DLOREAN: Dynamic LOcation-aware REconstruction of multiwAy Networks

Fredrik Johansson
Chalmers University
Göteborg, Sweden
frejohk@chalmers.se

Vinay Jethava
Chalmers University
Göteborg, Sweden
jethava@chalmers.se

Devdatt Dubhashi
Chalmers University
Göteborg, Sweden
dubhashi@chalmers.se

***Abstract*—This paper presents a method for learning time-varying higher-order interactions based on node observations, with application to short-term traffic forecasting based on traffic flow sensor measurements. We incorporate domain knowledge into the design of a new damped periodic kernel which leverages traffic flow patterns towards better structure learning. We introduce location-based regularization for learning models with desirable geographical properties (short-range or long-range interactions). We show using experiments on synthetic and real data, that our approach performs better than static methods for reconstruction of multiway interactions, as well as time-varying methods which recover only pair-wise interactions. Further, we show on real traffic data that our model is useful for short-term traffic forecasting, improving over state-of-the-art.**

Keywords*—*Traffic prediction, structure learning, higher-order, spatio-temporal, kernel-reweighting, hierarchical inclusion

I. Introduction

Understanding road traffic flow has been of fundamental interest to urban planners for several decades [1]. Modern urban traffic control systems (UTCS) collect large amounts of heterogenous data including flow rate, occupancy, camera feeds, etc. which is used in design of Intelligent Transport Systems (ITS). A key component of such systems is prediction of traffic flow over short-time intervals.

Several methods have been investigated for short-term traffic forecasting (STTF) e.g. time-series methods such as ARIMA [2] and seasonal ARIMA [3], state-space models [4], [5], nonparametric methods [6], neural networks [7], simulation models [8], Bayesian methods [9], [10], random forests [11], [12] etc. See [13], [7] for survey of earlier methods, and [6], [14], [15], [16] for more recent advances.

It is now recognized that a key aspect of above problem is finding the dependencies between traffic flow at different (nearby) locations [5], [17], [18]. This can be posed as an instance of *structure learning* within the framework of probabilistic graphical models [19], wherein the structure (dependencies between nodes) of the graphical model is learnt based on node observations. Structure learning is an extremely well-studied problem [19], which is computationally hard in general [20].

Further, modeling the dependencies between traffic at different nodes introduces two additional caveats: the dependencies between traffic at different nodes *vary over time*, while showing periodic trends [5]; and, the dependencies between several nodes might be coupled and consequently, *cannot be represented using pair-wise interactions alone* [21]. We explain the two problems in greater detail below.

The temporal variation in dependencies between traffic at different location is well-known. For example, the traffic flow patterns, and the resulting traffic jams, differ drastically between peak hours of $4-5$ pm, compared to other times. One possible interpretation is that the model structure itself changes over time, and therefore, the node observations cannot be assumed to be generated independently and identically distributed (i.i.d.) from some underlying static model. Several methods [22], [23], [24], [25], [26] have investigated learning of time-varying graphical models with pair-wise interactions based on node observations at different time points.

In several contexts [27], [28], pair-wise interactions alone are not representative of the complex interactions present in the system. Hypergraphs [29] present the natural extension of graphs for representing such multiway interactions. Within the graphical models framework, multiway interactions can be easily represented using factors involving several variables[1]. Tractable learning algorithms are obtained by making additional assumptions on the structure of the higher-order graphical model such as bounded tree-width restriction [30], [31], [32] and hierarchical constraints [21], [33]. A number of recent works [34], [32], [35], [21] have presented fast methods for tractable learning of higher-order graphical models, and have used resulting algorithms for inferring static dependencies between traffic at different locations based on sensor measurements. The resulting models with higher-order interactions show better likelihood compared to models based on pair-wise interactions alone.

All the methods discussed above [34], [32], [35], [21] yield a higher-order graphical model but ignore the temporal variation in model structure. In contrast, the time-varying methods [22], [23], [24], [25], [26] yield a time-varying graphical model having only pair-wise interactions.

The first contribution of this paper is to devise a procedure for learning time-varying higher-order graphical models, based on a kernel-reweighting method for aggregating influence of observations at nearby time points; followed by learning of sparse graphical model which respects hierarchical constraints at each time independently by extending the approach of

[1] In this paper, we use the terms *multiway interaction*, *higher-order interaction* and *hyperedge* interchangeably to mean a factor node connected to several variable nodes.

Schmidt and Murphy [21]. At a high level, the kernel-reweighting procedure [22], [36] multiplies the likelihood due to an observation at a nearby time instant with a higher factor compared to an observation at a distant time instant; thus reweighting based on interval between current time instant and observation time instant. It decouples the temporal dependence and structure learning steps, and therefore, can be used with other methods for learning higher-order graphical models [30], [32], [35]. We show using synthetically generated data and real traffic measurements that our hybrid approach does better than static higher-order structure learning [21] as well as time-varying structure learning with only pair-wise interactions [22].

We next focus on discovering repeated traffic flow patterns. For example, if we want to find the dependencies that occur on weekdays between $4-5$ pm given real traffic measurements, we should reweight the observations at $4-5$ pm on weekdays higher than observations at other times on weekdays, or observations on weekends. Our second contribution is designing a periodic damped kernel which captures daily variation in our approach, following similar ideas in signal processing [36]. We show using experiments on real traffic data, that our damped periodic kernel captures trends much better than simple kernels which just consider the interval between two time instants.

We then ask the following question with a focus towards interpretability of the learnt model: *What if we want to discover long-range interactions alone? Or alternatively, we are interested in learning graphical models where the interactions are between nearby nodes alone?* The traditional methods for traffic flow prediction *do not* handle this. We address this by introducing location-based regularization, i.e. the regularization term for a higher-order interaction depends on the geographical locations of the interacting nodes. For example, when one wishes to learn a short-ranged graphical model, we penalize the higher-order interactions involving nodes which are very far apart with a large regularization term. We show, using experiments on real traffic data, that our location-based regularization yields more interpretable models exhibiting desired geographical properties.

II. Related work

a) Short-Term Traffic Forecasting: Short-term traffic forecasting methods provide traffic flow forecasts typically for next $5-10$ minutes up to an hour into the future [9], [7]. Karlaftis and Stathopoulos [37] investigate spatial variation and different time resolutions (daily, monthly, yearly); and observe that traffic flow exhibits spatial variation but does not show much variation during most months of the year as well as during weekdays.

