FCCF: Forecasting Citywide Crowd Flows Based on Big Data

Minh X. Hoang[*]
University of California,
Santa Barbara
mhoang@cs.ucsb.edu

Yu Zheng[†]
Microsoft Research
Beijing, China
yuzheng@microsoft.com

Ambuj K. Singh
University of California,
Santa Barbara
ambuj@cs.ucsb.edu

ABSTRACT

Predicting the movement of crowds in a city is strategically important for traffic management, risk assessment, and public safety. In this paper, we propose the novel problem of predicting two types of flows of crowds in every region of a city based on big data, including human mobility data, weather conditions, and road network data. To have a practical solution for citywide traffic prediction, we first partition the map of a city into regions using both its road network and historical records of human mobility. Our problem is different from the predictions of each individual's movements and each road segment's traffic conditions, which are computationally costly and not necessary from the perspective of public safety at a citywide scale. To model the multiple complex factors affecting the crowd flows, we decompose the flows into three components: seasonal (periodic patterns), trend (changes in periodic patterns), and residual flows (instantaneous changes). The seasonal and trend models are built as intrinsic Gaussian Markov random fields, which can cope with noisy and missing data. Whereas, the residual model exploits the spatio-temporal dependence among different flows and regions, as well as the effect of weather. Experimental results on three real-world datasets show that our method is scalable and outperforms all baselines significantly in accuracy.

Categories and Subject Descriptors

H.2.8 [**Software Engineering**]: Database Applications—
data mining, spatial databases and GIS

Keywords

Urban computing, spatio-temporal data mining

[*]The research was done when the first author was an intern at Microsoft Research. Yu Zheng is the corresponding author of this paper.

[†]School of Computer Science and Technology, Xidian University, China; Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences.

1. INTRODUCTION

Predicting the movement of crowds in a city is strategically important for traffic management, risk assessment, and public safety. For example, 36 people died and 47 others were injured amid the bloody stampede in the Shanghai Bund in 2015, turning a New Year Celebration into a catastrophic accident. Massive flows of people streamed into a strip region, which could not really hold them, to watch the New Year's Eve Light Show, making the region overloaded and out of police control. Similar stampedes also happened in the 2010 German Love Parade. If we can predict the arrival of crowds in a region and know the crowd flows would exceed the region's safe capacity, we can launch emergency mechanisms (e.g., sending warnings to people and conducting traffic controls) or evacuating people in advance.

Prior research on crowd movements has focused on the predictions of each individual's movement (e.g., [23, 26]), and the traffic conditions on road segments (e.g., [16, 21]). While these problems provide a *detailed view* of a city traffic, they may have heavy computational costs due to the huge number of roads, vehicles, and people in a big city, and are also not necessary from the perspective of public safety at a citywide scale. Further, predicting each individual's movement is difficult to achieve given the diversity of individual life patterns and the randomness of human behavior.

Given the above limitations, in this paper, we investigate a *macro-level view* of crowd movements by predicting two types of *flows of crowds* in every region of a city based on big data, including human mobility data, weather conditions, and road network data. As shown in Figure 1a, a region (such as r_1) is bounded by major roads, and the two flows are: 1) *new-flow*, the traffic of crowds originating from a region at a given time interval (e.g., people start driving from a parking spot); and 2) *end-flow*, the traffic of crowds that is terminated in a region (e.g., people stop driving and park their cars). Intuitively, new-flow and end-flow track the origins and final destinations of the crowds. These two flows thus summarize the movements of crowds and are enough for traffic management and risk assessment.
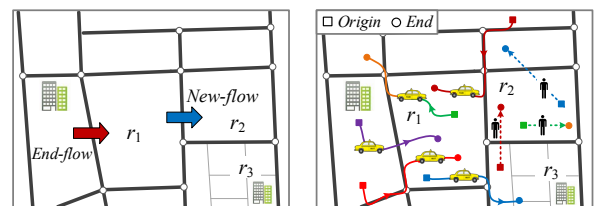
(a) Two types of flow to be predicted   (b) Illustration of measurement of flow

Figure 1: Crowd flows in a region

The two crowd flows can be measured *individually* by the number of vehicles driven on roads, or the number of people traveling in public transportation systems, or the number of pedestrians, or *all together* if data is available. The data representing human mobility can be the GPS trajectories of vehicles, or the mobile phone signals of users, or card swiping data in public transportation systems such as the subway or bike sharing systems. For example, in Figure 1b, according to the GPS trajectories and measured by the number of vehicles, the new-flow and end-flow of r_1 in the past 30 minutes are (2, 3) respectively. Likewise, the two types of flows are (2, 2) in r_2 in terms of mobile phone signals, measured by the number of pedestrians. If both the GPS and phone signals are tracked, we can consider the crowd flows of region r_3 to be (1, 1). During a time interval, if a person starts and ends her trajectory in the same region, she will be counted in both the new-flow and end-flow of that region. Note that our proposed framework can also be applied as is to other definitions of crowd flows.

The challenges of our research are three-fold. 1) *Multiple complex factors*: There are multiple complex factors affecting the crowd flows, which can be captured thanks to the advent of big data. For instance, the crowd flows in a region usually have a daily and weekly periodic pattern, which might change over time, as well as instantaneous changes due to noise, weather conditions, and other social events. 2) *Flow dependencies*: There are dependencies between different types of flows in a region (intra-region dependence) and those among different regions (inter-region dependence) over time. For example, the increase of end-flow in a region in the current hour may raise its new-flow over the next hour. Similarly, the end-flow of a region is influenced by the new-flows of its neighbors. 3) *City-scale prediction*: While we need the prediction instantly, a city-scale prediction is computationally intensive. Therefore, an efficient predictive model is needed. In addition, different regions could have different scales of crowd flows. Sparse flow data in some regions will prevent us from learning a stable periodic pattern inherent in crowd flows, thus reducing the prediction accuracy.

To tackle these challenges, we decompose each type of flow in a region into *three ingredients*: seasonal, trend, and residual flows, proposing a three-step predictive method to capture each of them. The contributions of our research:

- To deal with data sparsity and construct a practical citywide solution, we first divide a city into low-level regions using its road network, and then group adjacent low-level regions with similar crowd flow patterns using graph clustering. The obtained high-level regions have more stable (thus easier to predict) crowd flows, and also provide a meaningful and more manageable representation of the citywide crowd flows.

- Based on the Intrinsic Gaussian Markov Random Field (IGMRF), we propose a seasonal model to predict the periodic flow, and a trend model to predict the change of the seasonal pattern over time. Our IGMRF models are robust to noisy and missing data, and scalable to big data.

- We propose a spatio-temporal residual model to predict the instantaneous deviations from the periodic patterns of flows, based on the historical flow data of a region and those of its neighbors as well as weather information. The model uses a Bayesian network to capture the transition probability among the regions. We combine the seasonal

model, trend model, and residual model to obtain our **FCCF** model (**F**orecasting **C**itywide **C**rowd **F**lows).

- Experiments[1] on three real-world datasets (taxicab and bike data) show that our framework is scalable and outperforms baseline approaches significantly in accuracy.

The rest of this paper is as follows: Section 2 overviews our framework. Section 3 discusses the division of a city into regions. Section 4 proposes the seasonal and trend models. Section 5 proposes the spatio-temporal residual model. Section 6 reports our experimental results. Section 7 discusses related works and Section 8 concludes the paper.

2. OVERVIEW
2.1 Preliminaries

Regions: There are many definitions of a location in terms of different granularities and semantic meanings. In this study, we first partition a city into a number of low-level regions by city roads, using a map segmentation method [25]. Consequently, each region is bounded by the roads, carrying a semantic meaning of neighborhoods or communities, as illustrated in Figure 1. These regions are low-level, that is, they can be very small and have very little data for prediction. Therefore, we propose to group adjacent low-level regions with similar crowd flow patterns into high-level regions using a graph clustering approach. We will discuss the clustering step in Section 3. Denote the set of high-level regions as $\mathcal{R} = \{u_1, u_2, ..., u_m\}$, where m is the number of high-level regions. We then use the high-level regions as the minimal unit of location in the following study, though a region can be a uniform grid or defined by the governments in other applications.

