Predicting Urban Dynamics with GPS data by Multi-Order Poisson Regression Model

Yanru Chen[1,a)] Hayakawa Yuta[1] Tsubouchi Kota[2] Masamichi Shimosaka[1]

Abstract: Forecasting people flow in urban regions (urban dynamics) is playing an increasingly important role in urban planning, emergency management, public services, and commercial activities. In this paper, we propose a Multi-Order Poisson Regression Model for urban dynamics prediction based on an enriched and generalized feature representation. In the proposed method, new features are produced by employing a variety of polynomial combinations of multiple factors which greatly affect people flow (e.g., time-of-the-day, day-of-the-week, weather situation, holiday information). The results obtained from an experiment with a massive GPS dataset show that the proposed method is capable of producing models which have higher prediction accuracy compared to the state-of-the-art method.

Keywords: urban dynamics, GPS data, Poisson regression

1. Introduction

Forecasting population density transition in urban regions, i.e., urban dynamics, is playing an increasingly important role in many real-world aspects, such as urban planning [16], emergency management [12], public services [13], and commercial activities.

In conventional urban dynamics studies, it is useful to analyze urban dynamics by using data from questionnaire-based surveys [7]. However, it usually requires laborious work to conduct questionnaire-based surveys. Thanks to the rapidly popularized smart devices, a large amount of GPS data has been accumulated. Therefore, many studies on urban dynamics analysis have utilized mobility logs with GPS information without additional survey costs.

Data obtained from GPS logs has its unique spatial-temporal properties [17]. On account of the properties of GPS logs data across regions, time and days, approaches based on tensor factorization are frequently used for urban computing [2], [14], [19]. Tensor-factorization approaches have gained remarkable attention and adopted by many studies to extract urban dynamics patterns [2], [15]. Moreover, Mixture Modelling, which has high explanatory power with non-linear distributions, is another prominent approach for the analytics of human activity patterns [1], [3], [5], [6], [8]. Shimosaka et al. proposed a nonparametric Bayesian mixture model to extract the daily pattern of population transition [10]. However, in respect of urban dynamics forecasting, those approaches based on mixture modelling or factorization techniques are not capable of providing accurate prediction, which is similar to the well-known cold-start problem in recommendation systems [5].

1 Tokyo Institute of Technology
2 Yahoo! JAPAN Research
a) chen@miubiq.cs.titech.ac.jp

In contrast to previous work on urban dynamics patterns extraction, many studies based on discriminative approaches have been recently conducted and reported to achieve accurate prediction results [4], [9], [18]. Shimosaka et al. [9] proposed a predictive model called Bilinear Poisson Regression Model, which can avoid the cold-start problem by utilizing contextual information such as weather situation and calendar information.

However, when including multiple contextual information for richer feature representation, there is no systematic feature design method for urban dynamics forecasting. In this paper, we extend the idea of the Bilinear Poisson Regression Model proposed by [9], by using a novel feature encoding approach which includes polynomial combinations of multiple factors for feature representation, and finally produce new predictive models for urban dynamics, which are called Multi-Order Poisson Regression Models in this paper.

Our contributions of this paper can be summarized as follows:

(1) We propose an enriched and generalized representation of Multi-Order Poisson Regression Model for urban dynamics prediction.

(2) We propose a novel approach of feature encoding to produce new predictive models based on Multi-Order Poisson regression. Specifically, we employ various polynomial combinations of multiple factors in feature designing, and produce new predictive models for urban dynamics.

(3) We conducted experiments using a massive dataset of smartphone mobility logs with GPS information, and showed that the proposed method is capable of producing predictive models which achieve better predicition result compared to the state-of-the-art method.

(4) Predicition results of the various predictive models produced by our approach show interesting trends and could serve as an important benchmark for future work on urban dynamics prediction models.

2. Problem Setting

In this paper, the daily transition of the active population within a target urban region is modelled as urban dynamics. We divide one day into several time segments, and the active population in a target area of a certain time segment is defined by the total number of access logs in that segment. It can be assumed that the active population of each time segment follows a Poisson distribution.

We denote the number of time segments in one day as S. In a target urban region, h_s represents the active population of the s-th time segment, which is a non-negative integer. Therefore, $H = [h_1, h_2, \ldots, h_S]$ can represent a one-day active population transition. According to the prior work [9], we assume that each h_s follows a Poisson distribution, and thus the likelihood of h_s can be written as:

$$p(h_s) = \mathrm{Pois}(\mathrm{h_s}|\lambda_\mathrm{s}) = \frac{\lambda_\mathrm{s}^{\mathrm{h_s}} \mathrm{e}^{-\lambda_\mathrm{s}}}{\Gamma(\mathrm{h_s}+1)} \tag{1}$$

where λ_s is the mean parameter of the Poisson distribution. With this problem setting, in order to predict the true active population h_s, our task turns into formulating and estimating the mean parameter λ_s .

3. Single-Order Poisson Regression Models

3.1 Time-only Poisson Regression Model

It is shown by observations that the active population is greatly affected by time-of-the-day, so we can assume the mean parameter λ is regressed by the time factor with weight parameters. Specifically, λ can be formulated as a function of a time feature vector $\boldsymbol{t}$ shown as follows:

$$\lambda(\boldsymbol{t}) = \exp(\boldsymbol{t}^\top \boldsymbol{W}) \tag{2}$$

where $\boldsymbol{W}$ is a weight parameter vector, and $\boldsymbol{t}$ is an S-dimensional vector of which the s-th component corresponds to the s-th time segment.

