best prediction at any time for any content. For this reason, we introduce a new entity called Forecaster. The forecaster is in charge of computing a prediction based on the predictions received from the best $ARMA(\rho,q)$ models selected as described in Section IV-C. More precisely, this forecaster predicts the average value of the predictions provided by the k best experts, with $1 \leq k \leq 4$. This forecaster is called k-BE in short. There are several ways to define a Best-Expert. In this paper, we use:

- *the prediction error based forecaster*, also called the $k-BE$ on MSE forecaster: this forecaster uses at time t the experts minimizing the MSE up to time t.
- *the rank based forecaster*, also called the $k-BE$ on rank forecaster: the k best experts used are those occupying the first rank most often for the contents in the set considered.
- *the hit ratio based forecaster*, also called the $k-BE$ on hit ratio: the k best experts used are those providing the highest hit ratio at time $t-1$.

C. Accuracy of Forecaster predictions

We now compare the predictions given by each forecaster defined in Section V-B. Their prediction accuracy is illustrated by Figure 9. We observe that all the forecasters have a correct accuracy. However, the forecasters BE on rank and BE on MSE tend to underestimate the popularity of video contents. The best ones are those that compute an average value based on the advice of several experts.

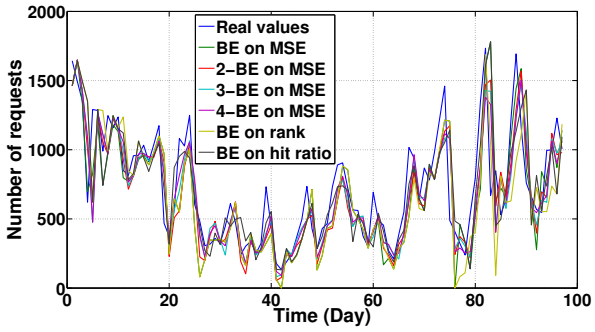


Fig. 9: Prediction accuracy of each Forecaster.

D. Caching based on Forecaster predictions

We now evaluate the performances of a caching based on forecaster prediction, itself based on $ARMA(\rho,q)$ predictions. For this performance evaluation of caching, we adopt the following assumptions:

- we consider chunks of contents of same size.
- the cache size allows to cache 40% of the contents chunks.

We evaluate the Hit Ratio obtained by each forecaster. Results depicted in Figure 10 show that $k-BE$ forecasters have an average hit ratio greater than all the hit ratios previously obtained by the different experts. This validates the use of forecasters in popularity prediction based on $ARMA(\rho,q)$ models.

When we look at the daily behavior of hit ratio in Figure 11, we notice that all the forecasters are able to follow the sudden change at time 55, unlike the experts (see Figure 7). Note that a sudden change in hit ratio corresponds to a sudden and very significant change in the popularity of certain video

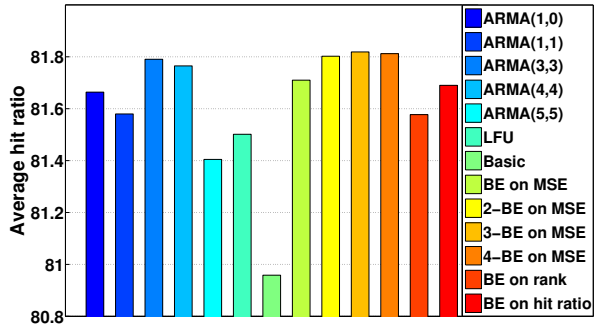


Fig. 10: Average Hit Ratio obtained by experts and forecasters.

contents to the point that the change is visible on all the videos in the test set. In addition, the BE on MSE and BE on rank forecasters have a hit ratio far from the best computed one. These two forecasters fail because both take prediction decision based on only one expert. So if these forecasters pick the best expert at time $t-1$ which becomes bad at time t in term of prediction accuracy, their prediction decision will be often incorrect. $k-BE$ forecasters succeed in following the second sudden variation of the hit ratio that occurred at time 80 because their decision is based on more than one expert.

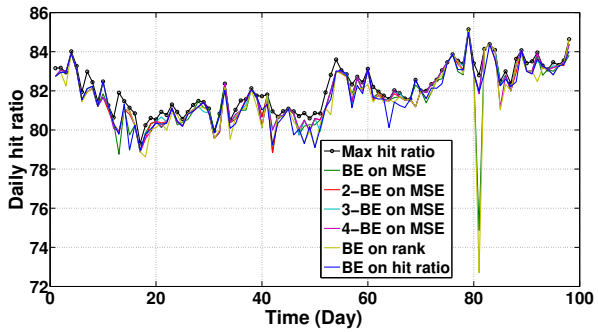


Fig. 11: Hit Ratio obtained by each Forecaster.