Definition 1 (Crowd flows) *The movement of an individual can be recorded as a spatial trajectory $\mathcal{T}$, which is a sequence of time-ordered points, $\mathcal{T} : p_1 \to p_2 \to ... \to p_{|\mathcal{T}|}$, where each point $p_i = (a_i, b_i, t_i)$ has a geospatial coordinate position (a_i, b_i) and a timestamp t_i, and $|\mathcal{T}|$ is the number of points in $\mathcal{T}$. Likewise, the movement of crowds can be represented by a collection of trajectories $\boldsymbol{P}$. Specifically, for a region u, the two types of flows of crowd (crowd flows) at timestamp t, namely new-flow and end-flow, are defined respectively as*

$$x_{u,t}^{new} = |\{\mathcal{T} \in \boldsymbol{P} : (a_1, b_1) \in u, t_1 = t\}|$$
$$x_{u,t}^{end} = \left|\{\mathcal{T} \in \boldsymbol{P} : (a_{|\mathcal{T}|}, b_{|\mathcal{T}|}) \in u, t_{|\mathcal{T}|} = t\}\right|$$

where $(a_i, b_i) \in u$ means that point p_i lies within region u.

Problem (Forecast Citywide Crowd Flows) *For $\forall u \in \mathcal{R}$ and $\forall \theta \in \{new, end\}$, given the historical crowd flows $x_{u,t}^{\theta}$ for $t = 0, ..., n-1$, predict $x_{u,n}^{\theta}$.*

2.2 A Case Study of Taxi Trajectories

We now analyze a case study of the Beijing's taxi GPS dataset **BJ** (detailed in Section 6.1). First, we partition Beijing into 372 low-level regions based on its road network as done in [25] and shown in Figure 4a. Since 372 regions are too many to monitor at city scale, we further cluster the low-level regions into 26 high-level regions with comparable crowd flow volumes, as shown in Figure 4b. The region IDs are also provided in this figure.

---

[1]Our data and code are available at http://cs.ucsb.edu/~mhoang/data/fccf.tar.gz
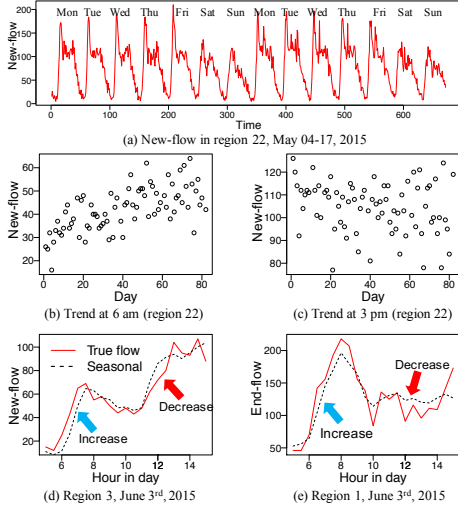
Figure 2: Beijing data (the regions are shown in Figure 4b). One timestamp is 30 minutes. (a) new-flow of region 22 during two weeks of May, 2015. (b, c) Trend of new-flow at 6am and 3pm for region 22 from March to June, 2015. (c, d) new-flow and end-flow of two neighboring regions (regions 3 and 1) during June 03, 2015.

We obtain the crowd flows by tracking the trajectories of taxis using their GPS signals. For example, Figures 2a shows the new-flow in region 22 during May 4th-17th, 2015, where each timestamp is 30 minutes. Clearly, the flows have a periodicity of day and week—a seasonal effect. Further, we can see a trend of change in this seasonal pattern over time, which may differ per region and per time in day. For example, as the weather got warmer, new-flows at 6am of region 22 in Figure 2b clearly got bigger on average. Whereas, new-flows at 3pm in this region (Figure 2c) got smaller possibly because it is less comfortable to travel outside when the temperature is too high.

Neighboring regions can affect each other due to the crowd flows among them. Figures 2d and 2e show an example of two neighboring regions 1 and 3 at the top right corner of Figure 4a. The new-flow of region 3 and the end-flow of region 1 deviate from their seasonal patterns at the same time and in the same direction (as marked by the blue and red arrows), suggesting their dependence on each other.

2.3 Prediction Framework

We discuss the segmentation of a city map into regions in Section 3. Then, we track and predict the crowd flows in these regions. Based on the observations in Section 2.2, we propose a prediction framework as shown in Figure 3.

Figure 3a shows the modelling framework for the crowd flows of a region. Specifically, we decompose a crowd flow time series $\boldsymbol{x} = (x_0, x_1, ..., x_{n-1})$ over n timestamps into three components: a seasonal component $\boldsymbol{s}$ capturing the periodic pattern, a trend component $\boldsymbol{y}$ capturing the offset from the periodic pattern for each timestamp in a period, and a spatio-temporal residual component $\boldsymbol{r}$ capturing the instantaneous changes. Thus,

$$\boldsymbol{x} = \boldsymbol{s} + \boldsymbol{y} + \boldsymbol{r} \qquad (1)$$

We use only temporal information to model $\boldsymbol{s}$ and $\boldsymbol{y}$. The seasonal model $\boldsymbol{s}$ is learned on the original flow $\boldsymbol{x}$, and the trend model $\boldsymbol{y}$ is learned on the residual $\boldsymbol{x} - \boldsymbol{s}$. After that, the residual $\boldsymbol{r}$ is learned for $\boldsymbol{x} - \boldsymbol{s} - \boldsymbol{y}$.

Assume that the periodicity in $\boldsymbol{x}$ has a length of period F, that is, $s_t = s_{t \bmod F} \ \forall t = 0, ..., n-1$, then we can divide
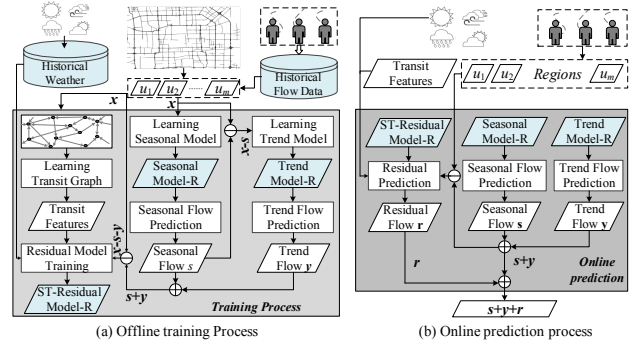

Figure 3: Framework overview.

$\boldsymbol{x}$ into a sequence of periods as shown in the first two lines of Table 1 for $n = 10$ and $F = 4$. In general, $\boldsymbol{x}$ will contain $n_y = \lfloor n/F \rfloor + 1$ periods, where period j contains timestamps in the range $[jF, (j+1)F - 1]$, for $j = 0, ..., n_y - 1$.

Since the timestamps within a period may have different evolutionary trends over time (see Figures 2b and 2c), we build a seperate trend model for each of them. Therefore, we decompose $\boldsymbol{x}$ into three components as in Table 1:

$$x_t = s_{t \bmod F} + y_{t \bmod F, \lfloor t/F \rfloor} + r_t \qquad (2)$$

where:

- $s_{t \bmod F}$ is the seasonal flow at the $(t \bmod F)$-th timestamp within a period.
- $y_{t \bmod F, \lfloor t/F \rfloor}$ is the offset from the seasonal flow of the $(t \bmod F)$-th timestamp in period $\lfloor t/F \rfloor$.
- r_t is the residual flow at time t.

In particular, we model $\boldsymbol{s}$ as a time series of length F: $\boldsymbol{s} = (s_0, s_1, ..., s_{F-1})$. For each i-th timestamp of a period ($i = 0, ..., F-1$), we model its trend across different periods as a time series $\boldsymbol{y}_i$ of length n_y: $\boldsymbol{y}_i = (y_{i0}, y_{i1}, ..., y_{i,n_y-1})$, where y_{ij} is the offset from the seasonal pattern of the i-th timestamp in period j. Finally, the residual $\boldsymbol{r}$ is modeled as a time series of length n: $r = (r_0, r_1, ..., r_{n-1})$.

Period 0				Period 1				Period 2	
x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
s_0	s_1	s_2	s_3	s_0	s_1	s_2	s_3	s_0	s_1
y_{00}	y_{10}	y_{20}	y_{30}	y_{01}	y_{11}	y_{21}	y_{31}	y_{02}	y_{12}
r_0	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9

Table 1: Decomposed flow $\boldsymbol{x} = \boldsymbol{s} + \boldsymbol{y} + \boldsymbol{r}$ for $n = 10$, $F = 4$.

Both $\boldsymbol{s}$ and $\boldsymbol{y}$ are built based on IGMRFs (Section 4), scalable to big data, and robust to noisy and missing data.

For the residual $\boldsymbol{r}$ (Section 5), we first propose a Bayesian network to model the transition probability of crowds among regions. By applying this transition probability into $\boldsymbol{r}$, we obtain the residual transit flows among regions—the transit features in Figure 3a—which capture the dependence among neighboring regions. Finally, we combine the transit features (inter-region dependence), the history of all types of flows of a region (intra-region dependence), and the weather data into a spatio-temporal residual model to predict $\boldsymbol{r}$.

