Predicting Bundles of Spatial Locations from Learning Revealed Preference Data

Truc Viet Le[#], Siyuan Liu[⋆], Hoong Chuin Lau[#], and Ramayya Krishnan[⋆]
[#]School of Information Systems, Singapore Management University
[⋆]Heinz College, Carnegie Mellon University
[#]{trucviet.le.2012, hclau}@smu.edu.sg
[⋆]{siyuan, rk2x}@cmu.edu

ABSTRACT

We propose the problem of predicting a bundle of goods, where the goods considered is a set of spatial locations that an agent wishes to visit. This typically arises in the tourism setting where attractions can often be bundled and sold as a package to visitors. While the problem of predicting future locations given the current and past trajectories is well-established, we take a radical approach by looking at it from an economic point of view. We view an agent's past trajectories as *revealed preference* (RP) data, where the choice of locations is a solution to an optimisation problem according to some unknown utility function and subject to the prevailing prices and budget constraint. We approximate the prices and budget constraint as the time costs to finish visiting the chosen locations. We leverage on a recent line of work that has established algorithms to efficiently learn from RP data (i.e., recover the utility functions) and make predictions of future purchasing behaviours. We adopt and adapt those work to our original setting while incorporating techniques from spatiotemporal analysis. We experiment with real-world trajectory data collected from a theme park. Our predictions show improved accuracies in comparison with the baseline methods by at least 20%, one of which comes from the spatiotemporal analysis domain.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Economics

Keywords

Mobile agents; trajectories; revealed preferences; utility learning; knapsack; decision modelling.

1. INTRODUCTION

We consider the problem of predicting a bundle of goods, where the goods are spatial locations that an agent wish to visit (a.k.a. "spatial bundle"), given knowledge of the costs of all goods considered and their budget constraint. This scenario typically arises in the travel and tourism industry where attractions in a certain geographical area can be packaged together by the developer and sold at a (discounted) bundled price. An example is CityPASS, where the company sells booklets (bundles) of attractions in 11 cities throughout North America. Bundles typically include transport passes and tickets to places of interest that can be redeemed for a specified duration of visit. Another is Eurail – a European-based company that markets bundled train passes to a defined set of European countries that share borders and for a specified period of travel. When prices of bundles differ depending on the combinations of the included goods and their quantities, the problem becomes that of classical *revealed preference* analysis pioneered by Paul Samuelson [13].

Revealed preference (RP) is a consumer behaviour theory built on the premise that intrinsic preferences are unobserved. However, a consumer's preferences can be *revealed* through their observed purchasing behaviours. That is, it is possible to predict consumer behaviours on the basis of variable prices and income (budget constraint). A consumer with a given income will buy a certain mixture of goods; but as their income changes, the mixture of goods will change accordingly. The theory assumes that a rational consumer has considered a set of all possible alternatives according to some well-defined utility function before making their decision. Thus, given a consumer chooses a option out of this set, this option must be the most preferred (i.e., the utility maximiser) that they can afford. The basic question of RP analysis is to recover a utility function that best explains or *rationalises* the observed behaviours [13, 3, 9, 15].

We consider the scenario where the developer allows buyers to "mix and match" a fixed quantity $Q > 0$ of items at *uniform* price p and the chosen bundle can be consumed during a specified period B. For instance, a theme park developer would sell a bundle of attractions that visitors can choose from a fixed set (e.g., choosing $Q = 4$ out of 16 attractions) at price p; and once chosen, the attractions can be visited in any order during period B. If we consider all those who go for bundles at price p, then RP analysis is no longer feasible because of price uniformity. In other words, the cost information has become *latent* or *unobserved*. In order to apply RP analysis, we need to find a proxy to the costs that consumers take into consideration when making decisions. In our setting, cost information may be approximated by the physical distances of the visited locations revealed from an agent's trajectory in the absence of any other sources of

Appears in: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Bordini, Elkind, Weiss, Yolum (eds.), May 4–8, 2015, Istanbul, Turkey.*

information (e.g., means of transport, queue length or congestion level at each location). This is because a rational agent would plan their visit such as to minimise the total distance travelled (or the time cost) over their chosen locations subject to budget constraint B. Finding such proxies is thus a challenge in RP analysis for spatial goods in the absence of complete information.

Given its spatial nature, the problem can benefit from the rich literature of the location prediction problem in spatiotemporal analysis. Motivated from the massive growth of spatiotemporal data generated by location-aware devices, the problem seeks to predict the next location(s) that an individual would travel to given their current and past trajectories. A trajectory is defined as an ordered sequence of timestamped locations. A common approach is to apply a wide variety of Markov models, most commonly Markov chains and hidden Markov models (HMMs), to model the sequential movements and make predictions. A common subtask is to cluster the locations using the hidden states of an HMM to discretise the space into finite points of interest (POIs) [10, 5]. Another is to cluster the trajectories into groups of similar mobility patterns and model each separately to reduce variance and improve predictions [10, 7].

In this paper, we integrate techniques from spatiotemporal analysis to solve the proposed problem. In particular, we use trajectory clustering to divide the agents into groups of similar preferences for variance reduction. We then use HMMs to extract clusters of locations that are frequently visited together in order to establish the *reference points* from where the agents based their decisions on to approximate the costs. Finally, we leverage on a recent line of work emerging from the intersection of algorithmic game theory and statistical learning theory [3, 15], that has established the conditions and algorithms for efficiently learning the utility functions from RP data. We evaluate our proposed solution using real-world data collected from a theme park. Our results show significantly more accurate predictions compared to the baseline methods, one of which was proposed in the context of the next location prediction problem [10].

Applications of the ability to make such predictions are plentiful and include predicting the aggregate demand in response to changes in costs of the goods (e.g., changing certain locations to further/nearer distances) or the set of all available goods itself (including/excluding some locations to/from the consumer's choice set), resource planning in anticipation of such changes in demand, and developing location-aware services or marketing just to name a few.

