Modeling the Interaction Coupling of Multi-View Spatiotemporal Contexts for Destination Prediction

Kunpeng Liu * Pengyang Wang◊ Jiawei Zhang † Yanjie Fu+‡ Sajal K. Das$

Abstract

Bike-Sharing Systems (BSSs) are being introduced to more and more cities recently, and therefore they have generated huge amounts of data. Mobike is a station-less BSS which is suffering from the chaotic parking problem. To solve this problem, it is necessary to predict where the bikes are going. Traditional works dealing with destination prediction mainly focus on station-based BSSs, and they merely leverages context-aware information technically. Thus it is naturally promising to investigate how to improve the destination prediction of station-less bikes by context information. To that end, in this paper, we develop a multi-view machine (MVM) method, by incorporating the context information from Point of Interest (POI) data and human mobility data into destination prediction. Specifically, we first describe three different views, namely start position, start time and destination by features extracted from POI data and human mobility data. Then, we capture the relationship between these three views' interactions and the trip's possibility by a multi-view machine. Finally, since multi-view machine contains too many parameters to be optimized, we leverage tensor factorization (TF) to reduce the computation costs. The experimental results show that the model can effectively capture the potential relationship of three views with trip's possibility and the approach is thus much more effective than traditional prediction methods for destination.

1 Introduction

People taking public tansportations may often encounter the 'last mile problem'[1], because most living and working places don't locate exactly adjacent to bus stops or subway stations. Recently, Bike Sharing System (BSS) has become a healthy and enrivornment-friendly traffic solution to overcome the inconveniences of the last walking mile. Mobike is a station-less BSS operated by one of China's top-funded bike-sharing companies and has become more and more popular in many big cities of China [26].

Even though the station-less property brings convenience for sharing bikes riders, it can also arouse several non-trivial new problems for the public traffic. For example, when large sum of bikes gather in a specific area without going out, the road can be blocked and further influence the normal public traffic [18]. Therefore, from the perspectives of city planners and the companies operating the Bike Sharing Systems, it is indispensable to predict where the bikes would go from specific region in specific time, which is the first step to solve traffic problems caused by sharing bikes.

However, due to the complex nature of Bike Sharing Systems, destination prediction is never an easy task. Three unique challenges arise in achieving this goal: (1) spatial and temporal view description: how to derive mathmatical representations to describe origin, destination and time of the bike trip; (2) interaction coupling: how to build a model to capture the interactions between spatial and temporal views; (3) parameter learning: how to obtain the pramaters in the model by optimization. To tackle these challenges, in this paper, we propose a multi-view spatiotemporal based method which considers the interation coupling effects between different views.

First, people in urban areas often leave from one POI and visit another POI for daily activities, which means that POI is an ideal indicator to describe spatial view. Meanwhile, the frequency of visited POI by some traffic modes, e.g., taxi, could reveal the whole city's mobility in particular time windows. Thus, taxi traces incorporated with POI data could be used to describe temporal view. If we describe every origin, destination, and trip time seperately, the data would be very sparse. To solve this problem, we partition the city into small grid cells and the whole day into small time windows, then we formulate their mathmatical representations and aggregate the raw data according to the coarse-grained origins, destinations and time. Then, we can transform the task of describing the specific origin, destination and time into describing their corresponding

*◊+$Department of Computer Science, Missouri S&T, MO, Rolla, USA. Email: {kl6ph, pwqt3, fuyan sdas}@mst.edu.

†IFM Lab, Department of Computer Science, Florida State University, FL, Tallahassee, USA. Email: jzhang@cs.fsu.edu.

‡Contact Author.

grid cells and time windows.

Second, after getting the mathmatical representation of the three views, we attempt to capture their potential connections with each other. Most frequently used algorithms are support vector machine and factorization machine [21]. Those two algorithms are efficient and effective in most cases, but in our case, they may suffer from overfitting problems because they would combine view features together instead of dealing with them seperately. We propose to adapt a multi-view machine (MVM) to model views' connections by taking all interactons into account and learn the corresponding weights through optimization.

Third, since our model considers all the possible interactions between view features, and the interactions would explode with the increase of the number of features from the views. Inspired by tensor factorization [16], we reorganize the parameters into a three dimensional tensor and factorize this tensor thereafter. This factorization procedure in the algorithm not only reduces the number of parameters to be optimized, but also captures the relationship between them.

To summarize, in this paper, we propose to adapt a multi-view machine model to learn the interactions between a trip's origin, destination and time. Specifically, the followings are our three main contributions: (1) We leverage context information, i.e., POI data and taxi trace data to build mathmatical representations of spatial and temporal views. (2) We propose to adapt a multi-view machine (MVM) model to capture the interactions between three views. (3) We leverage tensor factorization to reduce the number of parameters to be optimized as well as capturing their relationships.

2 Preliminaries

We first introduce some important definitions and formulate the problem. We then provide an overview of the proposed method.

2.1 Definitions and Problem Statement

Definition 1: *(Trip):* A trip of a Mobike is defined as $Trip = \{\text{origin,destination,time}\}$, where the 'origin' and 'destination' are the starting and ending positions respectively, which are denoted by their GPS coordinates, and 'time' is the time when the trip starts.

Definition 2: *(View):* A view is an aspect from which you can describe or understand the target object. In this paper, a Mobike trip could be described by three different views, i.e., the feature vectors of origin, destination, and time. While each view partially describes the trip from a particular perspective, the destination of a Mobike trip is closely related to the three views.