Following the prior work [9], this time feature vector $\boldsymbol{t}$ can be formulated as an indicator function of a normal distribution, in order to smooth the time effect:

$$\boldsymbol{t} = \{t_s | t_s = \mathcal{N}(s|\tau, \sigma), s = 1, \ldots, S\}, \tag{3}$$

where $\mathcal{N}(\tau, \sigma)$ is a normal distribution with mean parameter τ corresponding to the target time segment.

However, this Time-only Poisson Regression model does not utilize contextual information such as day of the week, weather situation and holiday information, and thus cannot capture the effect of external factors on active population. **Fig. 1** illustrates the prediction result by Time-only Poisson Regression Model, compared with the ground truth transition of a weekday and a weekend in Shinjuku. It shows that Time-only Poisson Regression Model provides the same prediction result, regardless of the significant difference of population transition between the weekday and weekend.

3.2 Linear Poisson Regression Model

Prior studies show that the active population transition is closely related to many external factors such as day of the week, weather situation and holiday information [9], [11]. To model the transition in urban dynamics with these external factors, a naive approach is to simply concatenate all the factors as the feature of the model:

$$\lambda(\boldsymbol{t}, \boldsymbol{d}) = \exp([\boldsymbol{t}^\top, \boldsymbol{d}_1^\top, \boldsymbol{d}_2^\top, \ldots, \boldsymbol{d}_N^\top]\boldsymbol{W}) \tag{4}$$

where $\boldsymbol{d} = \{\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_N\}$ are external factor vectors, and each of them is encoded by One-Hot encoding method.

However, this model cannot work properly because it is not capable to handle the interaction effect between factors. For example, **Fig. 2** shows the prediction results of population transition of a weekday and a weekend in Shinjuku by Linear Poisson Regression Model. Compared by the ground truth population, we can see that although this prediction model could capture the decrease of the active population in the weekend, it cannot combine time factor and day-of-the-week factor at the same time: In Fig. 2, the predicted transition also have two commute peaks for the weekend, which means Linear Poisson Regression Model is not feasible to capture the coupling effect between time factor and day-of-the-week factor, which is conforming to the discussion in prior work [9].

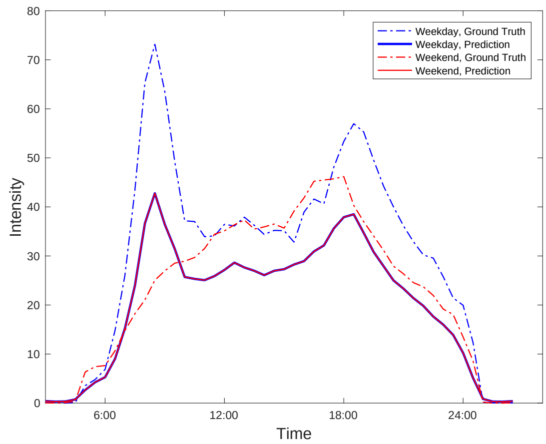

Fig. 1 Time-only Poisson Regression Model

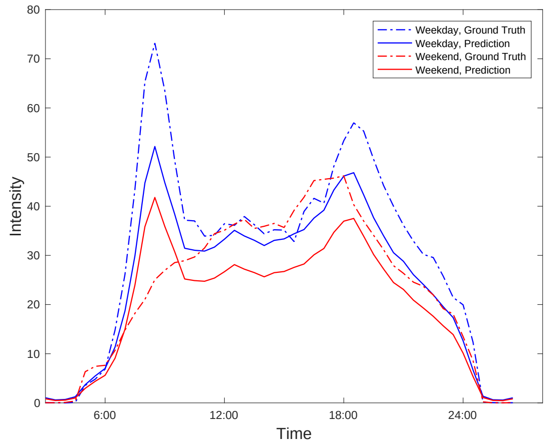

Fig. 2 Linear Poisson Regression Model

3.3 Bilinear Poisson Regression Model

In order to handle the interaction effect between time factor and external factors on active population transition, a Bilinear Poisson Regression Model is proposed in [9]:

$$\lambda(\boldsymbol{t}, \boldsymbol{d}) = \exp([\boldsymbol{d}_1^\top, \boldsymbol{d}_2^\top, \ldots, \boldsymbol{d}_N^\top]\boldsymbol{W}\boldsymbol{t}) \tag{5}$$

The weight parameter $\boldsymbol{W}$ here is in matrix representation. And previous work has shown its advantage in handling the interaction effect between time factor and external factors.

However, since all the external factors are simply concatenated, coupling effect within external factors cannot be captured.

3.4 Multilinear Poisson Regression Model

Actually, the interaction effect between all factors on active population transition can be handled by the following Multilinear Poisson Regression Model:

$$\lambda(\boldsymbol{t}, \boldsymbol{d}) = \exp(\mathrm{Vec}(\boldsymbol{t} \otimes \boldsymbol{d}_1 \otimes \boldsymbol{d}_2 \otimes \cdots \otimes \boldsymbol{d}_N)^\top \boldsymbol{W}) \tag{6}$$

where $\otimes$ is the Kronecker product operator, and $\mathrm{Vec}(\cdot)$ denotes a vectorization operation to expand the tensor-form feature into a vector, and $\boldsymbol{W}$ is a weight parameter vector. Theoretically, the idea of this Multilinear Poisson Regression Model is similar to the model proposed in [11] which adopted a tensor-form for representing weight parameter.