Our contributions are three-fold. First, we propose a new and interesting problem in the domain of RP analysis where goods are spatial locations and cost information is unobserved. Second, from the point of view of the proposed problem, our application of spatiotemporal analysis to tackle the emerged challenge is novel and practically doable. Third, from the point of view of the next location(s) prediction problem, our application of RP analysis as a viable solution is novel and the first of its kind. Finally, we have shown that there is a common ground between the two traditionally separate domains (i.e., revealed preference and spatiotemporal analysis) that can be explored and cross-benefited.

2. RELATED WORK

Since the seminal work of Samuelson [13], there has been a voluminous body of work in the economics literature on RP theory. See [14] for a comprehensive survey. A classic result is Afriat's theorem [1], which formulates a system of inequalities that has positive solution iff the demand data is rationalizable. A recent line of work emerging from the intersection of algorithmic game theory and statistical learning theory has established some theoretical groundwork for the problem of learning utility functions from RP data [3, 9, 15]. Beigman and Vohra [3] use statistical learning analysis to address the problem of learning utility functions from RP data with the explicit goal of prediction. They show that the sample complexity (in the probably approximately correct sense) of learning a utility function from RP data is infinite, assuming monotonicity and concavity of utility functions. Lahaie [9] applies kernel methods to rationalise RP data assuming non-linear prices and incomplete price information, where prices of non-demanded bundles are unknown. The proposed method reduces the problem to fitting utility function to observations in the transformed high-dimensional space using the "kernel trick".

Notably, Zadimoghaddam and Roth [15] recently propose a simple and efficient algorithm to learn utility functions from RP data for the class of linear and linearly separable concave utility functions in polynomial sample complexity. Because of its simplicity and efficiency, we adopt one of the learning algorithms in [15] for our own setting. Our work can therefore be viewed as a extension of [15] to the spatial setting. It is, however, not straightforward how the algorithm can be adapted to solve the proposed problem, especially where costs are unobserved.

The problem of predicting the next location(s) of a mobile agent has been the traditional domain of spatiotemporal analysis. In most of these work, some form of Markov model is proposed to learn the trajectories and make inferences of future locations. Mathew *et al.* [10] use hidden Markov models (HMMs) to identify clusters of locations from raw GPS data, where each cluster is a point of interest (POI) and corresponds to a hidden state of the HMM. They make inferences of the next locations using the forward algorithm of HMMs given the current trajectory. Trajectories are in turn clustered into groups of similar patterns to reduce variance and improve predictions. Jiang *et al.* [7] propose an advanced method to cluster spatiotemporal trajectories by combining principal component analysis (PCA) and K-means clustering. Gambs *et al.* [5] propose a mobility model call MMC (Mobility Markov Chain) to incorporate knowledge of the previous n visited locations and develop an inference algorithm based on n-th order Markov chains. Gao *et al.* [6] takes a Bayesian approach to the problem, but still within the framework of Markov models. Another recent work that uses some variation of HMMs to make predictions when location data is uncertain or missing is due to Qiu *et al.* [11]. A common thread along this line is: some variant of a Markov models and some kind of clustering (e.g., K-means, HMMs) to extract mobility patterns. We adopt both themes in this work to adapt traditional RP learning and prediction to the spatiotemporal setting.

3. PROBLEM STATEMENT & SOLUTION OVERVIEW

We consider a set $\mathcal{D}$ of agents and a finite set $\mathcal{G}$ ($|\mathcal{G}| = d$) of spatial locations of arbitrarily large capacity each. Each agent $i \in \mathcal{D}$ faces a cost vector p_i, where p_{ij} is the cost of

visiting location $j \in \mathcal{G}$ for i. Each i also has a personal value vector v_i over each $j \in \mathcal{G}$, where v_{ij} is the value of j for i. In other words, v_i reflects i's intrinsic preference over all $j \in \mathcal{G}$. Finally, agent i comes with a budget constraint B_i and i wishes to visit a subset $s_i \subseteq \mathcal{G}$ such that $|s_i| \leq Q$ for some $Q > 0$ and $\sum_{j \in s_i} p_{ij} \leq B_i$. WLOG, we suppose that i makes a vector of binary decisions $x_i \in \{0,1\}^d$ of which location j to include in the bundle s_i. The preference of i over all possible bundles is defined by a non-decreasing, non-negative concave utility function $U : \{0,1\}^d \to \mathbb{R}^+$. Throughout the paper, we assume that i's utility function belongs to the class of linear utility functions, i.e., $u(x_i) = x_i \cdot v_i$. Thus, i chooses his most preferred bundle s_i^* (or equivalently x_i^*) by solving the classic knapsack problem:

$$x_i^* = \arg\max_{x_i} \{u(x_i) : x_i \cdot \mathbf{1} \leq Q \wedge x_i \cdot p_i \leq B_i\}. \quad (1)$$

Following the conventions in machine learning, we derive a training set $\mathcal{S} = \{(p_i, B_i, x_i^*)\}_{i=1}^m$ drawn i.d.d. from $\mathcal{D}$ and a test set $\mathcal{T} = D - S$. Assuming linear utilities of the agents, we wish to learn the value vectors $\hat{v}_i$ from $\mathcal{S}$ in order to predict the chosen bundles in $\mathcal{T}$ with good enough accuracies. Let $x_i^*(p_i, B_i, v_i)$ be the chosen bundle and let $\hat{x}_i(p_i, B_i, \hat{v}_i)$ be the predicted one, our accuracies are good enough if for all i and for some $\delta > 0$:

$$\Pr(x_i^*(p_i, B_i, v_i) \neq \hat{x}_i(p_i, B_i, \hat{v}_i)) \leq 1 - \delta. \quad (2)$$

Fig. 1 illustrates the overall framework of our learning and predictive solution to the above problem. For learning, we first split all training agents in S into K clusters using trajectory clustering. For each cluster Cl_j ($1 \leq j \leq K$), we train a separate HMM_j that best describes the sequential movements of those in Cl_j. We then propose a heuristic to approximate the perceived costs faced by each agent called the "centroid heuristic". In short, we derive a set of "centroids" C_j for each Cl_j using the hidden states of HMM_j such that each agent $i \in \mathrm{Cl}_j$ can be mapped to each centroid (a.k.a. "reference point") $r_k \in C_j$ depending on their intention I_i (to be defined later). Each r_k corresponds to a perceived cost vector p_k shared by all the agents having the same intention. We can then efficiently learn the value ratio matrix R_j given the chosen bundle x_i^* and cost vector p_k of all agents $i \in \mathrm{Cl}_j$ using an RP learning algorithm, e.g., the recent one due to Zadimoghaddam and Roth [15].