Traditional time-series methods such ARIMA and variants focus on univariate prediction of traffic flow, ignoring the spatio-temporal correlations in the measurements. Subsequently, a number of techniques (e.g. [38], [17], [18]) have been developed which take into account the spatial correlations between traffic flow at "nearby" nodes. However, most existing work considers either a fixed number of nodes (e.g. [17], [18]) or distance dependent measure (e.g. [38]) to find "nearby" nodes, ignoring that the significant variation which might occur e.g. traffic at an expressway, and city network [39].

To the best of the authors' knowledge, none of the existing methods have approached both of the following two problems: Does the set of "nearby" nodes used for making short-term traffic predictions at a given node change with time? Do there exist complex interactions (captured using higher-order correlation terms) which can improve predictive performance?

Solving the first problem requires insight into how the set of "nearby" nodes (neighbouring nodes used in making traffic prediction) varies over time. Intuitively, the set of neighbours should not change too much in short intervals e.g. 5:00 and 5:10 p.m. Further, one might observe periodic trends e.g. traffic flow at a stadium affecting nearby stations before and after games.

For the second problem, a naive approach to modelling higher-order correlations is intractable since number of unknown parameters far exceeds the observations. Instead, selecting few higher-order interactions which give best predictive improvement under reasonable model assumptions seems more suitable. The remainder of this paper addresses above problems using the framework of probabilistic graphical models.

b) Learning time-varying graphical models: Several methods have focussed on inferring time-varying interactions based on the assumption that interaction networks *change slowly* over time [40], [22], [23]. Song et. al. [22] infer sparse time-varying gene interaction network under above assumption by first using kernel-reweighting to aggregate influence of observations at nearby time points, and subsequently inferring sparse network at each time independently using ℓ_1 regularization. Ahmed and Xing [24] and Kolar et. al. [25] have studied alternative losses for modeling piece-wise constant networks with sharp structural changes between different segments, as well as networks with smooothly (linearly) changing model parameters. Fu et. al. [23] explore dynamic version of Mixed Membership Stochastic Blockmodels (MMSB) [41] wherein the interaction dynamics is governed by a state-space model. Other methods have focussed on dynamic network inference under tighter assumptions on network structure based on domain knowledge [42], [26].

c) Learning higher-order graphical models: Chechetka and Guestrin [32] and Shahaf et. al. [35] present fast inference techniques for learning bounded tree-width graphical models [30], which they use to learn the static dependencies between traffic at different locations based on sensor measurements (of traffic at each location) at different time points.

Schmidt and Murphy [21] provide a tractable procedure for learning model structure under the assumtption of hierarchical inclusion constraints [43], [44], which specifies that a higher-order interaction of order k i.e. involving k nodes, is active only if all interactions involving subsets of the k nodes are also active. They use overlapping group ℓ_1 regularization [43] to enforce the hierarchical constraint. The resulting optimization is solved using spectral projected gradient where Dykstra's algorithm [45] is used to compute the projection. While the method supports learning of higher-order models, it has no inherent support for modeling dynamic networks.

III. Preliminary

We are given observations $\mathbf{x}^t = [x_1^t, \ldots, x_n^t]^\top$ at time instant $t \in \mathcal{T}$ where $x_i^t \in \mathcal{X}$ denotes the measurement at node

i for time t, and $\mathcal{X}$ is a discrete set. In the context of traffic flow prediction, node i denotes a physical sensor placed on some highway, while x_i^t represents the traffic flow measured at the sensor at time t.

The problem of (static) structure learning is to find a hypergraph $G = (V, \mathcal{E})$ with nodes $V = \{1, \ldots, n\}$ and subsets of vertices (hyperedges) $\mathcal{E} \subseteq 2^V$; with associated set of potential functions $\Phi = \{\phi_e\}_{e \in \mathcal{E}}$ which maximizes the log-likelihood:

$$\hat{\Phi} = \arg\max_{\Phi} \sum_{t \in \mathcal{T}} \log P(\mathbf{x}^t | \Phi) \tag{1}$$

In this paper, $P(\mathbf{x}|\Phi)$ is the log-linear model [46], given by

$$P(\mathbf{x}|\Phi) = \exp\left(\sum_{e \in \mathcal{E}} \phi_e(\mathbf{x}_e) - A(\Phi) \right) \tag{2}$$

where $A(\Phi)$ denotes the normalization function at time t respectively; and the potential function $\phi_e(\mathbf{x}_e)$ depends only on the node observations $\mathbf{x}_e = \{x_i\}_{i \in e}$ of nodes present in the hyperedge.

Following the notation of Schmidt and Murphy [21], we use $\mathbf{w}_e$ to denote the parameters associated with potential function ϕ_e; and $\mathbf{w}$ to denote the full set of parameters. In general, when $e \subseteq V$ contains m nodes, then $\mathbf{w}_e$ will have length $|\mathcal{X}|^m$. For example, if $e = \{1, 2\}$ and x_1 and x_2 are binary, then a possible over-complete representation is $\phi_{1,2}(\mathbf{x}_{1,2}) = \sum_{i,j=0}^{1} \mathbb{I}(x_1 = i \wedge x_2 = j) w_{i,j}$.[2]

A. Convex structure learning

This subsection reviews the approach by Schmidt and Murphy [21] which learns hierarchical log-linear models [46], [47] obeying the following constraint:

Definition 1 (Hierarchical inclusion restriction)**.** *A log-linear model with model structure $G = (V, \mathcal{E})$ and associated parameters $\{\mathbf{w}_e : e \in \mathcal{E}\}$ satisfies the hierarchical inclusion restriction if the following is true:*

$$\textit{If } \mathbf{w}_a = 0 \textit{ and } a \subset b, \textit{ then } \mathbf{w}_b = 0 \quad \forall a, b \in \mathcal{E}$$

Schmidt and Murphy [21] learn higher-order networks while enforcing the hierarchical inclusion restriction by introducing additional mixed-norm regularization terms based on overlapping groups of nodes [44], [43], resulting in the following constrained optimization problem:

$$\min_{\mathbf{w}, \mathbf{g}} \; -\sum_{t \in \mathcal{T}} \log p(\mathbf{x}^t | \mathbf{w}) + \sum_e \lambda_e g_e \;\; \text{s.t.} \;\; g_e \geq \|\mathbf{w}_e^*\|_2 \tag{3}$$

where λ_e is a per-edge regularization factor and $\mathbf{w}_e^*$ is a vector obtained by concatenation of all parameters $\mathbf{w}_{e'}$ which satisfy $e' \subset e$. $\mathbf{g}$ is a vector of auxilliary variables g_e, introduced to bound the norm of weights $\mathbf{w}_e$ of each group e.

