Learning and Controlling Network Diffusion in Dependent Cascade Models

Jiali Du, Pradeep Varakantham, Akshat Kumar and Shih-Fen Cheng
School of Information Systems
Singapore Management University

***Abstract*—Diffusion processes have increasingly been used to represent flow of ideas, traffic and diseases in networks. Learning and controlling the diffusion dynamics through management actions has been studied extensively in the context of independent cascade models, where diffusion on outgoing edges from a node are independent of each other. Our work, in contrast, addresses (a) learning diffusion dynamics parameters and (b) taking management actions to alter the diffusion dynamics to achieve a desired outcome in *dependent* cascade models. A key characteristic of such dependent cascade models is the flow preservation at all nodes in the network. For example, traffic and people flow is preserved at each network node. As a case study, we address learning visitor mobility pattern at a theme park based on observed historical wait times at individual attractions, and use the learned model to plan management actions that reduce wait time at attractions. We test on real-world data from a theme park in Singapore and show that our learning approach can achieve an accuracy close to 80% for popular attractions, and the decision support algorithm can provide about 10-20% reduction in wait time.**

I. Introduction

Diffusion processes describe how ideas, influence, and people spread over an underlying network, which for example may be a social network [1] or a transportation network [2]. Understanding diffusion dynamics is important as it helps predict and control the contagion spread in a network. The independent cascade (IC) model of [1] and its variants have been quite successful in modeling such a diffusion process over a network in a variety of domains [3],[4],[5].

In our work, we address several features of cascades in real-world that are not modeled in the existing IC model. For example, in many networks, it is not feasible to track the status of each entity in the cascade (e.g., tracking visitors moving in a theme park). Therefore, diffusion dynamics must be learned from the *aggregate* observed data for the underlying contagion. Some examples include finding the most probable path of migratory birds [6], understanding the evolution of traffic in a transportation network [7],[2], emotional contagion in a crowd evacuation scenario [8] and mobility pattern of visitors in a theme park. Therefore, we develop techniques based on mathematical optimization that can learn the underlying dynamics of the diffusion process using only aggregate data.

We incorporate realistic features, such as modeling queues at nodes in the underlying network (e.g., attractions in a theme park) while learning the diffusion dynamics based on aggregate data. We further augment the IC model with flow conservation that is required when modeling the diffusion process in problems such as traffic flow diffusion or visitor flow across theme park attractions. Incorporating such features results in a *dependent cascade* model, where the diffusion over the outgoing edges must satisfy flow conservation, and is no longer independent for each edge.

Given the learned model of the diffusion dynamics, the next key problem we address is how to take decisions within a given budget to alter the dynamics of the underlying diffusion process to optimize some performance criterion. We are motivated by the problem of placing sideshows in an overcrowded theme park to reduce congestion at different attractions. Sideshows alter the underlying flow of visitors by attracting some fraction of visitors to themselves, thereby reducing congestion at main attractions in the theme park. When and where to place sideshows using limited resources is the key decision making problem we address.

We validate our models in the context of a real theme park in Singapore. First, given historical aggregate information about the number of people waiting at theme park attractions provided by the theme park operator, we learn the mobility pattern of visitors moving across various attractions in the theme park. Empirically, our learning approach provides an accuracy of about 80% for popular attractions, providing good empirical support for our diffusion models. Our decision making approach for sideshow placement provides a reduction in wait times by about 10% over baseline approaches, and up to 20% over the case when no sideshows are present.

Related Work In the social network literature, learning the parameters of a IC model based diffusion process has become a flourishing research area [9],[5],[10],[11],[12],[13]. Myers and Leskovec [9] formulate the problem of parameter learning using convex optimization. Gomez *et al.* [5] address a similar problem using submodularity based optimization. Netrapalli and Sanghvi [10] address the complementary question of how many observed cascades are necessary to correctly learn the structure of a network. There has also been work on learning the parameters using features of the diffusion process, such as the language of tweets [11] in a Twitter network and geographical features to learn an endangered species movement parameters [14]. Daneshmand *et al.* [12] investigate the network structure inference problem using an l_1-regularized likelihood maximization framework recently. Qu *et al.* [13] utilized data summarization tools to learn the diffusions over large and dynamic online social networks. Our work is different in the sense that the underlying diffusion process has dependent outgoing flows from a node to maintain flow conservation unlike the IC model. We also assume that only aggregate information is available, rather than individual-level tracking information required in the IC model.

The dependent cascade model we use is closely related to the collective diffusion model in [2]. Our work addresses an enriched version of such a collective diffusion model as data requirements in our approach are much weaker than that of [2]. We concretely show later the differences between our approach and that of [2]. In addition, a significant contribution of our work is to formulate and validate diffusion models with real world data, which was not provided in [2].

Having learned a model of the diffusion process, the natural next step is decision-making in the context of the learned model. The goal is typically to find the set of actions that result in a diffusion process with certain desirable properties subject to operational constraints [1], [4], [15],[16]. This kind of decision-making has numerous applications. For example, in disease control, the decision is which nodes to vaccinate to curb the spread of disease while in advertising, the question is which nodes to target to encourage the adoption of a product. In traffic scenario, the decision is to route cooperate vehicles such that can ease congestions, the challenge then becomes how to capture the dynamic and congestion situation and control the vehicles. Our work proposes and develops an optimization formulation for similar decision making problem within the context of dependent cascades and aggregate flow.

II. Network Diffusion with Dependent Cascades

We provide an operational model to represent diffusion in a time indexed graph, $G(V, E, T)$ with dependent cascades. Before explicitly explaining our model, we start with the well known independent cascade model [1],[15] that is used to represent spread of ideas, influence. Every edge $(u, v) \in E$ at time t is associated with a transition probability $p^t_{u,v}$ representing the probability that node v will be activated if node u was previously activated. In the independent cascade model, probabilities associated with outgoing edges from a node are independent of each other and hence the cascades in different parts of the network are independent.

