Estimating Learning Dynamics in the Routing Game

Kiet Lam
UC Berkeley
kiet.lam@berkeley.edu

Walid Krichene
UC Berkeley
walid@eecs.berkeley.edu

Alexandre Bayen
UC Berkeley
bayen@berkeley.edu

ABSTRACT

The routing game models congestion on transportation and communication networks. We consider an online learning model of player dynamics: at each iteration, every player chooses a route (or a probability distribution over routes), then the joint decision of all players determines the costs of each path, which are then revealed to the players. We first review convergence guarantees of such online learning dynamics. Then, we consider the following estimation problem: given a sequence of player decisions and the corresponding costs, we would like to fit the learning model parameters to these observations. We consider in particular entropic mirror descent dynamics, and develop a numerical solution to the estimation problem.

We demonstrate this method using data collected from a routing game experiment: we develop a web interface to simulate the routing game. When players log in to the interface, they are assigned an origin and destination on the graph. They can choose, at each iteration, a distribution over their available routes, and each player seeks to minimize her own cost. We collect a data set using this interface, then we apply the proposed method to fit the learning model parameters. We observe in particular that after an exploration phase, the joint decision of the players remains within a small distance of the Nash equilibrium. We also use the estimated model parameters to predict the flow distribution over routes, and compare these predictions to the actual distribution. Finally, we discuss some of the qualitative implications of the experiment, and give directions for future research.

1. INTRODUCTION

The routing game is a non-cooperative game that models congestion in transportation and communication networks. The game is given by a directed graph that represents the network, and each player is given by a source node and destination node, and seeks to send traffic (either packets in a communication setting, or cars in a transportation setting) while minimizing the total delay of that traffic. The delay is determined by the joint decision of all players, such that whenever an edge has high load, it becomes congested and any traffic using that edge incurs additional delay.

This model of congestion is simple yet powerful, and routing games have been studied extensively since the seminal work of Beckman [2]. The Nash equilibria of the game are simple to characterize, and have been used to quantify the inefficiency of the network, using the price of anarchy [20]. However, the Nash equilibrium concept may not offer a good descriptive model of actual behavior of players, as argued by many authors, including for example [8]. Besides the assumption of rationality, which can be questioned, the Nash equilibrium assumes that players have a complete description of the structure of the game, their own cost functions, and those of other players. This model is arguably not very realistic for the routing game, as one does not expect users of a network to have an accurate representation of the cost function on every edge of the network, or of the other users of the network. One alternative to the Nash equilibrium concept is a model of repeated play, sometimes called learning models or adjustment models. In such models, one assumes that each player makes decisions iteratively, and uses the outcome of each iteration to adjust their next decision. Formally, if $x_k^{(t)}$ is the decision of player k at iteration t, and $\ell_k^{(t)}$ is the corresponding vector of costs, then player k faces a sequential decision problem in which she chooses $x_k^{(t)}$ then observes $\ell_k^{(t)}$. These sequential decision problems are coupled through the cost functions, since $\ell_k^{(t)}$ depends not only on $x_k^{(t)}$ but also on $x_{k'}^{(t)}$ for $k' \neq k$ (but players do not neces-

sarily model this coupling). Such models have a long history in game theory, and date back to the work of Hannan [10] and Blackwell [3]. In recent years, there has been a resurgence of research on the topic of learning in games using sequential decision problems, see for example [6] and the references therein.

When designing a model of player decisions, many properties are desirable. Perhaps the most important property is that the dynamics should be consistent with the equilibrium of the game, in the following sense: asymptotically, one should expect the learning dynamics to converge to the equilibrium of the full information, one-shot game (be it Nash equilibrium or other, more general equilibrium concepts). In this sense, players "learn" the equilibrium asymptotically. Much progress has been made in recent years in characterizing classes of learning dynamics which are guaranteed to converge to an equilibrium set [9, 12, 11, 8]. In particular for the routing game, different models of learning have been studied for example in [7, 4, 14, 16, 15].

In this paper, we briefly review some of the known models of learning in the routing game. We focus in particular on the mirror descent model used in [17]. This model describes the learning dynamics as solving, at each step, a simple minimization problem parameterized by a learning rate η. Formally, the decision at iteration $t+1$ is obtained by solving

$$x_k^{(t+1)}(\eta_k^{(t)}) = \arg\min_{x_k \in \Delta^{\mathcal{A}_k}} \eta_k^{(t)} \left\langle \ell_k^{(t)}, x \right\rangle + D_{\psi_k}(x_k, x_k^{(t)}),$$

where ψ_k is a distance generating function with corresponding Bregman divergence D_{ψ_k}, and $\eta_k^{(t)}$ is a learning rate. The model is reviewed in detail in Section 2. Intuitively, minimizing the first term $\left\langle \ell_k^{(t)}, x \right\rangle$ will assign traffic to the routes which currently have minimal cost, and minimizing the second term $D_{\psi_k}(x_k, x_k^{(t)})$ will keep the traffic assignment at its current value. Minimizing the linear combination trades-off both terms, and the learning rate $\eta_k^{(t)}$ determines how aggressive the player is when updating her strategy: a small learning rate results in a small change in strategy (i.e. $x_k^{(t+1)}$ is close to $x_k^{(t)}$), while a large learning rate results in a significant change.