We use the trained models to make online predictions for crowd flows of regions as described in Figure 3b.

For clarity, Table 2 lists the notations used in this paper.

3. FINDING REGIONS

We aim to divide a city into regions with two goals so that they are useful for high level traffic management: (i) the regions are are semantically meaningful, and (ii) the regions have comparable traffic volumes.

$\mathcal{R} = \{u_1, u_2, ..., u_m\}$	The set of all high-level regions
m, n	Number of regions, and number of timestamps
$x^{\theta}_{u,t}$	Crowd flow of region u at time t; $\theta \in \{new, end\}$
$\boldsymbol{x} = (x_0, ..., x_{n-1})$	Vector representing a flow time series
F	Length of a period in $\boldsymbol{x}$
$n_y = \lfloor n/F \rfloor + 1$	Number of periods in $\boldsymbol{x}$
$\boldsymbol{s} = (s_0, ..., s_{F-1})$	Seasonal component of $\boldsymbol{x}$
$\boldsymbol{y_i} = (y_{i0}, ..., y_{i,n_y-1})$	Trend component for the i-th timestamp in a period
$\boldsymbol{r} = (r_0, ..., r_{n-1})$	Residual component of $\boldsymbol{x}$
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	The graph of an IGMRF
$\boldsymbol{Q}$	$n \times n$ precision matrix for an IGMRF of size n
κ	Precision parameter of an IGMRF
d_{max}	Maximum transit duration between two regions
L	History length

Table 2: Symbols and notations.

For the first goal, we use the map segmentation method in [25] to partition the map of a city based on its road network. For example, the map of Beijing city can be divided into 372 low-level regions as shown in Figure 4a. Such regions are bounded by the roads and thus naturally capture the division of human activities, making them semantically meaningful. However, the number of low-level regions can be high, making it difficult to monitor all of them. In addition, these regions have highly varying areas and traffic volumes. On one hand, it is not straightforward for city managers to decide how to distribute their work force across the city. On the other hand, it is hard to predict the crowd flows of a tiny region due to the sparsity of data for such a small area. Many small regions are simply roundabounds, making their existence less meaningful. As a result, we propose to further group the low-level regions into bigger high-level regions that have comparable traffic volumes and contain low-level regions with similar crowd flow patterns. To do this, we cluster the region graph as defined below.

Definition 2 (Region graph) *A region graph is denoted as* $G = (V, E, N, W)$, *where*

- *Node set* $V = \{v_1, v_2, ...\}$ *is the set of low-level regions obtained using the map segmentation method in [25]*
- *Edge set* $E = \{(v_i, v_j)|v_i$ *and* v_j *are adjacent on the city map}*.
- *Node weights* N, *where* $N_{v_i} = \sum_{t=0}^{n-1}(x^{new}_{v_i,t} + x^{end}_{v_i,t})$ *is the sum of crowd flows in region* v_i *during the historical time period* $[0, n-1]$.
- *Edge weights* W, *where* W_{v_i,v_j} *is the similarity of crowd flow patterns between regions* v_i *and* v_j.

In this paper, we want to merge low-level regions with similar rise-and-fall crowd flow patterns, i.e., the plots of their crowd flows over time have similar shapes. Thus, we define the edge weight between two low-level regions v_i and v_j as the Spearman's rank correlation coefficient between their crowd flows during a historical time period. Specifically, each region can be represented as a vector $v_i = (x^{new}_{v_i,0}, ..., x^{new}_{v_i,n-1}, x^{end}_{v_i,0}, ..., x^{end}_{v_i,n-1})$. The correlation coefficients among the regions are computed on these vectors.

Figure 4c shows a subgraph of the region graph for our Beijing dataset. Here, each node represents a low-level region while its size represents the node weight. There is an edge between two regions if they share a boundary road. The edge widths are proportional to the edge weights.

Next, we cluster the region graph into m high-level regions $\mathcal{R} = \{u_1, u_2, ..., u_m\}$, where each high-level region u_i is a set of adjacent low-level regions, with two goals:

- Edge cut minimization: $\min_{\mathcal{R}} \sum_{\substack{v_i \in u_k; v_j \in u_l \\ u_k \neq u_l}} W_{v_i,v_j}$
- Cluster balancing: $\frac{\sum_{v_i \in u_j} N_{v_i}}{\sum_{v_i \in V} N_{v_i}/m} < 1 + \zeta \ \forall u_j \in \mathcal{R}$, where $\zeta > 0$ is a predefined imbalance factor.

(a) Road-based map segmentation (b) Region grouping using graph clustering

(c) Low-level region graph for the area in the red box in Fig. 4a (d) Region grouping for the graph in Fig. 4c
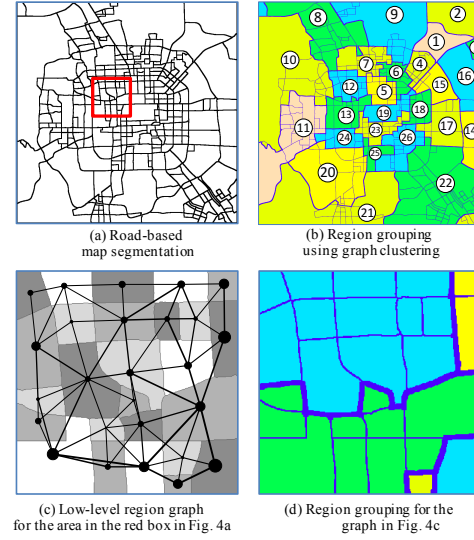
Figure 4: Finding regions in Beijing: (a) low-level regions based on city roads, (b) high-level regions based on crowd flow patterns, (c,d) the adjacency graph and region grouping for the red box in Figure 4a.
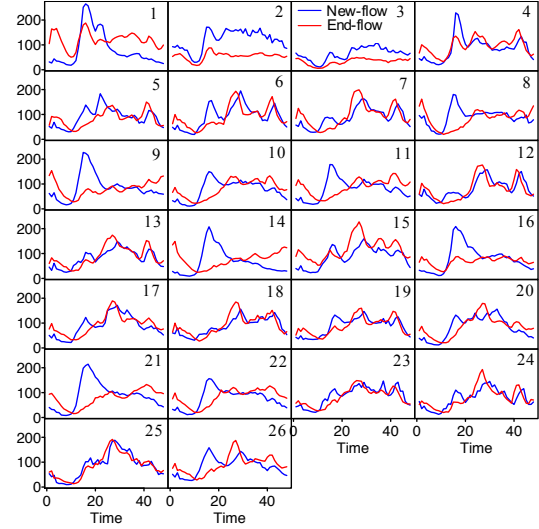

Figure 5: Average daily crowd flows of 26 regions in Fig. 5b. The region IDs are in the top right corners of the sub-figures.

The first goal helps us group highly similar low-level regions together. Whereas, the second goal constrains the sum of node weights in each cluster to be close to the attainable average. In other words, we want to balance the total traffic volumes among the clusters, which would be helpful for city planning and traffic management.

We use the graph clustering algorithm in [10] to cluster the region graph since it supports our two goals (we set $\zeta = 0.1$). To choose the number of clusters m, we use the elbow method [14] on the edge cut: $m = 26$ for the Beijing dataset and $m = 15$ for the NYC taxi dataset. The NYC bike dataset has only 23 regions, and there is thus no need to futher reduce the number of regions. Figure 4d shows the resulting high-level regions for the corresponding subgraph in Figure 4c. The whole high-level region map for Beijing is shown in Figure 4b, with the obtained region IDs. Further, Figure 5 summarizes the average daily crowd flows of the 26 high-level regions in Figure 4b. Clearly, these regions have comparable total traffic volumes. We can also see some distinctive traffic patterns, suggesting that the obtained clus-
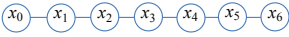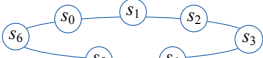
Table 3 (at top of page):

Model	Trend	Seasonal
Graph $\mathcal{G}$	(chain graph $x_0 - x_1 - x_2 - x_3 - x_4 - x_5 - x_6$)	(cyclic graph on nodes $s_0, s_1, s_2, s_3, s_4, s_5, s_6$)
Probability density π	$\pi(\boldsymbol{x}\vert\kappa) \propto \kappa^{(n-1)/2} exp\left(-\frac{1}{2}\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x}\right)$	$\pi(\boldsymbol{s}\vert\kappa) \propto \kappa^{(F-1)/2} exp\left(-\frac{1}{2}\boldsymbol{s}^T\boldsymbol{Q}\boldsymbol{s}\right)$
Precision matrix $\boldsymbol{Q}$	$\kappa \times \begin{pmatrix} 1 & -1 & & & & & \\ -1 & 2 & -1 & & & & \\ & -1 & 2 & -1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & -1 & 2 & -1 & \\ & & & & -1 & 2 & -1 \\ & & & & & -1 & 1 \end{pmatrix}$	$\kappa \times \begin{pmatrix} 2 & -1 & & & & & -1 \\ -1 & 2 & -1 & & & & \\ & -1 & 2 & -1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & -1 & 2 & -1 & \\ & & & & -1 & 2 & -1 \\ -1 & & & & & -1 & 2 \end{pmatrix}$

Table 3: Temporal IGMRF models: $\mathcal{G}$ is the graph ($n = 7$) and $\boldsymbol{Q}$ is the precision matrix; $\kappa \in \mathbb{R}$ is the precision parameter to be learned.

ters are meaningful. Neighboring regions may have similar patterns due to their geographhical proximity.