On the one hand, Multilinear Poisson Regression Model is hopefully to capture the interaction effect between all factors, but on the other hand, combining all the factors by Kronecker product would increase the feature dimension and greatly worsen the zero-frequency problem. For example, the rainy situation might be less common in some areas compared to other weather conditions, and a rainy Wednesday would be more uncommon. After combining weather factor, day-of-the-week factor and time factor, it would result in some extremely rare situation (e.g., raining at 13:00 on Wednesday) which might not appear in the training dataset, and thus the model is not feasible to provide accurate predictions.

Fig. 3 illustrates the prediction result of a population transition of a rainy day and a sunny day in Shinjuku. We can see that, compared with prediction result of the sunny day, the performance of this model exhibits a dramatic decline during the period from 10:00 to 21:00 in the rainy day, which is probably resulted from the worsened zero-frequency problem.

4. Multi-Order Poisson Regression Model

In respect of the polynomial terms composed by the independent variables $\{\boldsymbol{d}_0, \boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_N\}$ (time factor $\boldsymbol{t}$ is denoted as $\boldsymbol{d}_0$) of the regression model, all the models discussed above only include terms with the same order. Furthermore, these single-order regression models indicate a trend that models adopted low-order terms (e.g., Linear Poisson regression Model) is not feasible to capture the higher-order coupling effect, while models with high-order terms (e.g., Multilinear Poisson Regression Model) only focus on high-order interaction of independent variables (factors), and thus greatly worsen the zero-frequency problem.

The basic idea of the proposed Multi-Order Poisson Regression Model is to combine low-order pattern and high-order interaction effect simultaneously.

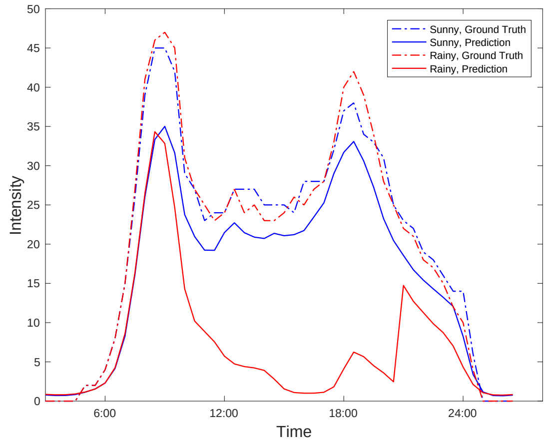

Fig. 3 Multilinear Poisson Regression Model

4.1 The Enriched and Generalized Representation

Analogous to multiple linear regression, all of the Single-Order Poisson regression models discussed above with independent variables $\{\boldsymbol{d}_0, \boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_N\}$ can be extended into an enriched and generalized representation:

$$\begin{aligned} &\ln \lambda(\boldsymbol{t}, \boldsymbol{d}) \\ &= W_0 + \sum_{i=0}^{N} \boldsymbol{d}_i^\top \boldsymbol{W}_1^{(i)} + \sum_{i=0}^{N}\sum_{j=i}^{N} \mathrm{Vec}(\boldsymbol{d}_i \otimes \boldsymbol{d}_j)^\top \boldsymbol{W}_2^{(i,j)} \\ &+ \sum_{i=0}^{N}\sum_{j=i}^{N}\sum_{k=j}^{N} \mathrm{Vec}(\boldsymbol{d}_i \otimes \boldsymbol{d}_j \otimes \boldsymbol{d}_k)^\top \boldsymbol{W}_3^{(i,j,k)} + \ldots \end{aligned} \tag{7}$$

where W_0 is a constant, $\boldsymbol{W}_1^{(i)}, \boldsymbol{W}_2^{(i,j)}, \boldsymbol{W}_3^{(i,j,k)}$ are all weight parameter vectors. Specifically, $\boldsymbol{W}_1$ is composed of weight parameter vectors corresponding to the first order terms, while components of $\boldsymbol{W}_2$ are weight parameter vectors of the second order terms, and so forth. Therefore, this generalized model can include infinite terms of the independent variables with different orders.

Actually, each of the Single-Order Poisson regression models discussed above (Time-only Poisson Regression Model, Linear Poisson Regression Model, Bilinear Poisson Regression Model and Multilinear Poisson Regression Model) can be treated as a special case of this generalized model, with a certain combination of terms with the same orders.

In practice, we do not consider all the terms of this generalized model due to the concern of computation cost. In this paper, we explore a number of combinations of different order terms for feature encoding, from which new predictive models are obtained and are generally called Multi-order Poisson regression models in our work. The advantage of these method is that the combination of low-order and high-order terms could capture the higher-order coupling effect as well as lower-order patterns, and the zero-frequency problem could thus be resolved, since the behavior patterns learned from lower-order terms could be utilized for compensation by this model.

4.2 Practical Feature Encoding Approach

As mentioned previously, features for Multi-Order Poisson Regression model can be produced by various polynomial combinations of factors (including the time factor and external factors).

In practice, we first determined several fundamental models and then create various combinations and derivatives based on these models.

