Network Inference and its Application to the Estimation of Crowd Dynamics from IoT Sensors

Nikhil Ravi, Raksha Ramakrishna, Hoi-To Wai, Anna Scaglione
School of Electrical Computer and Energy Engineering
Arizona State University, Tempe, AZ, United States
{nravi6, raksha.ramakrishna, htwai, Anna.Scaglione} @asu.edu

Abstract—In this paper, we explore the application of system identification techniques to the inference of a model that characterizes crowd dynamics, inspired by the social force model proposed by Helbing and Molnár. We focus then on sensor observations of pedestrians’ actions considering that wearables, smart mobile phones and other IoT devices embedded in the environment give significant insights on their expected mobility patterns. Previous work using IoT sensors to uncover social interactions is not based on mathematical models, while most models used for tracking mobility ignore the strong coupling between the model-agents as well as their surroundings. Our aim is to bridge these approaches, by capturing in the data model the swarming behavior of the network, including social interactions.

Keywords — non-linear graph filter, network inference, Viterbi training, crowd-dynamics, IoT;

I. INTRODUCTION

IoT promises to provide a window for observing individuals interacting with their surroundings in real time. In the realms of IoT technology, crowd analysis is typically based on machine vision, and pertains the analysis of crowds from camera networks. With the emergence of the IoT, a long line of algorithms have been proposed to harness sensor data from the sensors embedded in off-the-shelf mobile and IoT devices that people carry with them e.g., [1], [2]. A framework to collect real-time data that captures social interactions among the agents is presented in [3]. Community sensing also leverages the trends in the aggregate data collected from cliques of people who have similar goals (at least in the short term) [2]. Work has also been undertaken to utilize the knowledge of agents’ social relationships to automate the identification cliques of people to participate in group actions, e.g., [4]. The authors in [5] have proposed a paradigm where the concepts of social networking are integrated into the IoT along with a system architecture required for such a paradigm. The approach taken in this paper towards inference or relationship is very different: specifically, we introduce a crowd-dynamics model that captures social influence and uses a system identification approach and convert the IoT data in a tomographic image of the relative influence each agent exerts on other agents. The model builds on a long history of interdisciplinary research efforts, as surveyed in [6]. Related approaches can roughly be categorized into flow-based, entity-based and agent-based [7]. These approaches differ in terms of the complexity levels — the flow-based models describe the crowd as a simple, continuous flow of fluid, e.g., [8]; the entity-based models describe each individual in a crowd as homogeneous entities, where the resulting models are similar to a particle system, e.g., [9]; the agent-based models describe each individual in a crowd as an autonomous agent whose movements are decided independently by an agent-specific rule, e.g., [10].

Note that the entity based models have been studied theoretically to analyze emergent swarming behavior, e.g., [11]. The social force model proposed by Helbing and Molnár (HM) is perhaps the most popular entity-based model. The HM model easily accounts for various external effects such as road blocks, walls, etc. The model postulates that the movement of individuals is affected by three forces: a) a self-driving force (to reach his destination) prompting the agent to accelerate to a certain velocity, b) a repulsive force to keep a distance from the others and c) a repulsion force to avoid hitting obstacles, and d) an attractive force towards other pedestrians they may know, as well towards certain sites on the road side (like shops’ displays). However, in practicality a pedestrian doesn’t experience the same level of social force with all the others in the area. To accommodate this, in this paper we parameterize the attractions and repulsions pedestrians experience towards one another by weighted adjacency matrices. It is also worthwhile to mention the recent work in [12] which use an optimization based model for entity-based crowd dynamics.

In the context of network inference, a popular problem is inferring the network topology in the form of a binary adjacency matrix describing the inter-dependency of the agents’ actions. Typically, this problem is studied under the general framework of graphical model inference, e.g., the popular graphical LASSO method [13]. Also related is the literature on network topology recovery from online social networks [14]. The more difficult problem involves network identification, which aims at inferring the exact network dynamics including the strengths of interaction between agents. Recent efforts have been found on the topic, e.g., [15], [16]. An algorithm that estimates the underlying graph to structure and capture the correlations between large number of unstructured time series (like financial data from the stock market) is presented in [17].