Motivated by this interpretation of learning rates, we propose in Section 3, the following estimation problem: given a sequence of player decisions $(x_k^{(t)})$, and the sequence of corresponding losses $(\ell_k^{(t)})$, can we fit a learning model to these observations? A simple approach is to assume that the player is using a given distance generating function ψ_k, and estimate η_k for example by minimizing the distance between the observed decision $\bar{x}_k^{(t+1)}$, and the decision predicted by the model, $x_k^{(t+1)}(\eta_k)$. More precisely, we can choose $\eta_k^{(t)}$ to minimize $D_{\psi_k}(\bar{x}_k^{(t+1)}, x_k^{(t+1)}(\eta))$. We show that in the entropic case (when ψ_k is the negative entropy), this problem is convex, thus $\eta_k^{(t)}$ can be estimated efficiently e.g. by using gradient descent. This method allows us to estimate one parameter $\eta_k^{(t)}$ per iteration t. When we have a sequence of observations available, it can be desirable to control the complexity of the model by assuming a parameterized sequence of learning rates, instead of estimating each term separately. Thus, we propose a second method which assumes that the learning rate is of the form $\eta_k^{(t)} = \eta_k^{(0)} t^{-\alpha_k}$, with $\alpha_k \in (0,1)$. The resulting estimation problem is non-convex in general, but since it is a two dimensional problem, it can be solved efficiently. Finally, we consider a family of distance generating functions ψ_ϵ, parameterized by ϵ, that can be viewed as a generalization of the negative entropy function. These generalized entropy functions also offer some desirable properties that we discuss in Section 3. We also briefly discuss potential uses for the estimated model: for example, the model can be used simply to predict the decision of the players over the next few iterations, by propagating the model forward with the estimated values of the parameters; more generally, the model can be used to formulate a receding-horizon optimal control problem, by using the current estimate of the model as a plant in the control problem.

In the second part of the paper, we present an experimental setting which we developed to collect data on routing decisions. We developed a web interface in which a master user can create an instance of the routing game by defining a graph and cost functions on edges of the graph. Then other users can connect to the interface as players. The game then proceeds similarly to our learning model: at each iteration, every player chooses a flow distribution on their available routes (using a graphical user interface with sliders), then their decisions are sent to a backend server, which computes the total cost of each route, and sends back this information to each player. In Section 4, we describe the experimental setting, some implementation details, as well as the nature of the collected data. We then use this data to run the estimation tasks which were proposed in Section 3, and give some qualitative and quantitative insights into the behavior of players. In particular, we observed that in the first few iterations, the flow distributions oscillate, which corresponds to a high value of estimated learning rates. For later iterations, the flow distributions are, in general, close to equilibrium, and the learning rates are lower, although some players may occasionally move the system away from equilibrium by performing an aggressive update (corresponding to a high learning rate). It is also interesting

to observe that in some rare cases, the estimate of the learning rate is negative, which means that the player updated her strategy by assigning more traffic to routes with higher cost, a counter-intuitive behavior which is hard to model. We also solve the decay rate estimation problem and compare the decay rates of the different players, then use the estimated parameters to predict the traffic assignment over the next few iterations, and comment on the quality of the prediction. We conclude in Section 6 by summarizing our results and giving directions for future research.

2. THE ROUTING GAME AND THE LEARNING MODEL

In this section, we give the definition of the (one-shot) routing game, and the model of learning dynamics.

2.1 The routing game

The routing game is played on a directed graph $\mathcal{G} = (V, E)$, where V is a vertex set and $E \subset V \times V$ is an edge set. The players will be indexed by $k \in \{1, \dots, K\}$, where every player is given by an origin vertex $o_k \in V$, a destination vertex $d_k \in V$, and a traffic mass $m_k \geq 0$ that represents the total traffic that the player needs to send from o_k to d_k. The set of available paths connecting o_k to d_k will be denoted by $\mathcal{P}_k$, and the action set of player k is simply the probability simplex over $\mathcal{P}_k$, which we denote by $\Delta_k = \{x \in \mathbb{R}_+^{\mathcal{P}_k} : \sum_{p \in \mathcal{P}_k} x_p = 1\}$. In other words, each player chooses a distribution over their available paths, and their traffic is allocated to paths according to that distribution. We will denote by $x_k \in \Delta^{\mathcal{P}_k}$ the distribution of player k. Note that x_k is a distribution vector, so the vector of actual flows is the scaled vector $m_k x_k$. The joint decision of all players is denoted by $x = (x_1, \dots, x_K)$. The costs of the players are then determined as follows:

- The cost on an edge e is $c_e(\phi_e(x))$, where $c_e(\cdot)$ is a given, increasing function, and $\phi_e(x)$ is the total traffic flow on edge e induced by the distribution x, obtained simply by summing all the path flows that go through that edge, i.e. $\phi_e(x) = \sum_k \sum_{p \in \mathcal{P}_k} m_k x_{k,p}$.
- The cost on a path $p \in \mathcal{P}_k$ is denoted by $\ell_{k,p}(x)$, and is the sum of edge costs along the path, i.e. $\ell_{k,p}(x) = \sum_{e \in p} c_e(\phi_e(x))$.
- The cost for player k is the total path cost for all the traffic sent by player k, i.e. $\sum_{p \in \mathcal{P}_k} m_k x_{k,p} \ell_{k,p}(x)$. This is simply the inner product between the flow vector $m_k x_k$ and the path loss vector $\ell_k(x)$, which we denote by $\langle \ell_k(x), x_k \rangle$.

