Adaptive Caching in the YouTube Content Distribution Network: A Revealed Preference Game-Theoretic Learning Approach

William Hoiles, Omid Namvar Gharehshiran, Vikram Krishnamurthy, *Fellow, IEEE*, Ngọc-Dũng Đào, and Hang Zhang

(Invited Paper)

***Abstract*—Cognitive systems are dynamical systems that adapt their behavior to achieve sophisticated outcomes in nonstationary environments. This paper considers two cognitive aspects regarding the problem of distributed caching with limited capacity in a content distribution network that serves YouTube users with multiple edge servers: first, the theory of revealed preference from microeconomics is used to estimate human utility maximization behavior. In particular, a nonparametric learning algorithm is provided to estimate the request probability of YouTube videos from past user behavior. Second, using these estimated request probabilities, the adaptive caching problem is formulated as a non-cooperative repeated game in which servers autonomously decide, which videos to cache. The utility function tradesoff the placement cost for caching videos locally with the latency cost associated with delivering the video to the users from a neighboring server. The game is nonstationary as the preferences of users in each region evolve over time. We then propose an adaptive popularity-based video caching algorithm that has two timescales: The slow timescale corresponds to learning user preferences, whereas the fast timescale is a regret-matching algorithm that provides individual servers with caching prescriptions. It is shown that, if all servers follow simple regret minimization for caching, their global behavior is sophisticated—the network achieves a correlated equilibrium, which means that servers can coordinate their caching strategies in a distributed fashion as if there exists a centralized coordinating device that they all trust to follow. In the numerical examples, we use real data from YouTube to illustrate the results.**

***Index Terms*—YouTube social network, adaptive caching, Afriat's theorem, revealed preferences, non-cooperative games, correlated equilibrium, stochastic approximation algorithm.**

NOMENCLATURE

$\mathbf{p}$	Probe vector.
$\mathbf{x}^i$	Users response to $\mathbf{p}$ at server i.
$\mathbf{y}^i$	Users response to $\mathbf{p}$ observed in noise at server i.
$\mathcal{D}^i$	Dataset of probe signals and user responses collected at server i.
$\mathcal{D}^i_{\text{obs}}$	Dataset of probe signals and user responses observed in noise at server i.
$u^i(\cdot)$	Utility function of users at server i.
$\mathbf{x}^i_o$	User demand vector computed at server i.
$\boldsymbol{\mu}^i_o$	Request probability vector computed at server i.
$\mathcal{I}$	Set of servers.
$\mathcal{V}$	Set of video blocks to be cached.
$\mathcal{A}^i$	Set of caching decisions of server i.
s_j	Size of the j-th block of video.
N_i	Storage capacity of server i.
D	Distance matrix for the CDN topology.
a^i	Caching decision of server i.
$\mathbf{a}^{-i}$	Caching profile of all server excluding server i.
$C^i(\cdot)$	Cost function of server i.
ν^i	Aggregate rate of video requests at server i.
$\mathcal{Q}$	Set of correlated equilibria.
R^i	Regret matrix of sever i.
ε	Adaptation rate of adaptive caching algorithm.
$\boldsymbol{\psi}^i$	Caching strategy of sever i.
$\mathbf{z}$	Global caching behavior.

Manuscript received March 22, 2015; revised July 7, 2015; accepted September 2, 2015. Date of publication October 8, 2015; date of current version December 21, 2015. This work was supported by the NSERC Strategic Grant and the Canada Research Chairs program. The associate editor coordinating the review of this paper and approving it for publication was M. Zorzi.

W. Hoiles, O. N. Gharehshiran, and V. Krishnamurthy are with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC V6T 1Z4, Canada (e-mail: whoiles@ece.ubc.ca; omidn@ece.ubc.ca; vikramk@ece.ubc.ca).

N.-D. Đào and H. Zhang are with Huawei Canada Research Centre, Ottawa ON K2K 3J1, Canada (e-mail: Ngoc.Dao@huawei.com; Hang.Zhang@huawei.com).

Digital Object Identifier 10.1109/TCCN.2015.2488649

I. INTRODUCTION

ADAPTIVE caching of YouTube videos provides for an interesting example of cognitive processing in telecommunication networks. On the one hand, human behavior that determines the popularity of specific videos need to be modeled in order to determine what to cache. On the other hand, distributed caching algorithms must exploit this human-centric information and adapt to changes in popularity of videos. This paper deals with both these aspects using two powerful tools from microeconomics theory: revealed preferences and game-theoretic learning. Before proceeding with details on these methodologies, we first describe the adaptive caching problem in a YouTube content distribution network (CDN).

YouTube is a web-based service that provides video sharing via the Internet. Users can upload their own videos and make them available to the public. Viewers can search for videos and watch them on their computers or mobile devices. The

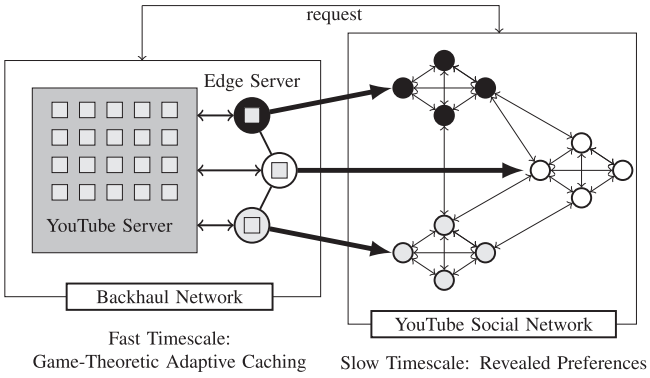

Fig. 1. Given a set of videos, how to select the subset to be stored on each edge server in the content distribution network so as to establish a proper balance between local placement costs of servers and the latency in delivering videos to the users in the YouTube social network? The content distribution network allows files, illustrated by □, to be transferred between the edge servers, depicted with circles filled with □.

YouTube social network is composed of users, a communication infrastructure which contains the server that stores all YouTube videos, and a series of edge servers in a CDN, which deliver the videos to users. The network topology is depicted in Fig. 1. Four steps are involved in delivering a requested video: First, a message is sent to the YouTube web server. Then, a redirect message is sent to the client which then relays this message to the CDN. Once the CDN receives the redirect message, it transfers the video to the client via an edge server.

The CDN contains several edge servers which cache content that may be requested by clients. Akamai[1] is an example of CDNs deployed by YouTube that contains over 135,000 servers in more then 2000 locations throughout the globe. If an edge server does not contain the requested video file in its cache, it will fetch and cache the file for delivery. The edge server can either retrieve the video from other edge servers in the CDN or directly from the YouTube server taking into account several factors such as latency, router speed, bandwidth between severs, and servers workload. Caching of videos on select edge servers has proven to enhance the performance—in terms of latency in delivering content—availability, and reliability of CDNs.

A. Main Ideas

This paper combines the tools of revealed preferences [1]–[3] and regret-matching game-theoretic learning [4]–[6], both from the economics literature, to devise and analyze an adaptive popularity-based video caching algorithm that prescribes edge servers in a CDN which videos to cache based on the request behavior of users. The proposed solution customizes the prescriptions to individual edge servers based on the preferences of users in its locale, and is adaptive in that the prescriptions are adapted to evolution of users preferences over time. It is a two timescale algorithm, each addressing a cognitive aspect as described below:

1) Slow Timescale: Learning Video Request Probabilities Using Revealed Preferences: The slow timescale corresponds to learning user preferences which can be used to estimate video request probabilities. As found in [7], [8], user preferences typically change on a slow timescale spanning several months, however changes in usage occur rapidly. This suggests that the request probabilities depends not only on the preferences of users but also on the quality of the YouTube videos. In Sec. II, the principle of revealed preferences [1], [9], [10] arising in microeconomics is applied for the first time to test if YouTube users are utility maximizers in their response to posted videos. Subsequently, their associated utility functions are constructed using the observed user behavior, using which a non-parametric learning algorithm is proposed to forecast the request probabilities of videos in Sec. III. The proposed scheme differs from existing works in the literature, which mostly use request statistics to detect for popular videos, in that it takes into account the videos' metadata (e.g., subscriber, thumbnail, author, etc.) and user responses to them to forecast request probabilities. In the numerical evaluations in Sec. VI, real data from YouTube is used to illustrate the results.

2) Fast Timescale: Regret-based Adaptive Video Caching: The fast timescale is an adaptive game-theoretic caching algorithm that relies on the regret-matching learning procedure [4], [11] in the economics literature. It exploits the human-centric information from the slow timescale to prescribe individual servers which videos to cache in a decentralized way, and adapts these prescriptions to popularities of videos by minimizing the experienced regrets based on a local cost function. This cost function trades-off the placement costs, associated with caching videos locally, with the access costs (latency) incurred by retrieving requested videos from neighboring edge servers or the YouTube main server. If a video is cached on a server nearby the server that has received the request, it may be better off accessing the remote replica. On the other hand, if all replicas are located far away, the server is better off caching the video itself. The interdependence of servers' caching decisions and their selfish behavior are the key features that characterizes our study as a non-cooperative game in Sec. IV. As the user preferences change over time, so does the servers' cost functions; therefore, we practically deal with a regime switching non-cooperative game [12].

We show in Sec. V that, if all servers follow the proposed algorithm independently, their collective caching behavior across the CDN will asymptotically reach the set of *correlated equilibrium* [13] and track its evolution as the user preferences change over time. Correlated equilibrium is a generalization of Nash equilibrium and describes a condition of competitive optimality. It is, however, more realistic in multi-agent learning scenarios since observation of the past history of decisions (or their associated outcomes) naturally correlates agents' future decisions. As described in the seminal work of Aumann [13], coordination among agents in correlated equilibrium can potentially lead to their higher social welfare than making decision independently as in Nash equilibrium. Reaching a correlated equilibrium implies the capability of servers to coordinate in a distributed fashion as if there exists a centralized coordinating device that they all trust to follow. We thus show that simple local caching behavior of servers and exchange of information among them can lead to the manifestation of a sophisticated and rational behavior at the network level.

[1] http://www.akamai.com/

B. Literature

How do large CDNs such as Akamai cache content in their network? An interesting feature of Akamai is that it does not perform any caching of content that has not been requested– this is known as the *pull model*. As requests for content are made, the network controller stores the statistics of the requests. Popular content is cached by the servers, and unpopular content is evicted to free space for high popularity content. In the caching scheme proposed in this paper, the request probability of content is predicted based on the preferences of users. This allows the content to be pre-fetched to the servers prior to the observation of the request statistics from users.

There has been considerable research in placement of replicas in CDNs and peer-to-peer systems; see [14] for a comprehensive survey. In [15], [16], the optimal object replication problem is shown to be NP-hard; therefore, they propose and compare four heuristics, namely, random, popularity, local greedy, and global greedy. Similar policies are proposed in [17], [18]. In [19], [20], joint optimization of surrogate placement, object replication, and request routing has been considered; however, they do not provide any heuristics to approximate the solution. Simulated annealing scheme have also been proposed in [21]. The interested reader is also referred to [22], [23] for recent comprehensive surveys of in-network caching mechanisms in information-centric networking. These mechanisms range from cache-everything-everywhere [24] and probabilistic schemes [25], [26] to more sophisticated intra-domain and inter-domain cache cooperation [27]–[29], centrality-based caching [30], heterogeneous storage size [30], and distributed cache management [31].

The recent past has witnessed a growing interest in employing game-theoretic methods for designing self-configurable adaptive networks [32]–[34]. There are only few works in the literature (e.g., [35], [36]) that, taking into account the local costs associated with replicating contents, consider selfish behavior of servers and propose non-cooperative game-theoretic caching mechanisms. All these works, however, consider Nash equilibrium as the solution concept for the formulated game, which makes the unrealistic assumption of independence among caching strategies of servers. In contrast, this paper considers the more realistic concept of game-theoretic equilibrium in adaptive learning, namely, correlated equilibrium [13], that captures the correlation amongst servers' learned caching strategies.