Inference of the underlying graphs from data generated by a linear graph-filter is explored in [18]. Non-linear kernel-based
models for inference of networks are presented in [19]. Models to infer interconnected multi-layered networks by learning the Supra-Laplacian matrix are presented in [20]. In contrast to the prior work listed above, this paper introduces and mathematically models the social-crowd dynamics to account for both phenomena of crowd and the emotional reactions of the individuals in the crowd. Specifically, rather than treating every individual as identical agents, we capture explicitly the social influence as an unknown parameter.

II. THE NETWORK INFERENCE PROBLEM

In casting our network inference problem it is useful to broaden the definition of autoregressive graph filters accounting for temporal dynamics, feedback and non-linearities. Let \( G(V, E) \) be the underlying directed network graph. The vector \( x(t) \) of data observed is the collection \( x(t) = (x_1^T(t), \ldots, x_N^T(t))^T \) of the blocks \( x_i(t) \in \mathbb{R}^k \) that represent the state of node \( i \in V \). This paper considers a first order non-linear feedback graph filter model:

\[
x_i(t + 1) = f_i(x_i(t)) + \sum_{j \in V} h_i(x_i(t), x_j(t)) g_{ij} + w_i(t),
\]

where \( h_i(x_i(t), x_j(t)) \in \mathbb{R}^{k \times k} \) is a non-linear interaction kernel matrix, whose columns define the type of coupling node \( i \) and node \( j \) as a linear combination of the \( k \) columns in \( h_i(x_i(t), x_j(t)) \); \( g_{ij} \in \mathbb{R}^k \) and \( g_{ij} \neq 0 \) iff \((i, j) \in E\) and \( g_{ii} = 0 \), \( \forall \ i \in V \) includes the coupling coefficients. The goal of the network inference problem is either detect the graph structure (i.e., finding which \( g_{ij} \neq 0 \) for \( i \neq j \)) or estimate all the coupling coefficients \( g_{ij} \). Our intention is to do the latter, by mapping the social-crowd dynamic model onto an instance of network inference problem. To put it in context with the popular literature on graph signal processing, \( h_i(x_i(t), x_j(t)) = x_i(t) - x_j(t) \) would make this a first order auto-regressive multi-layer graph filter [20]; but such model is not supported by any literature on crowd dynamics.

A. The social-crowd dynamics as a non-linear graph-filter

The model we adopt is inspired by the social force model introduced by Helbing and Molnár (HM) [9] for crowd movement dynamics highlighted in the introduction. At time \( t \), the latent state of pedestrian \( i = 1, \ldots, N \) is:

\[
x_i(t) = [\sigma_i^T(t), r_i^T(t), v_i^T(t)]^T
\]

where \( r_i(t) \in \mathbb{R}^2 \) and \( v_i(t) \in \mathbb{R}^2 \) are the continuous valued position and velocity vectors of agent \( i \) at time \( t \), respectively, and \( \sigma_i(t) \in \{0, 1\}^L \) is an unobservable component of the state variable. The set of the \( N \) pedestrians is denoted by \( V_p \), and the set of boundaries obstructing their motion is \( B \), so that \( V = V_p \cup B \). The positions of the pedestrians and boundaries are defined on a 2-D plane. The pedestrians can be attracted to each other or feel repelled. Hence, the network is parameterized by three weighted adjacency matrices \( \{A, R, B\} \), where \( A_{ij} > 0 \) (resp. \( R_{ij} > 0 \)) if pedestrian \( i \) is attracted to (resp. repelled by) pedestrian \( j \) when the pair is in each other’s proximity, and \( B_{ij} \) is the repulsive force on pedestrian \( i \) from obstacle \( j \). The mapping onto the generalized graph filter model in (1) works as follows:

\[
f_i(x) = \text{vec}([f_i^\sigma(x) \ f_i^r(x) \ f_i^v(x)])
\]

with the non-linear kernel matrix and coupling coefficients:

\[
h_i(x, x_j) = \begin{bmatrix} 0 \\ h_i(u) \end{bmatrix}, \quad g_{ij} = (A_{ij}, -R_{ij}, -B_{ij})^T,
\]

where \( u = (r_i - r_j) \) and the non-linear kernels are

\[
\bar{h}(u) \triangleq [h_a(u) \ h_r(u) \ h_b(u)] = [1 \ e^{-\lambda_a \|u\|} \ e^{-\lambda_b \|u\|}] \otimes \frac{u}{\|u\|}.
\]

The non-linear kernels for the HM model are defined in the following subsection.