Remark 1 (A note on the player model) *Some formulations of the routing game, e.g. [21, 16], formulate the game in terms of populations of players, such that each population is an infinite set of players with the same origin and destination. This assumes that each player contributes an infinitesimal amount of flow, so each player can pick a single path. In our model, each player is macroscopic, and can split its traffic across multiple routes. Both models are equivalent in terms of analysis, the only difference is in terms of interpretation of the model. We choose the finite player interpretation because it is more consistent with the experimental section of the paper, where we run the game with finitely many players.*

Definition 1 (Nash equilibrium) *A distribution $x^\star = (x_1^\star, \dots, x_K^\star)$ is a Nash equilibrium if it satisfies the following condition: for all other feasible distributions $x = (x_1, \dots, x_K)$ and for all k,*

$$\langle \ell_k(x^\star), x_k - x_k^\star \rangle \geq 0.$$

In words, $x^\star$ is a Nash equilibrium if for every player k, the expected cost under $x_k^\star$ is lower than the expected cost under any other distribution x_k. If we define the inner product $\langle x, \ell \rangle = \sum_k \langle x_k, \ell_k \rangle$, then this is equivalent to: $x^\star$ is an equilibrium if and only if $\langle \ell(x^\star), x - x^\star \rangle \geq 0$ for all feasible x. This variational inequality is, in fact, equivalent to the first-order optimality condition of the following potential function, usually referred to as the Rosenthal potential, in reference to [19]:

Proposition 1 (Existence of a convex potential) *Consider a routing game and define the following function*

$$f(x) = \sum_{e \in E} \int_0^{\phi_e(x)} c_e(u) du.$$

Then f is convex its gradient is $\nabla f(x) = \ell(x)$.

This result can be found for example in [20]. Due to the fact that the loss function $\ell(\cdot)$ coincides with the gradient field $\nabla f(\cdot)$ of the Rosenthal potential, the Nash condition can be rewritten as

$$\langle \nabla f(x^\star), x - x^\star \rangle \geq 0 \ \forall \text{ feasible } x,$$

and since f is convex, this is a necessary and sufficient condition for optimality of $x^\star$ (see e.g. Section 4.2.3 in [5]). Therefore the set of Nash equilibria is exactly the set of minimizers of the convex potential f. This is important both for computation (computing a Nash equilibrium can be done by minimizing a convex function), and for modeling: one can model player dynamics as performing a distributed optimization of the potential function. More precisely, if we adopt the point of view presented in the introduction, wherein each player faces a sequential decision problem, and plays $x_k^{(t)}$ then observes $\ell_k(x^{(t)})$, then this corresponds

to a first-order distributed optimization of the function f, where each player is responsible for updating the variables $x_k^{(t)}$, and observes, at each iteration, the partial gradient $\ell_k(x^{(t)}) = \nabla_{x_k} f(x^{(t)})$. Using this connection to distributed optimization, a model of player dynamics was proposed in [17]. We review the model in the next Section.

2.2 The learning model: mirror descent dynamics

We will consider the model of distributed learning proposed in [17]. Each player is assumed to perform a mirror descent update given by the following algorithm:

Algorithm 1 Distributed mirror descent dynamics with DGF ψ_k and learning rates $\eta_k^{(t)}$.

for each $t \in \{1, 2, \dots\}$ **do**
  Play $x_k^{(t)}$,
  Observe $\ell_k^{(t)} = \nabla_{x_k} f(x^{(t)})$,
  Update

$$x_k^{(t+1)} = \arg\min_{x_k \in \Delta^{\mathcal{P}_k}} \eta_k^{(t)} \left\langle \ell_k(x^{(t)}), x_k \right\rangle + D_{\psi_k}(x_k, x_k^{(t)}) \quad (1)$$

end for

In the update equation (1), $D_{\psi_k}(x, x_k^{(t)})$ is the Bregman divergence between the distributions x_k and $x_k^{(t)}$, defined as follows: $D_\psi(x, y) = \psi(x) - \psi(y) - \langle \nabla\psi(y), x - y\rangle$, for a strongly convex function ψ, called the distance generating function (DGF). Some special cases include

- The Euclidean case: if $\psi(x) = \frac{1}{2}\|x\|_2^2$, then $D_\psi(x, y) = \frac{1}{2}\|x - y\|_2^2$. In this case, mirror descent reduces to the projected gradient descent algorithm.
- The entropic case: if $\psi(x) = -H(x)$ where $H(x) = -\sum_p x_p \ln x_p$ is the negative entropy, then $D_\psi(x, y) = \sum_p x_p \ln \frac{x_p}{y_p}$ is the Kullback-Leibler (KL) divergence from x to y. In this case, the mirror descent algorithm is sometimes called the entropic descent [1], or exponentiated gradient descent [13].