On the other hand, since we primarily consider diffusion of agents (people/vehicles) where flow is preserved, the probabilities associated with edges going out of a node are dependent on each other. Specifically, we have the following flow preservation dependency for every node u and time t: $\sum_w p^t_{u,w} = 1$. Namely, every agent coming out of u move to one of the nodes w according to the diffusion dynamics, **p**. This paper focus on the following learning and planning problems in the context of such dependent cascade models of network diffusion:

(a) Learning: Compute the transition probabilities, **p** that maximize the likelihood of observing the aggregate observations **n** (ex: number of people waiting in queues at attractions in a theme park over multiple days).

(b) Planning: Given **p** and budget B, compute the plan for execution of management actions that achieve a desired objective.

A. Application to Theme Park Management

In recent years, theme parks have been an important driver in the growth of this industry. Unfortunately, a vibrant growth in the theme park industry comes hand-in-hand with worsening congestion and increased wait times. From field observations and our computational experiments, we notice that the wait times on weekends and holidays at popular attractions, particularly in Asian theme parks reaches 2-3 hrs. The key motivation for research in this paper is to improve visitor experience by reducing overall wait times.

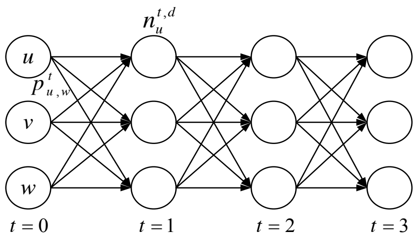


Fig. 1: Time indexed graph representing diffusion of visitors for 3 time steps.

Time indexed graph in figure 1 represents diffusion of people at theme park. Each node represents an attraction. An attraction being active at a time step t indicates that there are people who are coming out of the attraction after getting serviced at that time step. $p^t_{u,v}$ associated with an edge between attractions u and v indicates the probability that a visitor coming out of attraction u would move to the attraction v at time t. Since visitors stay within the theme park, summation of transition probability over all outgoing attractions v from u is 1, i.e., $\sum_v p^t_{u,v} = 1$.

At a theme park, it is impractical to track every visitor transitioning between attractions (referred to as **x** henceforth). Instead, there are both people and sensors to guide and track people at individual attractions. We use this information from multiple days, represented as $n^{t,d}_u$ (number of people waiting at attraction u at time t in day d). The **learning problem** can be formally defined as:

$$\max_{\mathbf{p}} \quad \mathcal{L}(\mathbf{p}|\mathbf{n}) \tag{1}$$

where $\mathcal{L}(\mathbf{p}|\mathbf{n})$ represents the likelihood of parameters **p** given the observed **n**.

Theme parks typically conduct moving road shows and/or photoshoots with cartoon characters near attractions that have high wait times to ensure people spend lesser time waiting at attractions. However, current practice of placement of such side shows is adhoc and does not consider the diffusion dynamics over time steps. To address this, our planning approach will compute a placement of sideshows, **l**, on edges to minimize wait times while respecting a budget available for side shows. Thus, the **planning problem** can be formally defined as:

$$\min_{\mathbf{l}} \sum_{d,t,u} \omega^t_u(\mathbf{l}, \mathbf{p}) \qquad \textbf{s.t.} \sum_{u,v,k,t} f(l^{t,k}_{u,v}) \le B$$

where $\omega^t_u(\mathbf{l}, \mathbf{p})$ is the wait time at time step t for attraction u given diffusion dynamics **p** and placement of side shows given by **l**. f represents the cost of placing a side show and B is the budget available.

III. Learning Diffusion Dynamics

We now describe our method for solving the learning problem described in Equation 1, when we only have access to aggregate observations **n**. We assume that the diffusion dynamics from any node u described by $\mathbf{p}_u$ is a standard probability distribution characterised by a few parameters. Specifically, we explore the most relevant ones, namely Multinomial, Multinomial Dirichlet and Poisson. The notation employed in this and subsequent sections is provided in Table I. Boldface letters are used to represent vectors of the items described by the corresponding normalface letter.

TABLE I: Notation

Variable	Definition
$p^t_{u,v}$	Transition probability between nodes u and v at time step t
$n^{t,d}_u$	Number of agents in node u at time t on day d
s_u	Service rate (number of agents serviced in one time step) of node u
D	Set of cascades or days on which observations about **n** are made
$x^{t,d}_{u,v}$	Number of agents moving from node u to node v at time t on day d
$l^{t,k}_{u,v}$	Binary variable that is set to 1 if side show of type k is placed on edge (u, v) at time t
β^k	Percentage of diffusion attracted by side show of type k

Before we describe our approach, we note that our learning problem is different than the collective flow diffusion (CFD) model of [2] that was developed to understand traffic flow. In the CFD model, it is assumed that the total number of agents that exit and enter each node are known. However, in many domains, such as theme parks, trade shows, travel tours, this type of data is not available. Instead, we only observe the total number of people waiting to be serviced at a node u. This makes learning more complex due to more hidden variables than the CFD model. We develop additional constraints to reflect such visitor queues as explained later using constraint 5 in Table II.

We provide multinomial distribution based diffusion model[1] and develop optimization based learning algorithm for computing the parameters.