Definition 3: *(Virtual Station):* We partition the city into $m*n$ small grid cells. Each grid cell is regraded as a 'virtual station'. For simplicity, we index all the stations from 0 to $m * n - 1$ respectively, and all the places referred in this paper are described by their virtual station indexes.

Definition 4: *(Mode-n Product):* Mode-n product $\mathbf{X} \times_n \mathbf{U}$, where tensor $\mathbf{X} \in \mathbb{R}^{I_1 \times \ldots \times I_{n-1} \times J \times I_{n+1} \times \ldots \times I_N}$ and matrix $\mathbf{U} \in \mathbb{R}^{J \times I_n}$ is defined as:

$$(\mathbf{X} \times_n \mathbf{U})_{i_1,\ldots,i_{n-1},j,i_{n+1},\ldots,i_N} = \sum_{i_n=1}^{I_n} x_{i_1,i_2\ldots i_N} u_{ji_n} \tag{2.1}$$

Definition 5: *(Problem Definition):* Formally, given the origin o, time t, the goal of our problem is to predict the most possible destination d. We have collected Mobike historical trips including the origin, destination, time, and corresponding frequencies, where the frequency f of a trip is statistically regarded as possibility p. Given the same origin, time and different potential destinations, we can obtain different trips and their possibilities. By ranking the probabilities, we can obtain the most likely candidate virtual station as our predicted destination. Contextual data, such as POI data and human mobility data, have encoded the unique spatial temporal pattern of Mobike's movements, and thus can be used to enhance the predictive model. Essentially, there are three major tasks: (1) Constructing the three views of origin, destination and time from contextual data; (2) Modeling the intercorrelations among the views of origin, destination, time to predict trip destination; (3) Solving the optimization problem.

2.2 Framework Overview

Figure 1 shows our proposed framework consisting of three steps. The focus of this paper is to develop a data mining approach for predicting Mobike destinations. In the pursuit of this general aim, we have three specific tasks: multi-view construction, modeling, and optimization.

To construct the views of origin, destination, and time, we aim to extract the feature vectors of each origin virtual station, each destination virtual station, and each time window. While the three views is difficult to be evaluated, context information provides a potential to circumvent this problem. Specifically, we use POI distributions over POI categories to construct the spatial view of origins and destinations; We exploit both POI data and taxi GPS data to extract visiting densities over POI categories for a specific time window to construct the temporal view of time windows.

To predict Mobike destinations, we aim to model the relationship between the three views (origin, destination, and time) and trip probabilities. Since we are

not clear about the exact form of their relationship, we propose to build a multi-view machine (MVM) that can capture all the interactions among different views and then train the model with the constructed view data.

To learn the parameters of the MVM model, we leverage a gradient descent method. Since this model is designed to capture and evaluate all the possible interactions among all the views, the number of its parameters exponentially increases along with the increase of the dimensionality of views. We propose to store the parameters in a tensor and use tensor factorization to reduce the number of parameters. We simultaneously optimize both the tensor factorization and the MVM problems.

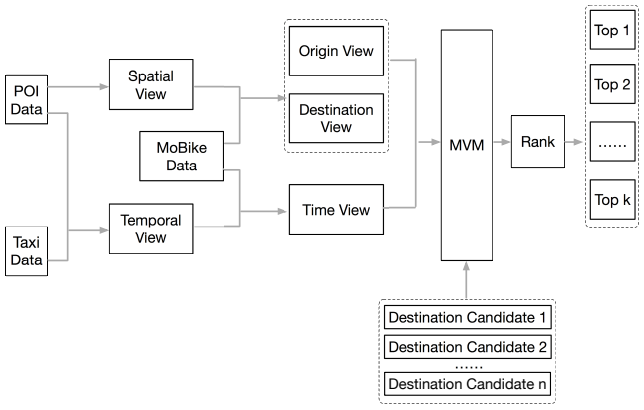

Figure 1: Framework overview

3 Constructing Spatiotemporal Multiviews

We use context information to construct mathmatical representations for multiviews.

3.1 Profiling Original Views and Destination Views

After partitioning the whole city into virtual stations, one trip's origin and destination would definitly fall into their corresponding virtual stations. Thus to quantify the origin and destination views, we can quantify the corresponding virtual stations and propogate the view vectors to the origin and destination views.

To quantify virtual stations, we make use of a POI distribution dataset. A POI record contains information of the POI's name, category and coordinates, which connects the POI together with virtual stations. The presumption is that, POIs have great impacts on the trips of Mobikes, and the impact of POIs in different category may vary. For example, subway stations are likely to attract Mobikes, either as origins or destinations in the whole day, while office buildings are likely to be destinations in morning and origins in afternoon.With this presumption, virtual station views, even though not observed directly, can be identified by strategically fusing the observable densities and diversities of POIs over various function categories, e.g., subway station, bus stop, office building, residental area, restaurant.

We propose to extract the density of POI categories in virtual stations. For each virtual station, we count the frequency of different POI categories as an estimation of its potential impact on Mobikes, and the obtained vector is the mathmetical representation of virtual stations. Suppose we have M POI categories, the view vector could be $\mathbf{v^s} = [v_1^s, v_2^s, ..., v_i^s, ..., v_M^s]$, where v_i^s is the frequency of the i_{th} POI.

As mentioned above, every origin and destination belong to their corresponding virtual stations, thus they can directly inherit the view vectors from virtual stations to describe themselves.

3.2 Profiling Temporal Views

We partition the whole day into several time windows, e.g., 24 time windows, correponding to 24 hours. We then quantify each time window to get view vectors of the temporal view.