To make predictions, for each agent $i \in \mathcal{T}$, we first predict which cluster Cl_k that i most likely belongs to – call this Cl_k^i. In this paper, we don't address the problem of class prediction due to restricted scope. We suppose it is feasible, and most of the time it is through some established method such as logistic regression and decision tree. Given i's intention I_i, we map I_i to the nearest reference point r_j^i, from where we can derive i's cost vectors p_j^i. Let $\hat{v_k}$ be any row vector of R_k corresponding to Cl_k. Given i's budget B_i and learned value vector $\hat{v}_k$, we predict i's chosen bundle when facing p_j^i by solving (1).

In the following sections, we elaborate on the components of the the proposed framework depicted in Fig. 1, beginning with trajectory clustering.

4. TRAJECTORY CLUSTERING

One of our challenges is that we cannot simply learn the preferences of each agent $i \in \mathcal{S}$ and predict for another $j \in \mathcal{T}$ because: (1) that is highly inefficient, and (2) it would most likely overfit the training data and lead to poor predictions. On the other hand, nor can we expect everyone to behave the same under the same prices and budget constraint as implied by RP theory because empirical data shows a great diversity of behaviours. We seek a solution in between where the agents can be divided into groups of similar behaviours such that we could learn the preferences from and predict the behaviours of those of the same group. This is the rationale for trajectory clustering.

For each agent $i \in \mathcal{S}$, let l_i be the sequence length of i (i.e., the number of locations visited by i), we denote the sequence of locations visited by i as $y^{(i)} = \{y_t^{(i)}\}_{t=1}^{l_i}$ and the sequence of timestamps for each $y_t^{(i)}$ as $\tau^{(i)} = \{\tau_t^{(i)}\}_{t=1}^{l_i}$. We define i's **trajectory** as $s^{(i)} = \{(y_t^{(i)}, \tau_t^{(i)})\}_{t=1}^{l_i}$. Hence, a trajectory is a spatiotemporal sequence of spatial locations and their corresponding timestamps. A spatial location is a place in the physical world that can be located using its coordinates. A timestamp τ_t indicates when the agent visited location y_t, but does *not* necessarily indicate how long they had stayed there (i.e., the duration of visit).

Suppose there exist an upper bound B_U and a lower bound B_L on the timestamps of all the trajectories, then the duration between B_U and B_L can be discretised into a finite number of T segments $T = \lceil (B_U - B_L)/\Delta_\tau \rceil$, where Δ_τ is an arbitrary duration of each time segment. We can derive a categorical vector a_i of finite and uniform length T for each agent i from their original trajectory $s^{(i)}$. Each element $a_{it} \in a_i$ ($1 \leq t \leq T$) indicates i's location at time t. If no location is recorded for i at t, then $a_{it} = 0$ by convention; otherwise, $a_{it} \in \{1, \ldots, |\mathcal{G}|\}$. We finally assume that i spends at least time Δ_τ and at most an integral multiple of Δ_τ at any location in its trajectory.

It is feasible to cluster the agents based on their similarity of behaviours by clustering the trajectories, or equivalently the vectors a_i for all $i \in \mathcal{S}$. To this end, there exist a wide variety of methods for sequence clustering. In this paper, we make use of the well-known method of hierarchical clustering because of its simplicity and the ability to incorporate domain knowledge in selecting the number of clusters K. In particular, we use the agglomerative approach that clusters the trajectories recursively in the bottom-up fashion. We use the edit distance to quantify the dissimilarity between any two vectors a_i and a_j with substitution cost being the physical distance between the pair of locations that differ in a_i and a_j and arbitrary insertion/deletion cost (because they are essentially two vectors of categorical variables of the same length). To select the number of clusters K, the hierarchy tree is "cut" at some height that would break up $\mathcal{S}$ into K clusters, which can be determined based on our domain knowledge, so we do not discuss it here.

There are many other more advanced methods for sequence and spatiotemporal clustering; we are using one of the simplest and most popular here because clustering is not our final goal, but a means to an end. An example of a more advanced method is [7] due to Jiang *et al.*, in which trajectories are clustered by combining both PCA and K-means clustering. Refer to [8] for a comprehensive survey on spatiotemporal clustering, or to [4] for a comprehensive survey on model-based clustering.

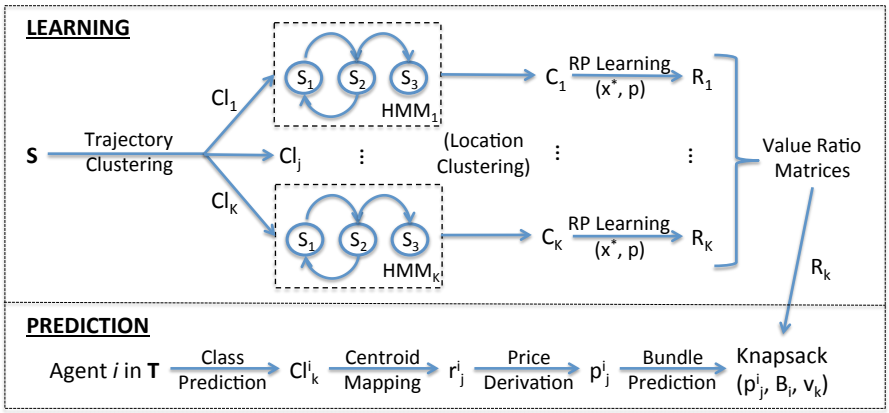

Figure 1: The proposed learning and predictive framework for the problem of spatial bundle prediction.

