Discovering Latent Cultural Patterns for City Planning

*The final project report of CPSC 8650

Zirou Qiu
School of Computing
Clemson University
Clemson, SC, U.S.A
zirouq@clemson.edu

Diejie Gao
School of Computing
Clemson University
Clemson, SC, U.S.A
diejieg@clemson.edu

Yixian Li
School of Computing
Clemson University
Clemson, SC, U.S.A
yixian@clemson.edu

***Abstract*—Ideally, a city should provide the most relevant cultural establishments to the people in the near regions while maintaining a reasonable cost of allocation. However, because of the geographic limitation of the urban landscapes and the complicated demographic structure of large cities, it is likely that the current planning of venues is not optimal. To enrich the cultural activities of the citizens while decreasing the cost, it is important to identify the hidden patterns of urban-cultural interactions of a given city, compute the active ranges of users and eventually, determine the current demand-supply ratios of venues in the city. In this paper, we analyze the longitudinal dataset of user check-ins in New York City. Based on the algorithms proposed in [1], we first implement the temporal Dirichlet allocation model to discover latent cultural patterns of people and venues. Then we identify the active range and centers of users using the POPTICS algorithm [1] which is an extension of the traditional OPTICS. At the same time, we compute the demand-supply ratios of venues to gain a broad understanding of the current allocations. Last, the optimization is formulated as a linear assignment problem and several solvers are suggested.**

***Index Terms*—urban planning, temporal data, latent Dirichlet allocation, OPTICS, demand-supply-ratio, optimization**

I. Introduction

Under the trend of urbanization, we are witnessing a migration of people from provincial areas to urban areas, seeking higher living qualities with richer cultural activities. As a result, a fully functional city should not only offer employment opportunities but also satisfies the cultural needs of its citizens. However, it could be complicated to devise a city for which satisfies demands of all citizens due to their different preferences. On top of that, reallocating old venues and building new ones all come at costs. To find a balance between the enrichment of cultural activities and the financial burden, there are many aspects which we should first consider. For example, the most frequent locations where users visited in the past few months, the hidden cultural patterns of those users and the latent pattern of the cultural venues for which they visited.

For this project, we focus on the cultural resource allocation of New York City. We use the data set provided by FourSquare which consists of the venue check-in history of users. After exploring many related works on pattern extraction, spatial data clustering, and optimization, we choose algorithms suggested by Zhou *et al* [1]. The main goal of our project is to discover the hidden cultural patterns of users and venues in New York City, identify which venues are mispositioned (the demand-supply ratio is too high/low), and give suggestions for further allocation of cultural establishments. In more details, we achieved the following:

- **Cultural pattern extraction**: First, we identify a total of nine cultural patterns in New York City. We implement the extended version of LDA mode, namely TLDA, proposed by Zhou *et al*. [1] which determines the patterns based on the temporal information on when a certain venue was visited. Given the user check-in history of New York, we feed the data to TLDA and identify such cultural trends. To quantify the quality of results, we employ the metric introduced by Zhou *et al*. [1].
- **Users check-ins clustering**: For each user, we partition his/her check-in data (coordinates with latitudes and longitudes) into clusters. Within a cluster, we determine its center and radius. On top of that, given a user with a particular cultural pattern (generated from the previous step), we extract all his/her check-ins whose corresponding venues fall under this cultural pattern and compute the sub active range. At an implementation level, we use the POPTICS algorithm proposed by Zhou *et al*. [1] which is an extension of the traditional OPTICS algorithm. The POPTICS algorithm automatically set the cluster threshold and calculate the radius and centroid of each cluster.
- **Demand-supply ratio computation**: Based on the central location and the sub-active ranges of each users check-ins, we use the Gaussian function suggested by Zhou *et al*. [1] to compute the demand level and supply level of each venue. Then we calculate each venues *demand-supply ratio* (DSR) to determine if its current location is close to optimal.
- **Urban Planning Optimization**: We formulate this prob-

lem into an optimization problem and suggests existing algorithms to solve this problem.

We finished all the steps and approaches for which we mentioned in the proposal. This report is organized as follows. Section II introduces some related works on pattern extraction and clustering. Section III provides detailed information about our data set. Section IV presents the TLDA model and its evaluation metric. Section V describes the algorithm for clustering and computing DSR score. Section VI consists of our optimization approach. Section VII presents the experimental results. Section VIII suggests the future work. Section IX concludes our report.

II. Related Work

Hidden pattern extraction and density-based clustering are topics that have been studied by many researchers. To start with, Blei and his colleges formalized the revolutionary latent Dirichlet allocation [3] with an intuitive document-topic model. Geurts suggested a pattern extraction model for time series data for which is used to construct the future classification rules [4]. Kurashima *et al*. introduced the Geo Topic Model which identifies patterns based on the geographical characteristics of venues and users [5]. It is also an extension of the traditional LDA model. Yin *et al*. proposed a recommendation system called *LCARS* for which recommends venues to users by studying users' interests [6].