II. Revealed Preferences on the YouTube Social Network

In this section we introduce how the revealed preferences can be used to estimate the request probability of videos based on observing the behavior of YouTube users. The estimated request probability will then be used as a measure of popularity of videos in Sec. V to devise an adaptive popularity-based caching algorithm that provides caching prescriptions to individual servers.

The request probability of a video is dependent on users behavior which can be modeled using *utility functions*. For each user, the utility function represents the amount of satisfaction he/she receives from watching a video. It is in the self interest of rational users to attempt to maximize their respective utilities. A major contribution of this paper is the use of revealed preferences to detect if YouTube users are utility maximizers in their response to metadata, e.g., subscribers, thumbnail, author. The test is non-parametric—that is, explicit knowledge of the users' utility function is not required. To test for utility maximizing behavior, one must observe the response of users $\mathbf{x}_d^i \in \mathbb{R}^J$ on each edge server i to a probe signal $\mathbf{p}_d \in \mathbb{R}^J$ over a series of observation periods $d \in \{1, \ldots, T\}$ with J denoting the number of video categories considered for analysis. The content users upload to YouTube is classified into 15 categories which include: "Music","Entertainment", and "People and Blogs". Additionally, the users define meta-level data including the title, description, author, and thumbnail. There are several ways for a user to respond to a video on YouTube including viewing the video, posting a comment to the video, rating the video, and subscribing to the user who posted the video. Denoting $\mathbf{p}_d \in \mathbb{R}^J$ as the meta-level data for videos in the observation period d, and $\mathbf{x}_d^i \in \mathbb{R}^J$ as the associated response from users to $\mathbf{p}_d$ on server i in the YouTube social network, one of the key questions considered is: Given a series of data $\mathcal{D}^i = \{(\mathbf{p}_d, \mathbf{x}_d^i) : d \in \{1, 2, \ldots, T\}\}$, is it possible to detect if the users are *utility maximizers*? To satisfy the utility maximization, the data must have originated from:

$$\mathbf{x}_d^i(\mathbf{p}_d) \in \arg\max_{\mathbf{p}_d'\mathbf{x} \le B_d^i} \left\{ u^i(\mathbf{x}) \right\} \tag{1}$$

with $u^i(\mathbf{x})$ a non-satiated utility function, and B_d^i the response constraint for users served by server i. Non-satiation means that an increase in any element of response $\mathbf{x}$ results in the utility function increasing. The non-satiated assumption rules out trivial cases such as a constant utility function which can be optimized by any response. As shown by Diewert in [1], without local non-satiation, the problem (1) may have no solution. In (1), the budget constraint $\mathbf{p}_d'\mathbf{x} \le B_d^i$ denotes the total viewing time available to users connected to server i for selecting the response $\mathbf{x}_d^i$ to the probe $\mathbf{p}_d$. A detailed discussion of $\mathbf{p}_d$, $\mathbf{x}_d^i$, and B_d^i is provided in Sec. VI-A.

If the response of users satisfies utility maximization (1), then the preferences of the users can be represented by a utility function. The non-parametric feasibility test and statistical test presented in Sec. III can detect if $\mathcal{D}^i$ originated from a utility maximization process, and can be used to construct the associated utility function of the users. Given the constructed utility function, a non-parametric learning algorithm is used to estimate the request probability vector $\boldsymbol{\mu}_d^i = [\mu_d^i(1), \ldots, \mu_d^i(J)]$ for edge server i associated with videos on YouTube. The request probability represents the estimated demand for a video based on the preferences of users and the quality (i.e. meta-data) of the videos.

Remark 2.1: In application, the request probability (4) provides a prediction of the estimated request for new videos. If the request statistics of a video in the network has already been identified as popular (see [37]–[39] for details), then a weighting scheme can be employed to select how much of the cache

should be dedicated to popular content, and the content preferred by users served by the server. We do not consider the application of such a weighting scheme in this paper.

As an example, let us consider computing the request probability for users with a Cobb-Douglas utility function

$$u^i(\mathbf{x}) = \sum\nolimits_{j=1}^{J} \alpha^i(j) \ln\left[x(j)\right], \tag{2}$$

where $\alpha^i(j)$ represents the preference of users connected to server i to request video j. For a user with utility function (2), and viewing budget B_d^i on day d, the user request $\mathbf{x}_d^i$ is computed by substituting (2) into (1) and is given by:

$$x_d^i(j) = \alpha^i(j) \frac{B_d^i}{p_d(j)}, \qquad \forall d \in \{1, 2, \ldots, T\}. \tag{3}$$

It is a formidable task to estimate the popularity of a video. Therefore, we compute the request probability of a video, which does not require the viewing budget B_d^i to be known. The request probability $\mu_d^i(j)$ on day d for videos in category j is computed as the ratio of requests for videos in category j over all the video requests of the user. The request probabilities associated with (3) are then given by

$$\begin{aligned} \mu_d^i(j) &= \left[1 + \sum\nolimits_{\substack{1 \le k \le J \\ k \ne j}} \frac{x_d^i(j)}{x_d^i(k)}\right]^{-1} \\ &= \left[1 + \sum\nolimits_{\substack{1 \le k \le J \\ k \ne j}} \frac{\alpha^i(k) p_d(j)}{\alpha^i(j) p_d(k)}\right]^{-1}. \end{aligned} \tag{4}$$

Note that the utility function (2) and the associated request function (3) are only introduced as examples. The non-parametric feasibility and detection test for utility maximization, and the non-parametric learning algorithm for request probabilities do not require the utility function of the users to be defined.

III. Request Probability From Revealed Preferences

In this section we provide a non-parametric feasibility and statistical test to detect utility maximizing behavior among a group of users served by the same edge server in the YouTube social network. A non-parametric learning algorithm is then provided for utility maximizing users which computes the request probabilities for YouTube videos. The request probabilities are then fed into the game-theoretic adaptive caching algorithm in Sec. V to optimally cache videos on edge servers based on their users behavior.

A. Feasibility Test for Utility Maximization

Here, a non-parametric feasibility test is provided to detect if a dataset $\mathcal{D}^i$ satisfies utility maximization (1). The celebrated "Afriat's theorem," summarized below provides a necessary and sufficient condition for a finite dataset $\mathcal{D}^i$ to have originated from an utility maximizer.

Theorem 3.1 (Afriat's Theorem): Given a dataset $\mathcal{D}^i = \{(\mathbf{p}_d, \mathbf{x}_d^i) : d \in \{1, 2, \ldots, T\}\}$, the following statements are equivalent:

1) The agent is a utility maximizer and there exists a non-satiated and concave utility function that satisfies (1).
2) For scalars u_d^i and $\lambda_d^i > 0$, the set of inequalities
$$u_\tau^i - u_d^i - \lambda_d^i \mathbf{p}_d'(\mathbf{x}_\tau^i - \mathbf{x}_d^i) \le 0 \tag{5}$$
has a feasible solution for $d, \tau \in \{1, \ldots, T\}$.
3) A non-satiated and concave utility function that satisfies (1) is given by:
$$u^i(\mathbf{x}) = \min_{d \in \{1,2,\ldots,T\}} \{u_d^i + \lambda_d^i \mathbf{p}_d'(\mathbf{x} - \mathbf{x}_d^i)\}. \tag{6}$$
4) The dataset $\mathcal{D}^i$ satisfies the "generalized axiom of revealed preference" (GARP), namely, for any $k \le T$,
$$\mathbf{p}_d' \mathbf{x}_d^i \ge \mathbf{p}_d' \mathbf{x}_{d+1}^i \quad \text{for all} \quad d \le k-1 \implies \mathbf{p}_k' \mathbf{x}_k^i \le \mathbf{p}_k' \mathbf{x}_1^i.$$

As pointed out in Varian's influential paper [10], a remarkable feature of Afriat's theorem is that, if the dataset can be rationalized by a non-trivial utility function, then it can be rationalized by a continuous, concave, and monotonic utility function. "Put another way, violations of continuity, concavity, or monotonicity cannot be detected with only a finite number of demand observations".

Verifying GARP (statement 4 of Theorem 3.1) on a dataset $\mathcal{D}^i$ comprising T points can be done using Warshall's algorithm with $O(T^3)$ [10], [40] computations. Alternatively, determining if Afriat's inequalities (5) are feasible can be done via a LP feasibility test, e.g., using interior point methods [41]). Note that the utility function in (6) is not unique and is ordinal by construction. Ordinal means that any monotone increasing transformation of the utility function will also satisfy Afriat's theorem. Therefore, the utility function mimics the ordinal behavior of humans. Geometrically, the estimated utility (6) is the lower envelop of a finite number of hyperplanes that is consistent with the dataset $\mathcal{D}^i$.

B. Statistical Test for Utility Maximization

In real world applications, a dataset $\mathcal{D}^i$ may fail the utility maximization test (6) as a result of the response $\mathbf{x}_d^i$ being corrupted by noise. Here, a statistical test is provided to detect utility maximizing behavior using a noisy dataset.

Consider the noisy dataset from M users served by the same edge server i:

$$\mathcal{D}_{\text{obs}}^i = \left\{\left(\mathbf{p}_d, \mathbf{y}_d^i\right) : d \in \{1, 2, \ldots, T\}\right\}, \tag{7}$$

consisting of probe signals $\mathbf{p}_d$, and noisy observations of responses $\mathbf{y}_d^i = \mathbf{x}_d^i + \mathbf{w}_d^i$, where $\mathbf{x}_d^i$ is the response of users on server i, and $\mathbf{w}_d^i$ represents the additive noise. The goal is to devise a feasibility test to determine if the clean dataset $\mathcal{D}^i = \left\{\left(\mathbf{p}_d, \mathbf{x}_d^i\right) : d \in \{1, 2, \ldots, T\}\right\}$ satisfies the utility maximization test (6). Let H_0 and H_1 denote the null hypothesis that the clean dataset $\mathcal{D}$ satisfies utility maximization, and the alternative hypothesis that $\mathcal{D}$ does not satisfy utility maximization, respectively. In devising a statistical test for H_0 vs H_1, there are two possible sources of error:

$$\begin{aligned} &\textbf{Type-I error:} \text{ Reject } H_0 \text{ when } H_0 \text{ is valid,} \\ &\textbf{Type-II error:} \text{ Accept } H_0 \text{ when } H_0 \text{ is invalid.} \end{aligned} \tag{8}$$

Given a noisy dataset $\mathcal{D}^i_{\text{obs}}$, the following statistical test can be used to detect if users connected to server i select responses that satisfy utility maximization (1):

$$\int_{\Phi^*\{\mathbf{y}^i\}}^{+\infty} f^i_\chi(\varphi)\mathrm{d}\varphi \underset{H_1}{\overset{H_0}{\gtrless}} \gamma. \tag{9}$$

Here,

(i) γ is the "significance level" of the statistical test;

(ii) the "test statistic" $\Phi^*\{\mathbf{y}^i\}$ is the solution of the following constrained optimization problem for scalars $\overline{u}^i_d, \lambda^i_d > 0$:

$$\begin{aligned} \min \quad & \Phi \\ \text{s.t.} \quad & \overline{u}^i_\tau - \overline{u}^i_d - \lambda^i_d \mathbf{p}'_d(\mathbf{y}^i_\tau - \mathbf{y}^i_d) - \lambda^i_d \Phi \le 0, \\ & \lambda^i_d > 0, \quad \Phi \ge 0 \ \text{ for } \ d, \tau \in \{1, 2, \ldots, T\}. \end{aligned} \tag{10}$$

and $\mathbf{y}^i$ denotes the time series of noisy observation of responses of all M users served by server i for the observation period $\{1, 2, \ldots, T\}$.