4. TEMPORAL MODELS

In this section, we build the seasonal model $\boldsymbol{s}$ and trend model $\boldsymbol{y}$ based on Intrinsic Gaussian Markov Random Fields (IGMRF) by capturing the temporal information in the crowd flows. We first give a brief introduction to IGMRF.

4.1 Temporal IGMRF Models

To model a time series $\boldsymbol{x} = (x_0, x_1, ..., x_{n-1})$ over n timestamps, we treat $\boldsymbol{x}$ as a temporal IGMRF, that is, a random vector $\boldsymbol{x}$ having an improper Gaussian density. The IGMRF model fits our city-scale prediction problem well since it is robust to noise and missing data, and scalable to big data.

The temporal IGMRF is specified by its precision matrix $\boldsymbol{Q}$ and undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ and $\mathcal{E}$ are the node set and edge set respectively. In essence, the structure of $\mathcal{G}$ visually summarizes the conditional dependence among timestamps, while the value of matrix $\boldsymbol{Q}$ decides the specific probability density $\pi(\boldsymbol{x})$ of the distribution of $\boldsymbol{x}$. The second column in Table 3 shows an example IGMRF with its $\mathcal{G}$, $\boldsymbol{Q}$, and $\pi(\boldsymbol{x}\vert\kappa)$, where κ is a parameter to be learned. In particular, each timestamp is represented by a row and a column in $\boldsymbol{Q}$, as well as a node in $\mathcal{V}$. Non-zero entries in $\boldsymbol{Q}$ correspond to edges between corresponding nodes in $\mathcal{E}$. Each zero entry in $\boldsymbol{Q}$—or equivalently, the absence of an edge in $\mathcal{G}$—signifies that two corresponding timestamps are conditionally independent given the other timestamps. While the general form of $\boldsymbol{Q}$ is given in Table 3, we still need to learn the parameter $\kappa \in \mathbb{R}$ from data to set a specific value for $\boldsymbol{Q}$.

The formal definition of IGMRF [15] is given below. First, let us define a symmetric matrix $\boldsymbol{Q} \in \mathbb{R}^{n \times n}$ as symmetric positive semi-definite (SPSD) iff $\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} \geq 0 \ \forall \boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{x} \neq \boldsymbol{0}$.

Definition 3 (IGMRF) *Let $\boldsymbol{Q}$ be a SPSD precision matrix with rank $n-k > 0$. A random vector $\boldsymbol{x} = (x_0, ..., x_{n-1})$ is an IGMRF of rank $n - k$ with parameters $(\boldsymbol{\mu}, \boldsymbol{Q})$ iff $\boldsymbol{x}$ follows an improper Gaussian distribution, that is, its probability density function $\pi(\boldsymbol{x})$ has the form:*

$$\pi(\boldsymbol{x}) = (2\pi)^{-\frac{n-k}{2}}(\vert\boldsymbol{Q}\vert^*)^{1/2} exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{Q}(\boldsymbol{x}-\boldsymbol{\mu})\right) \quad (3)$$

Further, $\boldsymbol{x}$ is an IGMRF wrt a labelled graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{0, ..., n-1\}$ and

$$Q_{ij} \neq 0 \Leftrightarrow (i, j) \in \mathcal{E} \ \forall i \neq j$$

Here, $\vert\boldsymbol{Q}\vert^*$ denotes the generalized determinant equal to the product of the $n-k$ non-zero eigenvalues of $\boldsymbol{Q}$. The first condition states that $\boldsymbol{x}$ follows an improper Gaussian distribution, the probabilistic nature of which makes the IGMRF robust to noise and missing data. Whereas, the second condition ($Q_{ij} \neq 0$ only if i and j are neighbors in $\mathcal{G}$) is the

Markov property that makes $\boldsymbol{Q}$ and $\mathcal{G}$ sparse, hence easier to store and faster to compute as shown in [15]. We next design $\boldsymbol{Q}$ and $\mathcal{G}$ based on the forward differences:

Definition 4 (Forward differences) *Given a time series $\boldsymbol{x} = (x_0, ..., x_{n-1})$, the first-order forward difference at time t is defined as*

$$\Delta x_t = x_{t+1} - x_t, \quad t = 0, ..., n - 2$$

Gaussian assumption: *To make $\boldsymbol{Q}$ and $\mathcal{G}$ sparse, we impose the following assumption on $\boldsymbol{x}$:*

$$\Delta x_t \overset{iid}{\sim} \mathcal{N}(0, \kappa^{-1}), \quad t = 0, ..., n - 2 \quad (4)$$

where $\kappa \in \mathbb{R}$ is the precision parameter learned from data.

The graphs $\mathcal{G}$ for the Gaussian assumption is shown in the second column of Table 3 in case $n = 7$. There is an edge in $\mathcal{G}$ for and only for pairs of consecutive timestamps.

The Gaussian assumption reduces the numbers of edges in $\mathcal{G}$ and non-zero entries in $\boldsymbol{Q}$ to $O(n)$, making them very sparse and our solution scalable. Further, it imposes a smooth change between consecutive timestamps in the time series, making the IGMRF robust to noisy and missing data. To find $\boldsymbol{Q}$, we form the probability density $\pi(\boldsymbol{x})$ as in Equation 3:

$$\pi(\boldsymbol{x}\vert\kappa) \propto \kappa^{(n-1)/2} exp\left(-\frac{\kappa}{2}\sum_{t=0}^{n-2}(\Delta x_t)^2\right)$$

$$= \kappa^{(n-1)/2} exp\left(-\frac{\kappa}{2}\sum_{t=0}^{n-2}(x_{t+1} - x_t)^2\right)$$

$$= \kappa^{(n-1)/2} exp\left(-\frac{1}{2}\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x}\right) \quad (5)$$

where the $n \times n$ precision matrix $\boldsymbol{Q}$ is shown in the second column of Table 3 (zero entries are not shown), and κ is the parameter to be learned.

Learning IGMRF: We learn the IGMRFs by the integrated nested Laplace approximations approach [2]. In essence, we find the parameter κ using maximum a posterior (MAP) estimation given some prior distribution $\pi(\kappa)$ of κ:

$$\arg\max_{\kappa} \pi(\kappa\vert\boldsymbol{x}) = \arg\max_{\kappa} \pi(\boldsymbol{x}\vert\kappa)\pi(\kappa) \quad (6)$$

Note that the computation of an IGMRF can be sped up using the Cholesky factorization $\boldsymbol{Q} = \boldsymbol{L}\boldsymbol{L}^T$, where $\boldsymbol{L}$ is a lower triangular matrix. With the Gaussian assumption, $\boldsymbol{Q}$ becomes sparse with $O(n)$ non-zero entries, reducing the factorization cost from $O(n^3)$ in general to $O(n)$ [15].

4.2 IGMRF Seasonal Models

Gaussian properties of crowd flows: Figure 6a shows the histograms and the fitted normal distributions using maximum-likelihood estimation for the forward differences of the square root of new-flow in a region in **BJ** and **BIKE** (see Section 6.1 for dataset descriptions). Visually, the fitted normal distributions closely match the histograms. Thus, we can use our Gaussian assumption for **BJ** and **BIKE**

to model the square root of new-flow. We note that the square root of flows follow Gaussian distribution but the raw flows do not. We obtain similar results for end-flows and other datasets and report the complete results in Section 6.1. Therefore, we propose to build an IGMRF seasonal model for the square-root of crowd flows.