For density-based clustering, one of the most well-known ones is DBSCAN [7]. DBSCAN clusters points that are close to each other (high density) well discard outliers (low density) [7]. To extend DBSCAN, OPTICS was built on the same foundation and that data set with non-uniformed density can be correctly clustered [8]. Sander *et al*. suggested another way to improve DBSCAN for which they also consider non-spatial attributes [10]. Campello and his colleagues proposed a density-based clustering algorithm with the hierarchical nature based on the tree of significant clusters [9]. Among all these works, we chose to study Zhou et al. [1] because it is an integration of both pattern extraction and clustering. Also, the algorithms they proposed works smoothly with our data set.

III. Dataset

Foursquare is a large location collection company who provides city guides and venue check-in functionality to users. For this project, we use the NYC Check-ins data set provided by FourSquare [2]. It contains check-ins history of users in NYC over 10 months from April 2012 to February 2013. In total, 227,428 check-ins of people in New York City are included. For each check-in datum, the corresponding time stamp, GPS coordinates, and category of the venue are also given. Here is the detailed information of each datum:

- User ID (anonymized)
- Venue ID
- Venue category ID
- Venue category name
- Latitude
- Longitude Time zone
- UTC time

For our project, we utilize the user IDs, venue categories, latitudes, longitudes and time.

IV. Cultural patterns extraction

In this section, we first justify the appropriateness of using TLDA on our data set, then present the way of extracting cultural patterns using TLDA. To determine the best number of patterns, we use the metric proposed by Zhou *et al*. All code can be found in the attached folder.

A. Temporal Factors Justification

The TLDA model extends the traditional LDA model for which it considers the temporal features of check-in data. To justify the correctness of the TLDA algorithm, Zhou *et al*.[1] made a key assumption such that cultural patterns are indeed related to temporal data. For example, in a day, venues are visited the most only during a specific time. To show that TLDA can be applied to our New York data set, we analyzed the temporal features of each users check-in data, studied their periodicity of cultural check-ins and constructed the heap maps of check-in distributions of May, July, October, and December, respectively. We chose these four months because each of them represents a meaningful period of the year. July and December correspond to summer break and winter break, respectively. May could potentially be the busiest month right before the summer break, and October is in the middle of the second half year.

The heat maps are shown in Figure 1. with dates lie on the x-axis and hours (time of the day) on the y-axis. The darker the grid, the higher volumes of the check-in activities. The periodicity of cultural check-ins is quite obvious which also conveys the nature of New York City. For each month, the most likely hours for cultural check-ins is during the second half of the day, roughly from 12 pm to 12 am. On the contrast, the least likely hours for check-ins are from 6am to 10am. This does not come as a surprise since New York City is known for its nightlife.

The periodicity of check-ins is less obvious in October[1] than other months. After Taking a closer look, we found that the heat map also correctly indicates periodicity of check-ins with respect to holidays. For example, the column corresponding to the Christmas day is nearly blank. This is explainable since people want to stay with families on that day instead of going out. Overall speaking, the temporal cultural patterns is significant which implies that we can apply TLDA on this data set.

B. Temporal Latent Dirichlet Allocation

Recall that in LDA, Given M documents with each containing N words (the number of words is determined by a Poisson distribution), two Dirichlet prior, α and β, are first generated, and then for each document d, the corresponding topic distribution θ_d is determined based on α [3]. For each

[1] the columns which correspond to the first several days in October is empty. This is due to data loss