A. Multinomial Distribution Based Diffusion

Multinomial distribution – a generalization of the binomial distribution – is a categorical distribution where each trial results in exactly one of k possible outcomes with probabilities $P_1, \cdots, P_k$ (so that $P_i \geq 0, \forall i = 1, \cdots, k$) and $\sum_{i=1}^{k} P_i = 1$. We represent diffusion from each node, u at time t as a multinomial distribution with probabilities given by $\{p^t_{u,v}\}_{v \in V}$ for each of the outcomes $v \in V$. Therefore, the probability of observing $x^{t,d}_{u,v}$ number of transitions[2] from node u to node v, $x^{t,d}_{u,w}$ number of transitions from node u to node w and so on for any day/cascade d is given by:

$$Pr(\mathbf{x}^{t,d}_u|\mathbf{p}) = \frac{(\sum_z x^{t,d}_{u,z})!}{\prod_z x^{t,d}_{u,z}!} \prod_z (p^t_{u,z})^{x^{t,d}_{u,z}}$$

where $\sum_z p^t_{u,z} = 1$ and $x^{t,d}_{u,z}$ represents the number of times (frequency) there was a transition from u to z at time t on day d. Since, we do not observe either the probabilities, **p** or frequencies, **x**. We learn them by maximizing the likelihood, $\mathcal{L}(\mathbf{p}|\mathbf{x}, \mathbf{n})$.

TABLE II: GetDiffusionDynamics($\boldsymbol{n}$, $\boldsymbol{s}$)

$$\textbf{max:} \sum_d \sum_u \sum_t \Big(\log\big((\sum_z x^{t,d}_{u,z})!\big) - \sum_z \log(x^{t,d}_{u,z}!) + \sum_z x^{t,d}_{u,z} \log(p^t_{u,z}) \Big)$$

$$\textbf{s.t.} \quad n^{t+1,d}_u = n^{t,d}_i + \sum_z x^{t,d}_{z,u} - \sum_z x^{t,d}_{u,z}, \quad \forall t, d, u \tag{5}$$

$$\sum_z x^{t,d}_{u,z} \leq min(s_u, n^{t,d}_u), \quad \forall t, d, u \tag{6}$$

$$\sum_z p^t_{u,z} = 1, \quad \forall t, u \tag{7}$$

$$x^{t,d}_{u,z} \in \mathbb{N}_0, \quad \forall t, d, u, z \tag{8}$$

$$0 \leq p^t_{u,z} \leq 1, \quad \forall t, u, z \tag{9}$$

More specifically, over all the attractions, likelihood is defined as follows:

$$\mathcal{L}(\mathbf{p}|\mathbf{x}, \mathbf{n}) = Pr(\mathbf{x}, \mathbf{n}|\mathbf{p}) = \prod_{d \in D} \prod_{t \in T} \prod_{u \in V} \frac{(\sum_z x^{t,d}_{u,z})!}{\prod_z x^{t,d}_{u,z}!} \prod_z (p^t_{u,z})^{x^{t,d}_{u,z}} \tag{2}$$

where

$$\sum_z x^{t,d}_{u,z} = \begin{cases} n^{t,d}_u & \text{if} \ \ n^{t,d}_u < s_u \\ s_u & otherwise \end{cases}$$

Given the equivalence of maximizing likelihood and maximizing log likelihood, we employ the following objective

$$\max_{\boldsymbol{p}} \log \sum_{\boldsymbol{x}} Pr(\boldsymbol{n}, \boldsymbol{x}|\boldsymbol{p}) \tag{3}$$

To make it computationally simpler, we use the following approximation:

$$\max_{\boldsymbol{p}, \boldsymbol{x}} \log Pr(\boldsymbol{n}, \boldsymbol{x}|\boldsymbol{p}) \tag{4}$$

This approximation is in the same spirit as the one in Sheldon *et al.* [17]. The main intuition is that for categorical distributions, such as Binomial distribution, the mode is very close to the mean. The above optimization problem can be formulated as a non-linear program as shown in Table II. The objective function is the logarithm ($\log$) of Eq.(2). The first and the second constraint jointly represent the flow conservation at each node. In the first constraint, the number of visitors at a node u at time $t + 1$ according to a cascade d ($= n^{t+1,d}_u$) is constrained to be equal to the number of visitors at the same node at time t with the addition of in-flow into the node ($= \sum_z x^{t,d}_{z,u}$) and subtraction of the out-flow from the node ($= \sum_z x^{t,d}_{u,z}$) at the same time step. The second constraint ensures that the out-flow is equal to the minimum of the service rate at node u and the number of people currently waiting to be serviced at the node at time step t. Rest of the constraints enforce basic properties of the diffusion model.

We solve the optimization problem in Table II using a commercial non-linear solver called Lingo (http://www.lindo.com).

IV. Controlling Diffusion Dynamics

We now describe our mechanism to compute plans of management actions that will be used to control diffusion dynamics. In this work, we consider management actions that

[1] We can also employ Dirichlet-Multinomial distribution and Poisson distribution based diffusion models. Similar optimization problems as Table II that maximize the likelihood can be provided for both distributions.

[2] Since **x** represents an observation, it can be different for different days or cascades, $d \in D$.

can be viewed as dampeners that are placed on an edge to absorb the diffusion on that edge for a certain time duration. In the context of a theme park, these management actions correspond to side shows that can be placed between attractions for a limited time. Depending on their type, such management actions have an associated cost and impact on the diffusion. For instance, a photo opportunity with a cartoon character only attracts a few people and is not typically expensive as only one actor is involved. In contrast, an elaborate road show attracts most visitors traveling on that edge, and is more expensive due to multiple actors being involved. .

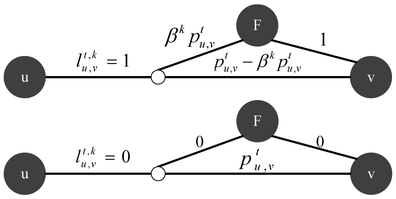


Fig. 2: Representation of Management Actions

Figure 2 provides a visual representation of how the diffusion dynamics are altered due to a management action. As explained earlier, $p^t_{u,v}$ represents the preference of visitors coming out of node u to move to node v; β^k is the proportion of visitors absorbed on an edge for one time step due to the management action of type k. When $l^{t,k}_{u,v} = 1$, there is a management action of type k executed on edge between u and v at time step t. In this case, β^k proportion of visitors are absorbed into the buffer node F for the current time step t. Such visitors at the buffer node move with probability 1 to their original intended destination v at time step $t+1$. Thus, introducing the buffer node via the action $l^{t,k}_{u,v} = 1$ reduced congestion at node v at time step t. Action $l^{t,k}_{u,v} = 0$ indicates that there is no management action of type k executed on edge between u and v at time step t. In this case, the diffusion probability remains the same as $p^t_{u,v}$.