5. REVEALED PREFERENCE LEARNING

In traditional RP analysis, we are given a sequence of observations $\mathcal{D} = \{(p_i, B_i, x_i^*)\}_{i=1}^N$, the problem is to recover the utility function that best explains or *rationalises* $\mathcal{D}$. Under our assumption of linear utility, we wish to recover the vector v_i for each agent $i \in \mathcal{D}$. Furthermore, not only do we wish to explain the observed data, we also wish to be able to predict future chosen bundles of the agents given the learned values. The latter goal is much broader and harder than the former, because being able to rationalise observations often does not mean being able to predict unobserved outcomes (i.e., to generalise) [15].

Suppose we are able to categorise $\mathcal{D}$ into K clusters ($K \ll N$). For simplicity, we also call an agent belonging to cluster k ($1 \leq k \leq K$) an agent of type k. The problem can be solved by applying the All Pairs Comparison (APC) algorithm due to Zadimoghaddam and Roth [15], where for each agent type k, there is a value ratio matrix R_k learned from $\mathcal{S}$ of dimension $d \times d$. The APC algorithm is a very simple and efficient algorithm to learn the value ratios v_i/v_j from RP data for all pairs of goods $i, j \in \mathcal{G}$. The main idea is to bound the pairs v_i/v_j such that if item i is *preferred to* item j in some chosen bundle x^*, then $v_i/p_i \geq v_j/p_j$, or equivalently $v_i/v_j \geq p_i/p_j$. In our setting, unlike an unordered bundle of goods, locations have to be visited in a sequential order; thus, x^* has an intrinsic ordering nature. Given locations $i, j \in x^*$, we denote $x_i^* > x_j^*$ if i was visited before j by the considered agent, which also connotes (approximately) the agent's preference of i over j in x^*.

Thus, v_i/v_j can be upper and lower bounded given the purchase decisions $x^{(k)}$ and price vectors $p^{(k)}$ of all agents of type k. Given a test agent of type k, we would choose any row i of R_k to obtain a value ratio vector $\hat{v} = v_i/v_j$ for all $1 \leq j \leq d$ (whose elements are arbitrarily in between the bounds) and predict an optimal bundle $\hat{x}(p, B, \hat{v})$ by solving (1) using the given price p and budget constraint B. Refer to [15] for full details of the APC algorithm.

Throughout the paper, we use physical distance to approximate the time cost of traveling from one location to another. Hence, our budget constraint B is defined as the total time cost required to go through the all the locations in the chosen bundle. Physical distances are different depending on from *where* they are measured, i.e., the *reference point*. One way to compute the total cost of a trajectory $s^{(i)}$ is to sum all the distances of the segments in $s^{(i)}$. This method does not scale because there are an exponential number of ways to choose Q locations from the set $\mathcal{G}$. On the other hand, suppose we know i's intention I_i of approximately where i would go, e.g., through survey or any other means, we would be able to map I_i to a particular reference point r_k in space from where we can approximate the total cost as the sum of distances from r_k to all the locations in $s^{(i)}$. An optimal reference point r_k for i is one that minimises i's total distance derived from r_k assuming i's goal is to minimise the total cost. However, this is not feasible in the absence of complete information of $s^{(i)}$. We define i's **intention** I_i as any form of incomplete information about $s^{(i)}$ that we may have. In the following, section, we propose heuristics to derive r_k given I_i.

6. HEURISTICS FOR COST DERIVATION

If we know for certain i's first visited location, call it $y_1^{(i)}$, then we can take $y_1^{(i)}$ as the reference point for $s^{(i)}$. The rationale is that the first location in a sequence is often the one having the highest priority (i.e., the most preferred) and a rational agent would plan their itinerary in such a way to minimise their total time cost as seen from $y_1^{(i)}$. By this, we are making other locations that are distant from $y_1^{(i)}$ costly and less likely to be included in the chosen bundle. This heuristic aligns with our assumption that agents try to minimise the total distance due to budget constraint. We call simply this the **first-location heuristic**.

Let d_i be the vector of physical distances from r_i, where r_i is i's reference point, to all the locations in $\mathcal{G}$. We can easily derive p_i, the vector of time costs to travel from r_i to all the locations in $\mathcal{G}$ from d_i. For example, if the primary means of travel is on foot, then d_i can be converted to p_i using the average human walking speed of 5 km/h. Thus, for each location $j \in \mathcal{G}$, p_{ij} is the average time cost for i to go from r_i to j. Furthermore, suppose we know the upper bound on the duration of visit at each location $j \in \mathcal{G}$, call this b_j (independent of i), then the proper price vector as seen by i, call it $\hat{p}_i$, should be $\hat{p}_{ij} = p_{ij} + b_j$ to reflect the true time cost at j (i.e., the total time of traveling to j and the duration of visit at j).

Often, we may not know for certain what an agent i may want to include in their itinerary due to incomplete information. Instead, we may only know i's intention I_i of such. In

such cases, we would want to divide our physical space (that covers all of $\mathcal{G}$) into non-overlapping subareas and map I_i to one of such subareas. For each subarea, we would derive a reference point from where we can compute the price vector p_i. Our rationale comes from the empirical observations that businesses of similar nature tend to cluster together geographically in an area in order to compete. Therefore, identifying such clusters of locations (or subareas) is the first step to identifying sensible reference points to infer costs in the absence of complete information.

To this end, we make use of HMM to derive clusters of locations. Locations within a cluster should be physically close to one another and tend to be visited together in short temporal sequence (i.e., without much delay). We use the hidden states of an HMM to identify those clusters such that each state corresponds to a cluster. We then derive the reference point of each cluster using its *centroid* (to be defined later). Given a centroid r_k, we use the nearest-neighbour method to assign locations to clusters: we assign location j to cluster k such that the physical distance from j to r_k is the nearest among all other centroids. For each agent i, given I_i, we map I_i to the nearest cluster centroid r_k and calculate p_i as before. We call this the **centroid heuristic**. The following subsections elaborate on the proposed method, beginning with the preliminaries of HMMs.