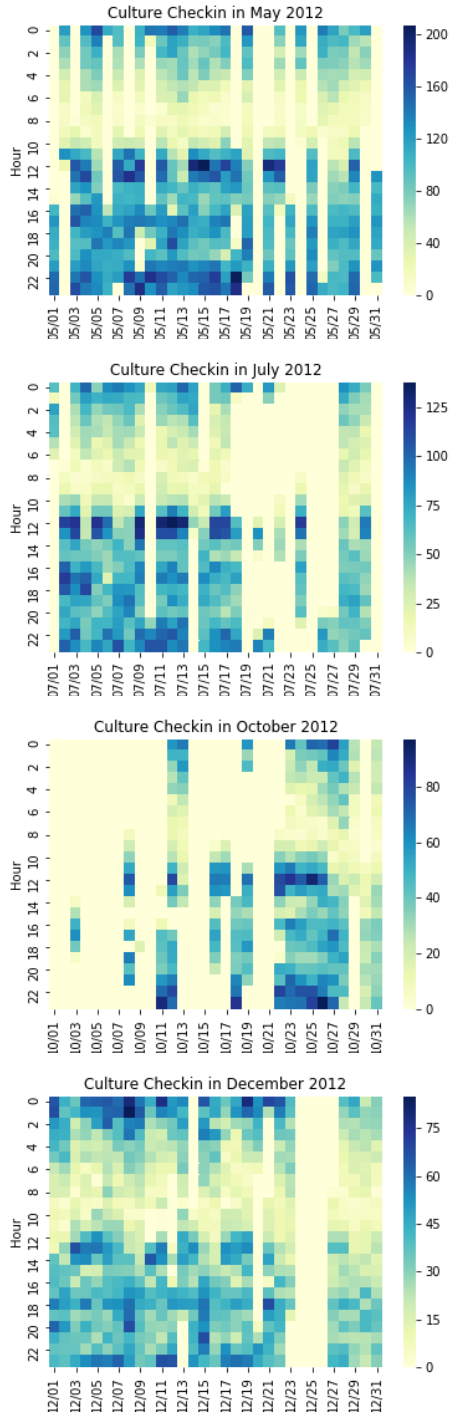

Fig. 1. Heap Maps of Check-in Distributions of Four Months in New York City

word w, a topic z is determined from the multinomial distribution over θ_d, then the content of this word is generated from the multinomial distribution conditioned on z and β [3]. Given a corpus with documents, the only observable data are words. Therefore, LDA discovers the two hidden distributions: document-topic distribution and topic-word distribution. To determine the distributions, we can use traditional methods such as Gibbs sampling.

Based on the algorithm suggested by Zhou *et al.*[1], we implemented the TLDA which incorporates the temporal factor into the LDA model and no longer built solely upon the bag-of-words assumption. In TLDA model, a pattern is determined by an extra Dirichlet prior γ over the time-pattern distribution [1]. As a result, each user/time is a mixture of cultural patterns (user/time-pattern distributions), and each cultural pattern is a mixture of venue categories (pattern-venue distribution). Figure 2 presents TLDA in plate notation [1].

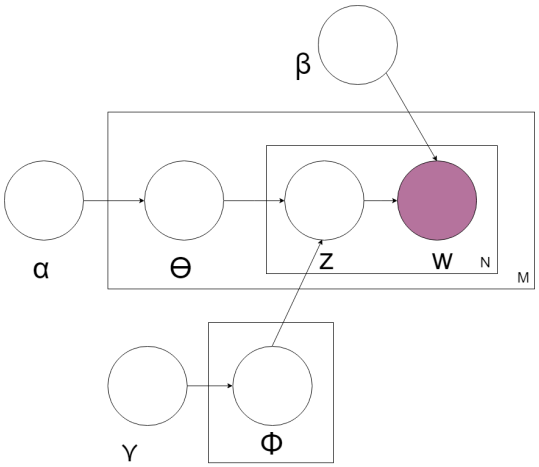

Fig. 2. Plate notation of TLDA

After applying the Gibbs sampling algorithm, we obtain three matrices: user-pattern matrix, time-pattern matrix and pattern-venue matrix.

C. Determine the Number of Patterns

Because of the unsupervised nature of TLDA (and LDA), we need to know the number of patterns beforehand. However, we do not possess such a knowledge of our data set (in fact, this is true for most cases). To evaluate the quality of our choice of the number of topics, we implement the algorithm for which computes the temporal coherence value (TCV) of a TLDA model. Intuitively speaking, TCV is the average value of the coherence level between cultural venue categories and time in each pattern. The score ranges from 0 to 1, and a TCV close to 1 indicates the high appropriateness of our number of clusters [1].

To evaluate the quality of a TLDA model, we first need to extract the set of top venue categories V^* and top time period T^* for each pattern. In the nutshell, each top venue category is segmented into a tuple consisting three elements: the venue category itself v^*, the set of top venue categories where v^* belongs to, and the set of top time periods [1]. Then we construct two vectors, w and W, out of each segmentation. For each segmentation, we compute the *normalised point-wise mutual information* (NPMI) between the top venue category and each time period in the set of top time periods. These

NPMI values are elements of vector w. The formula we used is suggested by Zhou et al.[3]:

$$NPMI(v^*, t_j^*)^a = (\frac{\log \frac{P(v^*, t_j^*)+b}{P(v^*)P(t_j^*)}}{-\log((v^*, t_j^*)) + b})^a \quad (1)$$

where t_j^* is a time period in the set of top time periods and $P()$ denote the probability. b is positive constant factor which always makes the numerator non-zero, and a increases the relative weight of high NPMI values [1]. Based on experimental data, we set b to 0.001 and a to 2.

At the same time, for each top time period t_j^* in a segmentation, we sum up its NPMI values with all $v*$ in $V*$ to obtain the accumulated NPMI value. These aggregated MPMI values are entries of vector W. Then we compute the cosine similarity between w and W. For clarity, each segmentation has a corresponding cosine similarity between its w and W. Finally, we accumulate the cosine similarities over all segmentations and compute their average. The resulted score is the TVC value for this specific TLDA model [1].