The constrained optimization in (3) is solved using a projected gradient method [48] where the projection is computed using Dykstra's algorithm [45]. Higher-order interactions are added to the model incrementally with the caveat that a higher-order interaction e is added only if all subsets of that hyperedge $e' \subset e$ are already part of the model (hierarchical constraint), and further, it satisfies the condition

$$\left\| \nabla_{w_e} \sum_{t \in \mathcal{T}} \log p(\mathbf{x}^t | \mathbf{w}) \right\|_2 > \lambda_e \tag{4}$$

For large graphs, pseudo-likelihood [49] is used instead of exact likelihood resulting in the following optimization:

$$\min_{\mathbf{w}} \; -\sum_{t \in \mathcal{T}} \log p(x_i^t | \mathbf{x}_{-i}^t, \mathbf{w}) + \sum_e \lambda_e \Big(\sum_{e' \subseteq e} \|\mathbf{w}_{e'}\|_2^2 \Big)^{1/2} \tag{5}$$

where $\mathbf{x}_{-i}^t := [x_1^t, \ldots, x_{i-1}^t, x_{i+1}^t, \ldots, x_n^t]^\top$ denotes the states of remaining nodes (except node i) at time t. This improves speed of learning for larger models since the higher-order moments do not need to be computed.

B. Kernel-reweighting

This section reviews the kernel-reweighting method introduced in Song et. al. [22], following similar ideas in time-series analysis [36]. The kernel-reweighting method allows learning different pair-wise interaction networks at each time t, by relaxing the i.i.d. observation assumption; yet taking into account the slowly changing nature of the time-varying network. More concretely, it infers a sparse network $\mathbf{w}^t$ at each time point $t \in \mathcal{T}$, using the observations (in terms of log likelihood) available at nearby time instant $s \in \mathcal{T}$, whose influence on the network inference is reweighted depending on the time gap between two time points $(t-s)$, using a symmetric kernel function $K_{h_w}(\cdot)$. This yields the optimization problem at each time t given by:

$$\min_{\mathbf{w}^t} \; -\sum_{s \in \mathcal{T}} \rho^t(s) \log p(\mathbf{x}^s | \mathbf{w}^t) + \lambda \|\mathbf{w}^t\|_1 \tag{6}$$

where $\rho^t(s)$ denotes the weight for observation at time s when inferring network $\mathbf{w}^t$ at time t, given by $\rho^t(s) = \frac{K_{h_w}(t-s)}{\sum_s K_{h_w}(t-s)}$ and λ is the parameter associated with ℓ_1-regularization term ($\|\mathbf{w}^t\|_1$), which governs the sparsity of the obtained network.

The kernel function $K_{h_w}(t)$ controls the degree of influence of nearby time instant s when inferring network at time t with increasing time gap $(t - s)$, where h_w is the kernel bandwith. In this paper, we investigate two kernel functions

- Gaussian kernel $K_{h_w}(t) = \exp(-t^2/h_w)$. This was originally used by Song et. al. [22] for inferring time-varying gene interaction network (see Figure 1 (a)).
- Laplace kernel $K_{h_w}(t) = \exp(-|t|/h_w)$. We find this kernel performs better in experiments reported here.

The model of Song et. al. [22] is used for comparison in Section VI and is denoted KELLER.

IV. Learning time-varying multiway interactions

This section presents our hybrid approach to learning time-varying multiway interactions satisfying hierarchical inclusion restriction at each time instant, using a natural extension of kernel-reweighting method to higher-order structure learning approach of Schmidt and Murphy [21].

[2] The function $\mathbb{I}(\varphi)$ equals 1 if φ holds true and 0 otherwise

We consider the problem of inferring the multiway interaction network $\mathbf{w}^t$ at time t based on the following optimization:

$$\min_{\mathbf{w}^t} \; -\sum_{s\in\mathcal{T}} \rho^t(s) \log p(\mathbf{x}^s|\mathbf{w}^t) + \sum_e \lambda_e \|w_e^{t^*}\|_2 \qquad (7)$$

where $\rho^t(s)$ denotes the reweighted influence of an observation at time s while learning the multiway interaction network at time t as in Equation (6). In this paper we focus on improving the network reconstruction by incorporating domain knowledge into the design of an appropriate kernel $K_{h_w}(\cdot)$ and choice of regularization parameter λ_e to obtain better-fitted and more interpretable networks.

The kernel-reweighting decouples the structure learning problem at each time instant t, which can then be solved using the existing approach of Schmidt and Murphy [21]. Notice that setting $\rho^t(s) = 1$ makes (7) equivalent to (3) for static multiway network reconstruction by treating measurements at all time instants as i.i.d. observations.

We show in Section VI using synthetic dataset as well as real traffic measurement that the hybrid approach performs better than static multiway reconstruction as well as learning time-varying pair-wise networks.

A. Damped periodic kernel

The section presents our damped periodic kernel for leveraging the periodic (daily) patterns in road traffic towards learning better networks. Suppose we are interested in learning a multiway interaction network at 5 p.m. on Friday (Feb. 22, 2013). The kernels described in Section III-B will assign high weight to nearby time instants, say 4.50 p.m. and 5.10 p.m. on the same day, which agrees with the intuition that the networks at those times should be very similar to the underlying network at 5.00 p.m. on the same day.

On the other hand, the kernels also assign higher weight to observations to e.g. 11.00 p.m. on the same day, compared to weight assigned to observations at 5.00 p.m. on Thursday, or those at 5.00 p.m. last Friday. However, it is well-known that traffic show periodic patterns e.g. in many countries, 5.00 p.m. is peak traffic time in general and more so on Fridays. Therefore, it makes sense to design a kernel which accounts for these trends.