To quantify time windows, we make use of the same POI distribution dataset as in profiling spatial views and a taxi record dataset. A taxi record contains dropoff time and dropoff place. Similarly to taxis, the movements of Mobikes are parts of the city's mobility, which tides periodically in different time windows of the day. The presumption is that in the same time, Mobikes tend to go to the same destinations as taxis do, and the POIs are the key factors that affect the destinations. For example, in the morning, taxis tend to go to working places and so do Mobikes; In the afternoon, taxis tend to go to residential or entertainment places and so do Mobikes. With this presumption, time window views, even though not observed directly, can be identified by strategically fusing the view vectors of virtual stations where taxis go to, e.g., in one time window, there exist k_1 taxi dropoff records in virtual station $\mathbf{v_1^s}$ and k_2 taxi dropoff records in virtual station $\mathbf{v_2^s}$, the time window's view vector could be $(k_1 * \mathbf{v_1^s} + k_2 * \mathbf{v_2^s})/(k_1 + k_2)$.

4 Fusing Coupled Multiview Interactions for Destination Prediction

Denote the constructed view vectors of origin, destinaiton and start time as $\mathbf{v}^o \in \mathbb{R}^{1\times M}$, $\mathbf{v}^d \in \mathbb{R}^{1\times M}$, $\mathbf{v}^t \in \mathbb{R}^{1\times M}$ respectively, we have:

$$\begin{aligned} \mathbf{v}^o &= [v_1^o, v_2^o, ..., v_{i1}^o, ..., v_M^o] \\ \mathbf{v}^d &= [v_1^d, v_2^d, ..., v_{i2}^d, ..., v_M^d] \\ \mathbf{v}^t &= [v_1^t, v_2^t, ..., v_{i3}^t, ..., v_M^t] \end{aligned} \tag{4.2}$$

where M is the POI categories.

Given a training dataset with n labeled records represented by 3 views: $\mathbf{D} = \{(\mathbf{v}_i, y_i)|i = 1, 2, ..., n\}$

in which $\mathbf{v}_i = (\mathbf{v}_i^o, \mathbf{v}_i^d, \mathbf{v}_i^t)$ and y is the trip's frequency.

Here, we consider all the possible intersections between the three views, and then we can have the multi-view machine model:

(4.3)
$$\begin{aligned} y =& \beta_0 + \sum_{i_1=1}^{M} \beta_{i_1}^o v_{i_1}^o + \sum_{i_2=1}^{M} \beta_{i_2}^d v_{i_2}^d + \sum_{i_3=1}^{M} \beta_{i_3}^t v_{i_3}^t \\ &+ \sum_{i_1=1}^{M}\sum_{i_2=1}^{M} \beta_{i_1,i_2}^{o,d} v_{i_1}^o v_{i_2}^d + \sum_{i_1=1}^{M}\sum_{i_3=1}^{M} \beta_{i_1,i_3}^{o,t} v_{i_1}^o v_{i_3}^t \\ &+ \sum_{i_2=1}^{M}\sum_{i_3=1}^{M} \beta_{i_2,i_3}^{d,t} v_{i_2}^d v_{i_3}^t + \sum_{i_1=1}^{M}\sum_{i_2=1}^{M}\sum_{i_3=1}^{M} \beta_{i_1,i_2,i_3}^{o,d,t} v_{i_1}^o v_{i_2}^d v_{i_3}^t \end{aligned}$$

5 Solving the Optimization Problem

The number of the parameters in the multi-view machine model is $N_p = 1+3*M+3*M^2+M^3 = (M+1)^3$, which would explode with the increase of POI category amount M. Thus we propose to adapt a factorization method to reduce it.

Firstly, to make it more clear, we transform Eq.(4.3) to a more intuitive form:

(5.4)
$$\begin{aligned} y =& \beta_0 * (1*1*1) + \sum_{i_1=1}^{M} \beta_{i_1}^o * (v_{i_1}^o * 1 * 1) \\ &+ \sum_{i_2=1}^{M} \beta_{i_2}^d * (1 * v_{i_2}^d * 1) + \sum_{i_3=1}^{M} \beta_{i_3}^t * (1*1*v_{i_3}^t) \\ &+ \sum_{i_1=1}^{M}\sum_{i_2=1}^{M} \beta_{i_1,i_2}^{o,d} * (v_{i_1}^o * v_{i_2}^d * 1) \\ &+ \sum_{i_1=1}^{M}\sum_{i_3=1}^{M} \beta_{i_1,i_3}^{o,t} * (v_{i_1}^o * 1 * v_{i_3}^t) \\ &+ \sum_{i_2=1}^{M}\sum_{i_3=1}^{M} \beta_{i_2,i_3}^{d,t} * (1 * v_{i_2}^d * v_{i_3}^t) \\ &+ \sum_{i_1=1}^{M}\sum_{i_2=1}^{M}\sum_{i_3=1}^{M} \beta_{i_1,i_2,i_3}^{o,d,t} * (v_{i_1}^o * v_{i_2}^d * v_{i_3}^t) \end{aligned}$$

