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

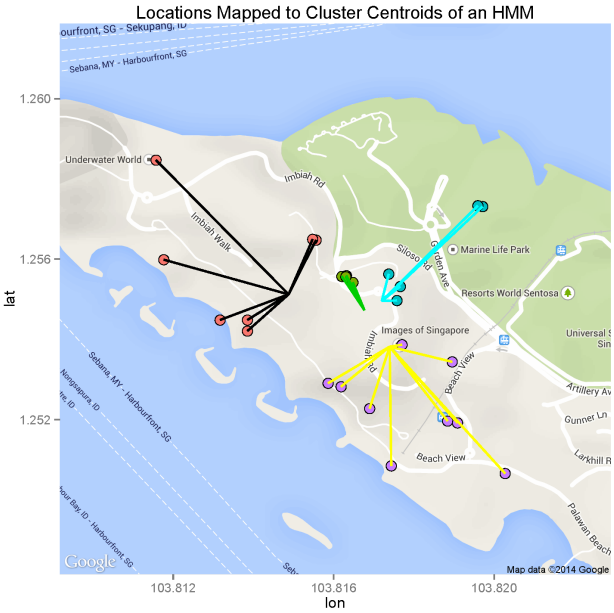

Figure 2: Visualisation of spatial locations (attractions) being mapped to their nearest centroids (reference points) derived from the hidden states of a 4-state HMM. Four states of the HMM form four distinct clusters of locations with each having a unique centroid as illustrated. Mappings are indicated 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.

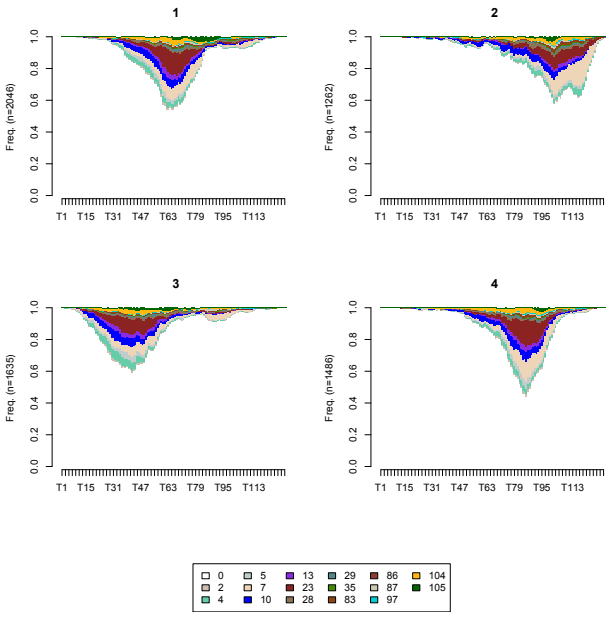

Figure 3: Visualisation of 4 clusters (a.k.a. "class labels") 1–4 of the trajectory data used in the experiments. Horizontal axes represent the timeline in discrete intervals of 5 minutes from 9 a.m. to 7 p.m. Vertical axes represent the probability of the visitors of each class being in each of the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

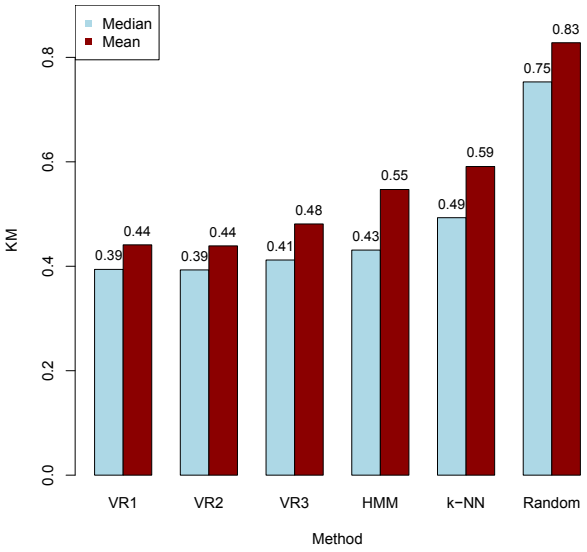

Figure 4: Accuracies of all the methods considered averaged over 10-fold cross-validation. Accuracies are measured by the mean (median) distance between predicted and actual bundles in kilometres (KM). Thus, the lower the distance, the higher the accuracy. Our proposed methods (VR1 – VR3) give better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