At the same time, the traffic pattern on Friday at 5.00 p.m. a few months back might be very different due to any of several reasons e.g. road blocks, construction, special events, etc.; and consequently, those observations should be considered with a grain of salt (assigned less weight).

We account for these patterns by using a damped periodic kernel $\tilde{K}^T_{h_w,\beta}(t)$ computed by convolving the basic kernel $K_{h_w}(\cdot)$ with a damped periodic delta train with period ΔT and damping function $l_\beta(\cdot)$, where β controls the degree of damping; following similar ideas in signal processing [36]. Mathematically, one can write

$$\tilde{K}^T_{h_w,\beta}(t) = K_{h_w}(t) \circledast \left(\sum_i l_\beta(t - iT)\delta(t - iT) \right) \qquad (8)$$

where $\delta(\cdot)$ denotes Dirac's delta function, and $\circledast$ denotes the convolution operator. Figure 1 presents an example of damped periodic kernel function, obtained using damping function $l_\beta(x) = \exp(-|x|/\beta)$.

In the remainder of this paper, we focus on daily trends by choosing $T = 24$ hours, and using damping function $l_\beta(x) = \exp(-|x|/\beta)$ with appropriately chosen β found using grid search. It is simple to design a kernel focussing on weekdays, weekends or monthly trends [37] by simply choosing $l_\beta(\cdot)$ appropriately.

Experimental results on traffic data (Section VI-B) show that the damped periodic kernel performs better (in terms of log-likelihood), compared to hybrid model compared in previous section.

V. Structured localization

This section presents our spatial (location-based) regularization for learning more interpretable networks. Specifically, one might want to learn networks with most interactions occurring between nearby sensors (which we call a short-range model), or conversely, models with long-range interactions. While hard restrictions has been made previously, e.g. using a simple distance threshold [18], to the best of our knowledge, this problem has not been approach by means of regularization. We formalize this notion below.

Let $\vec{p}_i = [p_{i_x} \; p_{i_y}]^\top$ denote the geographical location of sensor i respectively. Then, for any interaction $e \in \mathcal{E}$, we use the maximum Euclidean distance between any two nodes u and v which are part of the interaction $(u, v \in e)$ as a measure of the range of the interaction i.e.

$$d_e = \max_{u,v\in e} \; \|\vec{p}_u - \vec{p}_v\|_2 \qquad (9)$$

Under such a definition, we are interested in finding a short-range model i.e. a network for which d_e is small for most edges.

We enforce this by penalizing long-range interactions as part of the regularization by using a distance-dependent regularization function for each edge $e \in \mathcal{E}$ as:

$$\min_{\mathbf{w}^t} \; -\sum_{s\in\mathcal{T}} \rho^t(s) \log p(\mathbf{x}^s|\mathbf{w}^t) + \sum_e \lambda_e \|\mathbf{w}_e^{t^*}\|_2 \qquad (10)$$

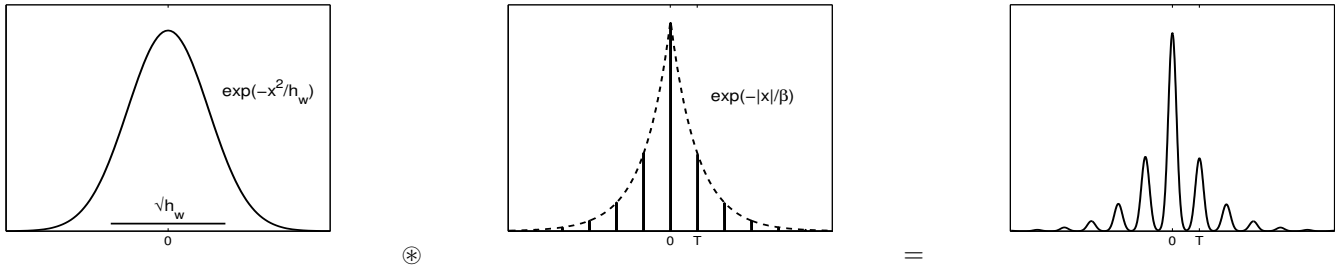
where λ_e is the regularization parameter for interaction e, which in turn depends on the range d_e of the interaction[3]. In this paper, we use regularization function of the form

$$\lambda_e = \eta \times (d_e)^\gamma \qquad (11)$$

where d_e represents the range of the interaction e, while η and γ are parameters governing the geographical properties of the learnt model. For example, choosing γ greater than zero favors short-range models, while negative values of γ lead to models with high number of long-range interactions.

Experimental results on traffic data (Section VI-B) show that location-based regularization can recover more interpretable models with desired geographical properties (long-range or short-range) with a trade-off in terms of model performance.

[3]Schmidt and Murphy [21] use the same λ for all interactions



(a) $K_{h_w}(t)$ $\circledast$ (b) $\sum_i l_\beta(t-iT)\delta(t-iT)$ $=$ (c) $\tilde{K}^T_{h_w,\beta}(t)$

Fig. 1: Example showing (a) Basic kernel, (b) Damped periodic operator, and (c) Damped periodic kernel.

VI. Results

In this section we present the results of evaluating our method on synthetic data and traffic data. For baseline comparison, we use our base model (HYBRID) as well as (a) static network construction using all available data (at different time points) as i.i.d. observations (STATIC) and (b) pair-wise network reconstruction using kernel-reweighting (KELLER).

We adapt code provided by Mark Schmidt[4] to implement STATIC, HYBRID and variations described in Section IV-A.

A. Synthetic experiment

We first evaluate our method on a synthetically generated data set (node observations), where the underlying model structure is changing slowly over time (Section VI-A).

1) Data set: We consider a hypergraph with $n = 10$ nodes and hyperedge probability $p = [0.15, 0.10, 0.05]$, where $p(i)$ denotes the probability of random hyperedge of size $(i+1)$. The potential functions are chosen with weights in $[1, 2]$ uniformly at random (analogous to procedure in [22]). We generate two hypergraphs $H^A = (V, \mathcal{E}^A)$ and $H^B = (V, \mathcal{E}^B)$ from the above model. If there are common edges i.e. $e \in \mathcal{E}^A \cap \mathcal{E}^B$, we remove them from the hypergraphs.