Seasonal model: For a periodic time series with period length F, we design an IGMRF $s = (s_0, s_1, ..., s_{F-1})$ as a seasonal model, with an additional assumption on the smooth change between s_{F-1} and s_0. The graph $\mathcal{G}$ thus becomes circular: there is an additional edge between the last timestamp and the first timestamp. Specifically, the circular graph $\mathcal{G}$ is shown in the last column of Table 3. To impose the circular property of $\mathcal{G}$, we modify the forward differences for $i = 0, ..., F-1$ as follows:

$$\Delta s_i = s_{(i+1) \bmod F} - s_i$$

Here the Gaussian assumption is $\Delta s_i \overset{iid}{\sim} \mathcal{N}(0, \kappa_s^{-1}) \ \forall i = 0, ..., F-1$, and $\kappa_s \in \mathbb{R}$ is the only parameter we need to learn. The corresponding circular precision matrix Q can be derived similarly to the case without the seasonal assumption, and is given in Table 3.

4.3   IGMRF Trend Models

Gaussian properties of trend: After the seasonal pattern is removed from the flows, we obtain the raw residual $x - s$. Figure 6b shows the histograms and the fitted normal distributions for the forward differences of these residuals for the same flows in Figure 6a. Again, it is clear that Gaussian distributions can be used to approximate these histograms. We have the same observations for all flow types and datasets, as statistically shown in Section 6.1. Thus, we can build an IGMRF trend model as follows.

Trend model: As discussed in Section 2.3, we propose to add a trend y_i to capture the change over time in the seasonal pattern of the i-th timestamp in a period. In particular, we want to model a time series $y_i = (y_{i0}, y_{i1}, ..., y_{i,n_y-1})$, where n_y is the number of periods. The temporal IGMRF model in Section 4.1 can be used directly for this purpose.
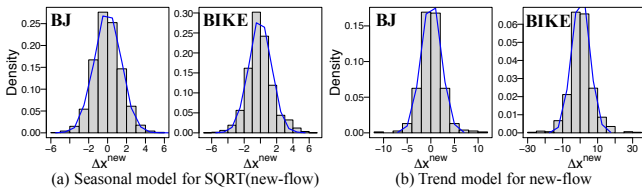

Figure 6: Gaussian properties of new-flow of one region.

5.   SPATIO-TEMPORAL RESIDUAL MODEL

In this section, we utilize the (intra-region and inter-region) dependencies among different flows and weather information to predict the residuals $r = x - s - y$. Denote $r_{u,t}^{\theta}$ as the residual flow of type θ at time t in region u, where $\theta \in \{new, end\}$. We design our final spatio-temporal residual model to predict $r_{u,t}^{\theta}$ as a regression problem:

$$r_{u,t}^{\theta} = \boldsymbol{\alpha}_{u,\theta}^T \boldsymbol{\delta}_{u,t} + \boldsymbol{\beta}_{u,\theta}^T \boldsymbol{z}_{u,t} + \phi_{u,w_t}^{\theta} + \sigma_{u,h_t}^{\theta} + \gamma_u^{\theta} \quad (7)$$

where the inputs are:

- $\boldsymbol{\delta}_{u,t}$: Transit features of region u at time t, capturing the *inter-region dependence* among flows (Section 5.1.2).

- $\boldsymbol{z}_{u,t} = (r_{u,t-i}^{new}, r_{u,t-i}^{end} | i = 1, 2, ..., L)^T$: Historical residual flows of region u at time t, capturing *intra-region dependence* among different flow types, where L is a chosen history length.

- w_t: The weather condition w_t at time t.

  Model parameters (for region u and flow type θ): $\boldsymbol{\alpha}_{u,\theta}$ and $\boldsymbol{\beta}_{u,\theta}$ are coefficient vectors for $\boldsymbol{\delta}_{u,t}$ and $\boldsymbol{z}_{u,t}$; ϕ_{u,w_t}^{θ} is a coefficient for weather condition w_t; σ_{u,h_t}^{θ} is a coefficient for the hour in day h_t of time t; and γ_u^{θ} is an intercept.

  Other factors: While we only consider weather and public holidays here, if more data is available (e.g., local social events in each region, traffic accidents, or traffic jams), we can easily and similarly include it into the residual model r.

  We next explain in details how to capture the inter-region dependence and the effects of weather and holidays.

5.1   Capturing Inter-region Dependence

5.1.1   Bayesian Network Transit Model

The flows of neighboring regions can affect each other due to the transition of objects among them. This inter-region dependence can be naturally captured by a Bayesian network transit model as shown in Figure 7a. For an individual that is moving between regions, when she gets out of a region R, we predict the next region R' she will move to and the time duration d that she will take to complete the transition. We assume that the next region R' depends on the current region R, different hours in a day h, and different types of days η. Similarly, the transit duration d is dependent on R, h, η as well as R', where she is traveling to. Hour in day h can take any integer value between 1 and 24. Days with similar transit patterns are grouped into one day type. We discuss how day types η are determined in Section 5.1.3.

Our Bayesian Network can be learned easily and fast by counting since it has known structure and full observability (R, R', h, and η are all pre-defined). Denote h_t and η_t as the hour in day and the day type for timestamp t respectively. Once we have learned the conditional probability functions $p(R'|R, h_t, \eta_t)$ and $p(d|R', R, h_t, \eta_t)$, we can compute the probability that an individual getting out from R at time t will transit to R' after d timestamps as:

$$g_{R,R',t,d} = p(R'|R, h_t, \eta_t) \times p(d|R', R, h_t, \eta_t) \quad (8)$$

5.1.2   Transit Features

To capture the influences among regions with regards to their deviations from the expected flows $(s + y)$, we apply g from Equation 8 to the residual flow $r = x - s - y$. Here, we assume that the transition probabilities g are the same for both the temporal components $(s + t)$ and the residual component r. While this is a strong assumption, it eliminates the need to model the flows between every pair of regions, which is costly and more susceptible to noise. With this assumption, we define the ***transit features*** in Equation 7 as $\boldsymbol{\delta}_{u,t} = (\delta_{u,t}^{new}, \delta_{u,t}^{end})^T$, where $\delta_{u,t}^{new}$ is the sum of crowd flows from other regions to u, and $\delta_{u,t}^{end}$ is the sum of crowd flows from u to other regions in the last d_{max} timestamps:

$$\delta_{u,t}^{new} = \sum_{d=1}^{d_{max}} \sum_{v \in \mathcal{R}} (r_{u,t-d}^{new} \times g_{u,v,t-d,d}) \quad (9)$$

$$\delta_{u,t}^{end} = \sum_{d=1}^{d_{max}} \sum_{v \in \mathcal{R}} (r_{v,t-d}^{new} \times g_{v,u,t-d,d}) \quad (10)$$

We define a maximum transit duration d_{max} since most transitions take a bounded amount of time in real life.
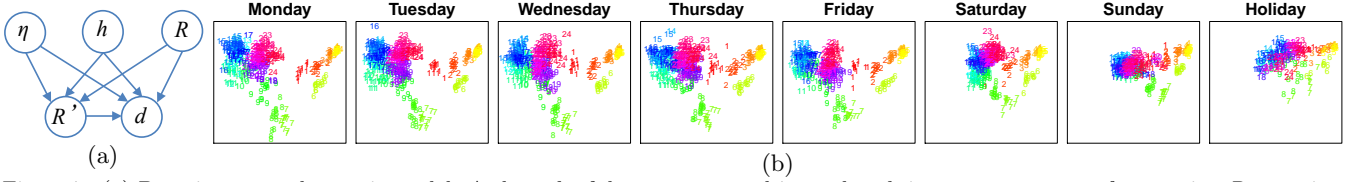
Figure 7: (a) Bayesian network transit model: At hour h of day type η, an object takes d timestamps to move from region R to region R' on average. (b) Mapping timestamps into 2D-space based on transit patterns using PARAFAC. Each point is for one hour in a day.