Based on the TVC values of TLDA models with a different number of patterns k, $k = 9$ yields the highest TVC value which indicates that cultural activities in New York City should be classified into nine groups. After running the TLDA with $k = 9$, we obtained the corresponding matrices and gained the knowledge of the users cultural preferences and cultural groups of venues. The detailed experimental results are presented in the experimental section of this report.

V. Cultural Venues Planning

In this section, we describe the approach of clustering the users' check-in data and determine the demand-supply ratio of venues. It is commonly accepted that the number of venues planned at a particular area should be proportional to the local population. In other words, the more people, the more venues. On top of that, we also want to make sure that the types (patterns) of venues are desired by the users. Therefore, it is important to understand the active ranges of users and their demands of venues. All code can be found in the attached folder.

A. Urban cultural planning based on culture pattern

The demand level of one venue is proportional to the probability of users going to that venue. Intuitively speaking, a user is more likely to visit venues lie in his/her active range. Therefore, we need to find such ranges of users based on their culture patterns. The data set we use have the check-in location information about latitude and longitude. Among the cluster methods, we can get the spatial range of the user activity based on those check-in data and calculate the center and radius of the activity range. OPTICS [8] is a suitable method to cluster the spatial information based on the density. And for our data set, it is not reasonable to use the same minimum points(a parameter of the OPTICS) due to the wide difference check-in numbers between users. Therefore we use the POPTICS [1] which is a modified version of OPTICS and can automatically decide the minimum points (the threshold) of clusters for user separately.

First, we clean the check-in data based on the user pattern. For each user, we delete the check-ins of the venue not belonging to the user culture pattern. And the number of minimum points is set in advanced based on the entire users check-ins and the experiment result. Then we run POPTICS to cluster the check-ins for each user. To calculate the core distance in the algorithm based on the latitude and longitude, we set the core distance as Euclidean distance between two venues using Haversine formula [13]:

$$CD(v_i) = Edist(v_i, v_j) : j = 1, 2, ..., N \quad (2)$$

$$Edist(v_i, v_j) = R \times 2\tan^{-1} \times \lambda \quad (3)$$

$$\lambda = \sqrt{\sin^2(\frac{a-c}{2}) + \cos(a) \times \cos(c) \times \sin^2(\frac{b-d}{2})} \quad (4)$$

where $v_i = (a, b)$, $v_j = (c, d)$. a, b, c, d are the radians of latitude and longitude of venue i and j, and R is the radius of earth who is set to 6,372,800 meters.

Then we set he reachability distance as [8]:

$$RD(v_i, v_j) = \max\{CD(v_i), Edist(v_i, v_j)\} \quad (5)$$

and

$$RD(v_j) = \min_{v_i \text{is a core point}} \{RD(v_i, v_j)\} \quad (6)$$

To automatically cluster the check-ins points, we calculate the values of all reachability distance and the lower the value the better the threshold is to detect outlier. Finally, we can get cluster of users. And the center and radius of the cluster is defined as:

$$center = \frac{\sum_{i=1}^{n}(a_i, b_i)}{n} \quad (7)$$

and

$$radius = \frac{\sum_{i=1}^{n} Edist(p_i, center)}{n} \quad (8)$$

where (a_i, b_i) are coordinates of the i^{th} point and p_i is the i^{th} point. The results of POPTICS are shown in the experiment section.

B. Demand-supply ratio computation

In this section, we construct the demand-supply model using Gaussian function. The demand and supply levels are based on the results from POPTICS. The Gaussian function we use is based on the work by Zhou *et al*, however, our detailed approach of determining which user should contribute to the demand/supply level of a particular venue is different from theirs.

The demand level of one venue is defined by the probability of users going to that venue. First we know the clusters of each user from POPTICS, and each cluster has a center μ and radius r_u. Then we determine that level based on the assumption suggested by Zhou et al such that for a certain user, his/her demand for a particular cultural pattern (his/her pattern) is highest at the cluster center and decrease as we move away

from the center[1]. In general, the demand level of the venue x for user u is computed using the formula suggested by Zhou *et al.* [1]:

$$d_u(x) = \frac{1}{\sqrt{2\pi r_u^2}} e^{-\frac{(x-\mu)^2}{2r_u^2}} \tag{9}$$

An example is shown in Figure 3 for which the user u has two clusters enclosed by blue circle and red circle. Each cluster contains several venues where user had been to. To calculate the demand level of venue x, we first determine the cluster the encloses the venue x. In this example, the venue x is in the blue cluster, then we use the blue clusters center and radius to compute the demand level.