We then append the constant value 1 to the three view vectors. We have $\mathbf{z}^o = (\mathbf{v}^o, 1) \in \mathbb{R}^{1\times M+1}$, $\mathbf{z}^d = (\mathbf{v}^d, 1) \in \mathbb{R}^{1\times M+1}$ and $\mathbf{z}^t = (\mathbf{v}^t, 1) \in \mathbb{R}^{1\times M+1}$, thus Eq.(5.4) could be reformed as:

$$y = \sum_{i_1=1}^{M+1}\sum_{i_2=1}^{M+1}\sum_{i_3=1}^{M+1} \omega_{i_1,i_2,i_3} * (z_{i_1}^o * z_{i_2}^d * z_{i_3}^t) \tag{5.5}$$

where

$$\begin{cases} \omega_{i_1,i_2,i_3} = \beta_0 & i_1,i_2,i_3 = M+1 \\ \omega_{i_1,i_2,i_3} = \beta_{i_1}^o & i_1 \le M; i_2,i_3 = M+1 \\ \omega_{i_1,i_2,i_3} = \beta_{i_2}^d & i_2 \le M; i_1,i_3 = M+1 \\ \omega_{i_1,i_2,i_3} = \beta_{i_3}^t & i_3 \le M; i_1,i_2 = M+1 \\ \omega_{i_1,i_2,i_3} = \beta_{i_1,i_2}^{o,d} & i_1,i_2 \le M; i_3 = M+1 \\ \omega_{i_1,i_2,i_3} = \beta_{i_1,i_3}^{o,t} & i_1,i_3 \le M; i_2 = M+1 \\ \omega_{i_1,i_2,i_3} = \beta_{i_2,i_3}^{d,t} & i_1,i_2 \le M; i_1 = M+1 \\ \omega_{i_1,i_2,i_3} = \beta_{i_1,i_2,i_3}^{o,d,t} & i_1,i_2,i_3 \le M \end{cases} \tag{5.6}$$

Eventually, we build a tensor $\mathbf{\Omega} = \{\omega_{i_1,i_2,i_3}\} \in \mathbb{R}^{M+1\times M+1\times M+1}$, and we use CANDECOMP/PARAFAC (CP) factorization [20] to reduce the parameters as well as to capture the potential connections between the parameters, suppose we factor the tensor into k sub-tensors, we have:

$$\mathbf{\Omega} = \mathbf{S} \times_1 \mathbf{A}^o \times_2 \mathbf{A}^{\mathbf{d}} \times_3 \mathbf{A}^t \tag{5.7}$$

where $\mathbf{A}^o = \{a_{i_1,f}^o\} \in \mathbb{R}^{M+1\times k}$, $\mathbf{A}^d = \{a_{i_2,f}^d\} \in \mathbb{R}^{M+1\times k}$, $\mathbf{A}^t = \{a_{i_3,f}^t\} \in \mathbb{R}^{M+1\times k}$, and $\mathbf{S} \in \mathbb{R}^{k\times k\times k}$ is an identity tensor.

Elementwise, we have:

$$\omega_{i_1,i_2,i_3} = \sum_{f=1}^{k} a_{i_1,f}^o * a_{i_2,f}^d * a_{i_3,f}^t \tag{5.8}$$

Now, the tensor $\mathbf{\Omega}$ is factorized into one identity tensor and three matrixs, whose parameter amount reduces to $N_p^{'} = (M+1)*k*3$. Generally, $k*3 \ll (M+1)^2$, thus $N_p^{'} \ll N_p$.

We use stochastic gradient descent (SGD) to learn the parameters in the model. Combine Eq.(5.8) with Eq.(5.5), we have:

(5.9)
$$y = \sum_{i_1=1}^{M+1}\sum_{i_2=1}^{M+1}\sum_{i_3=1}^{M+1} (\sum_{f=1}^{k} a_{i_1,f}^o * a_{i_2,f}^d * a_{i_3,f}^t) * (z_{i_1}^o * z_{i_2}^d * z_{i_3}^t)$$

The objective function is defined as:

$$\mathcal{L} = (\hat{y}(\mathbf{v}, \mathbf{\Theta}) - y)^2 + \sum_{i=1}^{N_p^{'}} \theta_i^2 \tag{5.10}$$

where $\mathbf{\Theta} = \{\theta_i \mid i \in [1, N_p^{'}]\} = \{a_{i_1,f}^o, a_{i_2,f}^d, a_{i_3,f}^t \mid i_1,i_2,i_3 \in [1, M+1], f \in [1,k]\}$ is the parameter set.

The gradient of the model $\frac{\partial \hat{y}(\mathbf{v},\mathbf{\Theta})}{\partial \theta}$ is:

(5.11)

$$\frac{\partial \hat{y}(\mathbf{v},\mathbf{\Theta})}{\partial a^o_{i_1,f}} = z^o_{i_1} * (\sum_{i_2=1}^{M+1} a^d_{i_2,f} * z^d_{i_2}) * (\sum_{i_3=1}^{M+1} a^t_{i_3,f} * z^t_{i_3})$$

$$\frac{\partial \hat{y}(\mathbf{v},\mathbf{\Theta})}{\partial a^d_{i_2,f}} = z^d_{i_2} * (\sum_{i_1=1}^{M+1} a^o_{i_1,f} * z^o_{i_1}) * (\sum_{i_3=1}^{M+1} a^t_{i_3,f} * z^t_{i_3})$$

$$\frac{\partial \hat{y}(\mathbf{v},\mathbf{\Theta})}{\partial a^t_{i_3,f}} = z^t_{i_3} * (\sum_{i_1=1}^{M+1} a^o_{i_1,f} * z^o_{i_1}) * (\sum_{i_2=1}^{M+1} a^d_{i_2,f} * z^d_{i_2})$$

The gradient of the objective function is:

$$\frac{\partial \mathcal{L}}{\partial \theta} = 2(\hat{y}(\mathbf{v},\mathbf{\Theta}) - y) * \frac{\partial \hat{y}(\mathbf{v},\mathbf{\Theta})}{\partial \theta} + 2\sum_{i=1}^{N_p^{'}} \theta_i \tag{5.12}$$

The learning process of Multi-view Machine is shown in Algorithm 1.