Fundamental models used in this experiment include the 4 Single-Order Poisson Regression Models mentioned above and an External-only Poisson Regression Model:

(1) **(Time-only):**
Time-only Poisson Regression Model, defined by Eq. (2)

(2) **(Linear):**
Linear Poisson Regression Model, defined by Eq. (4)

(3) **(Bilinear):**
Bilinear Poisson Regression Model, defined by Eq. (5)

(4) **(Multilinear):**
Multilinear Poisson Regression Model, defined by Eq. (6)

(5) **(External-only):**
External-only Poisson Regression Model only includes external factors, simply concatenated into its feature. Although this model alone may not achieve satisfying prediction result, it can be used as a baseline for comparison as well as a fundamental model for combination. It is defined as follows:

$$\lambda(\boldsymbol{d}) = \exp([\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_N]^\top \boldsymbol{W}) \tag{8}$$

A variety of new Multi-Order Poisson regression Models can be obtained from the derivatives based on the above fundamental models, by employing some of the following operations:

- **(Simple Combination):**
Simply combine features from some of the fundamental models into the feature of the new predictive model. For example, combination of **Linear** and **Bilinear** model, denoted as **Linear+Bilinear**, can be written as follows:

$$\begin{aligned}&\ln \lambda(\boldsymbol{t}, \boldsymbol{d}) \\ &= [\boldsymbol{t}^\top, \boldsymbol{d}_1^\top, \ldots, \boldsymbol{d}_N^\top, \mathrm{Vec}^\top([\boldsymbol{d}_1^\top, \ldots, \boldsymbol{d}_N^\top] \otimes \boldsymbol{t})]\boldsymbol{W}\end{aligned} \tag{9}$$

- **(Adding a Constant Feature):**
Add a constant into the combination of features from some of the fundamental model. For example, adding a constant feature to **Linear+Bilinear** model, denoted as **Linear+Bilinear+Constant**, can be written as follows:

$$\begin{aligned}&\ln \lambda(\boldsymbol{t}, \boldsymbol{d}) \\ &= [1, \boldsymbol{t}^\top, \boldsymbol{d}_1^\top, \ldots, \boldsymbol{d}_N^\top, \mathrm{Vec}^\top([\boldsymbol{d}_1^\top, \ldots, \boldsymbol{d}_N^\top] \otimes \boldsymbol{t})]\boldsymbol{W}\end{aligned} \tag{10}$$

In fact, a model with only a constant feature degenerates into the simple Poisson Regression Model. Therefore, we expect that the constant feature would be helpful to capture the basic rate of the active population transition behavior.

- **(Low-Rank Decomposition):**
The weight parameter vectors of all of the five fundamental models can be rewritten in matrix representation. Based on the assumption that the rank of weight parameter matrix is prone to decrease, we adopt a low-rank decomposition to achieve rank reduction. Be noted that since the intention of this operation is to reduce rank and make the parameter learning more stable, we did not conduct this decomposition on the models with quasi-diagonal weight matrix.
For example, the **Linear+Bilinear+Constant** model with Low-Rank decomposition can be written as follows:

$$\begin{aligned}\ln \lambda(\boldsymbol{t}, \boldsymbol{d}) &= [1, \boldsymbol{t}^\top, \boldsymbol{d}_1^\top, \ldots, \boldsymbol{d}_N^\top, \mathrm{Vec}^\top([\boldsymbol{d}_1^\top, \ldots, \boldsymbol{d}_N^\top] \otimes \boldsymbol{t})]\boldsymbol{W} \\ &= [1, \boldsymbol{d}_1^\top, \ldots, \boldsymbol{d}_N^\top]\boldsymbol{W}[1, \boldsymbol{t}]\end{aligned} \tag{11}$$

For low rank decomposition, the weight matrix $\boldsymbol{W} \in \mathbb{R}^{(M+1)\times(S+1)}$ shown above can be assumed to be a product with two low-rank matrices, $\boldsymbol{U} \in \mathbb{R}^{(M+1)\times K}$ and $\boldsymbol{V} \in \mathbb{R}^{(S+1)\times K}$. Note that K and M satisfy $K \ll M$. And the above equation turns into:

$$\begin{aligned}\ln \lambda(\boldsymbol{t}, \boldsymbol{d}) &= [1, \boldsymbol{d}_1^\top, \ldots, \boldsymbol{d}_N^\top]\boldsymbol{W}[1, \boldsymbol{t}] \\ &= [1, \boldsymbol{d}_1^\top, \ldots, \boldsymbol{d}_N^\top]\boldsymbol{U}\boldsymbol{V}^\top[1, \boldsymbol{t}]\end{aligned} \tag{12}$$

This shrinkage not only makes it more stable for learning the parameters, but also helpful for understanding the characteristics of urban dynamics.
In this paper, the model is optimized by maximizing the log likelihood against training data. Given L days of training data

$$\begin{aligned}&\boldsymbol{D} = (h^{(l,s)}, \boldsymbol{t}^{(l,s)}, \boldsymbol{d}^{(l,s)}), \\ &\boldsymbol{d}^{(l,s)} = [1, \boldsymbol{d}_1^{(l,s)\top}, \ldots, \boldsymbol{d}_N^{(l,s)\top}], \\ &l = 1, \ldots, L, s = 1, \ldots, S,\end{aligned} \tag{13}$$

then the log likelihood of the data is written as:

$$\begin{aligned}&\ln L(\boldsymbol{U}, \boldsymbol{V}) \\ &= \sum_l \sum_s \ln \mathrm{Pois}(h^{(l,s)} | \lambda(\boldsymbol{t}, \boldsymbol{d})) \\ &\propto \sum_l \sum_s \{h^{(l,s)} \boldsymbol{d}^{(l,s)} \boldsymbol{U}\boldsymbol{V}^\top [1, \boldsymbol{t}^{(l,s)}] - \exp(\boldsymbol{d}^{(l,s)} \boldsymbol{U}\boldsymbol{V}^\top [1, \boldsymbol{t}^{(l,s)}])\}\end{aligned}$$

A regularization term can be added to prevent overfitting problem, then the objective function is as follows:

$$\hat{\boldsymbol{U}}, \hat{\boldsymbol{V}} = \underset{\boldsymbol{U}, \boldsymbol{V}}{\arg\min} \{-\ln L(\boldsymbol{U}, \boldsymbol{V}) + \Omega(\boldsymbol{U}, \boldsymbol{V})\} \tag{14}$$

where $\Omega(\boldsymbol{U}, \boldsymbol{V}) = \gamma \|\boldsymbol{U}\|_2^2 + \gamma \|\boldsymbol{V}\|_2^2$ is the regularization term with a hyper parameter γ (> 0). We use L-BFGS method to alternatively optimize parameter matrix $\boldsymbol{U}$ and $\boldsymbol{V}$.