B. State dynamics

Dynamics of unobservable state \( \sigma_i(t) \): The unobservable state \( \sigma_i(t) \) is a coordinate vector indicating the location the agent aims to reach at time \( t \), among a set of possible corner points organized as a \( 2 \times L \) matrix \( \Theta \). The pedestrians who are attracted to each other will travel together, their latent states will also be tied together, receiving the identical updates. Formally, let \( C_1, \ldots, C_M \subseteq V_p \) be the sets of all weakly connected components in the graph induced by the adjacency matrix \( A \) such that \( V_p = C_1 \cup \cdots \cup C_M \) and \( C_m \cap C_{m'} = \emptyset \) for \( m \neq m' \). Specifically, we have

\[
\theta_i(t + 1) = \Theta \sigma_i(t + 1),
\]

where \( \sigma_i(t + 1) = \sigma_i - \Theta \sigma_i(t) \) if \( i \in C_m \), and for \( m = 1, \ldots, M \)

\[
\sigma_m(t + 1) = \Pi \sigma_m(t) + w_m^\sigma(t) = f_m^\sigma(x(t)) + w_m^\sigma(t),
\]

where \( \Pi \) is the state transition probability, \( w_m^\sigma(t) \) is the difference between the actual state and the column of the transition probability matrix corresponding to the previous state and \( \mathbb{E}[w_m^\sigma(t)]\sigma_m(t) = 0 \).

Dynamics of position \( r_i(t) \): The evolution of position for all agents, indexed by \( i \in V_p \), is:

\[
r_i(t + 1) = r_i(t) + \gamma v_i(t) + w_i^v(t) = f_i^v(x(t)) + w_i^v(t)
\]

where \( \gamma > 0 \) is a step size accounting for the sampling rate and \( w_i^v(t) \) is the error that comes from a not necessarily straight motion within the sampling interval.

Dynamics of velocity \( v_i(t) \): The velocity evolves as:

\[
v_i^\prime(t + 1) = f_i^v(x(t)) + \gamma F_i^v(x(t)) + w_i^v(t),
\]

\[
v_i(t + 1) = v_i(t + 1) \min\{1, v_i^\max/\|w_i^v(t + 1)\|\},
\]

where \( w_i^v(t) \) is a random fluctuating force to account for those random actions of a pedestrian that deviates from the system model and \( v_i^\max \) is the maximum speed the pedestrian can attain. The component \( f_i^v(x) \) is the driving force of pedestrian \( i \), which is modeled as:

\[
f_i^v(x) \triangleq \beta_i v_i + (1 - \beta_i) \frac{\theta_i - r_i}{\|\theta_i - r_i\|},
\]

where \( \beta_i = (1 - \gamma \tau_i^{-1}) \) and the second term captures the fact that, without any obstructions, a pedestrian tends to reach his destination in the shortest possible route at his desired speed \( v_i^\gamma \), and \( \tau_i \) models the ith pedestrian’s relaxation time, representing the time taken by pedestrian \( i \) to reach the desired velocity. The total external force on the \( i \)th pedestrian is:

\[
F_i^e(x) \triangleq \sum_{j \in V_p} A_{ij} h_a(r_i - r_j) - R_{ij} h_r(r_i - r_j) - \sum_{j \in B} B_{ij} h_b(r_i - r_j).
\]
Note that if \( B_{ij} \neq 0 \), then \( A_{ij} = -R_{ij} = 0 \) and vice versa.