Algorithm 1 Multi-view Machine Optimization by SGD

Input: Training dataset $\mathbf{D} = \{(\mathbf{v}_i, y_i)|i = 1, 2, ..., n\}$, number of sub-tensors k, regularization parameter λ, learning rate η, standard derivation σ **Output:** Model parameters $\Theta = \{\mathbf{A^o}, \mathbf{A^d}, \mathbf{A^t}\}$

Initialize Θ with small random values.

while *not converge* **do**
  for $\{\mathbf{v}, y\} \in \mathbf{D}$ **do**
    for i_1, i_2, i_3 *in* $[1, M+1]$ **do**
      for f *in* $[1, k]$ **do**
        $\theta = \theta - \eta(\frac{\partial \mathcal{L}(\hat{y}(\mathbf{v},\mathbf{\Theta}),y)}{\partial \theta} + \lambda \frac{\partial \Omega}{\partial \theta})$
      end
    end
  end
end

6 Experimental Results

We provide an empirical evaluation for the performances of the proposed method on real-world POI data and human mobility data.

6.1 Data Description

Table 1 shows the statistics of three data sources used in the experiment. We use a public Mobike trip dataset, a POI dataset and a taxi GPS trace dataset. The Mobike trip dataset is released by Mobike in the Mobike Big Data Challenge 2017. The POI data set is obtained from www.dianping.com, which is a commercial review and recommendation website. And also, we collect the taxi GPS trace data from a Beijing taxi company. All the places referred in the datasets lie in Beijing, the capital of China.

To get more insight into the Mobike trip dataset, we visualize it from the perspective of origion, destination, trip distance and start time, as shown in Figure 2 - Figure 5 respectively. From Figure 2 and Figure 3 we can see the Mobikes are very active in downtown areas and fade away with the radiation of these areas. The two maps are very similiar indicating that the distance of most trips may not be long enough to change the active areas of Mobikes. This insight is also proved by Figure 4, where the trips concentrate in the three nearest intervals, i.e., $[0 - 500m]$,$[500m - 1000m]$ and $[1000m - 1500m]$. From Figure 5 we can see that Mobikes are active in 7:00-9:00 and 17:00-19:00, when commuters are going to work and off duty. Mobikes also behave actively around 12:00-13:00, when we believe most people would go for lunch.

We partition the city into $266 * 288$ virtual stations with each aera of $500m * 500m$, and partition the whole day into 24 time windows with each window of 1 hour. To obtain the spatial views of Mobike trips, we firstly extract the feature description of every virtual station from POI data, and then embed the features to origins and destinations of Mobike trips with consistence of coordinates. To obtain the temporal views of Mobike trips, we merge POI data with taxi GPS trace data to get the feature descriptions of each time window, and then embed the features to start time of Mobike trips with consistence of time. Eventually, we get the feature description of three views, i.e., origin, destination, start time. We use 70% of Mobike trip data to train the model and the remaining 30% for testing.

Table 1: Statistics of the Datasets.

Data Sources	Properties	Statistics
Mobike Trips	Number of trip records	3,214,096
	Number of users	349,693
	Time period of records	05/14/2017 - 05/23/2017
Taxi Traces	Number of taxis	13,597
	Effective days	92
	Time period	Apr. - Aug. 2012
	Number of trips	8,202,012
	Number of GPS points	111,602
	Total distance(km)	61,269,029
POIs	Number of POIs	328668
	Number of POI categories	20

6.2 Baseline algorithms

To show the effectiveness of the multi-view machine, we compare the performances of it with several ranking algorithms.

We used five learning to rank (LTR) algorithms for comparison: (1) Multiple Additive Regression Trees (**MART**) [11]: It is a boosted tree model, specifically,

a linear combination of the outputs of a set of regression trees. (2) RankBoost (**RB**) [10]: It is a boosted pairwise ranking method, which trains multiple weak rankers and combines their outputs as final ranking. (3) LambdaMART (**LM**) [3]: it is the boosted tree version of LambdaRank, which is based on RankNet. LambdaMART combines MART and LambdaRank. (4) ListNet (**LN**) [5]: it is a listwise ranking model with permutation top-k ranking likelihood as objective function. (5) RankNet (**RN**) [2]: it uses a neural network to model the underlying probabilistic cost function.

For these 5 LTR algorithms, we use RankLib[1]. We set the number of trees = 1000, the number of leaves = 10, the number of threshold candidates = 256, and the learning rate = 0.01 for MART and LambdaMART both. We set the number of iteration = 500, the number of threshold candidates = 10 for RankBoost. We set learning rate = 0.0007, number of hidden layers = 1, the number of hidden nodes per layer = 10, and the number of epochs to train for ListNet and RankNet both.

6.3 Evaluation Metrics

After ranking the potential destinations, we get the top-k destinations separately. The intuitive metrics are to evaluate how far they are from the actual destination. Thus we calculate their spherical distances from the actual destination separately.

Top-k Average Distance(Top-k AD) This metric is to evaluate how the algorithm performances averagely in the top k predictions: Top-k ND $= \frac{\sum_1^k d_j}{k}$, where d_j is the distance between the j_{th} destination candidate and the actual destination. Thus the less the Top-k AD is, the more accurate the prediction would be. For a testing dataset, we calculate the Top-k AD of each testing data and use the numerical average of all the data as the dataset's Top-k AD.

Top-k Nearest Distance(Top-k ND) This metric is to obtain the best perfromance in the top k predictions: Top-k ND $= min(d_1, ..., d_j, ..., d_k)$, where d_j is the distance between the j_{th} destination candidate and the actual destination. When $k = 1$, this metric degenerates to the same metric as Top-k ND. Thus the less the Top-k ND is, the more accurate the prediction would be. For a testing dataset, we calculate the Top-k ND of each testing data and use the numerical average of all the data as the dataset's Top-k ND.