We generate a sequence of $nT = 50$ hypergraphs where the model structure changes slowly from H^A to H^B. Specifically, each hyperedge $e \in \mathcal{E}^A$ in hypergraph H^A switches off (i.e. is removed from the model structure) at random instant t_e chosen uniformly at random over the interval $[0, T]$ independent of other edges. The hyperedge $e \in \mathcal{E}^A$ with associated time t_e is present in all models $\mathcal{E}^t$ with $0 \le t \le t_e$. Conversely, each hyperedge $e \in \mathcal{E}^B$ in hypergraph H^B switches *on* at time instant t_e chosen uniformly at random over the interval $[0, T]$, and is present in all hypergraphs $\mathcal{E}^t$ having $t_e \le t \le T$.

The hypergraph $H^t = (V, \mathcal{E}^t)$ has the conditional distribution $P(e \in \mathcal{E}^t | \mathcal{E}^{t-1})$ depending on the previous hypergraph H^{t-1}, with $C = 1/(T - t + 1)$, given by

$$P(e \in \mathcal{E}^t \,|\, \mathcal{E}^{t-1}) = \begin{cases} 1 - C & \text{if } e \in \mathcal{E}^A \cap \mathcal{E}^{t-1} \\ 1 & \text{if } e \in \mathcal{E}^B \cap \mathcal{E}^{t-1} \\ C & \text{if } e \in \mathcal{E}^B \cap \bar{\mathcal{E}}^{t-1} \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

At each time t, we generate $m = 200$ i.i.d. node observations $\mathbf{x}^{t,1}, \ldots, \mathbf{x}^{t,m}$ from hypergraph H^t, out of which 50% are used as training data while the remainder are used as test samples for computing the log-likelihood of the trained model. This is the high-dimensional region where the number of unknown variables $p \sim O(n^4)$ is much larger than number of samples $m \sim O(n^2)$.

[4] http://www.di.ens.fr/~mschmidt/Software/

TABLE I: Edge recovery (F1-score) for edges of cardinalities 2, 3 and 4 averaged over different time instants $t \in \mathcal{T}$, and negative log-likelihood (NLL) values for network reconstruction.

	F1 (2)	F1 (3)	F1(4)	NLL
STATIC	0.2721	0.0800	0.2080	478.4193
KELLER	0.3144	–	–	465.5713
HYBRID	**0.3153**	**0.1386**	0.0512	**462.5704**

2) Results: We use the Laplace kernel $K_{h_w}(t) = \exp(-t/h_w)$ with kernel bandwidth $h_w = 10$ and compare our method using kernel-reweighted estimation of time-varying network with static network inferred using [21]. We choose (default) regularization parameter $\lambda_{\mathcal{E}} = 10$.

Table I (right) shows a comparison of average negative log-likelihood (NLL) at each time $t \in \mathcal{T}$ for the methods averaged over 10 trials. Both KELLER and HYBRID which take the temporal variation of the model into account perform significantly better (lower NLL) than STATIC method which ignores the temporal variation in the underlying model. Further, HYBRID outperforms KELLER slightly by taking into account the higher-order edges.

For each method (STATIC, KELLER, HYBRID) and at time t, we partition the recovered edges as $\hat{\mathcal{E}}^t = \hat{\mathcal{E}}^t_2 \cup \hat{\mathcal{E}}^t_2 \cup \hat{\mathcal{E}}^t_4$ where $\hat{\mathcal{E}}^t_i$ denotes the set of hyperedges of size i, i.e. $\hat{\mathcal{E}}^t_i := \{e \in \hat{E}^t : |e| = i\}$. We compute the recovery of edges w.r.t. original hyperedges $\mathcal{E}^t = \mathcal{E}^t_2 \cup \mathcal{E}^t_3 \cup \mathcal{E}^t_4$ (where $\mathcal{E}_i := \{e \in \mathcal{E} : |e| = i\}$) using F1-score $f_i = \frac{2p_i r_i}{p_i + r_i}$, where p_i, r_i and f_i denote the precision, recall and F1-score in recovery of hyperedges of size i respectively.

Table I (left) shows F1-score for recovered edges compared to true hyperedges $\mathcal{E}^t$ at different sizes ($|e| = 2, 3, 4$). The hybrid method (HYBRID) recovers higher-order interactions while KELLER (by definition) only returns pair-wise interactions.

TABLE II: Negative pseudo log-likelihood (NLL), and aggregate geographical distances within inferred interactions using different methods on the PEMS-SF dataset.

	NLL	d_{min}	d_{avg}	d_{max}
STATIC	73.15	0.0099	0.24	0.44
KELLER	71.67	0.0002	0.22	0.44
HYBRID	70.63	0.0002	0.27	0.44
DP-HYBRID	**69.65**	0.0002	0.28	0.44
DP-HYBRID/LR	70.06	**0.1147**	**0.30**	0.44
DP-HYBRID/SR	71.62	0.0002	**0.07**	**0.17**

B. Learning traffic flow networks

We evaluate our method on the dataset of measurements of traffic occupancy rates collected as part of the PeMS[5] project. We investigate the behaviour of the following models, DP-HYBRID, our hybrid approach with damped periodic kernel, DP-HYBRID/LR, hybrid approach with damped periodic kernel and location-based regularization for finding models having long-range interactions and DP-HYBRID/SR, same as previous but for short-range interactions.

1) Dataset: The PeMS dataset was compiled for UCI Machine Learning Repository[6] and consists of measurements of occupancy rates in different car lanes in the San Fransisco bay area. In the experiments, we have selected a subset of the data comprising 14 days of measurement from 32 different measurement stations, the same number commonly used in previous work [32], [21]. We discretized the data into 2 bins, enabling comparison with KELLER. The dataset was then split in two halves, the first 7 days used for training and the second for testing. We denote this dataset PEMS-SF. We trained and evaluated our model on 12 noon on each day, using hourly measurements.

2) Kernels & parameter selection: For the hybrid approaches, as well as KELLER and STATIC, we have chosen parameters using grid-search, to maximize the pseudo likelihood of the trained models with respect to the test set. The bandwidth parameter h_w was selected from the range $[50, 200]$. For β, we chose values from $[1000, 5000]$ and for λ from $[1, 20]$ based on grid search. The period T of the periodic kernels was set equal to the number of samples per day.