(iii) f^i_χ is the probability density function of the random variable

$$\chi \equiv \max_{d,\tau} \left[\mathbf{p}'_d \left(\mathbf{w}^i_d - \mathbf{w}^i_\tau\right)\right].$$

The following theorem characterizes the performance of the statistical detection test (9).

Theorem 3.2: Consider the noisy dataset $\mathcal{D}^i_{\text{obs}}$ in (7) of probe signal and responses, and recall H_0 and H_1 from (8). The probability that the statistical test (9) yields a Type-I error (rejects H_0 when it is true) is less then γ.

Proof: See [42] for the detailed proof. ■

Note that (10) is non-convex due to $\lambda^i_d \Phi$; however, since the objective function is given by the scalar Φ, for any fixed value of Φ, (10) becomes a set of linear inequalities allowing feasibility to be straightforwardly determined [42].

If the dataset $\mathcal{D}^i_{\text{obs}}$ satisfies (10), then it is desirable to estimate the associated utility function (6). From [43], it is possible to estimate the lower bound on the additive noise allowing the estimation of the parameters $\{u^i_d, \lambda^i_d\}$ in (6). This involves solving the following quadratically constrained quadratic program:

$$\begin{aligned} \min \quad & \textstyle\sum_{d=1}^{T} \sum_{j=1}^{J} (\mathbf{y}^i_d(j) - \boldsymbol{\eta}^i_d(j))^2 \\ \text{s.t.} \quad & u^i_\tau - u^i_d - \lambda^i_d \mathbf{p}'_d[(\mathbf{y}^i_\tau - \mathbf{y}^i_d) - (\boldsymbol{\eta}^i_d - \boldsymbol{\eta}^i_\tau)] \le 0, \\ & \lambda^i_d > 0 \quad \text{for} \quad d, \tau \in \{1, 2, \ldots, T\}. \end{aligned} \tag{11}$$

In (11), $\boldsymbol{\eta}^i_d$ denotes the minimum deviation of the observed data $\mathbf{y}^i_d$ necessary for $\mathbf{x}^i_d$ to have originated from users satisfying utility maximization (i.e. $\boldsymbol{\eta}^i_d$ is minimum estimate of $\mathbf{w}^i_d$). Note that (11) will always have a feasible solution. Therefore, prior to applying (11) to construct the utility function, the dataset $\mathcal{D}^i_{\text{obs}}$ must pass the statistical test (9).

C. Non-Parametric Learning Algorithm for User Request Probabilities

For a dataset that satisfies the feasibility test (5) or statistical test (10), this subsection provides an algorithm to compute the request probability of videos based on behavior of user served by a particular edge server. A non-parametric learning algorithm is provided which uses the estimated utility function to predict future demand for different blocks of videos from which the request probability for each block can be computed. These blocks may, for example, represent videos from different video categories such as music, entertainment, people and blogs, etc.

Using (6), the forecasted demand of a utility maximizing user is computed by substituting (6) into (1). Given a dataset that satisfies utility maximization (1), the parameters u^i_d and λ^i_d for each edge server i can be computed using (5) or (11). To compute the video demand of users connected to server i, denoted by $\mathbf{x}^i_o$, for a given resource budget B^i_o and video quality $\mathbf{p}_o$, the following optimization problem can be used:

$$\begin{aligned} \mathbf{x}^i_o = \mathbf{x}^i(\mathbf{p}_o, B^i_o) \in \arg\max_{\mathbf{x}} & \left[\min_{d\in\{1,\ldots,T\}} \{u^i_d + \lambda^i_d \mathbf{p}'_d(\mathbf{x} - \mathbf{x}^i_d)\}\right] \\ \text{s.t.} \quad & \mathbf{p}'_o \mathbf{x} \le B^i_o, \quad \mathbf{x} \ge \mathbf{0}. \end{aligned} \tag{12}$$

The optimization problem (12) is a linear program with a piecewise linear objective and can be solved in polynomial time complexity. The algorithm used to compute the request probabilities $\boldsymbol{\mu}^i_o$ based on user behavior is provided in Algorithm 1.

Algorithm 1. Learning Request Probabilities

Step 0: Select a set of videos to compute the probe $\mathbf{p}_o \in \mathbb{R}^J_+$ for estimating the associated request probability $\boldsymbol{\mu}^i_o \in \mathbb{R}^J$.

Step 1: For a dataset $\mathcal{D}^i$ compute the parameters u^i_d and λ^i_d using (5). If the dataset is measured in noise $\mathcal{D}^i_{\text{obs}}$, then test for utility maximization using (10). If $\mathcal{D}^i_{\text{obs}}$ satisfies utility maximization then compute the parameters u^i_d, λ^i_d, and $\boldsymbol{\eta}^i_d$ using (11).

Step 2: Solve the following linear programming problem given u^i_d, λ^i_d, and $\boldsymbol{\eta}^i_d$ from Step 1 and $\mathbf{p}_o$:

$$\begin{aligned} \max \quad & z \\ \text{s.t.} \quad & z \le u^i_d + \lambda^i_d \mathbf{p}'_d(\mathbf{x}^i_o - \mathbf{y}^i_d + \boldsymbol{\eta}_d), d^i \in \{1, \ldots, T\} \\ & \mathbf{p}'_o \mathbf{x}^i_o \le 1, \quad \mathbf{x}^i_o \ge \mathbf{0} \end{aligned} \tag{13}$$

to obtain the forecasted demand $\mathbf{x}^i_o$ for a user with resource budget $B^i_o = 1$. Note that, if the noise is zero, then $\boldsymbol{\eta}^i_d = \mathbf{0}$ and $\mathbf{y}^i_d = \mathbf{x}^i_d$ in (13).

Step 3: Compute the request probabilities $\boldsymbol{\mu}^i_o$ using $\mathbf{x}^i_o$ computed from Step 2 using (4).

A key question is when does the non-parametric learning algorithm (13) admit a sufficiently accurate estimate of the request probability? Equivalently, what demand functions $\mathbf{x}^i(\mathbf{p}, B^i)$ are *probably approximately correct* (PAC) learnable using the algorithm (13)? Let $\mathcal{C}$ denote the set of possible demand functions under consideration. Below, we formally defines the set of *efficiently* PAC-learnable demand functions [44].

Definition 3.1: A set of demand functions $\mathcal{C}$ is *efficiently* PAC-learnable if for any $\epsilon, \delta > 0$, $h \in \mathcal{C}$, and probability distribution $\mathbb{P}$ on the probes and responses, there exists a polynomial

time algorithm for a set of observations to find a demand function h such that

$$\mathbb{E}_{(\mathbf{p},\mathbf{x})}(\|h(\mathbf{p}, B) - \mathbf{x}(\mathbf{p}, B)\|_\infty^2) < \epsilon$$

with probability $1 - \delta$, where $\mathbf{x}(\mathbf{p}, B)$ is the actual demand, and $\mathbb{E}_X$ denotes expectation with respect to X.

As shown in [45], the set of all demand functions induced by monotone concave utility functions is too rich to be efficiently PAC-learnable. Further assumptions are necessary to ensure PAC-learnability of the demand function $\mathbf{x}(\mathbf{p}, B)$. It is shown in [45], [46] that the class of *income-Lipschitz* demand functions is efficiently PAC-learnable.

Definition 3.2: A demand function $\mathbf{x}(\mathbf{p}, B) \in \mathbb{R}^J$ is *income-Lipschitz* if for every strictly positive vector $\mathbf{p} \in \mathbb{R}_+^J$, there exist positive reals $K > 0$ and $\vartheta > 0$ such that if $(\mathbf{p}, \mathbf{p}') \in P \subset \mathbb{R}_+^J$, $\|\mathbf{p} - \mathbf{p}'\| < \vartheta$, then

$$\|\mathbf{x}(\mathbf{p}', B) - \mathbf{x}(\mathbf{p}', B')\| \leq K. \tag{14}$$

The requirement of the demand function to have the income-Lipschitz property does not impose any parametric assumption on the utility function of the users served by server i. The income-Lipschitz (3.2) property merely imposes a stability assumption on the demands which rules out dramatically different requests for similar video qualities.

IV. Limited Capacity Caching Problem in YouTube Content Distribution Network

The remainder of this paper is devoted to the second cognitive functionality, namely, a distributed game-theoretic caching algorithm that prescribes individual servers which videos to cache and adapts its prescriptions as the preferences of users in each region evolve over time. To this end, Sec. IV-A formulates the non-cooperative game-theoretic model for the limited capacity caching problem in a YouTube CDN. Section IV-B then elaborates on a prominent solution concept in non-cooperative games, namely, correlated equilibrium.

A. Competitive Caching Game

We consider a content distribution network comprising of I edge servers, indexed by $i \in \mathcal{I} = \{1, 2, \ldots, I\}$, which are equipped with cognitive learning and processing power. Each server i has a limited storage capacity of N_i Mb, and has to repeatedly make decisions as which blocks of videos to cache. These blocks may, for example, comprise of popular videos of different categories such as music, entertainment, people and blogs, etc., and are indexed by $\mathcal{V} = \{1, 2, \ldots, J\}$. Let s_j denote the size of the j-th block of videos. Then, taking into account the storage capacity constraint, the set of feasible caching strategies available to server i can be expressed by the set

$$\mathcal{V}_f^i = \left\{\mathcal{W} \in 2^{\mathcal{V}} - \emptyset; \textstyle\sum_{j\in\mathcal{W}} s_j \leq N_i\right\}, \tag{15}$$

where $2^{\mathcal{V}}$ denotes the power set of set $\mathcal{V}$. For simplicity of presentation, we index the elements of $\mathcal{V}_f^i$, and work with the indices set instead, denoted by $\mathcal{A}^i = \{1, 2, \ldots, A^i\}$. In what follows, use $a^i \in \mathcal{A}^i$ to denote a generic caching decision made by server i.

Most caching strategies in the literature are designed assuming edge servers are willing to cooperate with each other to optimize the overall CDN performance. These mechanisms assume no cognitive capability for servers. They fail to capture the local costs for replicating objects and the interest of servers to optimize performance of local users, which may indeed result in servers' selfish behavior. This section focuses on modeling and analysis of such selfish caching behavior of servers in a YouTube CDN.

Let $d_{ii'}$ denote the distance between any two servers i and i', which can be summarized in a distance matrix $D = [d_{ii'}]$ with dimension $I \times I$. Such a distance matrix models an underlying network topology, and satisfies the following properties:

1) Symmetry: $d_{ii'} = d_{i'i}$ for any $i, i' \in \mathcal{I}$;
2) Triangle inequality: $d_{ii'} + d_{i'i''} \geq d_{ii''}$ for all $i, i', i'' \in \mathcal{I}$.

Let the vector $\mathbf{a} = (a^1, \cdots, a^I) \in \mathcal{A}^{\mathcal{I}}$ represent a generic caching *decision profile* of all servers, where $\mathcal{A}^{\mathcal{I}} = \times_{i=1}^I \mathcal{A}^i$ and $\times$ denotes the Cartesian product. One can then rearrange this vector as $\mathbf{a} = (a^i, \mathbf{a}^{-i})$, where $\mathbf{a}^{-i}$ denotes the joint decision profile of all servers excluding server i. The cost function that each server aims to minimize is formulated as follows:

$$\begin{aligned} C^i(a^i, \mathbf{a}^{-i}, \boldsymbol{\mu}_o^i) = {} & k_c^i \textstyle\sum_{j\in\mathcal{S}(a^i)} c^i(j) \\ & + k_l^i \, \nu^i \textstyle\sum_{j\notin\mathcal{S}(a^i)} \mu_o^i(j) \cdot d_{i\ell_j^i(\mathbf{a}^{-i})}, \end{aligned} \tag{16}$$

where $c^i(j)$ is the storage cost of the j-th block of videos, and $\mathcal{S}(a^i)$ represents the set of video blocks to which the chosen action a^i refers. Further, ν^i is the aggregate rate at which server i receives video requests, and $\boldsymbol{\mu}_o^i = [\mu_o^i(1), \ldots, \mu_o^i(J)]$ is the request probability vector computed via Algorithm 1. Each element $\mu_o^i(j)$ in this vector reflects the *popularity* of the j-th block of videos at the region supported by server i. $\ell_j^i(\mathbf{a}^{-i})$ also returns the index of the server with the shortest distance to server i that has cached video j. Clearly, such a distance is a function of the caching decisions of other servers, namely, $\mathbf{a}^{-i}$. When no server chooses to cache the video, we set $d_{i\ell_j^i(\mathbf{a}^{-i})} = d_S$, where d_S denotes the distance to the main YouTube server.