III. DYNAMICS INFERENCE

We propose to infer the crowd social dynamics parameters through observing the position vectors \( r(t) \) only. Specifically, from (7) and (8) we can write
\[
z_i(t) \triangleq r_i(t+1) - r_i(t)
\]

\[
= \gamma f_i^w(x(t-1)) + \gamma^2 F_i^v(x(t-1)) + w'(t)
\]

where \( w'(t) \sim \mathcal{N}(0, \nu^2 I) \) and \( \nu^2 \) is the noise variance. From (8) and (9), the conditional density of \( z_i(t) \) given the latent portion of the state \( \sigma_i(t-1) \) is written as

\[
p(z_i(t)|\sigma_i(t-1)) = \frac{1}{\sqrt{2\pi \nu^2}} \exp \left( -\frac{1}{2\nu^2} \| z_i(t) - \mu_i(t) \|_2^2 \right),
\]

where

\[
\mu_i(t) = \gamma^2 F_i^v(x(t-1)) + \gamma \beta_i v_i(t-1)
\]

\[
+ \gamma(1-\beta) v_i^0 \left\| \theta_i(t-1) - r_i(t-1) \right\|_2
\]

and \( v_i(t) = z_i(t)/\gamma \) from (7).

Let \( \xi \triangleq \{ \Pi, A, B, R \} \) be the set of unknown parameters to be estimated. Also, the unobservable component, \( \sigma_i(t) \) of the state sequence for each agent, is unknown. Let \( z_{\xi} \triangleq [z_1(1) \cdots z_i(T)], \varpi_{\xi} \triangleq [\varpi_1(0) \cdots \varpi_i(T-1)] \) be the set of observations and the state sequence for agent \( i \), respectively, and \( z \triangleq [z_1 \cdots z_N], \varpi \triangleq [\varpi_1 \cdots \varpi_N] \).

To estimate \( \xi \) we employ Viterbi training or segmental \( k \)-means method [21] by tackling

\[
\hat{\xi}_k = \arg \max_{\xi} \left\{ \max_{\varpi} p(z, \varpi|\hat{\xi}_{k-1}) p(\hat{\xi}_{k-1}) \right\},
\]

(13)

using the standard alternating maximization paradigm, where \( p(\hat{\xi}_{k-1}) \) is the prior assumed on the parameters \( \hat{\xi}_{k-1} \) which helps to regularize the problem by imposing constraints on matrices \( A, B, R \). Also, the inner maximization is only with respect to \( \sigma \) since other components of \( \varpi \) are known.

The outer maximization in (13) maximizes the state-optimized joint likelihood of the data, where the inner maximization finds the most probable states sequence. As opposed to the expectation maximization (EM) algorithm which maximizes the expected likelihood (i.e., a finite sum over all possible state sequences), the Viterbi training method is computationally less expensive than EM and it allows to regularize the parameters by using a prior distribution to compute their maximum a posteriori estimate (MAP).

The inner maximization in (13) with respect to the state sequence \( \sigma \) for agent \( i \) is performed by equivalently solving

\[
\arg \max_{\sigma_i} \log p(z_i, \varpi_i|\hat{\xi}_{k-1}), \forall i \in \mathcal{V}_p,
\]

(14)

where

\[
\log p(z_i, \varpi_i|\hat{\xi}_{k-1}) = \sum_{t=2}^{T} \log p(z_i(t)|z_i(t-1)) + \log p(\sigma_i(t-1)|\sigma_i(t-2))
\]

(15)

This translates to using a dynamic programming approach such as the Viterbi algorithm to estimate the most likely state sequence \( \sigma_i \) for agent \( i \) and then updating the state transition matrix \( \Pi \) by counting the number of transitions made from one state to another in all the state sequences \( \sigma_1, \ldots, \sigma_N \).

The outer-maximization with respect to \( g \) is carried out after updating the transition matrix \( \Pi \). It is equivalent to Bayesian MAP estimation undertaken by solving a regularized recovery problem. Note that the network is typically large with \( N > 1 \), yet the amount of data acquired is limited in comparison. We exploit prior knowledge of the system and use it to regularize the recovery problem.