For the location-based regularization we used the best parameters found for DP-HYBRID, while selecting γ through grid-search in the range $[1, 8]$ for short-range interactions, and $[-8, -1]$ for long-range interactions. The factor η was chosen as 10^{γ}.

For all of the experiments, we used a Laplace base kernel, $K_{h_w}(t) = \exp(-|t|/h_w)$. This was found to have the best performance on synthetic data as well as in early experiments on PEMS-SF.

3) Results: The results of our experiments, in terms of negative pseudo log-likelihood on the test set and distances of inferred interactions, are presented in Table II. We used the following parameters to obtain these results: for all methods, $\lambda = 1$, for KELLER and HYBRID, $h_w = 100$, and for DP-HYBRID, DP-HYBRID/LR and DP-HYBRID/SR we used $h_w = 80$ and $\beta = 2500$. For DP-HYBRID/SR we used $\gamma = 8$ and for DP-HYBRID/LR, $\gamma = -8$.

We see that all hybrid methods outperform both benchmarks. Further, we note that HYBRID has better performance than KELLER, indicating that including higher-order interactions contribute to higher likelihood of the model. HYBRID also improves on STATIC, showing that accounting for temporal variations in the underlying networks allow for a better fit to the data. Moreover, adding periodicity to the kernel with DP-HYBRID improves the likelihood even more, indicating that a domain-specific adaptation (accuounting for day-cycles of traffic) increases the likelyhood of the model. In conclusion, incorporating both temporal dynamics and higher-order factors, into traffic structure prediction, offers an advantage over existing methods.

We also report the minimum, average and maximum interaction distances, denoted d_{min}, d_{avg} and d_{max} respectiveley. We define $d_{min} = \min_{e\in\mathcal{E}} d_e$ and analogously, $d_{max} = \max_{e\in\mathcal{E}} d_e$ and $d_{avg} = \frac{1}{|\mathcal{E}|}\sum_{e\in\mathcal{E}} d_e$ where d_e is defined as in (9). The numbers are meant to serve as indicators of the performance of our spatial regularization.

In Table II, we note a significant difference in the aggregate distances when using location-based regularization. For instance, the average and minimum distances are increased when using DP-HYBRID/LR and conversely, the average and maximum distances are decreased when using DP-HYBRID/SR. *We note that* STATIC *has a comparatively high minimum distance, yet low average distance, making it less suitable for geo-targetted structure prediction.* Lastly, we note that the likelihood of the models using spatial regularization, while increasing slightly in NLL compared to $hybrid$, still performs better than STATIC and KELLER.

C. Short-term traffic forecasting

We evaluate the predictive capabilities of our model by comparing it to state-of-the-art method by Chandra and Al-Deek [17], [18] for short-term traffic forecasting. Chandra and Al-Deek [18] use a Vector Autoregressive model (VAR) for predicting flow based on past $p = 6$ flow measurements at nearby nodes. A VAR model of n variables has the form,

$$\mathbf{x}^t = \mathbf{v} + A^{(1)}\mathbf{x}^{t-1} + ... + A^{(p)}\mathbf{x}^{t-p} + \mathbf{e}^t \qquad (13)$$

where $\mathbf{x}^t$ is an $(n \times 1)$ observation vector at time t, $A^{(p)}$ is a $(n \times n)$ coefficient matrix for lag p, $\mathbf{v}$ is a constant term and $\mathbf{e}^t$ is a time-dependent noise term. Specifically, (13) defines a VAR model of lag p, i.e. using p previous samples for prediction, denoted VAR(p).

Estimating a VAR model amounts to fitting all coefficients $\mathbf{A} = [A^{(1)}, \ldots, A^{(p)}]$ using least-squares estimation. Often, a restricted subset of the variables is used for predicting a distinguished variable resulting in reduced complexity and higher interpretability. Chandra and Al-Deek (and others, see e.g. [38]) have used spatial restriction by selecting 5 nearest neighbours to model traffic at a distinguished node.

In this experiment, we compare the results using standard VAR model with spatial restriction to select the neighbours;

[5]http://pems.dot.ca.gov

[6]http://archive.ics.uci.edu/ml

with results using predictions based on nodes selected using our HYBRID approach which takes into account the temporal variation in the relevant neighbourhood (set of "nearby" nodes), as well as higher-order interactions.

Straight-forward extension of VAR to handle higher-order correlations is computationally intractable since a naive approach would require modelling parameters (A_{ijk}) for each order 3 correlation leading to $O(n^3)$ variables. Our prediction method incorporates each higher-order interaction obtained by HYBRID approach into the VAR model by regressing on an augmented feature vector with each term corresponding to product of neighbours (vertices other than the distinguished vertex) in an hyperedge, as described below.

Let $G = (\mathcal{V}, \mathcal{E})$ denote the hypergraph inferred using HYBRID model with maximum order k (i.e. maximum size of any hyperedge in G). Let $\mathcal{E}_i = \{e \in \mathcal{E} : i \in e\}$ denote the neighbourhood of vertex i. We perform regression by considering the neighbours $\mathcal{E}_i$ where for each hyperedge $e \in \mathcal{E}_i$, we consider the corresponding feature as $y_{i,e} = \prod_{u \in e \wedge u \neq i} x_u$, yielding an augmented vector $\mathbf{y}_i^t = [y_{i,1}^t, \ldots, y_{i,m_i}^t]^\top$, where $m_i = |\mathcal{E}_i|$ denotes number of hyperedges incident to vertex i.

In order to predict traffic flow x_i^t at vertex i at time t, we use linear regression on time-series $\mathbf{y}_i^t$ as

$$x_i^t = v_i + e_i^t + \sum_{l=1}^{p} {\beta_i^{(l)}}^\top \mathbf{y}_i^{(t-l)} \tag{14}$$

where e_i^t denotes the noise term; and, $v_i \in \mathbb{R}$ and $\beta_i^{(1)}, \ldots, \beta_i^{(p)} \in \mathbb{R}^{m_i}$ are the unknown parameters which are estimated using the data.