6.1 Hidden Markov Model (HMM)

An HMM describes the relationship between two stochastic processes: an observed process and an unobserved (or hidden) underlying process. The hidden process is assumed to follow a Markov chain, and the observations are considered conditionally independent given the sequence of hidden states. Let $\{Y_t\}_{t=1}^T$ and $\{X_t\}_{t=1}^T$ be the time series representing the observations and the corresponding hidden states of an HMM respectively. We denote $f(y_t|\Theta_{x_t}) = \Pr(Y_t = y_t; \Theta|X_t = x_t)$ the probability density function of observation y_t parameterized over vector Θ given state x_t. An HMM with finite N hidden states is specified by:

1. The finite set of hidden states $S = \{S_1, S_2, \ldots, S_N\}$;
2. The state transition matrix $\mathbf{A} = \{a_{ij}\}$, where $a_{ij} = \Pr(X_t = S_j|X_{t-1} = S_i), 1 \leq i, j \leq N$;
3. The parameter vector Θ_i of the response (or emission) density function $f(y_t|\Theta_{x_t})$ for each S_i; and
4. The vector of initial (state) probabilities $\pi = \{\pi_i\}$, where $\pi_i = \Pr(X_1 = S_i)$ and $\sum_{i=1}^N \pi_i = 1$.

It is common to use the compact notation

$$\mathbf{\Lambda} = (\pi, \mathbf{A}, \{\Theta_i\}) \tag{3}$$

to represent the complete parameter set of an HMM. The problem of estimating the parameters of an HMM given an observed sequence $\{y_t\}_{t=1}^T$ can be formulated as a maximum likelihood (ML) problem:

$$\mathbf{\Lambda}^* = \arg\max_{\mathbf{\Lambda}} \prod_{t=1}^{T} \Pr(Y_t = y_t|\mathbf{\Lambda}). \tag{4}$$

The well-known method to estimate $\mathbf{\Lambda}^*$ is the Baum-Welch algorithm, which is a special case of the EM algorithm, which in turn makes use of the forward-backward algorithm [2] to compute the marginal log-likelihood. Refer to [12] for more details on HMMs.

6.2 Centroid Heuristic Using HMM

Because of the spatiotemporal nature of our trajectories, each response variable is a tuple (y_k, τ_k) with the spatial component y_k being the discrete locations drawn from $\mathcal{G}$ as a multinomial distribution, and the temporal component τ_k being the continuous timestamp drawn from a Gaussian distribution $\mathcal{N}(\mu_k, \sigma_k)$ $(1 \leq k \leq N)$. Timestamp can be modelled as a continuous random variable because we can set a continuous temporal range from the earliest timestamp B_L to the latest one B_U for all $i \in \mathcal{S}$.

We fit the HMM using the trajectories $s^{(i)}$ for all $i \in \mathcal{S}$ using (y_k, τ_k) as the bivariate response. To select the optimal number of states N^*, we use the Bayesian Information Criterion (BIC), a popular penalized likelihood criterion for model selection [4]. We begin fitting with the simplest model where $N = 2$. At each iteration, as long as the BIC_N of this step is still less than that of the previous BIC_{N-1} (i.e., BIC keeps decreasing as the fitness improves while accounting for model complexity), we keep incrementing N. We stop when the current BIC becomes greater than the previous, i.e., it has reached the "elbow". The optimal number of states N^* is that of the previous step.

We use the set of states S to define the clusters of locations, where each $S_k \in S$ forms a cluster. For each S_k, we extract the parameter vector $\Theta_k = (\theta_1, \ldots, \theta_d)$ of the discrete multinomial response, which is a vector of probabilities of each location $j \in \mathcal{G}$ being visited while the agent is in the cluster S_k. Let C_j be the coordinates (latitude and longitude) of each location $j \in \mathcal{G}$, we compute the coordinates of the **cluster centroid** r_k of S_k as the weighted sum $r_k = \sum_{j=1}^d \theta_j C_j$. As a result, locations with high probabilities (i.e., likely to be in the cluster) have more weights, while those with low probabilities (i.e., unlikely to be in the cluster) have less weights.

Fig. 2 illustrates the concept. It shows the real-world locations of attractions in the theme park considered in the experiments in the following section being mapped to their nearest centroids derived from the hidden states of a 4-state HMM. The HMM was fitted using real-world trajectories of visitors to the theme park collected over 4 months. In the figure, the attractions are indicated by filled circles and the mappings indicated by straight lines emanating from the centroids. Coordinates of the centroids are computed by the wighted sums as described above. Attractions filled with the same colours are in the same cluster (i.e., they having the same centroid mapping) according to the heuristic.

7. EXPERIMENTS

7.1 Dataset

We collaborated with a theme park developer in an Asian city to collect data from their visitors. Our dataset contains the visitors' trajectories for the first 4 months of 2014. The dataset comes from an attraction bundling scheme marketed by the developer under which visitors can select any $Q = 4$ attractions out of a set of 16 and pay upfront a fixed price (independent of their choice). Visitors can redeem their chosen attractions on a chosen day and during a specified period from 9 a.m. to 7 p.m. of the day only. Each chosen attraction can only be redeemed once.