Given the diffusion dynamics, $\mathbf{p}$, the goal is to compute the plan $\mathbf{l}$ of management actions that will minimize the total waiting time or latency over all the nodes across all time steps over different realisations of diffusion. We also assume that each management action has a cost, and there is a fixed available budget. Two existing methods that have been employed for controlling/influencing diffusion are:

- Exploiting submodularity: The problem of selecting a fixed number of nodes in a social network that will maximize influence in the context of independent cascade model [1] is solved by exploiting submodularity of the objective. Specifically, because of submodularity, nodes can be greedily selected one after another based on their marginal addition to the overall influence. Such an approach provides a solution that is at least $1 - \frac{1}{e}$ (=63%) of the optimal. Unfortunately, with dependent diffusion dynamics, the problem of minimizing wait time is not submodular[3]. Given the relevance, we experimentally benchmark the performance of our approach against the greedy approach, even though it does not provide quality guarantees.
- Employing Sample Average Approximation (SAA): The problem of buying parcels of land to maximize the population of rare species [15] is formulated as a stochastic optimization problem. The key idea is that instead of solving the stochastic optimization problem directly, a solution is computed for a few samples from the diffusion process. Because of independence in cascades, samples can be generated before the optimization.

  Unfortunately, when simulating a large population of visitors moving in a theme park, we would need to sample the trajectory for each of them in the context of SAA. This leads to a prohibitively large size of the decision problem when formulated using a mathematical program, and is not scalable.

Therefore, the main contribution of this work with respect to controlling diffusion dynamics is a scalable approach that substitutes the computation of expected wait time (expectation over trajectory samples of visitors) with wait time for expected numbers of visitors (expectation over diffusion dynamics) over all nodes and all time steps. Specifically, we denote $E_{\boldsymbol{P}}[\sum_{i,t} g(n^t_i)]$ as the expected wait time for the joint multinomial distribution $\boldsymbol{P}(= \prod_t \prod_i \frac{(\sum_j x^t_{i,j})!}{\prod_j x^t_{i,j}!} \prod_j (p^t_{i,j})^{x^t_{i,j}})$ over all nodes and all time steps. We use $\sum_{i,t} g(E_{q^t_i}[n^t_i])$ to represent the wait time of expected number of visitors with distribution q^t_i for each individual attraction i at time t.

Proposition 4.1: With liner function $g(x)$ measuring the wait time (i.e. $g(x) = ax + b$), $E_{\boldsymbol{P}}[\sum\limits_{i,t} g(n^t_i)] = \sum\limits_{i,t} g(E_{q^t_i}[n^t_i])$.

Proof: $E_{\boldsymbol{P}}[\sum_{i,t} g(n^t_i)] = \sum_{i,t} E_{q^t_i}[g(n^t_i)] = \sum_{i,t} E_{q^t_i}[an^t_i + b] = \sum_{i,t} aE_{q^t_i}[n^t_i] + b = \sum_{i,t} g(E_{q^t_i}[n^t_i])$. ■

The resulting optimization formulation is much smaller and scalable when compared with the sampling approach using SAA. It should however be noted that even with using expected numbers of visitors, the problem of minimizing wait time using management actions remains NP-Hard.

Proposition 4.2: The problem of minimizing wait time for expected numbers of visitors at nodes over all time steps by using management actions $\mathbf{l}$ and a given budget is an NP-Hard problem.

Due to space constraints, we omit the proof. We prove this by showing that 0/1 knapsack problem is a special case of our problem. Specifically, we reduce the 0/1 knapsack problem to our problem. The key insight is that minimizing wait time at all nodes at all times is equivalent to maximizing number of agents in buffer nodes at all times. By mapping items in knapsack to management actions, weight of an item, w^k to the cost, c^k of executing management action and value of an item and finally v^k to the overall increase in number of agents at buffer node due to execution of management action, we demonstrate this reduction. ■

Table III provides an optimization formulation to compute $\mathbf{l}$ that minimizes the average wait time for expected number of visitors at every node and at every time step. Let $|U|$ denote the total number of attractions in the theme park, and T denote total time steps. Each time step in our case denoted a block of 1 hour period during the day resulting in $T = 9$. The objective function is the average wait time over all the attractions u

[3] Due to space constraints, we do not include the proof here. However, it is easy to identify a counter example.

across all time steps. This particular metric was suggested to us by the theme park operator and is a key performance indicator for the theme park management. While this formulation contains non-linear constraints, we will subsequently provide linear equivalents. We refer to this approach as CDON (Controlling Diffusion through OptimizatioN).