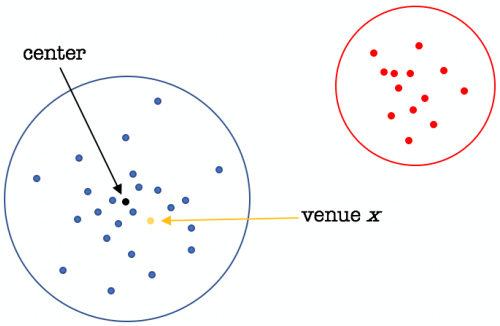

Fig. 3. The demand level of user u on venue x

Note that for each user, we filter out the venues that does not belong to the pattern of this user before passing them to the POPTICS. Therefore, r_u is the active range of user u for his/her pattern. The demand for venue x of its own pattern is the sum of demands of all users who have visited x.

The supply level of a venue x is defined by the service capability of that venue. The assumption is that the service of a venue x can reach a user u if u has visited x before [1]. Therefore, we determine the set of users who has visited x at least once and compute the average distance, denoted by σ, between their active centers (output of POPTICS) and the location of x. The Figure 4 illustrate an example of computing the distance between centers of uses and x. In this example, there are four users had check-ins at venue x. We calculate the distances between their centers with x.

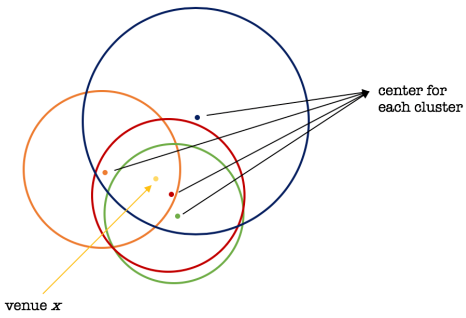

Fig. 4. An example of computing the supply level of x

The formula to compute the supply level of x for a user u is defined as:

$$d_u(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{10}$$

This definition is different from Zhou *et al*'s approach for which they defined the supply level of an area x based on the venues in that area. On the other hand, we define the supply level of a venue based on its capability of serving users. The overall supply level of a venue x is the sum of its supply levels to all users who have visited x before.

At last, we compute the demand-supply ratio (DSR) of a venue x based on the formula [1]:

$$DSR(x) = \frac{D(x)}{S(x)} \tag{11}$$

where $D(x)$ is the overall demand level of x and $S(x)$ is the overall supply level of x. If the venue is at the "perfect" location, the DSR is equal to or approximate to 1. The venue location is not ideal when the DSR is too high or too low. More specifically, a DSR who is significantly lower than 1 indicates the surplus of supply where a DSR higher than 1 indicates a shortage of supply.

VI. Urban Resource Optimization

In this section, we present ways to formulate the problem into a linear assignment problem. As the same time, we suggest candidate algorithms for solving the problem in stronly polynomial time. We approach this problem from the theoretical perspective since the real cost in USD of planning a venue cannot be easily determined. However, our model is universal.

A. Problem definition

In a nutshell, we have a set of venues A with corresponding DSR values that deviate from 1 significantly. Given a set of locations B for which $|A| = |B|$, we want to reallocate venues in A into locations in B such that the resulted DSR after reallocation for each venue is more optimal (closer to 1). Similarly, A could be the set of new venues that are waiting to be built, and we want to plan venues in A into locations in B.

The problem becomes less interesting and unrealistic if the costs of reallocating/building venues are zero. Therefore, we assume that there exist a unique cost of planning venue a_i to location b_i for every venue-location pair. Such a cost could be determined by multiple factors such that the geographic location, nature of the venue, targeted customer groups ect. Now we are ready to present the formal problem definition.

Problem definition 1: Given two sets, A and B for which $|A| = |B| = n$. Let $C : A \times B \longrightarrow \mathbb{R}$ be the weight function. We want to find a bijective mapping (both one-to-one and on-to) $M : A \longrightarrow B$ such that

$$\sum_{a \in A} C(a, M(a))$$

is minimized.

Note that we want to minimize the cost of construction while maximize the improvement of DSR for each venue, therefore, the cost function C should takes both factors into the account. One naive approach is to define the cost function C as:

$$C(a,b) = \frac{c(a,b)}{\Delta DSR(a,b)} \quad (1)$$

where $c(a,b)$ is the construction cost of moving venue a to location b and $\Delta DSR(a,b)$ is the improvement of DSR of a by moving to b. Note that because of the difference in units, the value of construction cost should be much greater than the value of $\Delta DSR(a,b)$ which makes it mathematically insignificant. Therefore, we could either normalized the construction cost over all construction costs, or somehow transform the improvement of DSR value into financial benefits in USD.

In the matrix notation, let C be the $n \times n$ real matrix for which C_{ij} is the cost of reallocating the i^{th} venue to the j^{th} location. Let $S^{n\times n}$ be the set of all $n\times n$ permutation matrices, then our objective function can be rewritten as:

$$\min_{X \in S^{n\times n}} trace(CX^T) \quad (2)$$

and this problem is a linear assignment problem.