The mirror descent method is a general method for convex optimization proposed in [18]. The model in Algorithm 1 is a distributed version of mirror descent, applied to the potential function f. To give some intuition of the method, the first term $\left\langle \ell_k^{(t)}, x_k \right\rangle$ in the minimization problem (1) can be thought of as a linear approximation of the potential function (since $\ell(x) = \nabla f(x)$), and the second term $D_\psi(x_k, x_k^{(t)})$ penalizes deviations from the previous iterate. The learning rate $\eta_k^{(t)}$ determines the tradeoff between the two terms, and can be thought of as a generalized step size: a smaller η_k results in a distribution which is closer to the current $x_k^{(t)}$. Thus, from the potential function point of view, the player minimizes a linearization of the potential plus a Bregman divergence term that keeps x_k close to $x_k^{(t)}$. From the routing game point of view, the first term $\left\langle \ell_k^{(t)}, x_k \right\rangle$ corresponds to putting weight on the paths that have smaller cost during the previous iteration, and the second term keeps the distribution at its current value.

The convergence of this distributed learning model is discussed in [17]. In particular, the learning dynamics is guaranteed to converge under the following assumptions:

Theorem 1 (Theorem 3 in [17]) *Consider the routing game with mirror descent dynamics defined in Algorithm 1, and suppose that for all k, $\eta_k^{(t)}$ is decreasing to 0. Then $f(x^{(t)}) - f^\star = \mathcal{O}\left(\sum_k \frac{1}{t\eta_k^{(t)}} + \frac{\sum_{\tau=1}^t \eta_k^{(\tau)}}{t}\right)$.*

In particular, if $\eta_k^{(t)} = \eta_k^{(0)} t^{-\alpha_k}$, with $\alpha_k \in (0, 1)$, then one can bound the sum $\sum_{\tau=1}^t \eta_k^{(t)} = \eta_k^{(0)} \sum_{\tau=1}^t \tau^{-\alpha_k} \leq \eta_k^{(0)} \int_0^t \tau^{-\alpha_k} d\tau = \frac{\eta_k^{(0)}}{1-\alpha_k} t^{1-\alpha_k}$. Therefore, $f(x^{(t)}) - f^\star = \mathcal{O}(t^{\alpha_k - 1}) + \mathcal{O}(t^{-\alpha_k}) = \mathcal{O}(t^{-\min(\alpha_k, 1-\alpha_k)})$.

While the specific convergence rate does not matter for the purposes of the estimation problem, the convergence guarantees for decaying learning rates motivates some of the modeling assumptions made in the next Section. To conclude this Section, we also point that a stochastic version of the distributed mirror descent dynamics has been proposed and studied in [15]. In that model, instead of observing the true loss vector $\ell_k^{(t)}$, a player observes a stochastic vector $\hat{\ell}_k^{(t)}$, the expectation of which (conditioned on all past information) is a.s. the true loss vector. A convergence result similar to Theorem 1 is obtained, but the convergence rate is that of $\mathbb{E}[f(x^{(t)})] - f^\star$. This results is important, as it shows that convergence is robust to noise and other stochastic perturbations. For a more detailed discussion, see [15].

3. LEARNING MODEL ESTIMATION

In this section, we assume that we have access to a sequence of observations of traffic distributions ($\bar{x}_k^{(t)}$), and a sequence of loss vectors ($\bar{\ell}_k^{(t)}$). We use the over bar to make a clear distinction between quantities which are observed and quantities which are estimated. Given this sequence of observations, we would like to fit a model of learning dynamics. From the previous section, the learning model in Algorithm 1 is naturally parameterized by the DGF ψ_k and the learning rate sequence $\eta_k^{(t)}$. First, we discuss how one can estimate a single term of the learning rate sequence, given a DGF ψ_k.

3.1 Estimating a single learning rate

Given the current flow distribution $\bar{x}_k^{(t)}$ and the current loss vector $\bar{\ell}_k^{(t)}$, the mirror descent model prescribes that the next distribution is given by

$$x_k^{(t+1)}(\eta) = \arg\min_{x \in \Delta^{\mathcal{P}_k}} \eta \left\langle \bar{\ell}_k^{(t)}, x_k \right\rangle + D_{\psi_k}(x_k, \bar{x}_k^{(t)}), \quad (2)$$

where ψ_k is assumed to be given in this Section. Therefore, $x^{(t+1)}$ can be viewed as a function of η, and to estimate η, one can minimize

$$d_k^{(t)}(\eta) = D_{\psi_k}(\bar{x}_k^{(t+1)}, x_k^{(t+1)}(\eta)).$$

The problem is then simply

$$\eta_k^{(t)} = \arg\min_{\eta \geq 0} d_k^{(t)}(\eta) \quad (3)$$

In the next proposition, we show that this problem is convex when the DGF is the negative entropy. In fact, one can explicitly compute the gradient of $d_k(\eta)$ in this case, which makes it possible to solve Problem (3) efficiently using gradient descent for example.