In (16), the first term is the *storage cost* and the second term is the *latency cost* associated with accessing remote replicas of the requested videos. The parameters k_c^i and k_l^i are the relative weights (importance) that server i places on the storage versus latency costs. The storage cost can further be defined as

$$c^i(j) = \beta_c^i \frac{s_j}{\mu_o^i(j)} + \beta_m^i \sum_{l\in\mathcal{I}} \mu_o^l(j)\nu^l \tag{17}$$

where the first term is the first time *caching cost*, which depends on the size of the videos and is inversely proportional to the popularity of videos in block j at the location of server i, and the second term is the *maintenance costs* associated with serving users at the location of server i as well as users redirected by other servers. The constants β_c^i and β_m^i further determine the relative weights of the caching and maintenance costs in the storage cost of server i. Note that the more popular the videos in

the j-th block are among users of server i, the more motivated will be server i to cache block j.

Remark 4.1: The second term in (16) could either represent the penalty for additional latency incurred when delivering videos to users from remote caches, or the bandwidth required for retrieving videos from them [47].

By minimizing the cost function (16), the servers aims to establish a balance between the latency in delivering videos from remote servers and the costs associated with caching videos. If a video is cached in a server nearby the server that has received the request, it may be better off accessing the remote replica. On the other hand, if all replicas are located far away, the server is better off caching the video itself. Therefore, the caching decisions of servers are interdependent. This is the key feature that characterizes our study as a non-cooperative game, defined by

$$\mathcal{G} = \left(\mathcal{I}, (\mathcal{A}^i)_{i\in\mathcal{I}}, (C^i)_{i\in\mathcal{I}}, \boldsymbol{\mu}_o\right). \tag{18}$$

Here, $\boldsymbol{\mu}_o = [\boldsymbol{\mu}_o^1, \ldots, \boldsymbol{\mu}_o^I]$ collects the request probability vectors of all servers, and is referred to as the *request probability profile*, $\mathcal{I}$ is the set of players, $\mathcal{A}^i$ denotes the action set for each player i, and the function C^i determines the cost associated with picking each action by player i.

B. Correlated Equilibrium

Here, we focus on correlated equilibrium [13] as the solution to the non-cooperative game formulated in Sec. IV-A. The set of correlated equilibria for each request probability profile $\boldsymbol{\mu}_o$ is defined as follows:

Definition 4.1 (Correlated Equilibrium) Let $\boldsymbol{\pi}$ denote a joint distribution on the joint action space $\mathcal{A}^{\mathcal{I}}$, i.e.,

$$\pi(\mathbf{a}) \geq 0, \ \forall \mathbf{a} \in \mathcal{A}^{\mathcal{I}}, \text{ and } \textstyle\sum_{\mathbf{a}\in\mathcal{A}^{\mathcal{I}}} \pi(\mathbf{a}) = 1.$$

The set of *correlated equilibria* $\mathcal{Q}(\boldsymbol{\mu}_o)$ for each $\boldsymbol{\mu}_o$ is the convex polytope[2] [see (19), shown at the bottom the page], where $\pi^i(l, \mathbf{a}^{-i})$ denotes the probability that server i picks action l and the rest of servers pick the caching profile $\mathbf{a}^{-i}$.

Several reasons motivate adopting the correlated equilibrium in large-scale cognitive CDNs. First, it is structurally and computationally simpler than the well-known Nash equilibrium. Second, the coordination among servers in the correlated equilibrium can lead to potentially lower individual costs of servers than if servers take their actions independently, as required by Nash equilibrium [13]. Finally, it is more realistic as observing the common history of cached videos in cognitive learning scenarios naturally correlates servers future caching decisions [11].

An intuitive interpretation of correlated equilibrium is "global coordination" in caching decisions of servers. Suppose a mediator observes the caching game being repeatedly played among servers in a CDN. The mediator, at each period, gives private recommendations as which videos to cache to each server. The recommendations are correlated as the mediator draws them from a joint probability distribution on the set of all feasible caching decisions of all servers; however, each server is only given recommendations about its own caching decision. Each server can then freely interpret the recommendations and decide if to follow. A correlated equilibrium results if no server wants to deviate from the provided recommendation. That is, in correlated equilibrium, servers' caching decisions are coordinated as if there exists a centralized coordinating device that all servers trust to follow.

[2]Note that (19) takes into account the fact that servers aim to minimize their cost functions, which is in contrast to the usual treatment in non-cooperative games where players aim to maximize their payoffs.

V. Adaptive Preference-Based Video Caching Algorithm

This section presents the adaptive popularity-based video caching algorithm that combines a regret-matching learning functionality [4], [5] with the non-parametric request probability learning procedure, summarized in Algorithm 1. This algorithm enables servers to adapt their caching decisions to their perception of the popularity of videos in their locale, and learn from their past caching experiences as well as interacting and exchanging information with the rest of the network. It will be shown that it facilitates coordination among the edge servers across the CDN to strategically minimize their cost functions, yet obtain a sophisticated and rational global behavior at the network level, namely, convergence to correlated equilibrium.

A. Regret-Based Adaptive Learning for Video Caching

Time is discrete $n = 1, 2, \ldots$. Each server i periodically makes caching decisions $a_n^i \in \mathcal{A}^i$ to strategically minimize local cost function (16) according to a strategy $\boldsymbol{\psi}_n^i$ that relies on the server's belief matrix $R_n^i = [r_n^i(l, l')]$ with dimension $A^i \times A^i$. Recall from Sec. IV-A that A^i represents the cardinality of the set $\mathcal{V}_f^i$, which denotes the set of feasible caching strategies available to server i. Each element $r_n^i(l, l')$ records the discounted time-averaged regrets—increase in the accrued costs—had the server selected caching decision l' every time it chose l in the past, and is updated via the recursive expression [see (20) at the bottom of the page]. In (20), $0 < \varepsilon \ll 1$ is a small parameter that represents the adaptation rate of the

$$\mathcal{Q}(\boldsymbol{\mu}_o) = \left\{\boldsymbol{\pi} : \textstyle\sum_{\mathbf{a}^{-i}} \pi^i\left(l, \mathbf{a}^{-i}\right)\left[C^i\left(l, \mathbf{a}^{-i}, \boldsymbol{\mu}_o^i\right) - C^i\left(l', \mathbf{a}^{-i}, \boldsymbol{\mu}_o^i\right)\right] \leq 0, \ \forall l, l' \in \mathcal{A}^i, i \in \mathcal{I}\right\} \tag{19}$$

$$r_{n+1}^i(l, l') = r_n^i(l, l') + \varepsilon\left[\left[C^i\left(l, \mathbf{a}^{-i}, \boldsymbol{\mu}_o^i\right) - C^i\left(l', \mathbf{a}^{-i}, \boldsymbol{\mu}_o^i\right)\right] \cdot I\left(a_n^i = l\right) - r_n^i(l, l')\right] \tag{20}$$

strategy update procedure, and is required as the user preferences slowly evolve over time [12]. Further, $I(X)$ denotes the indicator operator: $I(X) = 1$ if statement X is true, and 0 otherwise.

Let $\mathbf{1}_M$ denote a vector of ones of size M, and define $|x|^+ = \max\{0, x\}$. The adaptive popularity-based caching algorithm is summarized below in Algorithm 2. The two timescales in Algorithm 2 can be explained as follows: Since the user preferences change on a slow timescale spanning several months [7], [8], each server i collects responses of users over a month and runs Algorithm 1 to obtain the request probability vector $\boldsymbol{\mu}_o^i$ on a monthly basis. In contrast, the caching prescriptions a_n^i made by Algorithm 2 can be updated more frequently (even on an hourly basis) since no physical restriction limits the caching process in servers. The subscript n in Algorithm 2 represents the fast timescale, wherein $\boldsymbol{\mu}_{o,n}^i$ is a slow jump changing process. That is, it remains constant until T data points of user responses and probe signal are collected, at which point it jumps into its new value computed via Algorithm 1.

Algorithm 2. Adaptive Popularity-Based Caching Algorithm

Initialization: Set $\xi^i > A^i |C_{\max}^i - C_{\min}^i|$, where $C_{\max}^i$ and $C_{\min}^i$ denote the upper and lower bounds on the server i's cost function, respectively. Set the step-size $0 < \varepsilon \ll 1$, and T. Initialize

$$\boldsymbol{\psi}_0^i = (1/A^i) \cdot \mathbf{1}_{A^i}, \quad R_0^i = \mathbf{0}.$$

Step 1: Choose Action Based on Past Regret.

$$a_n^i \sim \boldsymbol{\psi}_n^i = \left(\psi_n^i(1), \ldots, \psi_n^i(A^i)\right),$$

where

$$\psi_n^i(l) = \begin{cases} \frac{1}{\xi^i} |r_n^i\left(a_{n-1}^i, l\right)|^+, & l \neq a_{n-1}^i, \\ 1 - \sum_{l' \neq l} \psi_n^i(l'), & l = a_{n-1}^i. \end{cases} \tag{21}$$

Step 2: Update Regret.

$$R_{n+1}^i = R_n^i + \varepsilon \left[F^i(a_n^i, \mathbf{a}_n^{-i}, \boldsymbol{\mu}_{o,n}^i) - R_n^i \right], \tag{22}$$

where $F^i = [f_{ll'}^i]$ is an $A^i \times A^i$ matrix with elements: [see (23) at the bottom of the page].

Step 3: Compute Request Probabilities.

If $|\mathcal{D}_{\text{obs}}^i| = T$, compute $\boldsymbol{\mu}_{o,n}^i$ via Algorithm 1, and set $\mathcal{D}_{\text{obs}}^i = \emptyset$. Otherwise, set $\boldsymbol{\mu}_{o,n+1}^i = \boldsymbol{\mu}_{o,n}^i$, collect new probe signal $\mathbf{p}_n$ and user responses $\mathbf{y}_n^i$, and update $\mathcal{D}_{\text{obs}}^i$:

$$\mathcal{D}_{\text{obs}}^i \leftarrow \mathcal{D}_{\text{obs}}^i \cup \left(\mathbf{p}_n, \mathbf{y}_n^i\right). \tag{24}$$

Recursion. Set $n \leftarrow n + 1$, and go Step 1.

B. Discussion and Intuition

Distinct properties of the adaptive popularity-based caching algorithm summarized in Algorithm 2 are as follows:

a) Adaptive behavior: In (22), ε introduces an exponential forgetting of the past caching experiences, and facilitates adaptivity to the evolution of the local user preferences over time. As servers successively make caching decisions, the effect of the old user preferences on their current decisions vanishes.

b) Caching decision strategy: Intuitively, the strategy (21) governs the caching decisions by propensities to depart from the current caching decision. It is natural to postulate that, if a server decides to switch its caching decision, it should be to those that are perceived as being better. The extent to which a caching decision l outperforms the last decision a_{n-1}^i is quantified with its regret $r_n^i(a_{n-1}^i, l)$. Positive regrets imply the opportunity to gain by switching from a_{n-1}^i to l. Therefore, the caching strategy (21) assigns positive probabilities to all actions l for which $|r_n^i(a_{n-1}^i, l)|^+ > 0$. In fact, the probabilities of switching to different actions are proportional to their regrets relative to the current action, hence the name "regret-matching" [4]. This procedure is viewed as particularly simple and intuitive as no sophisticated updating, prediction, or fully rational behavior is required by servers. Finally, equations (22)–(23) express the regret update recursion (20) in a closed matrix format.