To further understand the scope of the problem, consider a complete bipartite graph $G = (V, E)$ with bipartitions A and B of equal size. Let C be our weight function defined on edges, then the problem comes down to *finding a minimum-weight perfect matching in* G. Without loss of generality, we assume that every venue in A can be assigned to any location in B so the bipartite graph is complete. We also assume that B is constructed such that any assignment of a venue a in A to any location in B improves the DSR of a. Note that the optimal solution X from objective function (2) is the biadjacency matrix of the subgraph induced by the optimal perfect matching.

B. Algorithms for solving this problem

The problem of finding the minimum weight perfect matching in bipartite graph is known to be able to solve in strongly polynomial time. The most famous algorithm is the Hungarian algorithm proposed by Kuhn in 1955 [12]. From a high-level perspective, the Hungarian algorithm involves iteratively crossing rows and columns with zero entries of the weighted matrix. If the number of crosses is less than the dimension of the matrix, we update the matrix by subtracting the smallest uncrossed entry from all uncrossed entries, and adding twice the smallest uncrossed entry to all the crossed entries. The algorithm terminates when the number of crosses equals to the dimension of the matrix, and the zero entries made up the optimal matching. The running time of this algorithm is $O(n^4)$ [12].

We can also formulate this problem as a linear program. Given the bipartite graph $G = (V, E)$ with bipartition of equal size, let x be the $|E| \times 1$ incidence vector of a matching such that $x_e = 1$ if e is in the matching and $x_e = 0$ otherwise. Let w be the $|E| \times 1$ weight vector such that w_e is the weight of edge e. Let A be the incidence matrix of G, then we can formulate our problem into an integer programming problem:

$$\begin{aligned} \min \quad & w^T x \\ \text{s.t.} \quad & Ax = \boldsymbol{1} \\ & x_e \in \{0,1\} \ : \ e \in E. \end{aligned}$$

Note that $Ax = \boldsymbol{1}$ guarantees that x is a perfect matching. We can drop the integer constraint to obtain the relaxed linear program. Because the incidence matrix of a bipartite graph is totally unimodular, and $\mathbf{1}$ is integral, the LP formation of the program is an integral polyhedron which means that all extreme points of these this polyhedrons are integer vectors and there exists optimal solutions to this LP which are integral. It is obvious that these optimal solutions are also optimal solutions to the corresponding IP problems. Therefore, we can use traditional LP algorithms such as the simplex method to find the optimal perfect matching in polynomial time.

Because we cannot determine the real cost function of planning venues, only theoretical approaches are suggested. In the future, we would like to investigate the construction cost of building venues and apply the algorithms to optimize real-world problems

VII. Experiments

In this section, we present the experimental results. All code can be found in the attached folder.

A. Data Preprocessing

The NYC check-in data set consists of 227,428 check-ins over 1,083 distinct users. The venues spans over 178 venue categories. We performed some test runs and found that the number of categories is too large. After examining the data, we observed that many venues are categorized into some sub-categories of main categories. For example, other than the main category Restaurant, many venues belong to other categories of restaurants such as Mexican Restaurant, Sushi Restaurant, Italian Restaurant, French Restaurant etc. Indeed, such a detailed classification could provide more insight, however, the resulted TLDA model is less interpretable because the distribution is spread across over 200 categories. Therefore, we combine all sub-categories into the main category. On top of that, we identify that some venue categories gets visited less than the average, and we also removed those categories with corresponding check-ins.

At the same time, as Zhou *et al*. suggested, we removed users who has less than 100 check-ins. As a result, we end up with a data set consists of 175893 check-ins over 782 distinct users. The final venue categories is 40.

B. Discovery of cultural patterns

To determine the number of clusters k, we tested different number from 2 to 10. For each potential k, we perform 2000 iterations of TLDA and compute the corresponding TCV values shown in Figure 5.

It is clear that $k = 9$ is the best number of clusters. We run the TLDA algorithm with $k = 9$ until convergence and

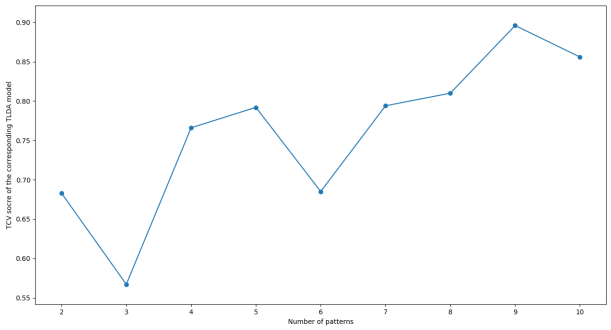

Fig. 5. TCV values of TLDA models with different number of clusters

obtain three matrices: user-pattern matrix (distribution), time-pattern matrix and pattern-venue matrix. Note that both user-pattern matrix and pattern-venue matrix are used to compute the demand-supply ratio. Given these distributions, we can determine which cultural pattern does a user prefer, and what venue categories belong to that pattern. The Figure 6 shows the top venue categories (with high probability) for each pattern.