Theorem 2 *If ψ_k is the negative entropy, then $d_k^{(t)}(\eta) = D_{\psi_k}(\bar{x}_k^{(t+1)}, x_k^{(t+1)}(\eta))$ is a convex function of η, and its gradient with respect to η is given by*

$$\frac{d}{d\eta} d_k^{(t)}(\eta) = \left\langle \bar{\ell}_k^{(t)}, \bar{x}_k^{(t+1)} - x_k^{(t+1)}(\eta) \right\rangle.$$

Proof. When ψ_k is the negative entropy, the solution of the mirror descent update (2) can be computed in closed form, and is given by

$$x_{k,p}^{(t+1)}(\eta) = \frac{\bar{x}_{k,p}^{(t)} e^{-\eta \bar{\ell}_{k,p}^{(t)}}}{Z_k^{(t)}(\eta)} \quad (4)$$

where $Z_k^{(t)}(\eta)$ is the appropriate normalization constant, given by $Z_k^{(t)}(\eta) = \sum_p \bar{x}_{k,p}^{(t)} e^{-\eta \bar{\ell}_{k,p}^{(t)}}$, see for example [1] for a proof of this result. Given this expression of $x_k^{(t+1)}(\eta)$, we can explicitly compute the Bregman divergence (which, in this case, is the KL divergence):

$$\begin{aligned} d_k(\eta) &= D_{KL}(\bar{x}_k^{(t+1)}, x_k^{(t+1)}(\eta)) \\ &= \sum_{p \in \mathcal{P}_k} \bar{x}_{k,p}^{(t+1)} \ln \frac{\bar{x}_{k,p}^{(t+1)}}{x_{k,p}^{(t+1)}(\eta)} \\ &= \sum_{p \in \mathcal{P}_k} \bar{x}_{k,p}^{(t+1)} \left(\ln \frac{\bar{x}_{k,p}^{(t+1)}}{\bar{x}_{k,p}^{(t)}} + \eta \bar{\ell}_{k,p}^{(t)} + \ln Z_k^{(t)}(\eta) \right) \\ &= D_{KL}(\bar{x}_k^{(t+1)}, \bar{x}_k^{(t)}) + \eta \left\langle \bar{\ell}_k^{(t)}, \bar{x}_k^{(t+1)} \right\rangle + \ln Z_k^{(t)}(\eta), \end{aligned}$$

where we used the explicit form of $x^{(t+1)}(\eta)$ in the third equality. In this expression, the first term does not depend on η, the second term is linear in η, and the last term is the function $\eta \mapsto \ln Z_k^{(t)}(\eta) = \ln \sum_p \bar{x}_{k,p}^{(t)} e^{-\eta \bar{\ell}_{k,p}^{(t)}}$, which is known to be convex in η (see for example Section 3.1.5 in [5]). Therefore $d_k^{(t)}(\eta)$ is convex, and its gradient can be obtained by differentiating each term

$$\begin{aligned} \frac{d}{d\eta} d_k^{(t)}(\eta) &= \left\langle \bar{\ell}_k^{(t)}, \bar{x}_k^{(t+1)} \right\rangle + \frac{\frac{d}{d\eta} Z_k^{(t)}(\eta)}{Z_k^{(t)}(\eta)} \\ &= \left\langle \bar{\ell}_k^{(t)}, \bar{x}_k^{(t+1)} \right\rangle + \frac{\sum_p -\bar{\ell}_{k,p}^{(t)} \bar{x}_{k,p}^{(t)} e^{-\eta \bar{\ell}_{k,p}^{(t)}}}{Z_k^{(t)}(\eta)} \\ &= \left\langle \bar{\ell}_k^{(t)}, \bar{x}_k^{(t+1)} \right\rangle - \left\langle \bar{\ell}_k^{(t)}, x_k^{(t+1)}(\eta) \right\rangle, \end{aligned}$$

which proves the claim. □

While we cannot prove that the problem is convex in the general case (when ψ_k is any DGF), since the problem is one-dimensional, one can apply any non-convex optimization method, such as simulated annealing, to find a local optimum of $d_k^{(t)}(\eta)$.

3.2 Estimating the decay rate of the learning rate sequence

In the previous section, we proposed a method to estimate one term of the learning rate sequence. One can of course repeat this procedure at every iteration, thus generating a sequence of estimated learning rates. However, the resulting sequence may not be decreasing. In order to be consistent with the assumption of the model, we can assume a parameterized sequence of learning rates (which is by construction decreasing), then estimate the parameters of the sequence, given the observations. Motivated by Theorem 1, we will assume, in this Section, that $\eta_k^{(t)} = \eta_k^{(0)} t^{-\alpha_k}$ with $\eta_k^{(0)} > 0$ and $\alpha_k \in (0, 1)$.

Given the observations $(\bar{x}_k^{(t)})$ and $(\bar{\ell}_k^{(t)})$, we can define a cumulative cost,

$$D_k^{(t)}(\alpha, \eta^{(0)}) = \sum_{\tau=1}^{t} d_k^{(\tau)}(\eta^{(0)} \tau^{-\alpha_k}),$$

then estimate $(\alpha_k, \eta_k^{(0)})$ by solving the problem

$$(\alpha_k, \eta_k^{(0)}) = \arg\min_{\alpha_k \in (0,1), \eta^{(0)} \geq 0} D_k^{(t)}(\alpha, \eta^{(0)}) \quad (5)$$

Note that this problem is non-convex in general, however, since it is two-dimensional, it can also be solved efficiently using non-convex optimization techniques.