The dataset contains the trajectories of $n = 6,400$ unique and independent visitors (i.e., if a visitor is observed to have

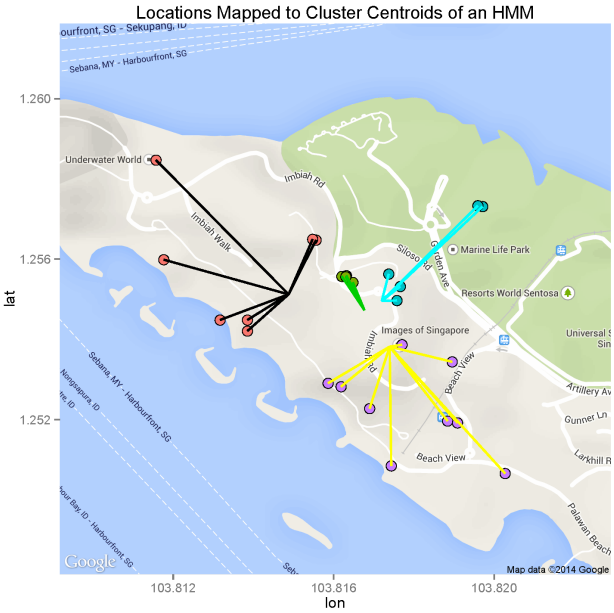

Figure 2: Visualisation of spatial locations (attractions) being mapped to their nearest centroids (reference points) derived from the hidden states of a 4-state HMM. Four states of the HMM form four distinct clusters of locations with each having a unique centroid as illustrated. Mappings are indicated by straight lines emanating from the respective centroids. The HMM was fitted using real-world trajectories collected from a theme park.

travelled in a group of the same trajectories, we take only one member of the group). We also have certain demographic features of the visitors, which are not discussed here for brevity. Table 1 summarizes the sequence length (l) and the first timestamp (τ_1) variables of the dataset. It shows that not everyone managed to redeem all 4 attractions they had chosen, although the majority did. Indeed, about 74.69% of our visitors managed to redeem all 4. Variable τ_1 measures the number minutes since the reference time (9 a.m.) to the first redemption, which can partially explain: while those who arrived early enough could redeem all 4, while those who came "late" couldn't – they had met their budget constraint (their ticket expired at 7 p.m.).

	Min.	Q_1	Median	Mean	Q_3	Max.
l	1.00	4.00	4.00	3.78	4.00	4.00
τ_1	8.57	173.60	254.70	259.70	343.10	604.90

Table 1: Summary statistics of the sequence length (l) and first timestamp τ_1 variables. Q_1 and Q_3 means the first and the third quartile, respectively.

7.2 Baseline Methods

We use the following baseline methods for comparison. In all of our experiments, we base our predictions on the knowledge of the first redemption of some form. The first baseline is to select 3 unique attractions randomly out of the set of 15 (16 less one) given the first attraction in the bundle. We call it the **Random** baseline. The second baseline is to choose $k = 3$ (physically) nearest attractions to the first redemption, which we call the **k-NN** baseline because it is essentially the k-nearest neighbours algorithm.

The third baseline is based on a recent method proposed by Mathew *et al.* [10] to predict future locations of a mobile agent based on past and current trajectories. The method can be concisely described as follows: (1) Cluster the set of trajectories into K clusters (something similar to Sect. 4); (2) Train a separate HMM_k for each cluster k; (3) Given a test agent i, his class label Cl_k^i, and the current trajectory, derive the most likely current state S_t^i of the HMM_k that i is in using Bayes' rule; and (4) Using the forward algorithm, derive the next sequence of 3 most likely locations conditioned on S_t^i. In our case, the current trajectory is simply the first known location and timestamp. We call this the **HMM** baseline because it is heavily based on HMM inference. Refer to [10] for full details of the method.

7.3 Proposed Methods

Our first two methods are the implementations the proposed framework using the two heuristics: first-location and centroid heuristic. We call them **VR1** and **VR2** respectively. ("VR" stands for value ratio, which is the central concept of the proposed solution.) For VR2, given an agent i's first location y_1^i, we map that to the nearest centroid r_j^i to derive p_j^i. By doing so, we do not need to know the explicit information of y_1^i, but which centroid it is nearest to. We call this the *implicit* information of y_1^i.

The third method is the partial implementation of our proposed framework using the centroid heuristic. Instead of using the full set of centroids derived from the hidden states of an HMM, we take randomly a fraction of that. In particular, given a fitted HMM_k, we select randomly 60% of the number of states of HMM_k to derive a partial set of centroids C_k'. Our rationale for this is to empirically estimate the optimality of the full set of centroids, i.e., we want to see how much the accuracy will be decreased (if any) if a partial set of centroids is used for predictions. In other words, we are asking whether the full set of centroids is an optimal set or can we achieve the same level of accuracy using less information? We call it **VR3** for convenience.

For these methods, we derive a test agent i's class label Cl_k^i using a decision tree trained on their demographic features and first timestamps. We do not discuss it here because it is off the focus of the paper. Budget constraint B_i is calculated as the remaining time from their first timestamp until 7 p.m. Finally, it is worth stressing that for all the above methods (including the baselines), except for VR2 and VR3, explicit information of the first location was used for make predictions; hence, the task reduces to predicting 3 locations out of 4. Whereas for VR2 and VR3, implicit information of the first location was used; thus, the prediction task remains that of predicting a full bundle given incomplete information.

7.4 Evaluation

For each agent $i \in \mathcal{T}$, let x_i^* and $\hat{x}_i$ be i's actual and predicted bundle, respectively. Note that x_i^* and $\hat{x}_i$ may not be of the same size. We construct a weighted complete bipartite graph $G = (U = x_i^*, V = \hat{x}_i, E)$ where each edge $e = (x_{ij}^*, \hat{x}_{ik}) \in E$ is weighted by the physical distance between any pair of locations $x_{ij}^* \in x_i^*$ and $\hat{x}_{ik} \in \hat{x}_i$. Denote

the weight of e as $w(e)$. Let $\delta(x_i^*, \hat{x}_i)$ be the distance between x_i^* and $\hat{x}_i$, we calculate $\delta(x_i^*, \hat{x}_i)$ using Algorithm 1. The rationale for using physical distance as the benchmark for prediction accuracy is because our costs are approximated by such distances. Also because businesses of similar nature tend to cluster geographically in real life, two locations are likely close semantically if they are physically close.