The choice of ξ^i also guarantees that there is always a positive probability of picking the same action as the last period. Therefore, ξ^i can be viewed as an "inertia" parameter, which plays a significant role in breaking away from the so-called "bad cycles". This inertia is, in fact, the important factor that makes convergence to the correlated equilibria set possible under (almost) no structural assumptions on the underlying game model [4].

c) Computational and communication requirements: The computational complexity (in terms of calculations per iteration) of the "regret-matching" procedure in Steps 1–2 of Algorithm 2 is negligible. It does not grow with the number of servers, hence, is scalable. At each iteration, each server needs to execute two multiplications, two additions, one comparison and two table lookups (assuming random numbers are stored in a table) to calculate the next decision. Servers exchange their caching decisions after they are made at the end of each iteration. Other variants of Algorithm 2 can be devised that require no communication or communication only with immediate neighbors. These variants, however, require more time to establish coordination among servers' caching strategies.

d) Markov chain construction: The sequence $\left\{a_n^i, R_n^i\right\}$ is a Markov chain with state space $\mathcal{A}^i$ and transition probability matrix

$$\Pr\left(a_n^i = l | a_{n-1}^i = l'\right) = P_{l'l}\left(R_n^i\right).$$

The above transition matrix is continuous, irreducible and aperiodic for each R_n^i. It is further (conditionally) independent of other servers' caching decision profile, which may be correlated. The sample path $\{a_n^i\}$ is fed back into the stochastic

$$f_{ll'}^i(a_n^i, \mathbf{a}_n^{-i}, \boldsymbol{\mu}_{o,n}^i) = \left[C^i\left(l, \mathbf{a}_n^{-i}, \boldsymbol{\mu}_{o,n}^i\right) - C^i\left(l', \mathbf{a}_n^{-i}, \boldsymbol{\mu}_{o,n}^i\right) \right] \cdot I(a_n^i = l). \tag{23}$$

approximation algorithm that updates R_n^i, which in turn affects its transition matrix. This interpretation is useful in deriving the limit dynamics associated with Algorithm 2 in Appendix.

C. Asymptotic Local and Global Caching Behavior

Here, we present the main theorem that characterizes both the local and global behavior emerging from each server individually following Algorithm 2. The regret matrices R_n^i, $i \in \mathcal{I}$, will be used as indicatives of servers' local experiences at each time n. The global behavior $\mathbf{z}_n$ of the CDN at each time n is defined as the *empirical frequency* of joint caching decision profile of all servers $\mathbf{a}_\tau$, for $\tau \leq n$. Formally,

$$\mathbf{z}_n = (1-\varepsilon)^{n-1}\mathbf{e}_{\mathbf{a}_1} + \varepsilon \sum_{2\leq\tau\leq n}(1-\varepsilon)^{k-\tau}\mathbf{e}_{\mathbf{a}_\tau}, \qquad (25)$$

where $\mathbf{e}_{\mathbf{a}_\tau}$ is a unit vector on the space of all feasible caching decision profiles $\mathcal{A}^{\mathcal{I}}$ with the element corresponding to $\mathbf{a}_\tau$ being equal to one. The small parameter $0 < \varepsilon \ll 1$ is the same as the adaptation rate in (22), and introduces an exponential forgetting of the past decision profiles to enable adaptivity to the changes in popularity of videos. It is more convenient to define $\mathbf{z}_n$ via the stochastic approximation recursion

$$\mathbf{z}_n = \mathbf{z}_{n-1} + \varepsilon\left[\mathbf{e}_{\mathbf{a}_n} - \mathbf{z}_{n-1}\right]. \qquad (26)$$

Remark 5.1: The global behavior $\mathbf{z}_n$ is a network "diagnostic" and is only used for the analysis of the emerging collective caching patterns of servers. However, a network controller can monitor $\mathbf{z}_n$ and use it to adjust servers' cost functions to achieve the desired global behavior, e.g., in a cache cluster [47].

We use stochastic averaging [48] in order to characterize the asymptotic behavior of Algorithm 2. The basic idea is that, via a 'local' analysis, the noise effects in the stochastic algorithm is averaged out so that the asymptotic behavior is determined by that of a 'mean' dynamical system [48], [49]. To this end, in lieu of working with the discrete-time iterates directly, one works with continuous-time interpolations of the iterates. Accordingly, define the piecewise constant interpolated processes

$$R^{i,\varepsilon}(t) = R_n^i,\ \mathbf{z}^\varepsilon(t) = \mathbf{z}_n \quad \text{for} \quad t \in [n\varepsilon, (n+1)\varepsilon). \qquad (27)$$

With slight abuse of notation, denote by R_n^i the regret matrix rearranged as vectors of length $(A^i)^2$, rather than an $A^i \times A^i$ matrix, and let $R^{i,\varepsilon}(\cdot)$ represent the associated interpolated vector processes. The following theorem characterizes the local and global behavior emerging from following Algorithm 2.

Theorem 5.1: Let t_ε be any sequence of real numbers satisfying $t_\varepsilon \to \infty$ as $\varepsilon \to 0$. Then, as $\varepsilon \to 0$, the following results hold:

(i) If server i follows Algorithm 2, the asymptotic regret $R^{i,\varepsilon}(\cdot + t_\varepsilon)$ converges in probability to the negative orthant. That is, for any $\rho > 0$,

$$\lim_{\varepsilon\to 0} P\left(\text{dist}\left[R^{i,\varepsilon}(\cdot + t_\varepsilon), \mathbb{R}_-\right] > \rho\right) = 0 \qquad (28)$$

where dist$[\cdot,\cdot]$ denotes the usual distance function, and $\mathbb{R}_-$ represent the negative orthant in the Euclidean space of appropriate dimension.

(ii) If every server follows Algorithm 2, the global behavior $\mathbf{z}^\varepsilon(\cdot + t_\varepsilon)$ converges in probability to the correlated equilibria set $\mathcal{C}(\boldsymbol{\mu}_o)$ in the sense that

$$\text{dist}\left[\mathbf{z}^\varepsilon(\cdot + t_\varepsilon), \mathcal{C}(\boldsymbol{\mu}_o)\right] = \inf_{\mathbf{z}\in\mathcal{C}(\boldsymbol{\mu}_o)} \left\|\mathbf{z}^\varepsilon(\cdot + t_\varepsilon) - \mathbf{z}\right\| \to 0, \quad (29)$$

where $\|\cdot\|$ denotes the Euclidean norm.

Proof: See Appendix for a sketch of the proof. ■

The first result in the above theorem simply asserts that, if a server individually follows Algorithm 2, it will asymptotically experience zero regret in its caching decisions. Here, $R^{i,\varepsilon}(\cdot + t_\varepsilon)$ looks at the asymptotic behavior of R_n^i. From the technical point of view, the requirement $t_\varepsilon \to \infty$ as $\varepsilon \to \infty$ means that we look at R_n^i for a small ε but large n with $\varepsilon n \to \infty$. For a small ε, R_n^i eventually spends nearly all of its time (with an arbitrarily high probability) in an ρ-neighborhood of the negative orthant, where $\rho \to 0$ as $\varepsilon \to 0$. Note that, when ε is small, R_n^i may escape from such an ρ-neighborhood. However, if such an escape ever occurs, it will be a rare event. The order of the escape time is often of the form $\exp(c/\varepsilon)$ for some $c > 0$; see [48, Section 7.2] for details. The second result in Theorem 5.1 states that, if now all edge servers in the CDN start following Algorithm 2 independently, their collective behavior for a small step-size ε is in an ρ-neighborhood of the correlated equilibria set. That is, servers can coordinate their caching strategies in a distributed fashion so that the distribution of their joint caching behavior is close to that determined by a correlated equilibrium. From the game-theoretic point of view, it shows that simple and self-oriented local behavior of servers can still lead to the manifestation of globally sophisticated and rational behavior at the network level.

VI. Numerical Study

This section provides numerical examples to illustrate the performance of the proposed algorithms. Section VI-A verifies that YouTube users are utility maximizers and shows how the preferences of users can be estimated using data from over one million YouTube videos. Section VI-B then compares the performance of the proposed adaptive caching scheme with existing heuristics in the literature, and illustrates its adaptation to changes in preferences of users. All computations in Sec. VI-A are preformed using the commercially available software package (MATLAB 8.4, The MathWorks Inc., Natick, MA, 2014).

A. Real-World Dataset From YouTube

We utilize the YOUStatAnalyzer [50] dataset[3] to detect for utility maximization and to estimate the preferences of users. The YOUStatAnalyzer dataset was constructed over the period of July 2012 to September 2013 and contains meta-level data for over 1 million videos. Given that "Music" and "Entertainment" are mostly viewed on YouTube [51], these categories are selected for analysis. To apply Afriat's utility maximization test (5), we construct the probe $\mathbf{p}_d$ and

[3]http://www.congas-project.eu/youstatanalyzer-database

response signals $\mathbf{x}_d$ as follows. The probe is defined by $\mathbf{p}_d(j) = [disreputability]$, where $disreputability = 1/\#\text{subscribers}$ and $j = 1$ refers to videos in the "Music" category, and $j = 2$ to videos in the "Entertainment" category. Intuitively, the probe gives a measure of the quality of videos that the user is likely to upload[4]. The associated response taken by users is given by $\mathbf{x}_d(j) = [viewcount]$. The response constraint is expected to satisfy $B_d = \mathbf{p}_d'\mathbf{x}_d$ as the total view time of the users is constrained by $\sum_j x_d(j)$ during each epoch d. Note that the response constraint B_d depends on the homophily of users, diffusion of information in the YouTube social network, and external parameters (e.g., news sources, twitter, reddit). Recall from Sec. II and Sec. III that B_d does not need to be known for computing the utility function of users or the request probabilities of videos. The parameter d denotes the epoch over which the mean disreputability and views for each category is computed. Given that the YOUStatAnalyzer dataset spans from July 2012 to September 2013 period, we compute the mean disreputability and views for a total of 30 bi-weekly epochs. Therefore, the constructed dataset for analysis is given by $\mathcal{D} = \{(\mathbf{p}_d, \mathbf{x}_d) : d \in \{1, 2, \ldots, 30\}\}$.

Does there exist a utility function that users are maximizing and rationalizes the observed responses in $\mathcal{D}$? The dataset $\mathcal{D}$ passes the utility maximization test (5). This establishes that users on the YouTube social network are utility maximizers, and that their preferences do not change over a period of approximately 15 months. To gain insight into the users' preferences, the associated utility function is provided in Fig. 2. Using the constructed utility function, which category ("Music" or "Entertainment") do users prefer to view? The *marginal rate of substitution*[5] (MRS) can be used to determine the preferred category. Formally, the MRS of $x(1)$ for $x(2)$ is given by

$$\text{MRS}_{12} = \frac{\partial u/\partial x(1)}{\partial u/\partial x(2)}.$$

From the constructed utility function shown in Fig. 2, we find that $\text{MRS}_{12} < 1$ suggesting that the users prefer the "Entertainment" category over "Music." Put differently, the users are not willing to give up a view in the "Entertainment" category for one additional view in the "Music" category.