3.3 A parameterized family of distance generating functions

In this Section, we propose to use a generalization of the entropy distance generating function, motivated by the following observation: according to the entropy update and its explicit solution 5, the support of $x_k^{(t+1)}(\eta)$ always coincides with the support of $\bar{x}_k^{(t)}$ (due to the multiplicative form of the solution). As a consequence,

if we observe two consecutive terms $\bar{x}_k^{(t)}, \bar{x}_k^{(t+1)}$ such that some p is in the support of $\bar{x}_k^{(t+1)}$ but not in the support of $\bar{x}_k^{(t)}$, the KL divergence $D_{KL}(\bar{x}_k^{(t+1)}, x_k^{(t+1)}(\eta))$ will be infinite since the support of $x_k^{(t+1)}(\eta)$ will not be a subset of the support of $\bar{x}^{(t+1)}$ (in measure theoretic terms, $x_k^{(t+1)}(\eta)$ is not absolutely continuous with respect to $\bar{x}^{(t+1)}$). This is problematic, as the estimation problem is ill-posed in such cases (which do occur in the data set used in Section 4).

To solve this problem, we consider the following DGF: for $\epsilon > 0$, let

$$\psi_\epsilon(x_k) = -H(x+\epsilon) = \sum_p (x_{k,p}+\epsilon)\ln(x_{k,p}+\epsilon).$$

The corresponding Bregman divergence is

$$D_{\psi_\epsilon}(x_k, y_k) = \sum_p (x_{k,p}+\epsilon)\ln\frac{x_{k,p}+\epsilon}{y_{k,p}+\epsilon},$$

and can be interpreted as a generalized KL divergence. In particular, for any $\epsilon > 0$, this Bregman divergence is finite for any $x_k, y_k \in \Delta^{\mathcal{P}_k}$, unlike the KL divergence. Finally, it is worth observing that when $\epsilon > 0$, the update equation (1) does not have a closed-form expression as in the special case of the KL divergence. Additionally, the support is not necessarily preserved, which was an undesirable property of the previous model. In our numerical simulations in Section 4, we use the generalized DGF proposed here.

3.4 Application to prediction

We conclude this Section by briefly discussing one important application of the proposed estimation problem. Once we have estimated a model of learning dynamics, we can propagate the model forward in order to predict the distributions of the players for the next time step. More precisely, if at iteration t, we have observed $\bar{x}^{(t)}$, $\bar{\ell}^{(t)}$, and we have estimated the terms $(\eta_k^{(1)}, \ldots, \eta_k^{(t-1)})$ for a player k, then we can use these terms to estimate $\eta_k^{(t)}$, and predict the next distribution by solving

$$\begin{aligned} x_k^{(t+1)} &= \arg\min x_k \in \Delta^{\mathcal{P}_k} \left\langle \eta_k^{(t)}, \ell(\bar{x}_k^{(t)}) \right\rangle + D_{\psi_k}(x_k, \bar{x}_k^{(t)}) \\ &\triangleq g(\bar{x}_k^{(t)}, \eta_k^{(t)}), \end{aligned}$$

where we defined the function g, which takes a distribution and a learning rate and propagates the model forward one step. We can inductively estimate the next terms by propagating the model further:

$$x_k^{(t+i+1)} = g(x_k^{(t+i)}, \eta_k^{(t+i)}).$$

Here, we assume that we can extrapolate the learning rate sequence to estimate the terms $\eta_k^{(t+i)}$. If we assume a particular form of the sequence, $\eta_k^{(t)} = \eta_k^{(0)} t^{-\alpha_k}$, then this can be done readily once we have an estimate of $\eta_k^{(0)}$ and α_k. However, if each term of the sequence is estimated separately, we need to use a simple model to predict the next terms. We propose a few simple methods in the numerical experiments, and evaluate their generalization performance.

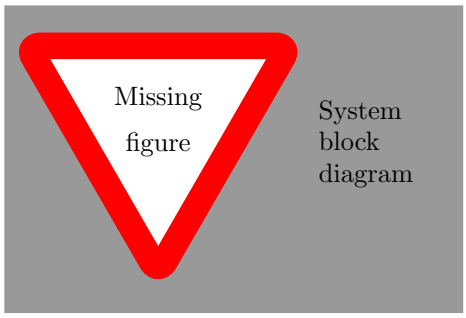

Figure 1: System block diagram to be added later.

4. EXPERIMENT

4.1 Setup

We simulate the proposed methods with a routing game system that implements the behavior of the network. The system is implemented with a web framework that is accessible to players through their browser. Players are assigned a unique origin and destination node pair and are tasked to assign a flow distribution to the paths between this pair at each turn of the game. The game calculates the cost along each edge of the network after all players have submitted their flow distribution for the turn. Each player then receives feedback with the local information of the cost for each path. The players then use this path cost history to assign a new flow distribution at the subsequent turns with the objective of minimizing their cumulative path cost.

To address a player who may be at a disadvantage due to having longer paths or more congested paths because of the topology of the network, we normalized the cumulative path cost of each player with their cumulative path cost at equilibrium calculated with Algorithm 1.

5. ANALYSIS

5.1 Convergence

At the beginning of the game, there is a clear exploration phase where players tend make aggressive assignments with their flow distribution. Towards the end of the game, the players consolidate their position and tend to make more conservative flow distribution assignments. An example of this behavior can be seen in Figure 3.