TABLE III: CDON($\boldsymbol{p}$, $\boldsymbol{s}$, $\boldsymbol{n}^{\mathbf{0}}$)

$$\begin{aligned}
&\textbf{min: } \frac{1}{|U|\cdot T}\sum_{u,t}\frac{n_u^t}{s_u} && \\
&\textbf{s.t.}\quad n_u^t + \sum_z x_{z,u}^t - \sum_z x_{u,z}^t = n_u^{t+1} && \forall u,t \quad (10)\\
&\sum_z x_{u,z}^t = y_u^t && \forall u,t \quad (11)\\
&y_u^t = min(s_u, n_u^t) && \forall u,t \quad (12)\\
&x_{u,z}^t = y_u^t \cdot \Big[p_{u,z}^t - \sum_k \beta^k \cdot l_{u,z}^{t,k} \cdot p_{u,z}^t\Big] && \forall u,z,t \quad (13)\\
&x_{u,F}^t = y_u^t \cdot \Big[\sum_k \beta^k \cdot l_{u,z}^{t,k} \cdot p_{u,z}^t\Big] && \forall u,z,t \quad (14)\\
&x_{F,z}^t = \sum_u \sum_k \beta^k \cdot l_{u,z}^{t-1,k} \cdot y_u^{t-1} \cdot p_{u,z}^{t-1} && \forall t,z \quad (15)\\
&\sum_k l_{u,z}^{t,k} \leq 1 && \forall u,z,t \quad (16)\\
&\sum_{t,u,z,k} l_{u,z}^{t,k} c^k \leq B && (17)\\
&l_{u,u}^{t,k} \in \{0,1\} && \forall u, v \neq u, k, t \quad (18)
\end{aligned}$$

Constraints (10) ensures that expected number of visitors in a node, u at time $t+1$ is equivalent to the expected number of visitors at u at time t plus the expected number of visitors transitioning out of u minus the expected number of visitors transitioning into u at time t. Constraints (11) are the flow preservation constraints, which ensure that all visitors coming out of a node go to one of the other nodes. This introduces dependencies across cascades. Constraints (12) ensures correct computation of number of visitors coming out of a node u. Service rate s_u indicates the number of visitors that are served in one time step. Therefore, the number of visitors served in any one time step is the minimum of s_u and number of visitors in u at time t, i.e., n_u^t. Constraints (13)-(15) ensure correct computation of the expected number of visitors[4] that move to other nodes and the buffer node due to placement of side shows. Constraints (16) and (17) are the constraint on for management actions. Constraints (16) ensures that for the same link at the same time, it is only allowed for one type of action. Constraint (17) enforces the budget constraints for the management actions.

As can be noted, Constraints (12),(13),(14),(15) all contain non-linear terms. The first non linear term is in Constraints (12). To provide a linearisation to this constraint, we use two binary variables, namely, d_u^t and e_u^t as follows:

$$\begin{aligned}
&d_u^t + e_u^t = 1; \quad y_u^t \leq n_u^t; \qquad y_u^t \leq s_u\\
&y_u^t \geq n_u^t - M\cdot(1-d_u^t); \\
&y_u^t \geq s_u - M\cdot(1-e_u^t)
\end{aligned} \qquad (19)$$

In these constraints M is a large positive number and the tightest bound for M is the largest value of n_u^t and s_u. The validity of these linear constraints can be ascertained by considering all possible values for the two binary variables, d_u^t and e_u^t. Next, we consider the non-linearity in constraints (13),(14) and (15), which is the term $y_u^t \cdot l_{u,z}^{t,k}$. Since y_u^t is positive and $l_{u,z}^{t,k}$ is a binary number, the linear equivalent constraints are:

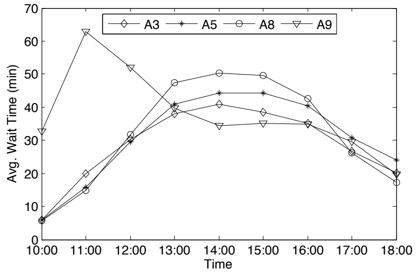


Fig. 3: Hourly wait time data for the four busiest attractions.

$$\begin{aligned}
&r_{u,z}^{t,k} \leq l_{u,z}^{t,k}\cdot M; \quad r_{u,z}^{t,k} \leq y_u^t;\\
&r_{u,z}^{t,k} \geq y_u^t - (1-l_{u,z}^{t,k})\cdot M; \quad r_{u,z}^{t,k} \geq 0
\end{aligned}$$

Again the validity of these constraints can be ascertained by considering both possible values for $l_{u,z}^{t,k}$.

V. Experiments: Learning Diffusion Dynamics

In order to demonstrate the utility of our approaches in computing diffusion dynamics, we use a 5-month long data set of wait times from a real theme park in Singapore which consists of 9 major attractions. Using the wait time data and service rate for each attraction in the data set, we get an estimate of how many visitors are currently waiting in queue at each attraction. We are unable to provide a map of the attractions due to confidentiality agreements.

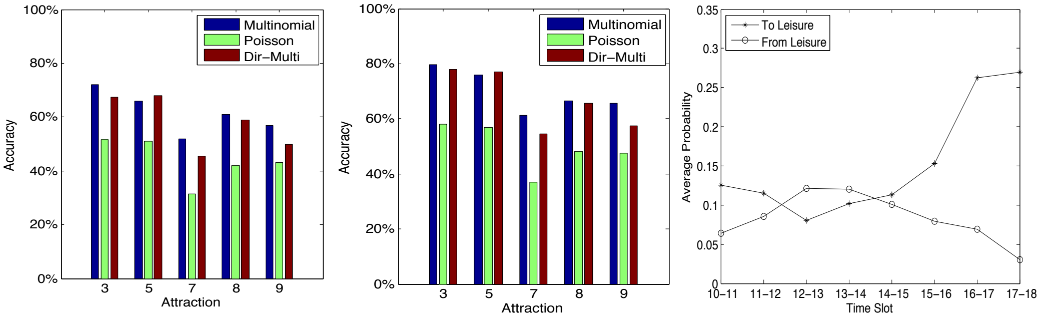
To account for lack of data on visitors entering and exiting the theme park as well as taking breaks, we introduce a new attraction called the 'leisure' node numbered 'A10'. This node is required to account for initial inflow of visitors and their exit. This node has infinite service rate and infinite capacity. We will show concretely how this can be captured using the leisure node.