Given the constructed utility function shown in Fig. 2, how do the preferences affect the request probabilities in (4)? Consider a collection of new "Music" and "Entertainment" videos with an associated probe $\mathbf{p}_o = [1.9 \times 10^{-3}, 0.45 \times 10^{-3}]$, which shows that videos in the "Music" category are produced by users with an average of 526 subscribers while in the "Entertainment" category have an average of 2222 subscribers. Using (13) and the constructed utility function, the computed request probability (4) is given by $\boldsymbol{\mu}_o = [0.18, 0.82]$. As expected from the number of subscribers, the videos in the "Entertainment" category should be cached by the edge servers. A *naive* scheme would be to conclude that the videos associated with users with the lowest disreputability should be cached. This method, however, does not account for user preferences. Consider another probe $\mathbf{p}_o = [2.4 \times 10^{-3}, 3.8 \times 10^{-3}]$ corresponding to 415 and 265 average subscribers in the "Music" and "Entertainment" category, respectively. If preferences are neglected, then the naive method would cache the videos in the "Music" category. Using (13) and utility function, the estimated request probability for this probe is $\boldsymbol{\mu}_o = [0.41, 0.59]$—even though there are more subscribers in the "Music" category, the request probability is 18% higher for the videos in the "Entertainment" category.

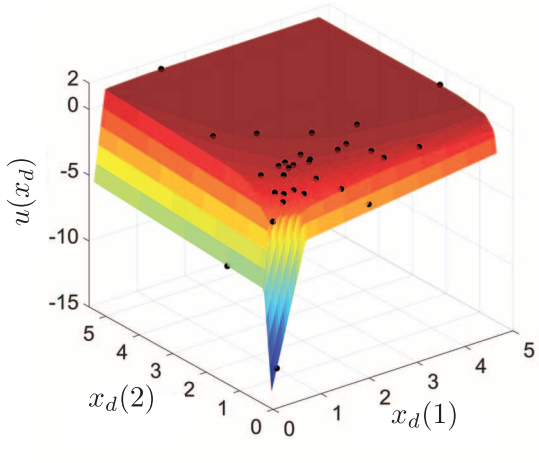

Fig. 2. Estimated utility function $u(\cdot)$ computed using (5) with the dataset $\mathcal{D}$ defined in Sec. VI-A. $\mathbf{x}_d(1)$ and $\mathbf{x}_d(2)$ denote the average number of views for the "Music" and "Entertainment" categories, respectively, with each unit denoting a thousand views. The black dots indicate the estimated utility at the experimentally measured views $\mathbf{x}_d$ in $\mathcal{D}$.

Can the Cobb-Douglas utility function (2) be utilized to compute the request probability of users? Using the dataset $\mathcal{D}$, the Cobb-Douglas preferences of users, $\alpha(1)$ and $\alpha(2)$ in (2), are estimated by minimizing the squared error of (3) with the observed responses $\mathbf{x}_d$ in $\mathcal{D}$. The results are provided in Fig. 3. As expected, the preference for the "Music" category is less than the "Entertainment" category with $\alpha(1) = 0.38$ and $\alpha(2) = 0.62$. From the results in Fig. 3 and the computed root mean squared error, the Cobb-Douglas utility provides a reasonable approximation of the preferences of users. The question is then: How accurate are the predicted request probabilities $\boldsymbol{\mu}_o$ from the Cobb-Douglas utility compared with the Algorithm 1? For $\mathbf{p}_o = [1.9 \times 10^{-3}, 0.45 \times 10^{-3}]$, the associated request probability from the Cobb-Douglas utility is $\boldsymbol{\mu}_o = [0.12, 0.88]$. Additionally, for $\mathbf{p}_o = [2.4 \times 10^{-3}, 3.8 \times 10^{-3}]$, the associated request probability is $\boldsymbol{\mu}_o = [0.5, 0.5]$. Generally, the request probabilities computed using the Cobb-Douglas utility has an error of approximately 10% compared to the request probabilities computed using Algorithm 1.

The analysis in this section suggests that users in the YouTube social network are utility maximizers. As we have shown, accounting for user preferences is critical for computing the request probability of videos. Note that about 32% of all YouTube videos tend to become popular by users in locally confined areas, and different YouTube categories exhibit different local popularity characteristics [54]. Therefore, for improved

[4]The only reliable method to assess the video quality perceived by a human observer is to ask human subjects for their opinion. Supervised machine learning techniques can then be applied to predict the perceived video quality for a YouTube video [52], [53].

[5]The amount of views that a user is willing to give up in exchange for another view while maintaining the same level of utility.

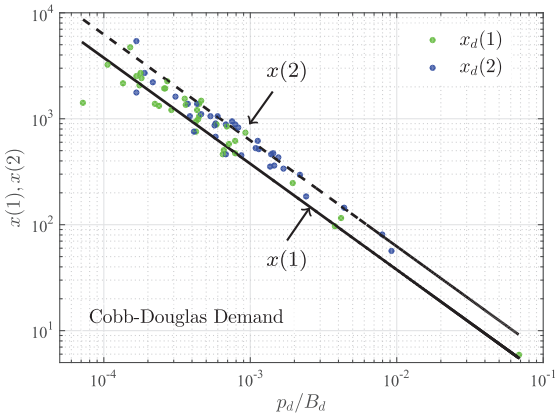

Fig. 3. The observed dataset $\mathcal{D}$ defined in Sec. VI-A and the numerically computed Cobb-Douglas demand function (3). The dots indicate the observed data, and the lines (solid for $\mathbf{x}(1)$ and dotted for $\mathbf{x}(2)$) the Cobb-Douglas demand functions. The units of $\mathbf{x}_d$ are the average number of video views in each category $j = 1$ for "Music" and $j = 2$ for "Entertainment", $\mathbf{p}_d/B_d$ is the normalized disreputability. The root mean squared error for $\mathbf{x}(1)$ is 815, and for $\mathbf{x}(2)$ is 516.

accuracy in application, the request probabilities should be computed locally at each edge server in the CDN using the methods outlined in Secs. II–III.

Remark 6.1: How can Algorithm 1 be applied to compute the request probabilities of content such as webpages, images, and audio? A key challenge for employing Algorithm 1 is how to compute the external influence $\mathbf{p}_d$ and the associated response $\mathbf{x}_d$. The response is straightforwardly defined as the number of requests for the content, however, specialized machine learning approaches are required to compute the external influence. Consider the quality of webpages as the external influence, then methods such as graph neural networks, Naive Bayes classification, and Random Forest learning methods can be utilized to estimate the quality of webages; see [55]–[57] for details. As the quality increases, the associated number of requests are expected to increase. With an ensemble dataset $\mathcal{D}$ of external influences and responses, the associated request probability for webpages can be computed using Algorithm 1.

B. Adaptive Video Caching

Here, we consider a CDN comprising of 10 edge servers equally distanced on a ring topology centered at the main server. The radius of the ring is chosen so that the main server is located at twice the distance between each edge server and the neighboring servers. To demonstrate adaptivity of Algorithm 2 and due to lack of enough real data, here it is assumed that the popularity of the categories of videos is governed by a Zipf-Mandelbrot distribution with shape parameter α and shift parameter q [27]. That is, the relative popularity of the j-th most popular category is given by:

$$\mu_o^i(j) = \frac{1}{H(\alpha, q, J)} \frac{1}{(q+j)^\alpha}, \quad j \in \mathcal{V}, i \in \mathcal{I}$$
$$\text{where} \quad H(\alpha, q, J) = \sum_{j=1}^{J} \frac{1}{(q+j)^\alpha}. \tag{30}$$

We set the shift parameter fixed at $q = 1$ throughout. It is assumed that there exist a collection of $J = 15$ categories of videos, and popular videos in each category are collected in batches of the same size $s_j = 20$ GB, $j = 1, \ldots, J$. Algorithm 2 is then used to decide which categories to cache in each server. We set $\nu^i = 0.1$ sec^{-1}, $k_c^i = 0.3$, $k_l^i = 7$, and the storage cost of all categories are normalized to $c^i(j) = 1$ since all categories consist of video blocks of the same size. Finally, we assume that each server has capacity $N_i = 1$ TB.

We use the following heuristics as benchmarks to compare the performance of Algorithm 2 in a static setting, where the request probability of videos are fixed:

e) Popularity-Based: Each server i sorts the videos in decreasing order of request probability $\mu_o^i(j)$, and stores as many videos in this order as its storage capacity N_i allows.

f) Myopic Greedy (Local) [47]: Each server i computes $\mu_o^i(j) \cdot d_{i\ell_j^i(\mathbf{a}_0^{-i})}$ for each video j given an initial caching decision of other servers $\mathbf{a}_0^{-i}$, and sorts them in decreasing order. Servers then cache as many videos as their storage capacity N_i allows according to this local ordering scheme.

g) Myopic Greedy (Global) [58]: The CDN controller first computes

$$\tilde{c}_{ij}(\mathbf{a}_0^{-i}) = \nu^i \mu_o^i(j) \cdot d_{i\ell_j^i(\mathbf{a}_0^{-i})}$$

for all server-video pairs (i, j) given an initial caching decision of other servers $\mathbf{a}_0^{-i}$. Once sorted in decreasing order, it picks the server-object pair (i, j) at the top of the list, informs server i to cache video j, and updates the caching decision profile $\mathbf{a}_{n+1}^{-i}$. Subsequently, the latency costs $\tilde{c}_{ij}(\mathbf{a}_{n+1}^{-i})$ are recomputed under the new placement $\mathbf{a}_{n+1}^{-i}$, and this procedure continues until the storage capacity of all servers are filled.

We conduct a semi-analytical study to evaluate performance of Algorithm 2. Recall from Theorem 5.1 that $\mathbf{z}_n$ converges to the set $\mathcal{C}(\boldsymbol{\mu}_{o,n})$ rather than a particular point in that set. In fact, even if the user preferences are fixed $\boldsymbol{\mu}_{o,n} = \boldsymbol{\mu}_o$, $\mathbf{z}_n$ can generally move around in the polytope $\mathcal{C}(\boldsymbol{\mu}_o)$. Since the game-theoretic adaptive caching mechanism is run on the faster timescale, its behavior in the slower timescale (learning user preferences) can be modeled by arbitrarily picking $\mathbf{z}_n$ from the polytope $\mathcal{C}(\boldsymbol{\mu}_{o,n})$. In our simulations, we use MATLAB's `cprnd` function, which draws samples $\boldsymbol{\pi}_n^c$ from the uniform distribution over the interior of the polytope $\mathcal{C}(\boldsymbol{\mu}_{o,n})$ defined by a system of linear inequalities[6]. Using a uniform distribution for sampling from $\mathcal{C}(\boldsymbol{\mu}_{o,n})$ is no loss of generality since it assumes no prior on the interior points of $\mathcal{C}(\boldsymbol{\mu}_{o,n})$, and matches the essence of convergence to a set. Once the sample $\boldsymbol{\pi}_n^c$ is taken, it is marginalized on each server to obtain individual server's activation strategy. This reduces the complexity of our numerical study to a one-level (rather than a two-level) stochastic simulation.

Figure 4 shows the average individual server's local costs (16) versus the local storage size of servers. As before, servers are assumed to have identical storage sizes. Further, it is assumed that all servers have identical shape parameters $\alpha = 1$

[6]Theoretically, finding a correlated equilibrium in a game is equivalent to solving a linear feasibility problem of the form $A\boldsymbol{\pi} < \mathbf{0}$; see (19) for details.

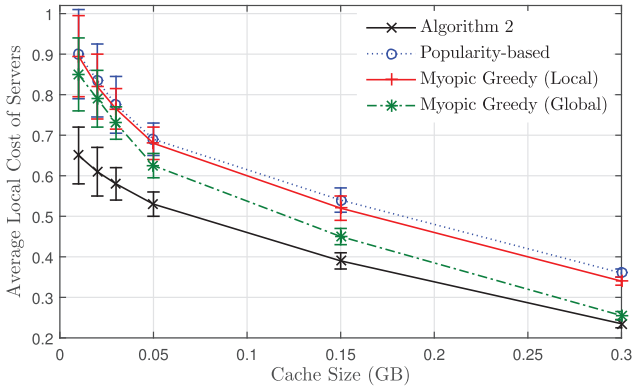

Fig. 4. Average local costs, defined by (16), versus the storage size of servers.