6.4 Overall Performances

Figure 6 and Figure 7 show the performance comparisions of our method, multi-view machine(MVM), with five learning to rank(LTR) algorithms in terms of Top-k AD and Top-k ND. In all cases, we observe a significant improvement of MVM with respect to baselines.

Basically, we can see Top-k AD increases with k increasing. This is because we introduce destination candicates into the metric sequentially by their ranks, which means with k increasing, more and more unaccurate predictions are introduced into the metrics. Since Top-k AD is based on numerical average calculation, the introduction of more unaccurate predicitons would definitely make it larger.

To the opposite, Top-k ND decreases with k increasing. This metric also accepts more and more unaccurate predictions sequentially as Top-k AD does, however, top-k AD is based on minimum operation instead of numerical average calculation, which means the introduction of more unaccurate predicitons would at least not increase it. Considering the possible unaccurate prediction in our experiment, e.g., the $4th$ destination candidate might be more closer to the actual destination than the $2nd$ and the $3rd$ candidate, the decreasing of Top-k ND is reasonable.

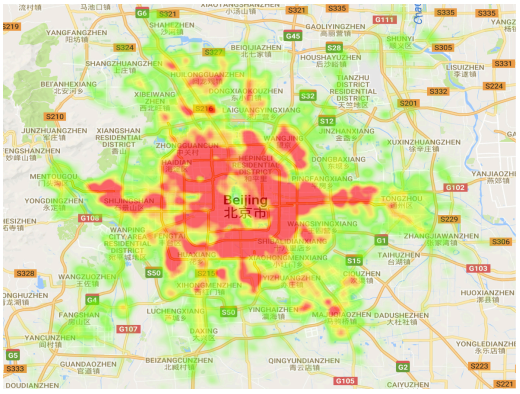

Figure 2: Heat map of origins.

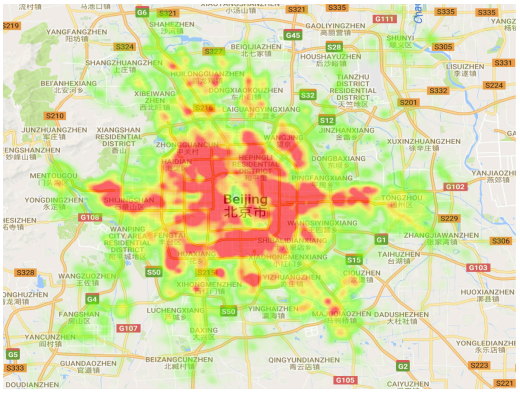

Figure 3: Heat map of destinations.

6.5 Robustness Check

We apply the learned multi-view machine model to dif-

[1] http://sourceforge.net/p/lemur/wiki/RankLib/

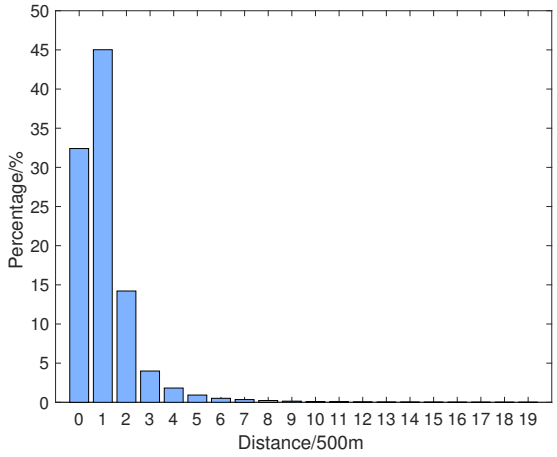

Figure 4: Percentage of trips over trip distance.

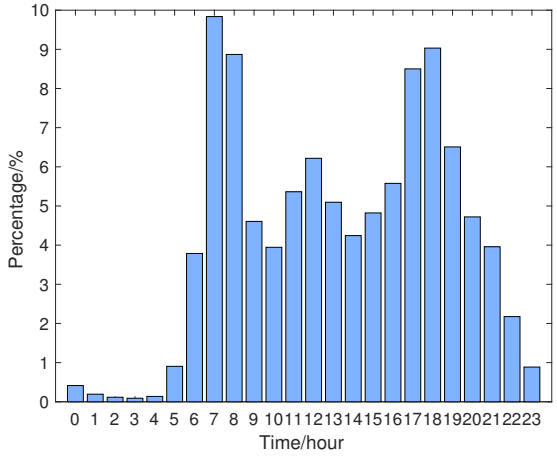

Figure 5: Percentage of trips over start time.

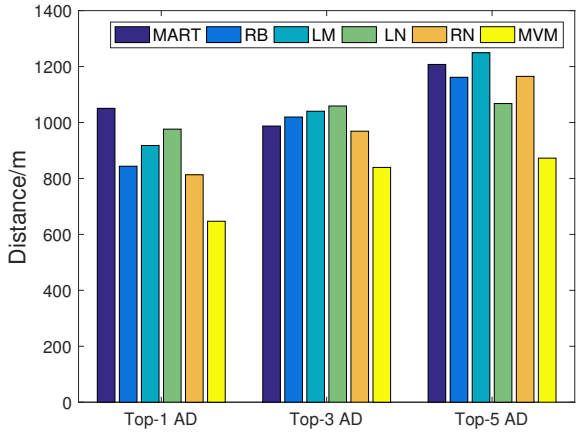

Figure 6: The overall performance comparisons of the MVM and five LTR algrithms in terms of Top-k Average Distance (AD).