In addition to the constraints that were specified in the previous section, the matrix of attraction coefficients, \( A \), is regularized to be sparse. The matrix of repulsion coefficients, \( R \), is restricted to be the sum of a rank one matrix \( R_e = r11^T \), which characterizes the inherent repulsion between agents and another matrix \( R_s = c1^T \) where \( c \in \mathbb{R}^N \) is a sparse vector. The support of \( c \) is indicative of agents found repulsive by everyone. Furthermore, the diagonal entries of both \( A \) and \( R \) are confined to be zero in accordance with the modeling assumption. Thus, \( R = R_e + R_s - \text{diag}(R_e + R_s) \) where diag \((X)\) is a diagonal matrix with the diagonal elements of matrix \( X \). The matrix of repulsion from boundaries, \( B \), is also assumed to have rank one, \( B = b11^T \). This is by assuming that all the boundaries are equally repulsive (such as walls). Hence, we solve the following least squares problem:

\[
\min_{A,c,r,b} \sum_{t=1}^{T_{max}-2} \sum_{\ell=0}^{\mathcal{V}_p} \| z_i(t+1) - \gamma f_i^w(x(t)) - \gamma^2 F_i^v(x(t)) \|_2^2 + \rho_1 \| \text{vec}(A) \|_1 + \rho_2 \| c \|_1
\]

(16)

s.t. \( A_{ii} = 0, \forall i \in \mathcal{V}_p \),

where \( \rho_1 \) and \( \rho_2 \) are regularization constants and the last two terms in the objective promote sparsity in \( A \) and \( c \) respectively.

IV. NUMERICAL RESULTS

In order to simulate the trajectory of pedestrians, a test area is setup as shown in Fig. 1 wherein pedestrians are allowed to move along the four corridors in either direction. The locations marked, I-VIII, in the figure are referred to as the corner points. There are \( N = 20 \) Pedestrians who are randomly assigned their initial location on the test area and based on state sequence generated from the state transition matrix, \( \Pi \), they follow paths along the edges of a polygon whose vertices (or corner points) constitute their intermediate destinations. The states here refer to the directed paths between intermediate points such as I-II, II-III, II-I etc. Therefore, we have a total of 20 states as shown in the Fig. 1. We impose a specific structure on transition matrix \( \Pi \) so that the probability of an agent staying in one of the states along the corridors (these are the states highlighted in red in Fig. 1) is higher, but the probability of the agent staying in the states in the junction between the corridors is low.

The positions, \( y_i(t) \), of all the agents \( i \in \mathcal{V}_p \) are tracked for \( t = 1, 2, \ldots, T_{max} \) instances of time. These measurements constitute the training set used to estimate the unknowns \( \xi \). Then, the trajectories of agents are predicted from \( t = T_{max} + 1, \ldots, T_{max} + T_I \) with the help of parameters inferred using the training set.
A. Initialization of parameters

To solve (13), we need to initialize the unknown parameters, i.e., finding $\xi_0 = \{A_0, B_0, R_0, \Pi_0\}$. We assume that we know the structure of the state transition matrix $\Pi_0$ and also know the most probable next state $j$ for every state $i$ i.e. the position $j$ in every row $i$ of the maximum value of transition probability. Using this, we initialize $\Pi_0$. This is a reasonable assumption as the layout of the area in which the data is collected is usually known.

To initialize $A_0, B_0,$ and $R_0$, we need to estimate an initial sequence of corner points $\theta_i(t), t = 1, \ldots, T_{\text{max}}$ for all agents $i \in V_p$ before we can solve the inference problem (16). In order to estimate the corner point, $\theta_i(t)$ a pedestrian $i$ is traveling to at time $t$, we first calculate the distances of these corner points from the pedestrian at time $t$, $u^p_i(t) = \|r_i(t) - \theta^p_i\|_2, p = 1, 2, \ldots L$ and choose the closest corner point to which the pedestrian is moving towards. This is done by calculating the difference in distance from corner points at times $t$ and $t-1$ and the minimum is chosen among corner points who have a negative first difference:

$$\theta_i(t) = \theta_i^p, \text{ where } u^p_i(t) = \min\{u^1_i(t), \ldots, u^L_i(t)\}$$

$$u^p_i(t) - u^p_i(t-1) \leq 0. \quad (17)$$

With the sequence of corner points, we estimate the boundary force value $b_0$ by considering measurements from those agents at particular time instances when they are closer to one of the boundaries, but isolated from all the other agents i.e.

$$b_0 = \min_b \sum_{(i,t) \in B_{\text{init}}} \|x_i(t) - \gamma f^a_i(x(t-1)) + \gamma^2 b F_b\|_2^2 \quad (18)$$

where $B_{\text{init}}$ contains the ordered pair of agents and times $(i,t)$ when they are far from everyone and close to boundaries and $F_b = \sum_{j \in B} h_b(y_j(t) - y_j(t))$. Then, we solve (16) by assuming the knowledge of $b_0$ to initialize $A_0$ and $R_0$ alone. Once $\xi$ is initialized, Viterbi training is employed iteratively to estimate $\xi$ as described in equation (13).