5. Experiment

To order to evaluate the predictive performance of the various Multi-Order Poisson regression models produced by the proposed method, we conducted an experiment with a massive dataset with GPS logs.

5.1 Dataset

The dataset of this experiment is extracted from a large number of mobility logs obtained from the disaster alert mobile application released from Yahoo! JAPAN[*1]. A mobility log was recorded only when mobile devices were moving instead of being stable, which is suitable for analysis of the active population transition. As for target regions, 300 urban area are selected. The size of the target areas was set to $900 \times 900 m^2$, and we count the number of access logs as active population within each target region

[*1] http://emg.yahoo.co.jp/

and each time segment of one year (from July 1, 2013 to June 30, 2014).

In this experiment, the width of each time segment is set to 30 minutes, and the fundamental period is 24 hours, starting from 3:00 and ended by 3:00 of the next day. We adopted the same experiment setting as that of the prior study [9] which proposed the Bilinear Poisson Regression Model.

5.2 Performance Measurement

The following two criteria were used as performance measurement for comparison of different Multi-Order Regression Models: mean absolute error (MAE), and rank number (Rank).

Give the test data and prediction value:
$h^{[l,s]}, \hat{\lambda}^{[l,s]}, (l = 1 \ldots L, s = 1 \ldots S)$,

The definitions of these two criteria are shown in Table 1

Table 1 Performance Measurement

Criteria	Definition
MAE	MAE = $\mathbb{E}(\lvert h^{[l,s]} - \hat{\lambda}^{[l,s]}\rvert)$, where $\mathbb{E}(x)$ is the expectation of x
Rank	For each target region, a five-fold cross validation(CV) was conducted for each model, and the average of MAE over all CVs is obtained for each model. Then all the models are ranked by average MAE on each target region.

5.3 Models For Comparison

In this part, we will introduce the Multi-Order Models produced by the proposed method. First, we selected three external factors along with time factor for feature design. Formulation of time factor has been introduced in the previous part of this paper. As for external factors, we selected weather situation, day-of-the-week and Is-Holiday-Or-Not information as external factors.

Day-of-the-week factor is denoted as a 7-dimentinoal vector $\boldsymbol{d}_1$ with one-hot encoding. And Is-Holiday-Or-Not factor is a 2-dimensional vector $\boldsymbol{d}_2$.

Weather data were collected from the Japan Meteorological Agency's website*2. The category of the weather was 1 to 4, {sunny(1), cloudy(2), rainy(3), or rough weather(4)}. Denotes weather factor as $\boldsymbol{d}_3$ with one-hot encoding, and then $\boldsymbol{d}_3$ is a 4-dimensional vector.

By employing the feature encoding approach proposed in Section **4.2**, we produced 28 predictive models and compared their performance in this experiment. A list of these models with formulation details is shown in **Fig. 4**.

5.4 Results

Fig. 5 exhibits the distribution of Rank for each model. We can see that the model **LowRank(Linear+Bilinear+Constant)**, which is generated by the proposed approach, is ranked first on 246 out of 300 regions, while **LowRank(Bilinear)** model ranked 4th on 240 out of 300 regions.

Table 2 shows the performance of 28 predictive models. In this table, MAE_Mean is the average MAE value over all 5 CVs and across all 300 target regions of each model. Similarly, Rank_Mean is the average Rank for each model.