1) Results: Using the extended model in (14), we predicted the occupancy rates of $(n = 32)$ stations at 12 (noon) around Berkeley, selected from the PeMS dataset. We trained both the STATIC and HYBRID models with maximum order $(k = 3)$ to obtain the hypergraphs G^{static} and G^{hybrid}, the difference being that in the HYBRID case, our inferred hypergraph is based on damped periodic kernel (as described in Section VI-B) and therefore, captures the neighbourhood which is relevant at 12 (noon).

The original VAR was used for comparison without modification, except for leaving out constant and error terms as in Chandra and Al-Deek [18]. Two restrictions are made on the variables used for predictions. VAR/SR includes only stations within $L = 10$ miles of the one being predicted, while VAR/FULL includes all stations.

We evaluate the predictions using the *Average Relative Error* [50] (ARE) defined as,

$$ARE_t = \frac{\sum_{i=1}^{k} |x_i^t - \hat{x}_i^t| / x_i^t}{k} \tag{15}$$

For all experiments, a lag of $p = 10$ was used. The results of the prediction experiments can be seen in Table III. The best parameters values for STATIC and HYBRID were $(\lambda = 10)$ and $(\lambda = 15, h_w = 50)$ respectively. The number of parameters fitted in the VAR models are stated, as a measure of complexity.

We make a note that both of our extensions perform better than the original VAR model, while retaining comparable complexity. This indicates that higher-order interactions does improve the quality of the prediction. Further, we note that the HYBRID model performed better than STATIC, indicating that the kernel-reweighting scheme also contributes to the performance. Lastly, we point out that there is no straight-forward way of adding parameters to VAR/FULL since it already uses all observations in prediction. The threshold distance L of VAR/SR was set to 20 values in $[2, 1000] \cup \{\infty\}$ and gave a monotone decrease in ARE with increasing distance (until all stations were included). The results for VAR/SR in Table III ($L = 10$) are thus reported for comparison only.

TABLE III: Prediction results using the VAR and modified VAR models in terms of ARE (see text). #PARAM denotes the average number of parameters. #2-EDGE and #3-EDGE denote the average number of 2-edges and 3-edges respectively.

	ARE	#PARAM	#2-EDGE	#3-EDGE
VAR/SR	0.1910	956	9.4	0
VAR/FULL	0.1599	10240	32	0
STATIC	0.1400	16600	25.9	14.0
HYBRID	**0.1263**	11193	24.7	8.16

VII. DISCUSSION

We present a hybrid method for inferring multiway time-varying interaction networks based on an extension of convex structure learning method of Schmidt and Murphy [21] using kernel-reweighting [22], with application to traffic data. We develop a domain-specific kernel which captures periodic traffic patterns; and introduce location-based regularization for learning networks with desired geographical properties.

We show using experiments on synthetic and real traffic data that our method is capable of inferring networks that better fit the data than existing methods by capturing the dynamic and higher-order interactions. Further, we show that our method improves performance over state-of-the-art in short-term traffic forecasting, while maintaining similar model complexity.

Our approach can be significantly improved by leveraging heterogenous data sources e.g. camera feeds, GIS information, vehicle co-location, etc. which are part of modern Intelligent Transport Systems (ITS). Another interesting aspect is integration and validation with large-scale simulation models [51]. Lastly, extension to large-scale networks remains a practical challenge, where distributed optimization techniques [52] and parallelization frameworks [53] might prove extremely useful.

REFERENCES

[1] F. A. Haight, *Mathematical theories of traffic flow*, 1965.

[2] M. S. Ahmed and A. R. Cook, "Analysis of freeway traffic time-series data by using box-jenkins techniques," *Transportation Research Record*, no. 722, 1979.

[3] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results," *Journal of transportation engineering*, vol. 129, no. 6, pp. 664–672, 2003.

[4] J. Whittaker, S. Garside, and K. Lindveld, "Tracking and predicting a network traffic process," *International Journal of Forecasting*, vol. 13, no. 1, pp. 51–61, 1997.

[5] A. Stathopoulos and M. G. Karlaftis, "A multivariate state space approach for urban traffic flow modeling and prediction," *Transportation Research Part C: Emerging Technologies*, vol. 11, no. 2, pp. 121–135, 2003.

[6] B. L. Smith, B. M. Williams, and R. Keith Oswald, "Comparison of parametric and nonparametric models for traffic flow forecasting," *Transportation Research Part C: Emerging Technologies*, vol. 10, no. 4, pp. 303–321, 2002.

[7] B. L. Smith and M. J. Demetsky, "Traffic flow forecasting: comparison of modeling approaches," *Journal of transportation engineering*, vol. 123, no. 4, pp. 261–266, 1997.

[8] A. Kotsialos, M. Papageorgiou, C. Diakaki, Y. Pavlis, and F. Middelham, "Traffic flow modeling of large-scale motorway networks using the macroscopic modeling tool metanet," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 3, no. 4, pp. 282–292, 2002.

[9] S. Sun, C. Zhang, and G. Yu, "A bayesian network approach to traffic flow forecasting," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 7, no. 1, pp. 124–132, 2006.

[10] E. Castillo, J. M. Menéndez, and S. Sánchez-Cambronero, "Predicting traffic flow using bayesian networks," *Transportation Research Part B: Methodological*, vol. 42, no. 5, pp. 482–509, 2008.

[11] M. Wojnarski, P. Gora, M. Szczuka, N. S. Hung, J. Swietlicka, and D. Zeinalipour, "Ieee icdm 2010 contest: Tomtom traffic prediction for intelligent gps navigation," in *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*. IEEE, 2010, pp. 1372–1376.

[12] B. Hamner, "Predicting travel times with context-dependent random forests by modeling local and aggregate traffic flow," in *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*. IEEE, 2010, pp. 1357–1359.

[13] B. Van Arem, H. R. Kirby, M. J. M. Van Der Vlist, and J. C. Whittaker, "Recent advances in the field of short-term traffic forecasting," *International journal of forecasting*, vol. 13, pp. 1–12, 1997.

[14] E. I. Vlahogianni, J. C. Golias, and M. G. Karlaftis, "Short-term traffic forecasting: Overview of objectives and methods," *Transport Reviews*, vol. 24, no. 5, pp. 533–557, 2004.