Accuracy Figures 4(a) and (b) show the average accuracy achieved by different diffusion models using the 5-fold cross validation. Using the learned model parameters, for example, transition probabilities $\{p_{u,z}^t\}$ for the multinomial diffusion, we predict the number of people $\boldsymbol{n}$ waiting at each attraction at hourly intervals for all the days in the test data.

To compute the accuracy, we consider a fixed confidence interval. A predicted aggregate value n is considered correct for an attraction if it is within a particular threshold, say 25%, of the true n. Using this definition, we count the total accuracy for all the predictions with one prediction for each time step for each test day per attraction. Figure 4(a) and (b) show the accuracy for 25% and 30% threshold. We make the following observations.

A key observation from figures 4(a) and (b) is that attractions that have high wait times (3, 5, 8, and 9 in Fig. 3) also have a high accuracy of prediction (≈ 70%-80%). The accuracy is lower for attractions that are relatively lightly congested, such as attraction 7. Attraction 7 as well as attractions 1, 4 and

[4] When there are no side shows, expected number of visitors moving from node u to z at time t is given by $y_u^t \cdot p_{u,z}^t$

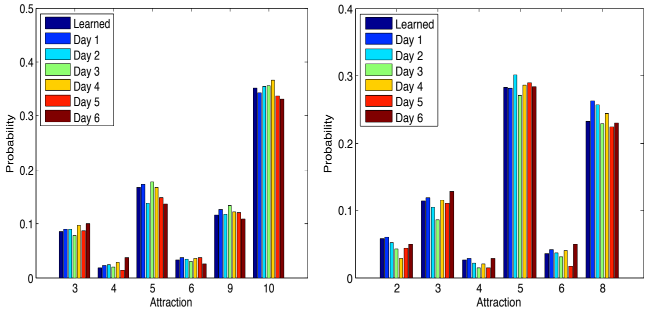


(a) 25% tolerance

(b) 30% tolerance

(c) Overview of leisure node

Fig. 4: Accuracy for 5 busiest attractions and leisure node



(a) Parameters for A8 for 4PM-5PM

(b) Parameters for A9 for 12PM-1PM

Fig. 5: Learned parameter verification for the multinomial based diffusion (figures best viewed in color)

6 have an average wait time of 15 minutes. This may contribute to lower accuracy as we use hourly wait times. Using finer grained reporting intervals (such as every 15 min.), we expect to increase the accuracy for such lightly congested attractions. We are currently in the process of collecting such fine grained data from the theme park operator. Importantly, our approach is able to provide good accuracy for all the heavily congested attractions validating our models and learning algorithms.

We also observe from figures 4(a) and (b) that the multinomial distribution consistently provides higher accuracy than the other two distributions. A key insight is that the poisson distribution performed significantly worse in comparison to the other two distributions. Poisson is extensively used to model individual queues. Our result indicates that it may not be an ideal distribution to represent a network of queues, where the status of one queue depends on the status of other queues. While not provided here, we also have graphs that show how the accuracy varies continuously with the tolerance threshold.

Learned Parameter Verification We now explain the transitions into the leisure node and provide a verification mechanism for the learned parameters obtained for the multinomial diffusion model ($\{p^t_{u,z}\}$). Figure 4(c) shows the average transition probability *from* and *to* the leisure node over all the attractions for each time interval of the day. The legend 10-11 refers to the interval 10AM to 11AM. This figure clearly shows that during the beginning of the day, the transitions from leisure node to major attractions gradually increases until reaches the peak at around 12PM to 1PM. Whereas the transition probability to the leisure node is quite low. This is expected as during the morning and early afternoon, visitors arrive in to the theme park and they are joining the queue of those major attractions. In contrast, the transition probability to the leisure node increases significantly towards the latter part of the day. This is also expected as visitors exit the theme park during late afternoon and evenings. Thus, the concept of leisure node is able to capture such visitor movements succinctly.

To verify the parameters generated using the optimization formulation in table II, we compare the learned parameters $\boldsymbol{p}$ against the $\boldsymbol{p'}$ calculated for each day from the $\boldsymbol{x}$ values for that specific day: $p'^{t,d}_{u,z} = \frac{x^{t,d}_{u,z}}{\sum_z x^{t,d}_{u,z}}$. Ideally, the learned parameters $\boldsymbol{p}$ and parameters $\boldsymbol{p'}$ for each test day should be as close as possible.

Figure 5(a) and (b) show these comparisons for attraction 8 for time interval 4PM-5PM and attraction 9 for time interval 12PM-1PM respectively. The x-axis denotes the attractions to which visitors can transition to. For example, for figure 5(a), $x = 3$ implies the parameter $p'^{t=7,\cdot}_{u=8,z=3}$, where the holder '$\cdot$' is for day number. Intuitively, this parameter represents the probability that a visitor currently at attraction 8 moves to attraction 3 during the time interval 7 (4PM-5PM). For each cluster on the x-axis, we show 7 bars. The first bar corresponds to the 'Learned' parameter from table II. Other bars show the computed parameter p' for different test days, 6 in total.

We make the following observations from figures 5(a) and (b). First, both the learned parameters $\boldsymbol{p}$ and the computed parameters $\boldsymbol{p'}$ are very close to each other. This is true for other busy attractions as well as rest of the time intervals. This further validates our approach. In addition, for figures 5(a), we see clearly that transition to the leisure node (x=10) dominates all the other transitions. This is as expected as during the evening hours, visitors exit the theme park. We also see a clear domination of a few attractions to which the visitors move from both the attractions 8 and 9. For example, figure 5(b) shows that visitors prefer to move to attractions 3, 5 and 8. Thus, our approach is able to extract meaningful visitor dynamics from the observed aggregate data.