Time-only:	$\ln \lambda_1(\mathbf{t}) = \mathbf{t}^\top \mathbf{W}_1$
Linear:	$\ln \lambda_2(\mathbf{d}, \mathbf{t}) = [\mathbf{d}_1^\top, \mathbf{d}_2^\top, \mathbf{d}_3^\top, \mathbf{t}^\top]\mathbf{W}_2$
Bilinear:	$\ln \lambda_3(\mathbf{d}, \mathbf{t}) = Vec^\top([\mathbf{d}_1^\top, \mathbf{d}_2^\top, \mathbf{d}_3^\top] \otimes \mathbf{t})\mathbf{W}_3$
Multilinear:	$\ln \lambda_4(\mathbf{d}, \mathbf{t}) = Vec^\top(\mathbf{d}_1 \otimes \mathbf{d}_2 \otimes \mathbf{d}_3 \otimes \mathbf{t})\mathbf{W}_4$
External-only:	$\ln \lambda_5(\mathbf{d}) = [\mathbf{d}_1^\top, \mathbf{d}_2^\top, \mathbf{d}_3^\top]W_5$
Linear+Bilinear:	$\ln \lambda_6(\mathbf{d}, \mathbf{t}) = \ln \lambda_2(\mathbf{d}, \mathbf{t}) + \ln \lambda_3(\mathbf{d}, \mathbf{t})$
Linear+Multilinear:	$\ln \lambda_7(\mathbf{d}, \mathbf{t}) = \ln \lambda_2(\mathbf{d}, \mathbf{t}) + \ln \lambda_4(\mathbf{d}, \mathbf{t})$
Bilinear+Multilinear:	$\ln \lambda_8(\mathbf{d}, \mathbf{t}) = \ln \lambda_3(\mathbf{d}, \mathbf{t}) + \ln \lambda_4(\mathbf{d}, \mathbf{t})$
Linear+Bilinear+Multilinear:	$\ln \lambda_9(\mathbf{d}, \mathbf{t}) = \ln \lambda_2(\mathbf{d}, \mathbf{t}) + \ln \lambda_3(\mathbf{d}, \mathbf{t}) + \ln \lambda_4(\mathbf{d}, \mathbf{t})$
Time-only+Constant:	$\ln \lambda_{10}(\mathbf{t}) = [1, \mathbf{t}^\top]\mathbf{W}_{10}$
Linear+Constant:	$\ln \lambda_{11}(\mathbf{d}, \mathbf{t}) = [1, \mathbf{d}_1^\top, \mathbf{d}_2^\top, \mathbf{d}_3^\top, \mathbf{t}^\top]\mathbf{W}_{11}$
Bilinear+Constant:	$\ln \lambda_{12}(\mathbf{d}, \mathbf{t}) = [1, Vec^\top([\mathbf{d}_1^\top, \mathbf{d}_2^\top, \mathbf{d}_3^\top] \otimes \mathbf{t})]\mathbf{W}_{12}$
Multilinear+Constant:	$\ln \lambda_{13}(\mathbf{d}, \mathbf{t}) = [1, Vec^\top(\mathbf{d}_1 \otimes \mathbf{d}_2 \otimes \mathbf{d}_3 \otimes \mathbf{t})]\mathbf{W}_{13}$
External-only+Constant:	$\ln \lambda_{14}(\mathbf{d}) = [1, \mathbf{d}_1^\top, \mathbf{d}_2^\top, \mathbf{d}_3^\top]\mathbf{W}_{14}$
Linear+Bilinear+Constant:	$\ln \lambda_{15}(\mathbf{d}, \mathbf{t}) = \ln \lambda_2(\mathbf{d}, \mathbf{t}) + \ln \lambda_{12}(\mathbf{d}, \mathbf{t})$
Linear+Multilinear+Constant:	$\ln \lambda_{16}(\mathbf{d}, \mathbf{t}) = \ln \lambda_2(\mathbf{d}, \mathbf{t}) + \ln \lambda_{13}(\mathbf{d}, \mathbf{t})$
Bilinear+Multilinear+Constant:	$\ln \lambda_{17}(\mathbf{d}, \mathbf{t}) = \ln \lambda_3(\mathbf{d}, \mathbf{t}) + \ln \lambda_{13}(\mathbf{d}, \mathbf{t})$
Linear+Bilinear+Multilinear+Constant:	$\ln \lambda_{18}(\mathbf{d}, \mathbf{t}) = \ln \lambda_2(\mathbf{d}, \mathbf{t}) + \ln \lambda_3(\mathbf{d}, \mathbf{t}) + \ln \lambda_{13}(\mathbf{d}, \mathbf{t})$
Bilinear+Time-only:	$\ln \lambda_{19}(\mathbf{d}, \mathbf{t}) = \ln \lambda_3(\mathbf{d}, \mathbf{t}) + \ln \lambda_1(\mathbf{t})$
Bilinear+Time-only+Constant:	$\ln \lambda_{20}(\mathbf{d}, \mathbf{t}) = \ln \lambda_3(\mathbf{d}, \mathbf{t}) + \ln \lambda_{10}(\mathbf{t})$
Bilinear+External-only:	$\ln \lambda_{21}(\mathbf{d}, \mathbf{t}) = \ln \lambda_3(\mathbf{d}, \mathbf{t}) + \ln \lambda_5(\mathbf{d})$
Bilinear+External-only+Constant:	$\ln \lambda_{22}(\mathbf{d}, \mathbf{t}) = \ln \lambda_3(\mathbf{d}, \mathbf{t}) + \ln \lambda_{14}(\mathbf{d})$
LowRank(Bilinear):	$\ln \lambda_{23}(\mathbf{d}, \mathbf{t}) = [\mathbf{d}_1^\top, \mathbf{d}_2^\top, \mathbf{d}_3^\top]\mathbf{U}_{23}\mathbf{V}_{23}\mathbf{t}$
LowRank(Multilinear):	$\ln \lambda_{24}(\mathbf{d}, \mathbf{t}) = Vec^\top(\mathbf{d}_1 \otimes \mathbf{d}_2 \otimes \mathbf{d}_3)\mathbf{U}_{24}\mathbf{V}_{24}\mathbf{t}$
LowRank(Bilinear+Multilinear):	$\ln \lambda_{25}(\mathbf{d}, \mathbf{t}) = [\mathbf{d}_1^\top, \mathbf{d}_2^\top, \mathbf{d}_3^\top, Vec^\top(\mathbf{d}_1 \otimes \mathbf{d}_2 \otimes \mathbf{d}_3)]\mathbf{U}_{25}\mathbf{V}_{25}\mathbf{t}$
LowRank(Bilinear+Time-only):	$\ln \lambda_{26}(\mathbf{d}, \mathbf{t}) = [1, \mathbf{d}_1^\top, \mathbf{d}_2^\top, \mathbf{d}_3^\top]\mathbf{U}_{26}\mathbf{V}_{26}\mathbf{t}$
LowRank(Bilinear+External-only):	$\ln \lambda_{27}(\mathbf{d}, \mathbf{t}) = [\mathbf{d}_1^\top, \mathbf{d}_2^\top, \mathbf{d}_3^\top]\mathbf{U}_{27}\mathbf{V}_{27}[1, \mathbf{t}]$
LowRank(Linear+Bilinear+Constant):	$\ln \lambda_{28}(\mathbf{d}, \mathbf{t}) = [1, \mathbf{d}_1^\top, \mathbf{d}_2^\top, \mathbf{d}_3^\top]\mathbf{U}_{28}\mathbf{V}_{28}[1, \mathbf{t}]$