Algorithm 1 The evaluation procedure

1: $\delta(x_i^*, \hat{x}_i) \leftarrow 0$
2: **while** $|U| > 0$ and $|V| > 0$ **do**
3: $\quad e^* \leftarrow \min_e(E)$
4: $\quad \delta(x_i^*, \hat{x}_i) \leftarrow \delta(x_i^*, \hat{x}_i) + w(e^*)$
5: $\quad E \leftarrow E - e^*$
6: **end while**

Using Algorithm 1, we calculate the distance $\delta(x_i^*, \hat{x}_i)$ for each agent $i \in \mathcal{T}$. To evaluate all the predictions, we take the mean and median distance ($\bar{\delta}$ and $\tilde{\delta}$ respectively) over all $\delta(x_i^*, \hat{x}_i)$. Hence, the lower $\bar{\delta}$ (or $\tilde{\delta}$) is, the more accurate our predictions are on the whole.

7.5 Results

Our trajectory clustering results in $K = 4$ clusters (or class labels) using the interval $\Delta_\tau = 5$ minutes (refer to Sect. 4) for all the agents. The value of K was chosen based on our domain knowledge of the dataset. Fig. 3 visualises those 4 clusters. The horizontal axis of each cluster represents the discretised timeline (by Δ_τ) from 9 a.m. to 7 p.m. and the vertical axis represents the probability of each agent belonging to each class being in any one of the 16 attractions at any time interval. The attractions are identified by their unique ID's and colour codes shown in the legend at the bottom of the figure. We denote "0" (white) when we don't know the precise location of an agent during a period (i.e., he was not at any particular attraction during the time interval according to the data).

Fig. 3 shows that the 4 clusters have rather distinct temporal behaviours: Cl_3 has its peak of activities the earliest, which is followed by Cl_1, then Cl_4, and finally Cl_2. This suggests the existence of 4 different "waves" of visitors that flow through the attractions in the park, from entering, peaking, and exiting, one after another. Visually, Cl_3 are the "early birds" and Cl_2 are "latecomers". We also observe certain differences in the preferences for the attractions across the clusters represented by the probabilities of attraction visits. However, these differences are not very distinguishing on the whole: popular attractions remain (more or less) popular across the clusters and unpopular ones remain unpopular. This is particular true for clusters 1, 2, and 4; while for cluster 2, there is a sudden surge in demand for attraction 7 towards the end, which distinguishes it more from the rest.

For each of the methods described in Sect. 7.3, we perform a 10-fold cross-validation to measure its accuracy on predicting bundles. For each fold, we compute the mean $\bar{\delta}$ and median distance $\tilde{\delta}$ of the predictions as described in Sect. 7.4. We finally compute the average accuracy (i.e., the mean of both $\bar{\delta}$ and $\tilde{\delta}$) over the 10 folds for each method. Fig. 4 shows the mean and median accuracies of all the methods considered averaged over their 10-fold cross-validation.

Fig. 4 shows that our proposed methods (VR1 – VR3) have the most accurate predictions (lowest distances) on average. In particular, the proposed method (VR2) is more accurate than the baselines by at least 20% (i.e., comparing to HMM). The baseline methods are (in the order to decreasing accuracy): HMM, k-NN, and Random, which is not surprising because that is also the decreasing order of their sophistication. Remarkably, using implicit information (VR2), we have achieved as much accuracy as using explicit information (VR1). This empirically supports our centroid heuristic: we only need to know implicitly where an agent intends to visit to make a good enough prediction. At the same time, the centroid heuristic requires much less information to make inferences (i.e., N^* cluster centroids as opposed to the full 16 first locations as in VR1, where N^* is in the range 7–9 in our experiments).

Another notable observation Fig. 4 is that randomly selecting 60% of the set of centroids (VR3) does make predictions less accurate, even though by a small amount (for both the mean and median distance). This shows that the full set of centroids is indeed an optimal one such that using less information (VR3) leads to decreased accuracy and using more information (VR1) does not increase the accuracy. On the other hand, while VR3 is technically less accurate than VR2, the difference is really small (as shown in the figure) compared to the reduction in information requirement (VR3 requires 40% less information than VR2). This suggests that our proposed centroid heuristic is also quite resilient to missing information as long as we get most of the reference points right.

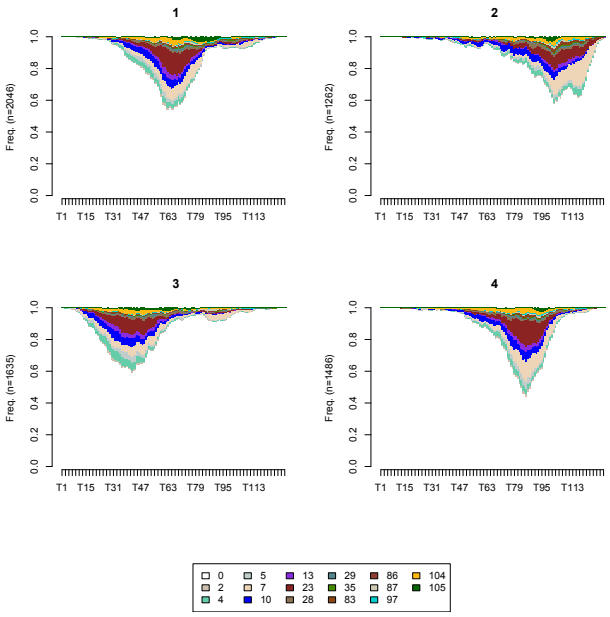

Figure 3: Visualisation of 4 clusters (a.k.a. "class labels") 1–4 of the trajectory data used in the experiments. Horizontal axes represent the timeline in discrete intervals of 5 minutes from 9 a.m. to 7 p.m. Vertical axes represent the probability of the visitors of each class being in each of the 16 attractions (or at some unknown location "0"), represented by their corresponding colour codes whose legend are shown at the bottom of the figure.