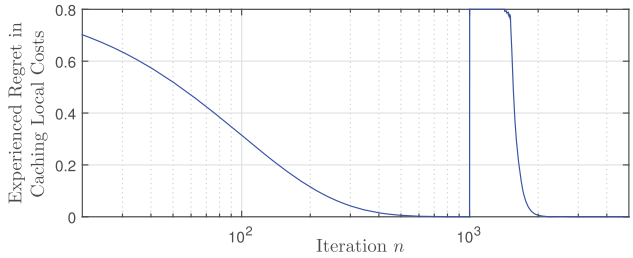

Fig. 5. Adaptive caching behavior of servers. The request probability of videos change at $n = 10^3$. The server adapts its caching behavior by learning from interaction with other servers.

in (30). Each point on the graph is an average over 100 independent runs of the algorithms. As the storage size of servers increases, the average cost of individual server decreases since the latency in retrieving videos from the main server decreases. As can be seen, the average caching behavior of individual servers determined by the correlated equilibria set of the game (18) incurs the least local cost to severs as compared with the heuristic methods existing in the literature. Recall that the local cost function of servers incorporates the local content placement cost and the latency cost associated with accessing remote replicas of the requested videos. Therefore, Fig. 4 confirms that the coordination among servers' caching strategies achieved in the correlated equilibrium results in faster delivery of videos and, hence, improved user experience. The following observations have been made while performing simulation in Fig. 4: On the lower end of the cache size axis, some video categories are cached in two servers across the diameter in the ring topology while others are cached in only one server. In contrast, on the upper end, some video categories are cached in every other server while others are cached in every server on the ring topology.

Figure 5 illustrates the average regret that a server experiences along the way to reach a correlated equilibrium, and further demonstrates adaptivity of the caching strategy implemented by Algorithm 2. At $n = 10^3$, the shape parameter α in (30) jumps from 1 to 3, and so does the request probabilities of videos. Therefore, the previous joint caching strategy of servers no longer belongs to the set of correlated equilibrium. As seen in Fig. 5, each server individually adjust its caching strategy so as to minimize its experienced regret once it realizes a change in popularity of videos. At the change time, the cached videos are no longer popular; hence, the experienced regret peaks at $n = 10^3$. As time passes by, servers detect such change and adapt to it through interaction with other servers; therefore, the experienced regret diminishes. Lower regret translates to more coordinated caching behavior of servers and, hence, more efficient use of cache resources. Recall that the speed of convergence of Algorithm 2 depends largely on parameters ξ and ε. Increasing one or both of these parameters will generally slow down the learning process; therefore, more time is required to achieve a ρ-distance of the zero axis in Fig. 5 or correlated equilibria set in the strategy space. Recall further that Algorithm 2 is run on the faster timescale; therefore, thousand iterations may only take up to several minutes after which the caching decisions will be fixed until the user preferences change, which is typically on the order of several days or months.

VII. Summary and Future Work

This paper presented a non-cooperative game-theoretic approach to the problem of distributed caching in a YouTube content distribution network, and considered two intertwined cognitive aspects: First, the user behavior that determines popularity of videos was studied using the theory of revealed preferences from the microeconomics literature. Second, a game-theoretic distributed caching algorithm was proposed that provided prescriptions as which videos to cache to individual servers taking into account the popularity of videos in its locale. It was shown that if all servers follow the proposed two timescale algorithm, their collective behavior at the network level tracks the correlated equilibria set of the underlying game as the popularities of videos change over time. This result is quite remarkable as it shows edge servers are capable to coordinate their caching strategies in a distributed fashion as if there exists a centralized coordinating device that they all trust to follow. In the numerical examples, using data from over 1 million YouTube videos, we illustrated how user behavior and quality of videos can be utilized to compute the request probability of videos—a critical component in the game-theoretic distributed caching algorithm.

The current paper focused on caching videos in a CDN. In practice, however, there are many CDNs that support various types of content such as webpages, images, and audio. A direction for future research is to consider the application of the proposed adaptive caching algorithm to such CDNs as well as information-centric networking. This involves measuring the quality of content using supervised machine learning approaches and observing the associated request statistics for the content. Another direction for future research is to extend the proposed formulation by considering transport energy and routing energy to develop energy-efficient caching mechanisms that the increasingly strict environmental standards and rapidly rising energy costs asks for in the context of green information-centric networking.

APPENDIX
PROOF OF THEOREM 5.1

The proof is based on [12], [33], [59]. For brevity, we only provide a sketch of the proof that comprises of three steps:

Step 1: We show that the limit system associated with the interpolated process $R^{i,\varepsilon}(\cdot)$ for each server i, as $\varepsilon \to 0$, is the differential inclusion[7]:

$$\frac{dR^i}{dt} \in H^i\left(R^i, \boldsymbol{\mu}_o^i\right) - R^i \tag{31}$$

where $H^i = [h_{ll'}^i]$ is an $A^i \times A^i$ matrix with elements: [see (32) at the bottom of the page]. In (32), $\boldsymbol{\Psi}^{-i}$ denotes the joint caching strategy of all servers excluding server i, and $\Delta\mathcal{A}^{-i}$ represents the simplex of all such strategies. Further, $\boldsymbol{\sigma}(R^i) = [\sigma_1(R^i), \ldots, \sigma_{A^i}(R^i)]$ is an invariant measure for the transition probabilities (21) satisfying

$$\textstyle\sum_{l\neq l'} \sigma_l\left(R^i\right)\left|r_{ll'}^i\right|^+ = \sum_{l\neq l'} \sigma_{l'}\left(R^i\right)\left|r_{l'l}^i\right|^+, \tag{33}$$

where $|x|^+ = \max\{0, x\}$. Note that the request probability vector $\boldsymbol{\mu}_o^i$ remains constant in (31). This is mainly due to the two timescale nature of the proposed scheme. It is well-known in the analysis of two timescale stochastic recursive algorithms that the slow timescale (non-parametric learning of request probabilities) is quasi-static while analyzing the fast component (regret-based adaptive caching); see [48, Chapter 8] for further details and related matters.

Step 2: We show global asymptotic stability of the limit system (31) via Lyapunov function methods, and characterize its set of global attractors by the negative orthant $\mathbb{R}_-$. More precisely,

$$\lim_{t\to\infty} \operatorname{dist}\left[R^i(t), \mathbb{R}_-\right] = 0, \tag{34}$$

where $\operatorname{dist}[\cdot,\cdot]$ denotes the usual distance function. This result is then combined with the results from Step 1 to show asymptotic stability of the interpolated process by showing that

$$\lim_{\varepsilon\to 0} \left\| |R^{i,\varepsilon}(\cdot + t_\varepsilon)|^+ \right\| = 0, \tag{35}$$

where t_ε represents any sequence of real numbers satisfying $t_\varepsilon \to \infty$ as $\varepsilon \to 0$, and the maximum in $|\cdot|^+$ is taken element-wise.

Step 3: The final step shows that, for each request probability vector $\boldsymbol{\mu}_o^i$, convergence of the interpolated regrets $R^{i,\varepsilon}(\cdot)$ to $\mathbb{R}_-$ for all servers $i \in \mathcal{I}$ provides the necessary and sufficient condition for convergence of interpolated global behavior vector $\mathbf{z}^\varepsilon(\cdot)$ to the set of correlated equilibria $\mathcal{C}(\boldsymbol{\mu}_o)$. Here, $\boldsymbol{\mu}_o = [\boldsymbol{\mu}_o^1, \ldots, \boldsymbol{\mu}_o^I]$ denotes the aggregate vector containing the request probability of all servers across the network. Finally, combining this result with the last result in Step 2 completes the proof.

For details of each step of the proof, the interested reader is referred to [60, Appendix B].

[7] A differential inclusion is of the form $dx/dt \in \mathcal{F}(x)$, where $\mathcal{F}(x)$ specifies a family of trajectories rather than a single trajectory as in the ordinary differential equations $dx/dt = f(x)$. Differential inclusions arise naturally in game-theoretic learning, since the strategies according to which others play are unknown.

REFERENCES

[1] W. Diewert, "Afriat and revealed preference theory," *Rev. Econ. Stud.*, vol. 40, no. 3, pp. 419–425, 1973.

[2] R. Blundell, "How revealing is revealed preference?," *J. Eur. Econ. Assoc.*, vol. 3, no. 2–3, pp. 211–235, 2005.

[3] W. Diewert, "Afriat's theorem and some extensions to choice under uncertainty," *Econ. J.*, vol. 122, no. 560, pp. 305–331, 2012.

[4] S. Hart and A. Mas-Colell, "A simple adaptive procedure leading to correlated equilibrium," *Econometrica*, vol. 68, no. 5, pp. 1127–1150, 2000.

[5] S. Hart and A. Mas-Colell, "A general class of adaptive strategies," *J. Econ. Theory*, vol. 98, no. 1, pp. 26–54, 2001.

[6] S. Hart, A. Mas-Colell, and Y. Babichenko, *Simple Adaptive Strategies: From Regret-Matching to Uncoupled Dynamics*. Singapore: World Scientific, 2013.

[7] R. Blundell, M. Browning, I. Crawford, B. Rock, F. Vermeulen, and L. Cherchye, "Sharp for SARP: Nonparametric bounds on the behavioural and welfare effects of price changes," Available: SSRN 2158658, 2012.

[8] L. Cherchye, B. Rock, and F. Vermeulen, "Opening the black box of intra-household decision making: Theory and nonparametric empirical tests of general collective consumption models," *J. Polit. Econ.*, vol. 117, no. 6, pp. 1074–1104, 2009.

[9] S. Afriat, "The construction of utility functions from expenditure data," *Int. Econ. Rev.*, vol. 8, no. 1, pp. 67–77, 1967.

[10] H. Varian, "The nonparametric approach to demand analysis," *Econometrica*, vol. 50, no. 1, pp. 945–973, 1982.

[11] S. Hart and A. Mas-Colell, "A reinforcement procedure leading to correlated equilibrium," in *Economics Essays: A Festschrift for Werner Hildenbrand*, Berlin, Heidelberg: Springer, 2001, pp. 181–200.

[12] O. N. Gharehshiran, V. Krishnamurthy, and G. Yin, "Distributed tracking of correlated equilibria in regime switching noncooperative games," *IEEE Trans. Autom. Control*, vol. 58, no. 10, pp. 2435–2450, Oct. 2013.

[13] R. J. Aumann, "Correlated equilibrium as an expression of Bayesian rationality," *Econometrica*, vol. 55, no. 1, pp. 1–18, 1987.

[14] A. Passarella, "A survey on content-centric technologies for the current internet: CDN and P2P solutions," *Comput. Commun.*, vol. 35, no. 1, pp. 1–32, 2012.

[15] J. Kangasharju, K. W. Ross, and J. W. Roberts, "Performance evaluation of redirection schemes in content distribution networks," *Comput. Commun.*, vol. 24, no. 2, pp. 207–214, 2001.

[16] L.-C. Chen and H.-A. Choi, "Approximation algorithms for data distribution with load balancing of web servers," in *Proc. IEEE Int. Conf. Cluster Comput.*, Newport Beach, CA, USA, 2001, p. 274.

[17] B. Wu and A. D. Kshemkalyani, "Objective-optimal algorithms for long-term web prefetching," *IEEE Trans. Comput.*, vol. 55, no. 1, pp. 2–17, Jan. 2006.

[18] Y. Guo, Z. Ge, B. Urgaonkar, P. Shenoy, and D. Towsley, "Dynamic cache reconfiguration strategies for cluster-based streaming proxy," *Comput. Commun.*, vol. 29, no. 10, pp. 1710–1721, 2006.

[19] T. Bektas, O. Oguz, and I. Ouveysi, "Designing cost-effective content distribution networks," *Comput. Oper. Res.*, vol. 34, no. 8, pp. 2436–2449, 2007.