5.1.3 Choice of Day Type η

To decide the day type η, we group days with similar transit patterns using a data-driven approach. Denote $\mathcal{Y} \in \mathbb{N}^{n \times m \times m \times d_{max}}$ as the (timestamp, source region, destination region, transit duration) tensor, where $\mathcal{Y}_{tuvd}$ is the number of trajectories that leave region u at timestamp t, and arrive at region v at timestamps $t+d$. To compare the transit patterns at two timestamps t_1 and t_2, we can compare the two sub-tensors $\mathcal{Y}_{t_1\ldots}$ and $\mathcal{Y}_{t_2\ldots}$, which are high-dimensional and sparse. Thus, to avoid the curse of dimensionality, we perform dimension reduction using PARAFAC tensor factorization [11]. In particular, $\mathcal{Y}$ can be factorized into four matrices $\boldsymbol{M} \in \mathbb{R}^{n \times q}$, $\boldsymbol{H} \in \mathbb{R}^{m \times q}$, $\boldsymbol{I} \in \mathbb{R}^{m \times q}$, and $\boldsymbol{J} \in \mathbb{R}^{d_{max} \times q}$, where q is the number of lower dimensions, such that:

$$\mathcal{Y}_{tuvd} = \sum_{i=1}^{q} M_{ti} H_{ui} I_{vi} J_{di} \qquad (11)$$

Matrix $\boldsymbol{M}$ is the low-dimension representation of the timestamps (each row is one timestamp, and each column is one dimension). We can use this low-dimension representation to compare the transit patterns at different timestamps.

Figure 7b shows the mapping result of timestamps into a 2D-space ($q = 2$) for dataset **BJ-B** (Section 6.1). Timestamps are separated into different weekdays and holidays. Each point in a subfigure is for one hour in a day (marked as 1 to 24). From these figures, we can visually put the daily transit patterns into five groups: (i) Monday to Friday, (ii) Saturday, (iii) Sunday, and (iv) holidays. We thus use these four day types for **BJ-B**. Similar results are obtained for other datasets but not shown due to space limitation.

5.2 Effects of Weather and Holidays

Weather and holidays can affect crowd flows. For example, Figure 8a shows that thunderstorm may increase the use of taxi while Figure 8b shows that heavy rain may reduce the crowd flows at a region compared to its seasonal pattern. Figure 8b also shows that crowd flows during a holiday can be significantly different from the flows during normal days.
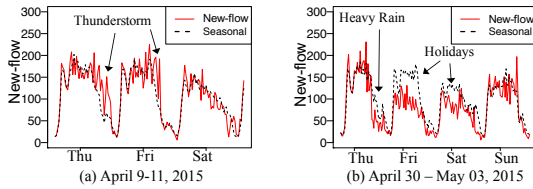

Figure 8: Effects of weather and holiday in region 2 in Beijing.

We include weather and holidays into our models in two ways. First, we build a separate seasonal model for holidays with the period of a day ($F_{holiday} = 1$ day). Second, we add a coefficient ϕ_{u,w_t}^{θ} for each region u, flow type θ, and weather condition w_t at time t in the regression (Equation 7).

6. EXPERIMENTS

6.1 Settings

Datasets: We use three different sets of data as summarized in Table 4. Each dataset contains three sub-datasets: trajectory, region map, and weather data, as detailed below.

BJ: Trajectory data is the taxi GPS data in Beijing in 2015. We categorize weather data into good weather (sunny, cloudy) and bad weather (rainy, storm, dusty). With our clustering framework in Section 3, we partition Beijing into 26 high-level regions (Figure 4a). Using Definition 1, we obtain two types of crowd flows. Data for 350 timestamps were missing for all regions due to system glitches. We choose data from the last three weeks as testing data, and all data before that as training data.

NYC: We partition NYC into 15 high-level regions using its road network and traffic data. The trajectory data is generated by taxis in NYC in 2013. Trip data includes: taxi ID, pick-up and drop-off locations and times. The new-flow and end-flow are thus the number of pick-ups and drop-offs in a region respectively. Weather conditions include good weather (sunny, or no available data) and bad weather (foggy, rainy, snowy). We pick the last-week data for testing, and data before that for training.

BIKE: Trajectory data is taken from New York City Bike system in 2014. Trip data includes: trip duration, start and end station IDs, start and end times. Following [12], we group bike stations into clusters using their bipartitie clustering method, and treat each obtained station cluster as a region, instead of using our own clustering framework in Section 3. For each region, the new-flow is the number of checked-out bikes, and the end-flow is the number of checked-in bikes. We use data from Apr. 1st to Sep. 10th for training and Sep. 10th to 30th for testing.

Gaussian properties of crowd flows: To statistically verify if a data sample follows a Gaussian distribution, we perform the Kolmogorov-Smirnov test (KS-test). If the KS-test returns a p-value greater than 0.01, the Gaussian hypothesis is acceptable. Table 5 reports the proportion of regions whose crowd flows follow Gaussian distributions for each dataset. Note that seasonal models are tested on the square root of flows, while trend models are tested on the

Dataset	BJ	NYC	BIKE
Data type	Taxi GPS	Taxi pickup	Bike rent
Location	Beijing	New York	New York
Start time	3/1/2015	1/1/2013	4/1/2014
End time	6/28/2015	9/8/2013	9/30/2014
#holidays	10	20	9
Timestamp bin size	30 minutes	1 hour	1 hour
	Trajectory data		
#taxis/bikes 34K	33.6K	16K	6.8K
#trips/records	11.7M	206.6M	5.4M
#effective timestamps	2,753	5,880	4,392
#missing timestamps	350	0	0
	Region map data		
#roads/bike stations 193,663	193,663	32,210	344
#low-level regions	372	215	-
#high-level regions	26	15	23
	Weather data		
#good-weather timestamps	3,812	4,081	5,398
#bad-weather timestamps	1,208	358	448
#missing timestamps	486	53	34
Temperature	$[-7,34]^{\circ}$C	$[32,90]^{\circ}$F	$[10,97]^{\circ}$F

Table 4: Datasets (holidays include adjacent weekends).

Dataset	BJ		NYC		BIKE	
	Seasonal	Trend	Seasonal	Trend	Seasonal	Trend
new-flow	0.96	1	0.93	1	1	1
end-flow	1	1	1	1	1	1

Table 5: Proportions of regions whose crowd flows satisfy the Gaussian assumption.

residual of the corresponding seasonal models. We emphasize that, for the seasonal models, the Gaussian assumptions are poorly satisfied by the raw crowd flows.

Parameter settings: Each timestamp corresponds to 1 hour for **NYC** and **BIKE**, and 30 minutes for **BJ**. Due to the differences in the crowd flows between weekdays and weekends, and between different weekdays (e.g., see Figure 2), we choose $F = 1$ week for the seasonal model s^w during normal days, and $F = 1$ day for the seasonal model s^h during holidays. Log-gamma prior is used for the precision parameter κ in Equation 6 as suggested in [15].

Evaluation metric: For evaluation, we use the Root Mean Squared Error (RMSE), as defined below:

$$RMSE(\theta) = \frac{1}{m} \sum_{t=1}^{m} \sqrt{\frac{1}{n} \sum_{u=1}^{n} (x_{u,t}^{\theta} - \hat{x}_{u,t}^{\theta})^2} \qquad (12)$$

where n is the number of regions, m is the number of timestamps, $x_{u,t}^{\theta}$ and $\hat{x}_{u,t}^{\theta}$ are respectively the true and predicted values of flow type θ in region u at time t.

Experiments are run on a Debian machine with Intel i7, 3.50GHz CPU and 15GB RAM. The IGMRF models are implemented using the R-inla package [2].

Baselines: Table 6 lists all compared methods. **FCCF** is our final spatio-temporal model. **SARIMA** is the seasonal ARIMA model, using only temporal data. We choose the best parameters for the SARIMA models using the "forecast" package in R language [6]. **LmNei** is a naive linear regression spatio-temporal model. **VAR** (vector auto-regressive model) and **STARMA** (space-time auto-regressive moving average model) are more advanced spatio-temporal models. **VAR** captures the pairwise relationships among all flows, and has heavy computational costs due to the large number of parameters. **STARMA** [7] has fewer parameters thanks to the spatial constraints, but requires an ad hoc definition of the weight matrices capturing the relationships among flows from the same or neighboring regions. **HP-BC-MSI** [12], the state-of-the-art prediction framework for bike-sharing systems, is the most similar to our problem. **HP-BC-MSI** predicts the in/new-flows for clusters of bike stations instead of the noisy individual station flows. Further, it first predicts the aggregated flow for the whole city, and then distributes this flow into each cluster (hierarchical prediction). However, it does not decompose the flows into three components as we do. To compare with **HP-BC-MSI**, we also predict the flows for the same clusters of stations by treating them as high-level regions. Finally, we break down our spatio-temporal model to investigate the contributions of each component.