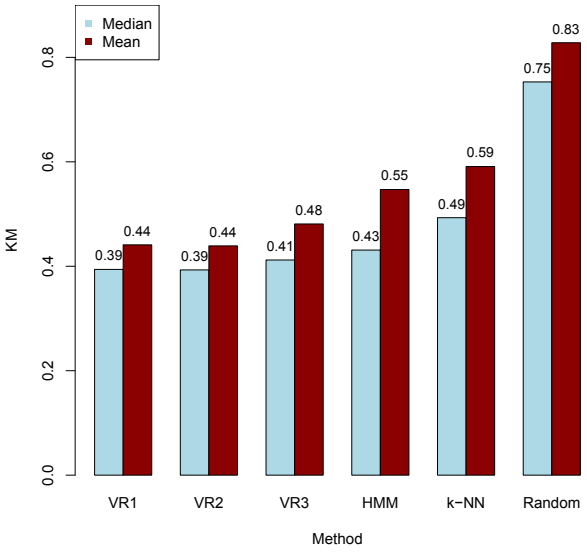

Figure 4: Accuracies of all the methods considered averaged over 10-fold cross-validation. Accuracies are measured by the mean (median) distance between predicted and actual bundles in kilometres (KM). Thus, the lower the distance, the higher the accuracy. Our proposed methods (VR1 – VR3) give better accuracies on the whole compared to the baseline methods (HMM, k-NN, and Random).

8. CONCLUSION

In this paper, we have introduced the problem of predicting a bundle of goods, where the goods here are a set of spatial locations that an agent wishes to visit. We look at the problem from an economic point of view where agents choose their bundles by optimising the values of the goods considered over some utility function subject to their budget constraints. To this end, there exists a rich literature to address the problem called *revealed preference* (RP) analysis. The fundamental problem of RP analysis is to recover the unknown utility functions of the agents given observations of their purchased bundles at the prevailing prices and budget constraints. In this paper, we assume the agents have linear utility functions so that the problem reduces to recovering the vector of values of the agents for the goods considered. Motivated by a recent line of work that has established efficient algorithms for learning values from RP data, we adopt and adapt one such algorithm to solve our problem. We also blend in two important techniques from spatiotemporal analysis: trajectory clustering and location clustering in order to make the problem feasible in our particular setting where cost information is unobserved. For location clustering, we propose the centroid heuristic, in which we use HMMs to derive the reference points as cluster centroids based on where the agents use to infer their perceived costs. We experiment our proposed methods with real-world data collected from a theme park, our predictions are significantly more accurate than the baseline methods. We also see that the proposed centroid heuristic not only requires less information, but it is also resilient to missing information artificially induced in the experiments.

There are limitations to our work. First, we have only considered *unordered* sets of spatial locations (i.e., bundles); however, in reality, agents consume those spatial goods by visiting them in sequence, one after another. There is an intrinsic ordering nature of the goods that we have yet taken into account. As a result, comparing between predicted and actual bundles should also consider the sequential order of the goods. Second, the proposed problem and solution may not be applicable to predicting long sequences (both in quantity and geographically) as in such cases, agents typically decide their next future location based on the current one only and not on past locations (i.e., the Markov property). Such situations have been the traditional playground of Markov models, and particularly HMM. Moreover, long sequences would require solving a large knapsack problem, which is NP-hard. Third, we have yet to consider other classes of utility functions besides the linear class, such as the class of separable piecewise linear concave functions often used to model decreasing marginal utility. Finally, we have not been able to establish the relationship between the amount of information required to make predictions and the prediction accuracy. There is a potential benefit in knowing less and yet being able to predict well enough because of the cost of information acquisition. These are left for future work.

9. ACKNOWLEDGMENTS

This research is supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office, Media Development Authority.

Siyuan Liu was also supported by the Basic Research Program of Shenzhen (Grant No. JCYJ20140610152828686) and the National Basic Research Program (Program 973, Grant No. 2015CB352400).

REFERENCES

[1] S. N. Afriat. The construction of utility functions from expenditure data. *International Economic Review*, 8(1):67–77, 1967.

[2] L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171, 1970.

[3] E. Beigman and R. Vohra. Learning from revealed preference. In *Proceedings of the 7th ACM Conference on Electronic Commerce*, pages 36–42. ACM, 2006.

[4] C. Fraley and A. E. Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458):611–631, 2002.

[5] S. Gambs, M.-O. Killijian, and M. N. del Prado Cortez. Next place prediction using mobility markov chains. In *Proceedings of the First Workshop on Measurement, Privacy, and Mobility*, page 3. ACM, 2012.

[6] H. Gao, J. Tang, and H. Liu. Mobile location prediction in spatio-temporal context. In *Nokia Mobile Data Challenge Workshop*. Citeseer, 2012.

[7] S. Jiang, J. Ferreira, and M. C. González. Clustering daily patterns of human activities in the city. *Data Mining and Knowledge Discovery*, 25(3):478–510, 2012.

[8] S. Kisilevich, F. Mansmann, M. Nanni, and S. Rinzivillo. *Spatio-temporal clustering*. Springer, 2010.

[9] S. Lahaie. Kernel methods for revealed preference analysis. In *ECAI*, pages 439–444, 2010.

[10] W. Mathew, R. Raposo, and B. Martins. Predicting future locations with hidden Markov models. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 911–918. ACM, 2012.

[11] D. Qiu, P. Papotti, and L. Blanco. Future locations prediction with uncertain data. In *Machine Learning and Knowledge Discovery in Databases*, pages 417–432. Springer, 2013.

[12] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[13] P. A. Samuelson. Consumption theory in terms of revealed preference. *Economica*, pages 243–253, 1948.

[14] M. Szenberg, L. B. Ramrattan, and A. A. Gottesman. *Samuelsonian economics and the twenty-first century*. Oxford University Press, 2006.

[15] M. Zadimoghaddam and A. Roth. Efficiently learning from revealed preference. In *Internet and Network Economics*, pages 114–127. Springer, 2012.