Fig. 4 List of Models for Comparison

Table 2 Performance Comparison

Models		MAE_Mean	Rank_Mean
Time-only	Plain	**166.29**	**21.9**
	+ C	167.25	23.3
Linear	Plain	134.95	21.7
	+ C	**134.71**	**20.8**
Bilinear	Plain	118.37	15.8
	+ C	116.95	10.4
	LR	**115.60**	**4.2**
Multilinear	Plain	250.18	25.8
	+ C	156.65	24.4
	LR	**130.87**	**21.4**
External-only	Plain	331.78	27.9
	+ C	**331.58**	**27.0**
Linear+Bilinear	Plain	116.30	7.9
	+ C	116.07	6.5
	+ C & LR	**115.07**	**1.3**
Linear+Multilinear	Plain	119.87	17.4
	+ C	**119.61**	**16.5**
Bilinear+Multilinear	Plain	122.52	19.1
	+ C	118.17	14.9
	LR	**117.64**	**13.6**
Linear+Bilinear+Multilinear	Plain	117.26	11.7
	+ C	**116.97**	**9.9**
Bilinear+Time-only	Plain	116.16	7.5
	+ C	115.80	5.1
	LR	**115.16**	**2.0**
Bilinear+External-only	Plain	117.41	13.0
	+ C	117.21	11.7
	LR	**115.46**	**3.2**

The first column shows the combinations of 5 fundamental Single-Order models, the second column shows the operation based on such combinations: "Plain" here means plain model which is same as shown in the first column, "+C" means combined with a constant feature, "LR" means conducted Low-Rank decomposition on weight parameter matrix, "+C & LR" means conducted Low-Rank decomposition based on the model which has included a constant into its feature.

We can conclude from this table that:

- **LowRank(Linear+Bilinear+Constant)**, **LowRank(Bilinear+Time-only)**, **LowRank(Bilinear+External-only)**, which are produced by the proposed method, all out-

*2 http://www.data.jma.go.jp/obd/stats/etrn

Model	MAE_Mean	Rank_Mean	Distribution of Rank																											
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
LowRank(Linear + Bilinear + Constant)	**115.07**	**1.29**	246	43	3	3	0	3	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LowRank(Bilinear + Time-only)	115.16	2.01	36	242	14	4	2	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LowRank(Bilinear + External-only)	115.46	3.22	2	4	252	25	8	7	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LowRank(Bilinear)	115.60	4.22	4	1	9	240	25	8	8	4	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bilinear + Time-only + Constant	115.80	5.14	7	3	14	17	200	34	12	4	4	1	2	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Linear + Bilinear + Constant	116.07	6.49	0	1	3	4	24	161	71	18	8	3	2	3	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
Bilinear + Time-only	116.16	7.50	0	1	3	6	24	54	63	61	50	29	8	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Linear + Bilinear	116.30	7.94	0	0	0	0	0	4	87	164	30	10	1	0	3	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
Bilinear + Constant	116.95	10.38	0	0	0	0	0	0	0	0	100	71	81	29	8	9	1	0	0	0	0	0	1	0	0	0	0	0	0	0
Linear + Bilinear + Multilinear + Constant	116.97	9.86	0	0	0	0	10	18	41	31	50	44	33	30	18	6	3	3	5	3	2	1	0	2	0	0	0	0	0	0
Bilinear + External-only + Constant	117.21	11.74	0	0	0	0	0	1	0	0	9	56	61	99	43	20	8	2	0	0	0	0	0	0	0	1	0	0	0	0
Linear + Bilinear + Multilinear	117.26	11.69	0	0	0	0	0	1	8	11	22	68	41	42	56	25	8	2	4	6	3	1	2	0	0	0	0	0	0	0
Bilinear + External-only	117.41	13.04	0	0	0	0	0	0	0	0	1	4	35	59	105	51	37	5	1	1	0	0	0	1	0	0	0	0	0	0
LowRank(Bilinear + Multilinear)	117.64	13.56	0	0	1	0	1	2	1	3	23	11	27	21	35	46	73	31	9	5	2	9	0	0	0	0	0	0	0	0
Bilinear + Multilinear + Constant	118.17	14.91	0	0	0	0	0	0	1	0	0	0	1	7	22	104	77	68	3	2	6	6	1	2	0	0	0	0	0	0
Bilinear	118.37	15.81	0	0	0	0	1	2	5	1	1	0	3	8	5	20	58	84	29	83	0	0	0	0	0	0	0	0	0	0
Linear + Multilinear + Constant	119.61	16.47	0	0	0	0	1	0	0	1	0	0	1	0	1	10	24	90	139	29	3	0	1	0	0	0	0	0	0	0
Linear + Multilinear	119.87	17.43	0	0	0	1	0	0	0	0	0	1	0	1	0	0	7	11	108	168	0	2	1	0	0	0	0	0	0	0
Bilinear + Multilinear	122.52	19.13	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	270	5	14	1	7	0	0	0	0	0
LowRank(Multilinear)	130.87	21.43	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	155	15	39	6	84	0	0	0	0
Linear + Constant	134.71	20.77	0	3	1	0	1	2	0	1	1	1	0	1	1	5	0	0	0	1	1	32	124	79	41	5	0	0	0	0
Linear	134.95	21.65	3	1	0	0	2	1	1	0	0	0	2	0	2	2	3	0	1	0	1	4	33	123	74	46	1	0	0	0
Multilinear + Constant	156.65	24.42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	80	6	211	1	0	0
Time-only	166.29	21.91	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	11	79	48	13	61	76	9	0	0	0
Time-only + Constant	167.25	23.26	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	6	58	40	29	78	79	9	0	0
Multilinear	250.18	25.79	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	2	4	0	282	1	7
External-only + Constant	331.58	27.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	283	9
External-only	331.78	27.95	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	284

Fig. 5 Distribution of Rank of each model

performed the state-of-the-art model **LowRank(Bilinear)** in this experiment.