B. Simulation results

The parameters for the model are as follows: the step size $\gamma$ is set at 0.1s, the relaxation time $\gamma$ is set at 0.5s, the initial velocities of the agents are assumed to be Gaussian distributed with mean 1.34ms$^{-1}$ and standard deviation 0.26ms$^{-1}$ [9]. The training set contains $T_{\text{max}} = 500$ data points.

In the simulation conducted, the attraction adjacency matrix $A$ contains only two pairs of edges, $(2,3), (7,8)$ and their reciprocal, $V_p$ can be partitioned into $M = 18$ weakly connected components, which are mostly singleton except for two which are given as $C_m = \{2,3\}$ and $C_{\text{mt}} = \{7,8\}$. Fig. 2 represents the actual and the inferred $\Pi$ matrices. As seen in the inferred matrix, state transitions with low probability i.e. the states in the junction (V-VI to VI-V in Fig.1) which occur with very low probability are not identified as the agents stay in these states for very short durations of time and hence these states are observed quite rarely in the training set.

Fig. 3 presents the actual and the inferred $A$ matrices, where the edges $(2,3), (7,8)$ are successfully inferred. In addition, we observe a few other artifacts, but at significantly lower strengths. Fig. 4 presents the actual and inferred matrices for parameter $R$. The pedestrian who was generated to be highly repulsive to all other agents in the area has been identified.

To quantify the network inference performance, the normalized error of $X$ is defined as:

$$e_X(X) = \frac{\|X - \hat{X}_k\|_{\text{norm}}}{\|X\|_{\text{norm}}}. \quad (19)$$

where $X \in \xi$ and $\hat{X}_k$ is the estimated value of $X$ at the end of iteration $k$. The normalized errors made while inferring the elements of $\xi$ at the end of each Viterbi training iteration are plotted in Fig. 5. We observe an expected reduction in the error at each iteration $k$. It is noted that the errors may be further reduced by increasing the size of the training dataset.

Using the inferred coefficients, the trajectories of agents are forecasted for $T_F = 150$ time steps ahead i.e. from $t = T_{\text{max}} + 1, \ldots, T_{\text{max}} + T_F$. Fig. 6 shows the forecasted paths for a few agents. The solid lines represent the path traced by agents for the duration of $t = 1, \ldots, T_{\text{max}}$, while the dashed and the dotted lines represent their actual and forecasted paths for the duration of $t = T_{\text{max}} + 1, \ldots, T_{\text{max}} + T_F$, respectively.

V. Conclusion

In this paper, we have shown that the data collected by IoT-enabled wearables may be used to infer the social behavior of pedestrians via an approach based on the HM social force model, which accounts for social interactions. We have also incorporated the strong coupling between a pedestrian and his/her fellow pedestrians as well as the boundaries and obstacles in his/her path. This paves way for the usage of the model...
described in various fields where the a-priori knowledge of the agents’ behavior leads to more efficient endeavors such as in the case of spectrum allocation by a base station to its users. If we have knowledge of how an user is going to behave, i.e., if s/he is going into a region of blockage or a region of better channel, we can use this knowledge to prioritize or neglect that agents for the time being, respectively. Agents’ predicted behaviors can also be employed to obtain the best path to reach their destination in the shortest time. Future work involves the modeling of the time dependent, fading forces acting on the agents by various attractions in the surroundings. The state transition matrix can also be modeled to accommodate the possibility that an agent’s transition probability may also depend on the actions or mere presence of certain agents in his proximity; thereby adding in layers of opinion dynamics to the model.

REFERENCES