Pattern (represented by the number)	Top venue categories
0	Outdoors, Park, Bus Station, Deli / Bodega
1	College Academic Building, Residential Building (Apartment / Condo), Medical Center, Neighborhood
2	Restaurant, Bar, Food & Drink Shop, Coffee Shop
3	Home (private), Bus Station
4	Plaza, Movie Theater, Museum, Coffee Shop
5	Office, Coffee Shop
6	Train Station, Bus Station, Subway
7	Gym / Fitness Center, Airport, Hotel, Coffee Shop
8	Clothing Store, Salon / Barbershop,

Fig. 6. Top venue categories for each pattern

Because TLDA has the fuzzy nature, some venue categories have high probabilities in multiply patterns. At the same time, some venue categories such as Electronics Store does not have high probability in any of our nine patterns.

Patterns are meaningful. For example, pattern 2 consists of venue categories that are highly related to dinning. Therefore, if a users main pattern is pattern 2, then it is likely that his/her primary demand of venues would be restaurants. Similarly, if a users main pattern is pattern 6, then he/she might be a working class who takes subway everyday. We also observe that the category coffee shop occurs with high probability in 4 patterns. This can be explained by the large number of coffee drinkers in the U.S. such that users from different patterns all drink coffee. Also, people visit coffee shops for social purposes.

TLDA gives us further insight about the connection between cultural patterns and temporal factors. As shown in the Figure 7 and Figure 8, each pattern is plotted by the compositions of time (day, time of the day). For better visualization, we group hours into time of the day: morning (6am - 10am), noon (11am - 3pm), afternoon (4pm - 7pm), evening (8pm - 11pm), night (midnight - 5am).

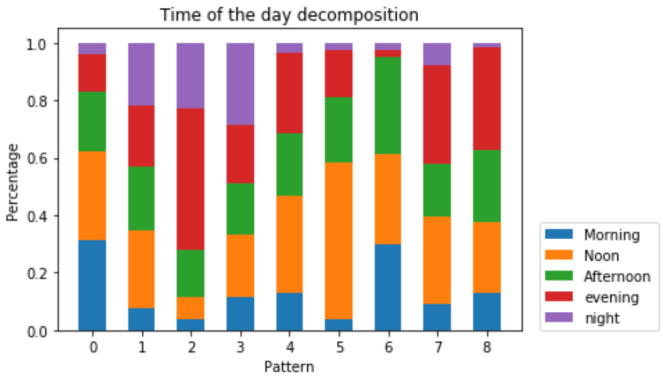

Fig. 7. Time of the day decomposition of patterns

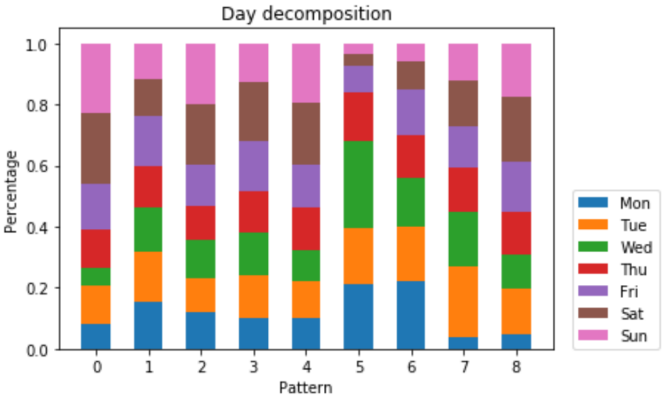

Fig. 8. Day decomposition of patterns

The time decomposition of patterns are explainable, and we select some representative patterns to justify its correctness. The group 0 (pattern 0) consists of users who like outdoor activities. For them, the highest visiting frequency occurs during the morning and the noon (the first column in the Figure 7). During a week, they are more likely to go outdoors on weekend than weekdays (the first column of the Figure 8). This is reasonable because people have free time on weekends and they are less likely to go outdoor during the night time (except for night running). The group 2 consists of people who like restaurants and bars. Interestingly, the most check-in time in a day for them is the evening and the night time (third column of the Figure 7). This could be due to the fact that many bars in NYC open at the night time and close in the morning. During a week, the probabilities are almost evenly distributed through the weekdays with a slight increase on weekends. People who work in offices fall into group 5. As we expected, noon and afternoon take up the majority of the fraction of pattern 5 (column five of the Figure 7). Also, the

frequency of check-ins is significantly lower on weekends than weekends. (column five of the Figure 8). Group 7 consists of gym lovers and travelers. Their most active time for them is the evening (column 7 of the Figure). This could because that many people get off the plane and check in the hotel at the night time. We also obtain some surprising findings. For example, the ninth group (pattern 8) are people who love shopping, and their most active times of the day are evening.