[20] T. Bektaş, J.-F. Cordeau, E. Erkut, and G. Laporte, "Exact algorithms for the joint object placement and request routing problem in content distribution networks," *Comput. Oper. Res.*, vol. 35, no. 12, pp. 3860–3884, 2008.

[21] T. Bektaş, J.-F. Cordeau, E. Erkut, and G. Laporte, "A two-level simulated annealing algorithm for efficient dissemination of electronic content," *J. Oper. Res. Soc.*, vol. 59, no. 11, pp. 1557–1567, 2008.

$$h_{ll'}^i\left(R^i, \boldsymbol{\mu}_o^i\right) = \left\{\left[C^i\left(l, \boldsymbol{\Psi}^{-i}, \boldsymbol{\mu}_o^i\right) - C^i\left(l', \boldsymbol{\Psi}^{-i}, \boldsymbol{\mu}_o^i\right)\right]\sigma_l\left(R^i\right); \boldsymbol{\Psi}^{-i} \in \Delta\mathcal{A}^{-i}\right\}. \tag{32}$$

[22] C. Fang, F. R. Yu, T. Huang, J. Liu, and Y. Liu, "A survey of energy-efficient caching in information-centric networking," *IEEE Commun. Mag.*, vol. 52, no. 11, pp. 122–129, Nov. 2014.
[23] M. Zhang, H. Luo, and H. Zhang, "A survey of caching mechanisms in information-centric networking," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 3, pp. 1473–1499, Apr. 2015.
[24] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proc. 5th Int. Conf. Emerging Netw. Exp. Technol.*, Rome, Italy, 2009, pp. 1–12.
[25] J. Ni and D. H. K. Tsang, "Large-scale cooperative caching and application-level multicast in multimedia content delivery networks," *IEEE Commun. Mag.*, vol. 43, no. 5, pp. 98–105, May 2005.
[26] N. Laoutaris, H. Che, and I. Stavrakakis, "The LCD interconnection of LRU caches and its analysis," *Perform. Eval.*, vol. 63, no. 7, pp. 609–634, 2006.
[27] S. Borst, V. Gupta, and A. Walid, "Self-organizing algorithms for cache cooperation in content distribution networks," *Bell Labs Tech. J.*, vol. 14, no. 3, pp. 113–125, 2009.
[28] J. M. Wang, J. Zhang, and B. Bensaou, "Intra-AS cooperative caching for content-centric networks," in *Proc. 3rd ACM SIGCOMM Workshop Inf. Centric Netw.*, 2013, pp. 61–66.
[29] J. Rajahalme, M. Särelä, P. Nikander, and S. Tarkoma, "Incentive-compatible caching and peering in data-oriented networks," in *Proc. ACM CoNEXT Conf.*, 2008, pp. 62:1–62:6.
[30] W. K. Chai, D. He, I. Psaras, and G. Pavlou, "Cache "less for more" in information-centric networks (extended version)," *Comput. Commun.*, vol. 36, no. 7, pp. 758–770, 2013.
[31] V. Sourlas, L. Gkatzikis, P. Flegkas, and L. Tassiulas, "Distributed cache management in information-centric networks," *IEEE Trans. Netw. Serv. Manage.*, vol. 10, no. 3, pp. 286–299, Sep. 2013.
[32] A. M. Zoubir, V. Krishnamurthy, and A. H. Sayed, "Signal processing theory and methods [in the spotlight]," *IEEE Signal Process. Mag.*, vol. 28, no. 5, pp. 152–156, Sep. 2011.
[33] O. N. Gharehshiran, V. Krishnamurthy, and G. Yin, "Distributed energy-aware diffusion least mean squares: Game-theoretic learning," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 5, pp. 821–836, Oct. 2013.
[34] M. Maskery, V. Krishnamurthy, and Q. Zhao, "Decentralized dynamic spectrum access for cognitive radios: Cooperative design of a non-cooperative game," *IEEE Trans. Commun.*, vol. 57, no. 2, pp. 459–469, Feb. 2009.
[35] B.-G. Chun, K. Chaudhuri, H. Wee, M. Barreno, C. H. Papadimitriou, and J. Kubiatowicz, "Selfish caching in distributed systems: A game-theoretic analysis," in *Proc. 23rd Annu. ACM Symp. Principles Distrib. Comput.*, St. John's, NF, Canada, 2004, pp. 21–30.
[36] M. X. Goemans, L. Li, V. S. Mirrokni, and M. Thottan, "Market sharing games applied to content distribution in ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 5, pp. 1020–1033, May 2006.
[37] M. Cha, H. Kwak, P. Rodriguez, Y. Ahn, and S. Moon, "Analyzing the video popularity characteristics of large-scale user generated content systems," *IEEE/ACM Trans. Netw.*, vol. 17, no. 5, pp. 1357–1370, Oct. 2009.
[38] H. Pinto, J. Almeida, and M. Gonçalves, "Using early view patterns to predict the popularity of YouTube videos," in *Proc. 6th ACM Int. Conf. Web Search Data Mining*, 2013, pp. 365–374.
[39] J. Famaey, F. Iterbeke, T. Wauters, and F. Turck, "Towards a predictive cache replacement strategy for multimedia content," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 219–227, 2013.
[40] A. Fostel, H. Scarf, and M. Todd, "Two new proofs of Afriat's theorem," *Econ. Theory*, vol. 24, no. 1, pp. 211–219, 2004.
[41] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[42] V. Krishnamurthy and W. Hoiles, "Afriat's test for detecting malicious agents," *IEEE Signal Process. Lett.*, vol. 19, no. 12, pp. 801–804, Dec. 2012.
[43] H. Varian, "Non-parametric tests of consumer behaviour," *Rev. Econ. Stud.*, vol. 50, no. 1, pp. 99–110, 1983.
[44] M. Anthony and P. Bartlett, *Neural Network Learning: Theoretical Foundations*. Cambridge, U.K.: Cambridge Univ. Press, 1999.
[45] M. Zadimoghaddam and A. Roth, "Efficiently learning from revealed preference," in *Internet and Network Economics*, vol. 7695, P. W. Goldberg, Ed., Berlin, Heidelberg: Springer, 2012, pp. 114–127.
[46] E. Beigman and R. Vohra, "Learning from revealed preference," in *Proc. 7th ACM Conf. Electron. Commerce*, Ann Arbor, MI, USA, 2006, pp. 36–42.
[47] S. Borst, V. Gupta, and A. Walid, "Distributed caching algorithms for content distribution networks," in *Proc. IEEE INFOCOM*, San Diego, CA, USA, 2010, pp. 1–9.
[48] H. J. Kushner and G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*, 2nd ed. Berlin, Germany: Springer-Verlag, 2003.
[49] H. J. Kushner, "Stochastic approximation: A survey," *Wiley Interdiscip. Rev. Comput. Statist.*, vol. 2, no. 1, pp. 87–96, 2010.
[50] M. Zeni, D. Miorandi, and F. Pellegrini, "YOUStatAnalyzer: A tool for analysing the dynamics of YouTube content popularity," in *Proc. 7th Int. Conf. Perform. Eval. Methodol. Tools*, Torino, Italy, 2013, pp. 286–289.
[51] S. A. Chowdhury and D. Makaroff, "Characterizing videos and users in YouTube: A survey," in *Proc. 7th Int. Conf. Broadband, Wireless Comput. Commun. Appl.*, Victoria, BC, Canada, 2012, pp. 244–251.
[52] A. Bovik, "Automatic prediction of perceptual image and video quality," *Proc. IEEE*, vol. 101, no. 9, pp. 2008–2024, Sep. 2013.
[53] C. Li, A. Bovik, and X. Wu, "Blind image quality assessment using a general regression neural network," *IEEE Trans. Neural Netw.*, vol. 22, no. 5, pp. 793–799, May 2011.
[54] A. Brodersen, S. Scellato, and M. Wattenhofer, "YouTube around the world: Geographic popularity of videos," in *Proc. 21st Int. Conf. World Wide Web*, Lyon, France, 2012, pp. 241–250.
[55] P. Dhiman, "Empirical validation of website quality using statistical and machine learning methods," in *Proc. 5th Confluence Next Gener. Inf. Technol. Summit*, 2014, pp. 286–291.
[56] X. Qi and B. Davison, "Web page classification: Features and algorithms," *ACM Comput. Surveys*, vol. 41, no. 2, p. 12, 2009.
[57] M. Richardson, A. Prakash, and E. Brill, "Beyond Pagerank: Machine learning for static ranking," in *Proc. 15th Int. Conf. World Wide Web*, Edinburgh, Scotland, 2006, pp. 707–715.
[58] J. Kangasharju, J. Roberts, and K. W. Ross, "Object replication strategies in content distribution networks," *Comput. Commun.*, vol. 25, no. 4, pp. 376–383, 2002.
[59] M. Benaïm, J. Hofbauer, and S. Sorin, "Stochastic approximations and differential inclusions—Part II: Applications," *Math. Oper. Res.*, vol. 31, no. 4, pp. 673–695, 2006.
[60] V. Krishnamurthy, O. N. Gharehshiran, and M. Hamdi, "Interactive sensing and decision making in social networks," *Found. Trends Signal Process.*, vol. 7, no. 1–2, pp. 1–196, 2014.

William Hoiles received the M.A.Sc. degree in engineering science from Simon Fraser University, Burnaby, BC, Canada, and the Ph.D. degree in electrical and computer engineering from the University of British Columbia, Vancouver, BC, Canada, in 2012 and 2015, respectively. He is currently a Postdoctoral Researcher with Electrical and Computer Engineering, University of British Columbia. His research interests include social sensors and the bioelectronic interface.

Omid Namvar Gharehshiran received the Ph.D. degree from the University of British Columbia, Vancouver, BC, Canada, in 2015, where he was a Member of the Statistical Signal Processing Group. He currently holds the NSERC Postdoctoral Fellowship at the Actuarial Science and Mathematical Finance Group, Department of Statistical Sciences, University of Toronto, Toronto, ON, Canada. His research interests include stochastic optimization and control, games, and learning theory.

Vikram Krishnamurthy (S'90–M'91–SM'99–F'05) received the bachelor's degree from the University of Auckland, Auckland, New Zealand, and the Ph.D. degree from the Australian National University, Canberra, A.C.T., Australia, in 1988 and 1992, respectively. He is currently a Professor and holds the Canada Research Chair with the Department of Electrical Engineering, University of British Columbia, Vancouver, BC, Canada. His research interests include statistical signal processing, computational game theory, and stochastic control in social networks. He served as Distinguished Lecturer for the IEEE Signal Processing Society and Editor-in-Chief of the IEEE JOURNAL ON SELECTED TOPICS IN SIGNAL PROCESSING. He was the recipient of an Honorary Doctorate from KTH (Royal Institute of Technology), Sweden, in 2013.

Ngọc-Dũng Đào received the Ph.D. degree from the University of Alberta, Edmonton, AB, Canada, in 2006. He is a Senior Research Engineer with Huawei Technologies Canada Co. Ltd. He was with Siemens Communications Networks, Vietnam, from 1995 to 2000, and with Toshiba Research Europe, U.K., from 2007 to 2010. His research interests include SDN and NFV for radio access networks, information-centric networking (ICN), heterogeneous and dense networks, and reliable video communications for 5G mobile networks. He is an Editor of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, an Associate Technical Editor of the *IEEE Communications Magazine*, and a Guest Editor of 5G Special Issue of *Eurasip Journal on Wireless Communications and Networking*.

Hang Zhang received the doctoral degree in technology from Tampere University of Technology, Tampere, Finland. She is currently leading Huawei future 5G wireless network architecture research with the Ottawa Research and Development Center, Huawei Technologies, Ottawa, ON, Canada. Previously, she was with Nortel in Canada, Nokia in Finland, and Beijing University of Post and Telecommunications, China. Her research interests include algorithms, protocols, and architectures of wireless networks.