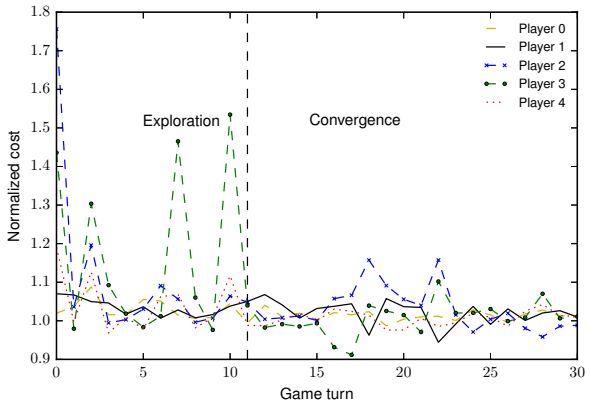

Figure 2: Player cost normalized by equilibrium cost .

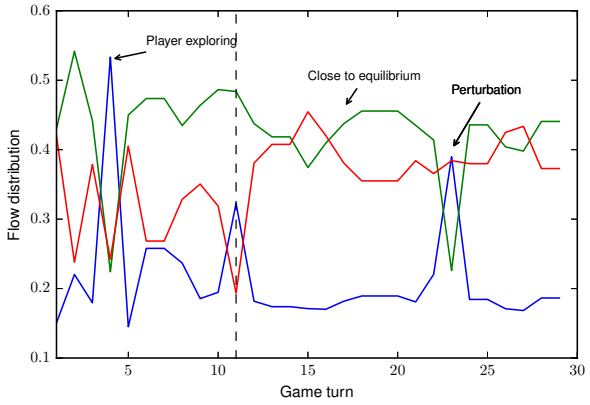

Figure 3: A player flow distribution assignment for the game.

We can describe the convergence of the system more concretely by evaluating the Rosenthal potential of the game at each turn from Prop **??**. From Figure 4, the potential of the system oscillates rapdily in the exploration phase and then converges to the minimal potential. The system also shows robustness seen in Figure 4 where it recovers from perturbations in the convergence phase.

5.2 Estimation and prediction

We can estimate the learning rate of each player with the update in Eq 3. We can see that the learning rates of some players are negative at certain turns, which may be explained by the players assuming a contrarian play style at flow distribution assignments for those certain turns.

With these estimated sequence of learning rates for

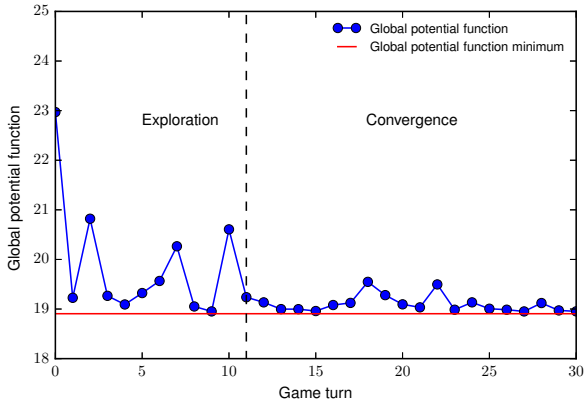

Figure 4: Rosenthal potential values at each iteration.

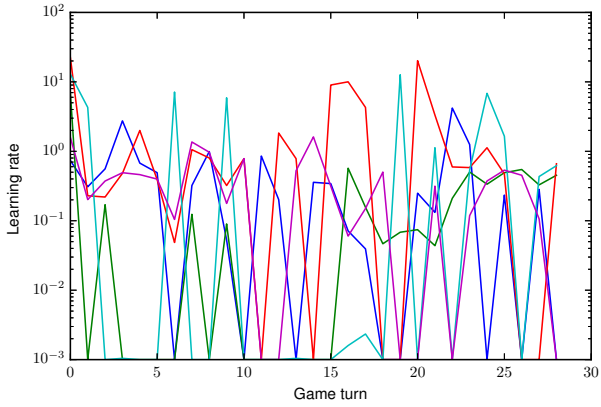

Figure 5: Learning rate for each player.

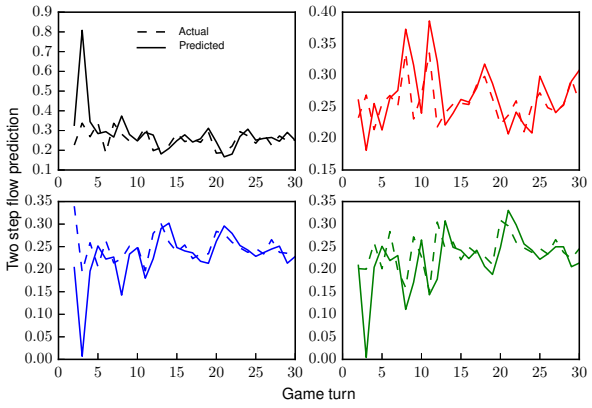

Figure 6: Two step prediction ahead for parameterized learning rate for a single player.

each player, we can predict the flow distribution at turn $t+1$ with the learning rate η_t. We propose three methods for calculating η_t: first we let $\eta_t = \eta_{t-1}$, second we compute a moving average

$$\eta_t = \frac{1}{N} \sum_{i=t-N-1}^{t-1} \eta_i$$

and finally we compute η_t as a linear regression from $\eta_i,\ i \in [0,\ t-1]$.