6.2 Results

Complete framework: Table 7 shows the RMSE of all methods. Our complete framework consistently and significantly outperforms all baselines. Specifically, **FCCF** is 22% to 52% better than **LmNei**, 25% to 50% better than **SARIMA**, 10% to 30% better than **VAR**, and 27% to 70% better than **STARMA**. **VAR** exploits the relationship among flows and is clearly better than other baseline

Method	Description
	Temporal models
SARIMA	Seasonal ARIMA model, frequency = 24.
S	$x = s^w$, weekly seasonal model, where s^w follows Section 4.2 with $F = 168$.
SH	$x = s = s^w + s^h$, where s^w is a seasonal model ($F = 128$) for normal days, s^h is a daily seasonal model ($F = 24$) for holidays (Section 4.2).
SHT	$x = s + y$, **SH** and trend model.
	Spatio-temporal models
LmNei	Linear regression of the historical flows of a region and its neighbors, as well as weather information.
VAR(p)	Vector Auto-Regressive model with lag p.
STARMA(p,q)	Space Time Auto-Regressive Moving Average model [7], p and q are the AR and MA lags.
HP-BC-MSI	(**BIKE** only [12]) Hierarchical prediction + bipartite clustering + multi-similarity-based inference.
SHT+intra	$x = s + y + r$; r follows Equation 7 without transit features $\boldsymbol{\alpha}^T \boldsymbol{\delta}_{u,i}$ and weather ϕ_{u,w_t}^θ.
FCCFnoWea	$x = s + y + r$; r follows Equation 7, without weather ϕ_{u,w_t}^θ.
FCCF	$x = s + y + r$; r follows Equation 7.

Table 6: Baselines

Model	BJ		NYC		BIKE	
	New	End	New	End	New	End
SARIMA	21.20	18.85	132.82	142.27	20.50	19.38
lmNei	19.17	18.18	154.38	146.92	22.26	20.62
STARMA(3,1)	42.46	19.57	287.34	161.20	26.94	21.01
VAR(5)	15.83	15.83	106.81	101.32	15.36	13.05
HP-MSI-BC					14.70	15.60
S	17.54	16.38	192.77	190.56	18.27	18.09
SH	17.18	16.34	159.90	155.10	17.56	17.33
SHT	16.60	15.80	156.06	153.84	15.24	14.84
SHTIntra	14.63	14.28	89.04	84.84	11.55	10.92
FCFCNoWea	14.19	**14.14**	87.93	84.45	10.83	9.83
FCFC	**14.17**	**14.14**	**87.18**	**83.89**	**10.79**	**9.80**

Table 7: RMSE.

methods. While both **LmNei** and **STARMA** use spatial information, they are far worse than **VAR**, and even worse than **SARIMA**, suggesting that the ad hoc assignments of the weight matrices in **STARMA** or the naive way of incorprating spatial information in **LmNei** can actually hurt performance. Moreover, this observation also hints that the prediction of a future crowd flow depends heavily on its own history. Finally, **FCCF** decreases the error by 26% for new-flow and 37% for end-flow in **BIKE** compared to **HP-BC-MSI**, the state of the art for prediction in bike-sharing systems, showing the clear benefits of our decomposing flows into three components.

Temporal components: As seen in Table 7, the seasonal model **SH** that considers holidays are clearly better than the one without holidays (**S**). The accuracy is further increased when trend is added (**SHT**). Our seasonal models **S** and **SH** are better than **LmNei** and **SARIMA** for the **BJ** and **BIKE** datasets, while worse for the **NYC** dataset. This is possibly due to the different levels of noise in different datasets. Specifically, the region crowd flows in the **BJ** and **BIKE** datasets are significantly smaller than those in **BIKE**, leading to noisier data. Thanks to its probabilistic nature, our IGMRF models are robust to noise, and thus give better prediction in the two more noisy datasets.

Spatial-temporal components: **SHT+intra** combines the intra-region dependence into **SHT**, leading to an outstanding improvement in accuracy. The addition of the transit features (**FCCFnoWea**) further reduces RMSE, which is more significant for **BIKE** than for the other three datasets, since bike trips are generally longer than taxi trips, and often take more than one 1-hour timestamp to complete. In

other words, the bigger the ratio between the average trip duration and the timestamp duration, the bigger the impact of inter-region dependence for short-term flow predictions.

Weather effect: The addition of weather (**FCCF**) improves the accuracy for all datasets except the end-flow of **BJ**, possibly due to the high number of timestamps with missing weather data in **BJ**.

Multi-step-ahead prediction: To predict the crowd flows for multiple steps ahead, we change the left hand side of Equation 7 from $r_{u,t}^\theta$ to $r_{u,t+\Delta}^\theta$, for $\Delta \in \{1, 2, 3, 4\}$. Figure 9a shows the results. Clearly, the farther in the future, the harder the prediction and the higher the error.

Training period: Figure 9b shows the errors as we vary the length of the training period from 1 month to all available months (4 months for **BJ**, 8+ months for **NYC**, and 5 months for **BIKE**). While more training data generally leads to higher accuracy, the addition of more training data after 3 months does not improve the results significantly.

Missing data: We evaluate the robustness of our framework against missing and noisy data by making predictions when a proportion (20%, 50%, and 70%) of the timestamps is randomly removed from the training datasets for all regions. As shown in Figure 9c, even when 50% of the training data is missing, the performance of **FCCF** is still exceptionally good. Specifically, **FCCF** with 70% missing data is still better than **VAR** with complete data, and significantly outperforms **SARIMA**, **LmNei**, **STARMA**, and **HP-BC-MSI** with complete data (as shown in Table 7).

Efficiency: Figure 9d shows the running times without any parallelization. As can be seen, the total time for offline training is less than 10 minutes for all three datasets. More importantly, online prediction takes less than 1 minute for all datasets, showing that our framework is practical for real-time prediction of citywide crowd flows. In practice, we can train the temporal models x and y in parallel for all flows.

6.3 Case Studies

Figure 10a shows the region map for lower Manhattan and Brooklyn, New York, for **BIKE** in 2014. To show that our framework can capture the sudden deviations of crowd flows from their usual patterns, we investigate two anomalous case studies: suddenly decreased and suddenly increased flows.

During a rainy day (Figure 10b, region $R9$, Sept. 13th,

2014), the flows of bikers were significantly reduced. From 1pm to 3pm, the weather turned from sunny to foggy with strong wind, making people wary of traveling by bike. Thus, the true end-flow (red solid line) into region $R9$ became smaller than its seasonal pattern (**SHT**, brown circles). From 4pm, it started to rain, leading to a big decrease of flows and deviation from **SHT**. By including the recent history of $R9$, **SHT+intra** (blue empty circles) better tracks the true flow but is still far from the truth. **FCCF** (diamonds) further notices the reduction of crowd flows from other regions into $R9$ due to bad weather conditions, and improves the prediction significantly compared to **SHT+intra**.

On Sept. 17th, 2014, an enormous flow of people traveled to Zuccotti Park in region $R1$ to celebrate the three-year anniversary of Occupy Wall Street protest at 8am. Figure 10a shows the large crowd flows from different origins traveling to $R1$ before 8am. As a result, the end-flow of $R1$ at 8am, as well as the new-flows of these origins (e.g., $R2$, $R3$, $R5$, $R7$) during the previous hours were anomalously higher. For example, an increase in the new-flow of region $R2$ at 7am (Figure 10c, red solid line) led to a later increase in end-flow of $R1$ at 8am (Figure 10d, red solid line), as annotated by the arrows. Model **SHT+intra** (blue circles), which only considers the history of region $R1$, fails to predict this sudden increase at 8am, as pointed out by the red arrow in Figure 10d. Whereas, **FCCF** (diamonds) captures the sudden increase in new-flows from other regions in the previous hours and thus much better tracks the ground truth of $R1$. Note that there is still room for improvement; for example, if we had known that Occupy Wall Street would happen beforehand, we may include one more coefficient for local social events in the residual model r to further improve prediction.