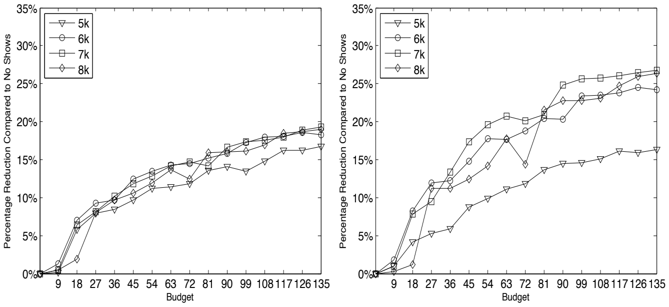


Fig. 6: Average and Peak Wait Time Reduction for CDON

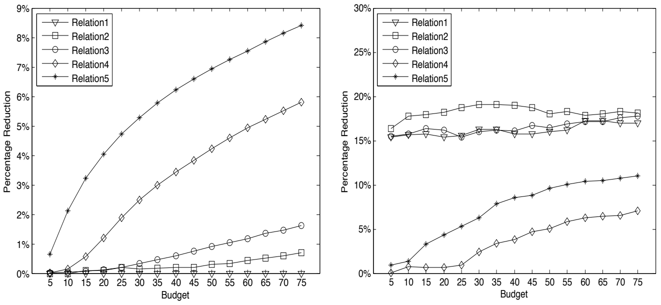
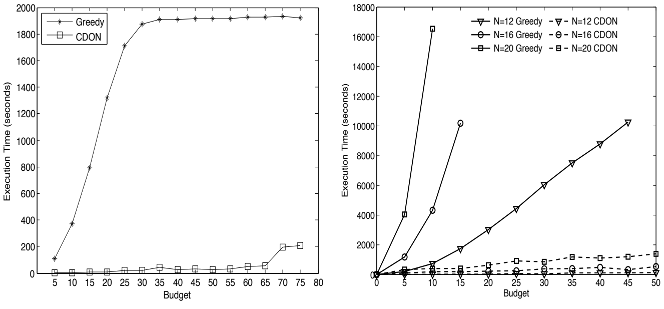


Fig. 7: Average (Left figure) and Peak (Right figure) Wait Time Reduction due to CDON in Comparison with greedy on Real Data

VI. Experiments: Controlling Diffusion Dynamics

We now provide an empirical evaluation to demonstrate the utility of our approach presented in Table III to control diffusion compared to having no side shows and the greedy baseline described below. We first consider the real data set of theme park along with the diffusion dynamics obtained using our learning mechanism (described in previous section) and then consider synthetic problems to demonstrate scalability of our approach in comparison to the greedy baseline.

Greedy We employ a greedy algorithm as a baseline in our experiment. It starts with an empty action set $\mathcal{S}$ and iteratively adds the *best* management actions until cost of management actions exceeds the budget. In each round, the objective is evaluated by simulating the cascade when adding every possible management action of type k to $\mathcal{S}$. The action with the largest reduction in wait time is added to the set $\mathcal{S}$.



(a) Real Dataset

(b) Synthetic Dataset

Fig. 8: Runtime Comparison of CDON with greedy on Real and Synthetic Datasets

A. Real Dataset

Since side shows are placed on the edges, the complexity of controlling diffusion is dependent on the number of edges, which is 729 in the time indexed graph for the real data set.

Single Type of Side Show Initially, we show the utility of using side shows by considering only one type of side shows and $\beta = 1.0$ for that side show type. Since there is only one type, we consider the budget to be the maximum number of available side shows that can be employed across different time steps. Figure 6 shows the average[5] and peak wait time[6] reduction when compared to wait times without any side shows for total population ranging from 5k-8k. As expected, both average and peak wait time reduction increased with the budget, irrespective of the population size and this reduction in wait time is as much as 20% for the average and up to 25% for the peak wait time.

Multiple Type of Side Shows We now consider multiple types of side shows, where the cost, c^k and diffusion absorption parameter, β^k for the type k are connected to each other according to one of the five relations below: (1) $c^k = 10\sqrt{\beta^k}$; (2) $c^k = 10\beta^k$; (3) $c^k = 10(\beta^k)^2$; (4) $c^k = 10(\beta^k)^3$; and (5) $c^k = 10(\beta^k)^4$. Their relations are based on the fact that a side show with low attractiveness (or low β) should be cheaper to deploy than a popular one. We consider $k = 4$ different types of shows are available, with β^k values given by $\{0.2, 0.5, 0.75, 1.0\}$ and a fixed initial population of 7000.

In figure 7, we compare the wait time reduction due to CDON in comparison with greedy approach on real dataset or $(W_{greedy} - W_{cdon}) * 100/W_{greedy}$, W_{alg} denotes the objective for the corresponding approach. A higher value of this percentage reduction denotes better performance by CDON over greedy. As the budget is increased, CDON performed consistently better than greedy on all five relations. We observe that as power of β is increased in the relation, the average wait time reduction is increased. Because CDON coordinates the placement of shows at different edges, if we have more available shows to place, the difference with greedy increases. With higher powers for beta, cost values for side shows are smaller resulting in more available side shows. This explains the reason for up to 9% wait time reduction provided by CDON in comparison with greedy. We similarly compare the wait time reduction for the peak attraction due to CDON in comparison with greedy approach in Figure 7(b). We observe that for all relations, the wait time reduction percentage is positive and in the best case it goes up to 20%.

We also record the runtime for the 5 different relationships for both CDON and greedy approach, the results are shown in figure 8. Due to space constraints, we only provide the results for relation 4. In each relation, greedy takes significantly more time to run than our approach, with the difference increasing

[5] Average wait time is the objective of CDON

[6] Peak wait time is the current wait time with CDON strategy for the attraction that had the highest wait time previously.