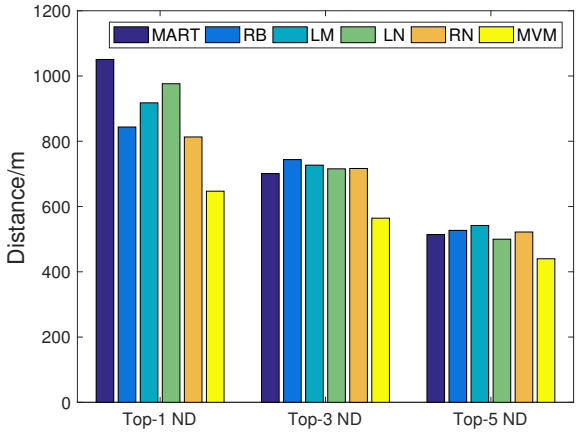

Figure 7: The overall performance comparisons of the MVM and five LTR algrithms in terms of Top-k Nearest Distance (ND).

ferent regions and time windows, to test the robustness of our method with variances in spatial and temporal views.

In spatial view variances testing, besides the whole city, we choose other two regions in Beijing, Zhongguancun and Yancunzhen representing downtown and uptown regions separately. The performance over Zhongguancun is better than the overall performance not only because there're more POIs in this region, but also because history records in Mobike dataset are also relatively abundant, which directly affects the model tending to perform better over downtown regions. To the opposite, in the uptown regions such as Yancunzhen the model performs relatively worse. We can see from Table 2 that even if in uptown regions, the prediction is still acurate.

In temporal view variances testing, besides the whole day, we choose other two time windows 7:00-8:00 and 22:00-23:00, representing busy hours and free hours separately. The performance over busy hours is better than the overall performance not only because there're more taxi trace data in this time window, making the mathmatical representation more accurate, but also because history records in Mobike dataset are also relatively abundant, which directly affects the model tending to perform better over busy hours. To the opposite, in free hours the model performs relatively worse. We can see from Table 3 that even if in free hours, the prediction result is still satisfying.

7 Related Work

Related work can be grouped into two categories: (i) application studies related to sharing bikes and (ii) methodological studies to exploit multi-view informa-

Table 2: The performance variances over different regions,evaluated by Average Distance(AD) and Nearest Distance(ND)

Region	Top-1 AD	Top-3 AD	Top-5 AD	Top-1 ND	Top-3 ND	Top-5 ND
Zhongguancun	563.75	687.45	729.73	563.75	371.60	359.16
Yancunzhen	955.08	1076.51	1099.79	955.08	757.25	689.44
Beijing	**646.99**	**839.43**	**872.62**	**646.99**	**564.36**	**439.95**

Table 3: The performance variances over different time windows, evaluated by Average Distance(AD) and Nearest Distance(ND)

Time	Top-1 AD	Top-3 AD	Top-5 AD	Top-1 ND	Top-3 ND	Top-5 ND
7:00-8:00	595.28	687.45	729.73	563.75	371.60	359.16
22:00-23:00	1020.19	1276.51	1280.19	1020.19	926.35	901.49
00:00-23:59	**646.99**	**839.43**	**872.62**	**646.99**	**564.36**	**439.95**

tion to improve destination prediction.

In prior literature, some methods have been proposed to deal with shared-bike destination prediction issues. For instance, Faghih-Imani et al. studied the decision process involved in identifying destination locations after picking up a bicycle at a shared-bike station, leveraging a random utility maximization approach in the form of a multinomial logit model [9], Zhang et al. introduced a new trip destination prediction and trip duration inference model on the basis of analyzing individuals' bike usage behaviors [29]. Prediction on station demands for station-based bike also catches great attention recently.

There are some other studies related to sharing bike systems, e.g., system planning [1, 19], pattern analysis [17], and bike repositioning [6, 22]. In the field of system planning, the work in [1] proposed a data-driven approach to develop bike lane construction plans based on large-scale real world bike trajectory data. The work in [19] found a significant correlation between the presence of bicycle lanes and Capital Bike share usage. In the field of bike repositioning, the work in [6] solved the many to many pickup and delivery problem which is motivated by operating self-service bike sharing systems leveraging branch-and-cut algorithm in the framework of graph theory.

Multi-view learning or learning with multiple distinct feature sets is a rapidly growing direction in machine learning with well theoretical underpinnings and great practical success. Sun et al. presented a method for finding high-informative examples for manual labeling based on extremely limited labeled data available during training [23]. Xu et al. proposed a new algorithm to improve the performance of adaboost by the theory of multi-view learning [28]. Cao et al. proposed a general predictor, named multi-view machines (MVMs), that could effectively explore the full-order interactions between features from multiple views [4].

Finally, nonlinear learning, human mobility learning as well as ranking spatial data are also related to our work. Wu et al. explored kernal methods in nonlinear learning and proposed the first analysis of Random Binning(RB) from the perspective of optimization, which by interpreting RB as a randomized block coordinate descent in the infinite-dimensional space[7, 27]. Wang et al. studied human mobility from probablistic point of view[24], and the work in [25] leveraged Mixture of Hawkes Processes to detect human mobility synchronization and trip purpose . Fu et al. studied the effect of geographic dependencies in real estate ranking [13, 14, 15] , and the work in [12] dealt ranking problem with sparse data from both online user reviews and offline moving behaviors. Fabozzi et al. modeled real estate equilibrium with a focus on a short-run view and study the explosiveness in the rental market by analyzing the impact of housing market exuberance on rents[8].