7. RELATED WORK

Human Mobility Prediction: Prior research [4, 18, 19, 26] has been done to predict an individual's movement based on their location history, in order to enable context-aware computing that can facilitate the individual's daily life, such as suggesting driving directions, pushing promotion coupons, or predicting human mobility under disaster scenarios. Unlike such research, we forecast the aggregated crowd flows in a region rather than millions of individuals' mobility traces. The latter is very difficult, computationally expensive, and not necessary for the application scenario of public safety.
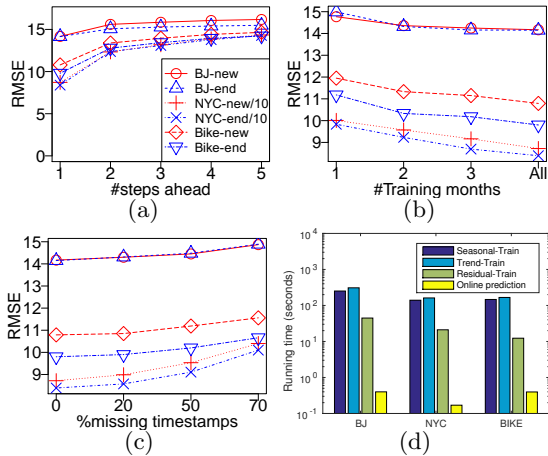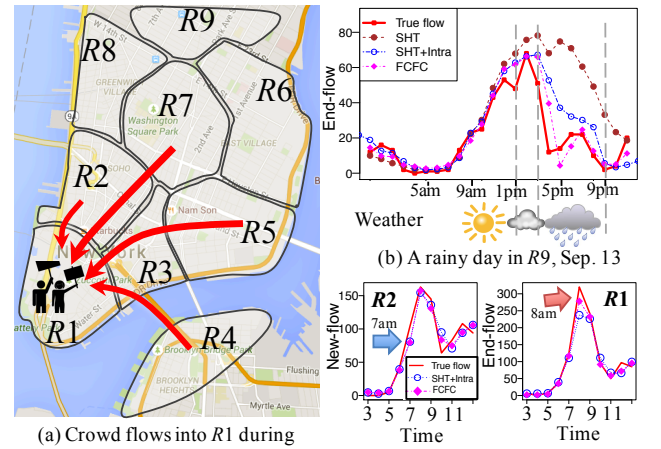

Figure 9: **FCFC**: (a) Multi-step ahead prediction; Effects of (b) fewer training data and (c) randomly missing data. (We plot RMSE/10 for NYC dataset for clearer figures). (d) Running time for training and predicting.

(a) Crowd flows into $R1$ during Occupy Wall Street (8am, Sep. 17)
(b) A rainy day in $R9$, Sep. 13
(c) $R2$, Sep. 17
(d) $R1$, Sep. 17

Figure 10: Crowd flow prediction for **BIKE**, in NYC, 2014.

Traffic Condition Prediction on Roads: Another branch of research has been conducted to predict travel speeds and traffic volume on road networks. The majority of such research [1, 8, 9, 17] focuses on the prediction on a single (or a few) road segment(s), rather than a city-scale prediction. A number of works also use Bayesian network approach [20, 22] and Markov Random Fields [3] for road traffic forecasting. Some recent studies [16, 21] try to scale up the prediction throughout an entire city, with a diversity of models, such as matrix factorization and tensor decomposition. [5] presents research on developing models that forecast traffic flow and congestion in a deployed traffic forecasting service.

Our method differs from the above problems in the following aspects. First, we study the crowd flows in a region rather than traffic conditions on a road segment. The region-based flows provide a macro-level view of city traffic, which is important not only for traffic management but also for public safety. The four types of flows we considered are only meaningful within a region setting. In addition, people can cross regions without being constrained by road networks, for example, by walking or subway systems. Second, given the four types of flows, our problem becomes more difficult, as there are dependencies between different types of flows in a region and dependencies between flows of different regions. Third, we cluster regions into groups based on flow patterns, predicting the crowd flows in a cluster. The latter can deal with the data sparsity and help adjust the flow in each individual region belonging to the cluster. That is, we have flow prediction at both fine and coarse granularities.

Urban Computing: Recently, the proliferation of big data in cities has fostered new research on urban computing [27], which aims to tackle urban challenges (such as traffic congestion and air pollution) by using data science and computing technology. A branch of research also partitions a city by major roads [25], and then studies the traffic flow between regions, for example, detecting traffic anomalies [13] and problematic urban design [28], or understanding the latent function of a region [24]. Our research is also a step towards urban computing, but different from them by the problem setting. To the best of our knowledge, in the field of urban computing, forecasting the crowd flows has never been done at the scale of a city and in a data-driven way.

8. CONCLUSION

In this paper, we propose the novel problem of predicting the flows of crowds in a city using big data, which is strategically important for traffic management and public safety. We propose a scalable prediction framework that exploits multiple complex factors affecting the crowds and decomposes the crowd flows into three components: seasonal, trend, and residual flows. Thanks to the IGMRF models and the cluster-based adjustment, our framework is robust to both noise and missing data. Experiments show that our approach is scalable and outperforms baselines significantly.

While we treat each type of trajectory data separately in our experiments due to the restriction of available data, the crowd flows can be measured as an aggregation of all types of trajectories if available (e.g., phone signals, GPS data, and subway card swiping data). Our framework still applies to such cases as is. Last but not least, if more information is available (e.g., local social events, traffic jams, or traffic accidents in each region), it can be easily incorporated into our residual model $\boldsymbol{r}$ to further improve prediction accuracy.

10. REFERENCES

[1] A. Abadi et al. Traffic flow prediction for road transportation networks with limited traffic data. *IEEE Transactions on ITS*, 16(2), 2015.

[2] M. Blangiardo and M. Cameletti. *Spatial and Spatio-temporal Bayesian Models with R-INLA*. John Wiley & Sons, 2015.

[3] P.-T. Chen, F. Chen, and Z. Qian. Road traffic congestion monitoring in social media with hinge-loss markov random fields. In *ICDM*, 2014.

[4] Z. Fan et al. CityMomentum: an online approach for crowd behavior prediction at a citywide level. In *UbiComp*, 2015.

[5] E. J. Horvitz et al. Prediction, expectation, and surprise: Methods, designs, and study of a deployed traffic forecasting service. *arXiv preprint arXiv:1207.1352*, 2012.

[6] R. J. Hyndman et al. Automatic time series for forecasting: the forecast package for R. Technical report, Monash University, Department of Econometrics and Business Statistics, 2007.

[7] Y. Kamarianakis and P. Prastacos. Spatial time series modeling: A review of the proposed methodologies. *The Regional Economics Applications Laboratory*, 2003.

[8] Y. Kamarianakis and P. Prastacos. Space–time modeling of traffic flow. *Computers & Geosciences*, 31(2), 2005.

[9] Y. Kamarianakis, W. Shen, and L. Wynter. Real-time road traffic forecasting using regime-switching space-time models and adaptive lasso. *Applied Stochastic Models in Business and Industry*, 28(4), 2012.

[10] G. Karypis and V. Kumar. Multilevel k-way partitioning scheme for irregular graphs. *Journal of Parallel and Distributed computing*, 48(1):96–129, 1998.

[11] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM review*, 51(3), 2009.

[12] Y. Li, Y. Zheng, H. Zhang, and L. Chen. Traffic prediction in a bike-sharing system. In *SIGSPATIAL*, 2015.

[13] W. Liu et al. Discovering spatio-temporal causal interactions in traffic data streams. In *KDD*, 2011.

[14] G. W. Milligan and M. C. Cooper. An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50(2):159–179, 1985.

[15] H. Rue and L. Held. *Gaussian Markov random fields: theory and applications*. CRC Press, 2005.

[16] J. Shang et al. Inferring gas consumption and pollution emission of vehicles throughout a city. In *KDD*, 2014.

[17] R. Silva, S. M. Kang, and E. M. Airoldi. Predicting traffic volumes and estimating the effects of shocks in massive transportation systems. *PNAS*, 112(18), 2015.

[18] X. Song et al. Modeling and probabilistic reasoning of population evacuation during large-scale disaster. In *KDD*, 2013.

[19] X. Song et al. Prediction of human emergency behavior and their mobility following large-scale disaster. In *KDD*, 2014.

[20] S. Sun, C. Zhang, and G. Yu. A bayesian network approach to traffic flow forecasting. *IEEE Transactions on ITS*, 7(1), 2006.

[21] Y. Wang, Y. Zheng, and Y. Xue. Travel time estimation of a path using sparse trajectories. In *KDD*, 2014.

[22] Y. Xu et al. Accurate and interpretable bayesian mars for traffic flow prediction. *IEEE Transactions on ITS*, 15(6), 2014.

[23] Y. Ye, Y. Zheng, Y. Chen, J. Feng, and X. Xie. Mining individual life pattern based on location history. 2009.

[24] J. Yuan et al. Discovering regions of different functions in a city using human mobility and POIs. In *KDD*, 2012.

[25] N. J. Yuan, Y. Zheng, and X. Xie. Segmentation of urban areas using road networks. Technical report, MSR-TR-2012-65, 2012.

[26] J. Zheng and L. M. Ni. An unsupervised framework for sensing individual and cluster behavior patterns from human mobile data. In *UbiComp*, 2012.

[27] Y. Zheng et al. Urban computing: concepts, methodologies, and applications. *TIST*, 5(3), 2014.

[28] Y. Zheng, Y. Liu, J. Yuan, and X. Xie. Urban computing with taxicabs. In *UbiComp*, 2011.