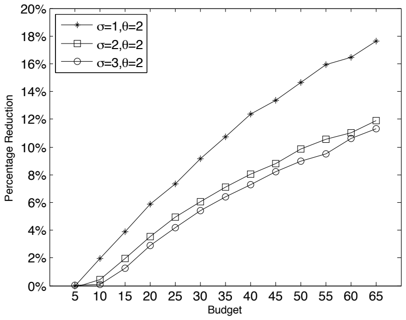


Fig. 9: Synthetic Dataset: Reduction in Wait Time with CDON in Comparison to the greedy Approach

as power of β is increased. This is because greedy strategy needs to evaluate many options as more number of side shows can be placed within the same budget.

B. Synthetic Data Set

To further demonstrate the performance improvement provided by CDON in comparison with greedy, we generate synthetic problems. Our goal is to identify scalability limits of CDON and greedy. The diffusion dynamics are skewed and are generated by using a gamma distribution (with different values of shape parameter σ and scale parameter θ). The sum of diffusion probabilities out of an attraction are normalised to 1. We show results for relation 4.

Figure 8(b) provides the runtime results as the number of nodes (N=12,16,20) and budget are increased. We have runtime on y-axis and budget on x-axis. There is an order of magnitude reduction in runtime provided by CDON in comparison with greedy. For most cases, greedy does not compute a solution within our threshold of 10000 seconds. This is because greedy has to evaluate placement of a side show of each type on each of the edges in the time indexed graph, and the number of edges has been increased from 729 (real problem) to 1296 (for N=12), 2304 (for N=16) and 3600 (for N=20). Thus, our CDON approach is highly scalable w.r.t. the number of attractions or network nodes as opposed to greedy. Figure 9 provides the percentage reduction in average wait time as the budget is increased for the same relation with different values of the gamma distribution's parameters. We again observe that gain by CDON increases up to 15% as the budget is increased.

VII. Conclusion

Managing diffusion in networks is an important and challenging problem with applications in ecology, leisure and entertainment, and marketing among others. Existing work has primarily focused on phenomena that diffuse independently on all outgoing edges of a node. We augmented the basic independent cascade model with important features required to model real-world problems, such as learning from aggregate data, modeling queues at network nodes and addressing flow conservation at network nodes. We also developed an optimization based approach that provided a plan for management actions to control the underlying diffusion process in a theme park for reducing the average wait time. We also demonstrated the efficiency and effectiveness of our learning and planning approaches through extensive evaluations on both real world and synthetic problems.

References

[1] D. Kempe, J. Kleinberg, and E. Tardos, "Maximizing the spread of influence through a social network," in *ACM SIGKDD*, 2003, pp. 137–146.

[2] A. Kumar, D. Sheldon, and B. Srivastava, "Collective diffusion over networks: Models and inference," in *UAI*, 2013, pp. 351–360.

[3] J. Leskovec, L. A. Adamic, and B. A. Huberman, "The dynamics of viral marketing," *ACM Trans. Web*, vol. 1, no. 1, May 2007.

[4] A. Krause, J. Leskovec, C. Guestrin, J. VanBriesen, and C. Faloutsos, "Efficient sensor placement optimization for securing large water distribution networks," *Journal of Water Resources Planning and Management*, vol. 134, no. 6, pp. 516–526, 2008.

[5] M. Gomez-Rodriguez, J. Leskovec, and A. Krause, "Inferring networks of diffusion and influence," *ACM TKDD*, vol. 5, no. 4, pp. 1–21, 2012.

[6] D. Sheldon, M. A. S. Elmohamed, and D. Kozen, "Collective inference on markov models for modeling bird migration," in *NIPS*, 2007.

[7] C. Daganzo, "The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory," *Transportation Research Part B: Methodological*, vol. 28, no. 4, pp. 269–287, 1994.

[8] J. Tsai, E. Bowring, S. Marsella, and M. Tambe, "Empirical evaluation of computational fear contagion models in crowd dispersions," *Autonomous Agents and Multi-Agent Systems*, vol. 27, no. 2, pp. 200–217, 2013.

[9] S. A. Myers and J. Leskovec, "On the convexity of latent social network inference," in *NIPS*, 2010, pp. 1741–1749.

[10] P. Netrapalli and S. Sanghavi, "Learning the graph of epidemic cascades," in *SIGMETRICS*, 2012, pp. 211–222.

[11] L. Wang, S. Ermon, and J. E. Hopcroft, "Feature-enhanced probabilistic models for diffusion network inference," in *ECML PKDD*, 2012, pp. 499–514.

[12] H. Daneshmand, M. Gomez-Rodriguez, L. Song, and B. Schölkopf, "Estimating diffusion network structures: Recovery conditions, sample complexity & soft-thresholding algorithm," in *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, 2014, pp. 793–801.

[13] Q. Qu, S. Liu, C. S. Jensen, F. Zhu, and C. Faloutsos, "Interestingness-driven diffusion process summarization in dynamic networks," in *Machine Learning and Knowledge Discovery in Databases*. Springer, 2014, pp. 597–613.

[14] X. Wu, A. Kumar, D. Sheldon, and S. Zilberstein, "Parameter learning for latent network diffusion," in *IJCAI*, 2013, pp. 2923–2930.

[15] D. Sheldon, B. N. Dilkina, A. N. Elmachtoub, R. Finseth, A. Sabharwal, J. Conrad, C. P. Gomes, D. B. Shmoys, W. Allen, O. Amundsen, and W. Vaughan, "Maximizing the spread of cascades using network design," in *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*, 2010, pp. 517–526.

[16] S. Liu, Y. Yue, and R. Krishnan, "Adaptive collective routing using gaussian process dynamic congestion models," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013, pp. 704–712.

[17] D. Sheldon, T. Sun, A. Kumar, and T. Dietterich, "Approximate inference in collective graphical models," in *ICML*, May 2013, pp. 1004–1012.