We also compare these methods with a parameterized sequence of learning rate in Section 3.2

6. CONCLUSION

We proposed a problem of model estimation in the routing game, to fit a distributed learning model to sequential observations of player decisions. The estimated model can then be used to predict the decisions at future iterations, or, more generally, as a plant model in an optimal control problem.

We considered in particular a model based on the mirror descent algorithm, parameterized by a DGF ψ_k and a sequence of learning rates $(\eta_k^{(t)})$, and gave an intuitive interpretation of how this model can describe player behavior. We showed that the problem of estimating one term of the learning rate sequence is convex in the case of the KL divergence (it remains open to prove this result for general Bregman divergences). To control the complexity of the model and to make the estimation consistent with the theoretical assumptions (decreasing learning rates), we proposed to parameterize the sequence with an initial term $\eta_k^{(0)}$ and a decay rate $\alpha_k \in (0,1)$. When we tested these methods on data collected from our routing game interface, the parameterized sequence estimation outperformed the single term estimation on the prediction task. Our test results suggest that the mirror descent model can be a good descriptive model of player behavior, although in some rare cases, a player decision can be hard to model (e.g. when a player increase traffic assignment to previously bad routes).

This estimation problem can be extended in several ways: first, in our method, we fixed the DGF to be the negative entropy (regularized in order to avoid situations in which the estimation problem is ill-posed). One could also estimate the DGF itself, in addition to estimating the learning rates. One natural way to pose the estimation problem is to consider a finite collection of distance generating functions $\{\psi_i\}_{i\in\mathcal{I}}$, then to assume that each player k uses a linear combination with weights θ_k $\psi = \sum_i \theta_{k,i}\psi_i$, then estimate the parameter vector θ_k. One other natural approach is to pose the problem as a Bayesian estimation problem, by setting a prior on the model parameters (e.g. $\eta_k^{(0)}$ and α_k), then define a likelihood model of observing a player decision given her DGF and learning rate.

7. REFERENCES

[1] A. Beck and M. Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Oper. Res. Lett.*, 31(3):167–175, May 2003.

[2] M. J. Beckmann, C. B. McGuire, and C. B. Winsten. Studies in the economics of transportation. 1955.

[3] D. Blackwell. An analog of the minimax theorem for vector payoffs. *Pacific Journal of Mathematics*, 6(1):1–8, 1956.

[4] A. Blum, E. Even-Dar, and K. Ligett. Routing without regret: on convergence to nash equilibria of regret-minimizing algorithms in routing games. In *Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, PODC '06, pages 45–52, New York, NY, USA, 2006. ACM.

[5] S. Boyd and L. Vandenberghe. *Convex Optimization*, volume 25. Cambridge University Press, 2010.

[6] N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.

[7] S. Fischer and B. Vöcking. On the evolution of selfish routing. In *Algorithms–ESA 2004*, pages 323–334. Springer, 2004.

[8] M. J. Fox and J. S. Shamma. Population games, stable games, and passivity. *Games*, 4(4):561–583, 2013.

[9] Y. Freund and R. E. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1):79–103, 1999.

[10] J. Hannan. Approximation to Bayes risk in repeated plays. *Contributions to the Theory of Games*, 3:97–139, 1957.

[11] S. Hart. Adaptive heuristics. *Econometrica*, 73(5):1401–1430, 2005.

[12] S. Hart and A. Mas-Colell. A general class of adaptive strategies. *Journal of Economic Theory*, 98(1):26 – 54, 2001.

[13] J. Kivinen and M. K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1 – 63, 1997.

[14] R. Kleinberg, G. Piliouras, and E. Tardos. Multiplicative updates outperform generic no-regret learning in congestion games. In *Proceedings of the 41st annual ACM symposium on Theory of computing*, pages 533–542. ACM, 2009.

[15] S. Krichene, W. Krichene, R. Dong, and A. Bayen. Convergence of heterogeneous distributed learning in stochastic routing games. In *53rd Allerton Conference on Communication, Control and Computing*, 2015.

[16] W. Krichene, B. Drighès, and A. Bayen. Learning nash equilibria in congestion games. *SIAM Journal on Control and Optimization (SICON)*, 2015.

[17] W. Krichene, S. Krichene, and A. Bayen. Convergence of mirror descent dynamics in the routing game. In *European Control Conference (ECC)*, 2015.

[18] A. S. Nemirovsky and D. B. Yudin. *Problem complexity and method efficiency in optimization*. Wiley-Interscience series in discrete mathematics. Wiley, 1983.

[19] R. W. Rosenthal. A class of games possessing pure-strategy nash equilibria. *International Journal of Game Theory*, 2(1):65–67, 1973.

[20] T. Roughgarden and É. Tardos. How bad is selfish routing? *Journal of the ACM (JACM)*, 49(2):236–259, 2002.

[21] W. H. Sandholm. Potential games with continuous player sets. *Journal of Economic Theory*, 97(1):81–108, 2001.