[15] C. van Hinsbergen, J. van Lint, and F. Sanders, "Short term traffic prediction models," in *Proceedings of the 14th World Congress on Intelligent Transport Systems (ITS), Beijing, October 2007*, 2007.

[16] E. Bolshinsky and R. Freidman, "Traffic flow forecast survey," Tech. Rep. CS-2012-06, 2012.

[17] S. R. Chandra and H. Al-Deek, "Cross-correlation analysis and multivariate prediction of spatial time series of freeway traffic speeds," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2061, no. -1, pp. 64–76, 2008.

[18] ——, "Predictions of freeway traffic speeds and volumes using vector autoregressive models," *Journal of Intelligent Transportation Systems*, vol. 13, no. 2, pp. 53–72, 2009.

[19] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. The MIT Press, 2009.

[20] D. Chickering, D. Geiger, D. Heckerman *et al.*, "Learning bayesian networks is np-hard," Tech. Rep., 1994.

[21] M. Schmidt and K. Murphy, "Convex structure learning in log-linear models: Beyond pairwise potentials," in *Proc. of AISTATS*, 2010.

[22] L. Song, M. Kolar, and E. P. Xing, "KELLER: estimating time-varying interactions between genes," *Bioinformatics*, vol. 25, pp. i128–i136, 2009.

[23] W. Fu, L. Song, and E. P. Xing, "Dynamic mixed membership blockmodel for evolving networks," in *Proc. of ICML*, 2009, pp. 329–336.

[24] A. Ahmed and E. Xing, "Recovering time-varying networks of dependencies in social and biological studies," *PNAS*, vol. 106, no. 29, pp. 11 878–11 883, 2009.

[25] M. Kolar, L. Song, A. Ahmed, and E. P. Xing, "Estimating time-varying networks," *Annals of Applied Statistics*, vol. 4, no. 1, pp. 94–123, 2010.

[26] V. Jethava, C. Bhattacharyya, D. Dubhashi, and G. N. Vemuri, "Netgem: Network embedded temporal generative model for gene expression data," *BMC Bioinformatics*, vol. 12, no. 327, 2011.

[27] S. Klamt, U. Haus, and F. Theis, "Hypergraphs and cellular networks," *PLoS computational biology*, vol. 5, no. 5, p. e1000385, 2009.

[28] G. Ghoshal, V. Zlatic, G. Caldarelli, and M. E. J. Newman, "Random hypergraphs and their applications," *CoRR*, vol. abs/0903.0419, 2009.

[29] C. Berge, *Graphs and hypergraphs*. Elsevier, 1976.

[30] N. Srebro, "Maximum likelihood bounded tree-width markov networks," in *Proc. of UAI*, 2001, pp. 504–511.

[31] F. Bach, M. Jordan *et al.*, "Thin junction trees," in *Proc. of NIPS*, 2001, pp. 569–576.

[32] A. Chechetka and C. Guestrin, "Efficient principled learning of thin junction trees," in *Proc. of NIPS*, 2007.

[33] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach, "Proximal methods for sparse hierarchical dictionary learning," in *Proc. of ICML*, 2010.

[34] A. Krause and C. Guestrin, "Near-optimal nonmyopic value of information in graphical models," in *Conference on Uncertainty in Artificial Intelligence (UAI)*, July 2005.

[35] D. Shahaf, A. Chechetka, and C. Guestrin, "Learning thin junction trees via graph cuts," *Artificial Intelligence and Statistics (AISTATS)*, vol. 3, no. 3, p. 2, 2009.

[36] S. Nawab and T. Quatieri, "Short-time fourier transform," *Advanced topics in signal processing*, pp. 289–337, 1988.

[37] A. Stathopoulos and M. Karlaftis, "Temporal and spatial variations of real-time traffic data in urban areas," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1768, no. -1, pp. 135–140, 2001.

[38] Y. Kamarianakis, A. Kanas, and P. Prastacos, "Modeling traffic volatility dynamics in an urban network," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1923, no. 1, pp. 18–27, 2005.

[39] W. Min and L. Wynter, "Real-time road traffic prediction with spatio-temporal correlations," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 4, pp. 606–616, 2011.

[40] S. Zhou, J. Lafferty, and L. Wasserman, "Time varying undirected graphs," *Machine Learning*, vol. 80, no. 2, pp. 295–319, 2010.

[41] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing, "Mixed membership stochastic blockmodels," *JMLR*, vol. 9, pp. 1981–2014, 2008.

[42] F. Guo, S. Hanneke, W. Fu, and E. Xing, "Recovering temporally rewiring networks: A model-based approach," in *ICML*, 2007, pp. 321–328.

[43] P. Zhao, G. Rocha, and B. Yu, "The composite absolute penalties family for grouped and hierarchical variable selection," *The Annals of Statistics*, vol. 37, no. 6A, pp. 3468–3497, 2009.

[44] F. Bach, "Exploring large feature spaces with hierarchical multiple kernel learning," in *Proc. of ICML*, 2008.

[45] R. Dykstra, "An algorithm for restricted least squares regression," *Journal of the American Statistical Association*, vol. 78, no. 384, pp. 837–842, 1983.

[46] Y. Bishop, S. Fienberg, and P. Holland, "Discrete multivariate analysis," 1975.

[47] J. Whittaker, "Graphical models in applied multivariate analysis," *Chichester New York et al.: John Wiley & Sons*, 1990.

[48] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.

[49] J. Besag, "Statistical analysis of non-lattice data," *The statistician*, pp. 179–195, 1975.

[50] B. Zhang, K. Xing, X. Cheng, L. Huang, and R. Bie, "Traffic clustering and online traffic prediction in vehicle networks: A social influence perspective." in *INFOCOM*, A. G. Greenberg and K. Sohraby, Eds. IEEE, 2012, pp. 495–503.

[51] C. Barrett, R. Beckman, K. Berkbigler, K. Bisset, B. Bush, K. Campbell, S. Eubank, K. Henson, J. Hurford, D. Kubicek *et al.*, "Transims: Transportation analysis simulation system," *Los Alamos National Laboratory Unclassified Report*, 2001.

[52] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[53] Y. Low, J. Gonzalez, A. Kyrola, D. Bickson, C. Guestrin, and J. M. Hellerstein, "Graphlab: A new framework for parallel machine learning," *arXiv:1006.4990*, 2010.