8 Conclusion Remarks

In this paper, we studied the problem of predicting the destination of Mobike. We presented a multi-view machine model that captures the interactions among spatial (i.e., origins and destinations) and temporal views (i.e., time periods). The construction of this model contains three critical steps. We started with extracting features from context information to build the feature vectors of three views, i.e., origin, destination and time. We then modeled the relationship of three views by incorporating all the possible interactions among each view into a multi-view machine model. In addition, we introduced a tensor factorization method to factorize the parameter tensor to reduce the number of parameters to be optimized. We applied the proposed model to predict Mobike destination by ranking destination candidates. The experimental results show the proposed multi-view model can effectively learn the relationship of the spatial and temporal views and substantially enhance the predictive performances.

Acknowledgements

This research was partially supported by University of Missouri Research Board (UMRB) via the proposal number: 4991. This research was partially supported by the Natural Science Foundation of China (NSFC) via the grant numbers: 71701007 and 61773199.

References

[1] J. Bao, T. He, S. Ruan, Y. Li, and Y. Zheng. Planning bike lanes based on sharing-bikes' trajectories. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1377–1386. ACM, 2017.

[2] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pages 89–96. ACM, 2005.

[3] C. J. Burges. From ranknet to lambdarank to lambdamart: An overview. *Learning*, 11(23-581):81, 2010.

[4] B. Cao, H. Zhou, G. Li, and P. S. Yu. Multi-view machines. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pages 427–436. ACM, 2016.

[5] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136. ACM, 2007.

[6] D. Chemla, F. Meunier, and R. W. Calvo. Bike sharing systems: Solving the static rebalancing problem. *Discrete Optimization*, 10(2):120–146, 2013.

[7] J. Chen, L. Wu, K. Audhkhasi, B. Kingsbury, and B. Ramabhadrari. Efficient one-vs-one kernel ridge regression for speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 2454–2458. IEEE, 2016.

[8] F. J. Fabozzi and K. Xiao. Explosive rents: The real estate market dynamics in exuberance. *The Quarterly Review of Economics and Finance*, 66:100–107, 2017.

[9] A. Faghih-Imani and N. Eluru. Analysing bicycle-sharing system user destination choice preferences: Chicago's divvy system. *Journal of transport geography*, 44:53–64, 2015.

[10] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *Journal of machine learning research*, 4(Nov):933–969, 2003.

[11] J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[12] Y. Fu, Y. Ge, Y. Zheng, Z. Yao, Y. Liu, H. Xiong, and J. Yuan. Sparse real estate ranking with online user reviews and offline moving behaviors. In *Data Mining (ICDM), 2014 IEEE International Conference on*, pages 120–129. IEEE, 2014.

[13] Y. Fu, G. Liu, S. Papadimitriou, H. Xiong, Y. Ge, H. Zhu, and C. Zhu. Real estate ranking via mixed land-use latent models. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 299–308. ACM, 2015.

[14] Y. Fu, H. Xiong, Y. Ge, Z. Yao, Y. Zheng, and Z.-H. Zhou. Exploiting geographic dependencies for real estate appraisal: a mutual perspective of ranking and clustering. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1047–1056. ACM, 2014.

[15] Y. Fu, H. Xiong, Y. Ge, Y. Zheng, Z. Yao, and Z.-H. Zhou. Modeling of geographic dependencies for real estate ranking. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 11(1):11, 2016.

[16] F. L. Hitchcock. The expression of a tensor or a polyadic as a sum of products. *Studies in Applied Mathematics*, 6(1-4):164–189, 1927.

[17] C. D. Keenan. Chicago's shared bikes: how big data technology can assess ridership. 2016.

[18] A. V. Khezerlou, X. Zhou, L. Li, Z. Shafiq, A. X. Liu, and F. Zhang. A traffic flow approach to early detection of gathering events: Comprehensive results. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(6):74, 2017.

[19] D. Kim, H. Shin, H. Im, and J. Park. Factors influencing travel behaviors in bikesharing. In *Transportation Research Board 91st Annual Meeting*, 2012.

[20] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[21] S. Rendle. Factorization machines. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 995–1000. IEEE, 2010.

[22] J. Schuijbroek, R. Hampshire, and W.-J. van Hoeve. Inventory rebalancing and vehicle routing in bike sharing systems. 2013.

[23] S. Sun and J. Shawe-Taylor. Sparse semi-supervised learning using conjugate functions. *Journal of Machine Learning Research*, 11(Sep):2423–2455, 2010.

[24] P. Wang, Y. Fu, G. Liu, W. Hu, and C. Aggarwal. Human mobility synchronization and trip purpose detection with mixture of hawkes processes. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 495–503. ACM, 2017.

[25] P. Wang, G. Liu, Y. Fu, Y. Zhou, and J. Li. Spotting trip purposes from taxi trajectories: A general probabilistic model. 9:1–26, 12 2017.

[26] F. Wu and Y. Xue. Innovations of bike sharing industry in china: A case study of mobike's station-less bike sharing system, 2017.

[27] L. Wu, I. E. Yen, J. Chen, and R. Yan. Revisiting random binning features: Fast convergence and strong parallelizability. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1265–1274. ACM, 2016.

[28] Z. Xu and S. Sun. An algorithm on multi-view adaboost. *Neural Information Processing. Theory and Algorithms*, pages 355–362, 2010.

[29] J. Zhang, X. Pan, M. Li, and S. Y. Philip. Bicycle-sharing system analysis and trip prediction. In *Mobile Data Management (MDM), 2016 17th IEEE International Conference on*, volume 1, pages 174–179. IEEE, 2016.