- Among most of the models compared in this experiment, adding a constant term in feature encoding could be helpful to improve prediction performance. This is probably because models including the constant term could capture the basic constant behavior of population transition, and thus help to obtain more accurate prediction results.
- Compared to the original plain model, those models using Low-rank approximation achieve better performance.

6. Conclusion

In this paper, We propose a Multi-Order Poisson Regression Model with an enriched and generalized representation for urban dynamics prediction. We have explored a variety of polynomial combinations of multiple factors for feature designing. Experiments are conducted using a massive dataset of smartphone mobility logs with GPS information, and the results showed that the proposed method is capable of producing predictive models which achieve better prediction result compared to the state-of-the-art model.

The spatial properties of mobility data could be taken into account for further study. In this work, we treated each region as an independent learning task. Multi-task learning techniques could be employed to improve predicition performance (especially for regions have low active population intensity), by considering the spatial correlation among tasks. Furthermore, inductive approaches could be developed to automatically produce the best-performed combinations based on the enriched and generalized representation proposed by this paper.

References

[1] Adams, R., Murray, I., and MacKay, D. : *Tractable nonparametric Bayesian inference in Poisson processes with Gaussian process intensities.*, In Proc. of ICML ’09 916 (2009).

[2] Fan, Z., Song, X., and Shibasaki, R.: CitySpectrum: A non-negative tensor factorization approach. *In Proc. of UbiComp.*, (2014).

[3] Ihler, A., and Smyth, P.: Learning time-intensity profiles of human activity using non-parametric Bayesian models. *In Proc. of NIPS ’07*, 625632 (2007)

[4] Jiang, R., Song, X., Fan, Z., Xia, T., Chen, Q., Shibasaki, R.: Deep ROI-based modeling for urban human mobility prediction. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2(1) (2018).

[5] Lika, B., Kolomvatsos, K., and Hadjiefthymiades, S.: Cold start problem,recommender systems. *Expert Systems with Applications.*, (2014).

[6] Nishi, K., Tsubouchi, K., and Shimosaka, M.: Extracting land-use patterns using location data from smartphones. *In Proc. of URB-IOT ’14*, 3843 (2014).

[7] Sekimoto, Y., Shibasaki, R., Kanasugi, H., Usui, T., and Shimazaki,Y.: Pflow: Reconstructing people flow recycling large-scale social survey data. *IEEE Pervasive Computing*, 10(4) (2011).

[8] Shimosaka, M., Ishino, T., Noguchi, H., Sato, T., and Mori, T.: Detecting human activity profiles with Dirichlet enhanced inhomogeneous Poisson processes. *Proc. of ICPR ’10* 43844387 (2010).

[9] Shimosaka, M., Maeda, K., Tsukiji, T., Tsubouchi, K.: Forecasting urban dynamics with mobility logs by bilinear Poisson regression. *Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing* (pp. 535-546). ACM. (2015).

[10] Shimosaka, M., Tsukiji, T., Tominaga, S., Tsubouchi, K.: Coupled hierarchical Dirichlet process mixtures for simultaneous clustering and topic modeling. *In Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, (pp. 230-246). Springer, Cham. (2016).

[11] Shimosaka, M., Tsukiji, T., Wada, H., Tsubouchi, K.: Predictive population behavior analysis from multiple contexts with multilinear poisson regression. *In Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, (pp. 504-507). ACM. (2018).

[12] Song, X., Zhang, Q., Sekimoto, Y., Horanont, T., Ueyama, S., and Shibasaki, R.: Modeling and probabilistic reasoning of population evacuation during large-scale disaster. *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1231–1239 (2013)

[13] Sun, L., Lee, D.-H., Erath, A., and Huang, X.: Using smart card data to extract passenger’s spatio-temporal density and train’s trajectory of MRT system. *Proc. of UrbComp’12*, 2154–2158 (2012).

[14] Takeuchi, K., Tomioka, R., Ishiguro, K., Kimura, A., Sawada, H.: Non-negative multiple tensor factorization. *In Proc. of ICDM*, (2013).

[15] Yao, Z.; Fu, Y.; Liu, B.; Hu, W.; and Xiong, H.: Representing urban functions through zone embedding with human mobility patterns. *In Proc. of IJCAI.*, (2017)

[16] Yuan, J., Zheng, Y., Xie, X.: Discovering regions of different functions in a city using human mobility and POIs. *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 186-194 (2012).

[17] Zhang, J., Zheng, Y., Qi, D., Li, R., Yi, X.: DNN-based prediction model for spatio-temporal data. *In Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, (p. 92). ACM. (2016).

[18] Zhang, J., Zheng, Y., Qi, D.: Deep spatio-temporal residual networks for citywide crowd flows prediction. *In Proc. of AAAI.*, (2017)

[19] Zheng, Y., Liu, T., Wang, Y., Zhu, Y., Liu, Y., Chang,E.: Diagnosing new york city ' s noises with ubiquitous data. *In Proc. of UbiComp.*, (2014).