Once more, although the update ratio varies very often from time t to time $t+1$ (Figure 11), $k-BE$ forecasters, with $k>1$ provide the smallest update ratio as depicted in Figure 13. They outperform both the experts and the forecaster using a single expert, even if this expert varies over time.

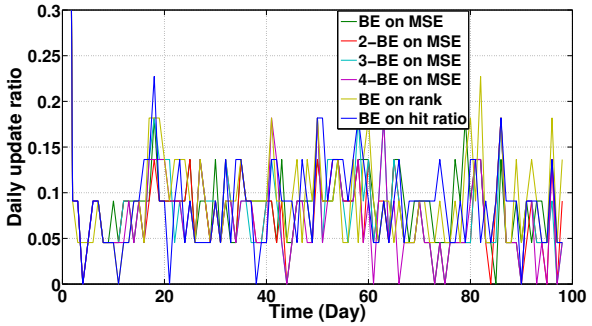


Fig. 12: Update Ratio obtained by each Forecaster.

VI. Conclusion

In this paper we focus on predicting the popularity of video contents using Auto-Regressive Moving Average (ARMA)

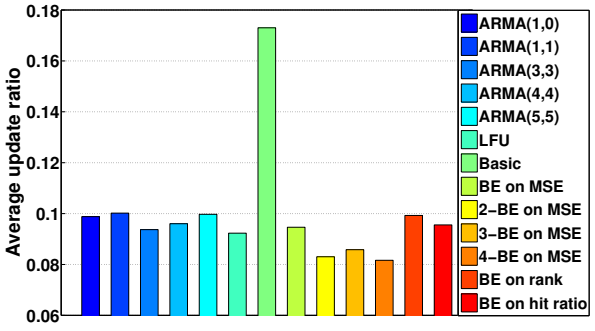


Fig. 13: Average Update Ratio obtained by experts and forecasters.

methods applied on a sliding window. These predictions are used to put the most popular video contents into caches. After having identified the parameters of ARMA experts, we compare them with an expert predicting the same number of requests as the previous day. Results show that ARMA experts improve the accuracy of the predictions. Nevertheless, there is no ARMA model that provides the best prediction for all the video contents over all their lifetime. We combine these statistical experts with a higher level of experts, called forecasters. By combining the experts prediction, some forecasters succeed in predicting more accurate values which helped to increase the hit ratio while keeping a correct update ratio. Hence, improving the accuracy of the predictions succeeds in improving the hit ratio. However, the most challenging task is to properly identify the subset of the most solicited contents the next day.

References

[1] N. Megiddo and D. S. Modha, Outperforming LRU with an adaptive replacement cache algorithm, Computer, vol. 37, no. 4, pp. 58-65, 2004.

[2] N. Ben Hassine, D. Marinca, P. Minet, D. Barth, *Caching strategies based on popularity prediction in content delivery networks*, the 12th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications. October 2016, New York, USA.

[3] L. Ljung, T. Söderström, (1983). *Theory and Practice of Recursive Identification*. MIT Press. 1987.

[4] K.Deng, A.W. Moore, M.C. Nechyba, *Learning to recognize time series: combining ARMA models with memory-based learning*, IEEE Int. Symposium on Computational Intelligence in Robotics and Automation, Monterey, USA, 1997.

[5] J. Lu, F. Valois, M. Dohler, M.Y. Wu, *Optimized data aggregation in WSNs using adaptive ARMA*, IARIA SENSORCOMM, Venice, Italy, July 2010.

[6] J. Cui, O. Lalami, J. Lu, F. Valois, A^2*: Agnostic aggregation in wireless sensor networks*, IEEE CCNC, Las Vegas, USA, January 2016.

[7] R. Milocco, S. Boumerdassi, *An efficient adaptive method for estimating the distance between mobile sensors*, Wireless Networks, The Journal of Mobile Communication, Computation and Information, ISSN 1022-0038, November 2015, Volume 21, Issue 8, pp 2519-2529.

[8] G. Gursun, M. Crovella, I. Matta, *Describing and forecasting video access patterns* INFOCOM, Shanghai, 2011, pp 16–20.

[9] Dennis J. Beal, *SAS Code to Select the Best Multiple Linear Regression Model for Multivariate Data Using Information Criteria*, Science Applications International Corporation

[10] L.A. Belady, *A study of replacement algorithms for virtual storage computers*, IBM Systems J., vol.5, n.2, 1966, pp. 78-101.

[11] K. H. Ang, G. Chong, and Y. Li, *Pid control system analysis, design, and technology,* IEEE Transactions on Control Systems Technology, vol. 13, no. 4, pp. 559-576, 2005