The above results suggest that patterns discovered by our TLDA model are meaningful. In the next section, we present the demand-supply ratio of venues for which are computed using the user-pattern matrix and pattern-venue matrix.

C. Clustering

There are 789 users in our dataset and the number of check-ins changes a lot between different user. As the POPTICS is sensitive to the number of minimum points, we have to do an experiment to get the correct the number of minimum points. But the relationship between the number of minimum points and the number of all check-ins are not liner. So, we can only get a balanced result of all users. Here are two examples of the cluster result of two users as Figure 9 shows, one has a little check-in and the other has a lot. Through the score we get, we can detect the outlier of the cluster, which is not in the user active range. The radius and center have been calculated while running POPTICS.

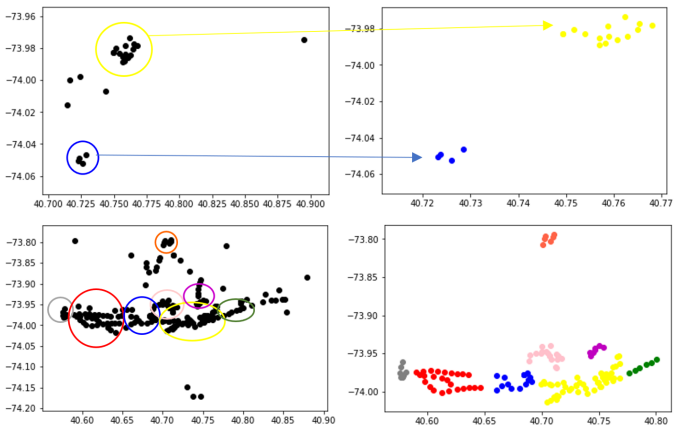

Fig. 9. Two example results of clustering

D. Demand-supply ratios

We compute the demand-supply ratio of 4512 venues as shown in Figure 10. Each venue is identified by its unique longitude and latitude.

To make a foundation for optimization in the next step, we analyze the DSR along with check-ins venues in the New York City google map. Parts of DSR are equal to 1, which means these venues are at the appropriate locations. One such an example is shown in figure 11. Some DSR are less than 1, which means there are too many this type of venues, and the demand level is smaller than the supply level. Figure 12 illustrate an example. Many DSR are larger than 1, which

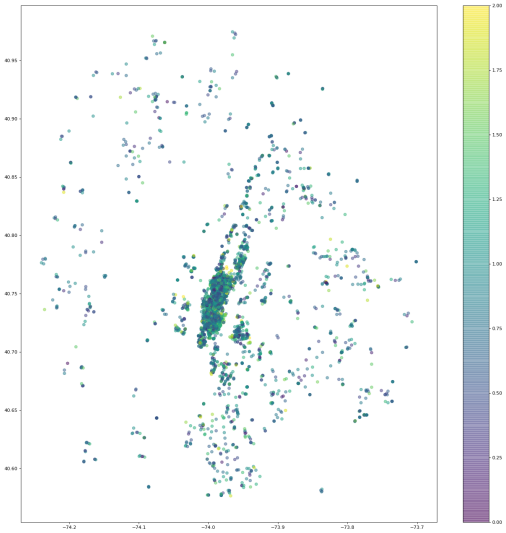

Fig. 10. DRS for 4512 venues

means there is too few this type of venues, and the demand level is larger than the supply level (Figure 13).

index	venue	DSR	category	pattern
		.....		
427	(40.755211, -73.827912)	0.5182932172554308	Restaurant	2
428	(40.757149, -73.836522)	0.5291915489692253	Food & Drink Shop	2
429	(40.751744, -73.836923)	1.0000000000002305	Park	0
430	(40.757083, -73.975952)	0.5826512058629260	Coffee Shop	2
431	(40.755052, -73.977077)	0.5637751161583770	Hotel	7
		.....		

Fig. 11. An example venue with perfect DSR

The venue at point (40.751744, -73.836923) has the DSR value around 1. This venue is a park and it has a good balance between the demand and supply.

index	venue	DSR	category	pattern
.....				
1722	(40.767477, -73.922882)	0.9999999999998779	Coffee Shop	5
1723	(40.777042, -73.921360)	0.3283310044871588	Restaurant	2
1724	(40.774640, -73.918439)	0.8237213791766961	Bar	2
1725	(40.780519, -73.920682)	0.6469824297555303	Park	0
.....				

Fig. 12. An example venue with low DSR

The venue at point (40.777042, -73.921360) has a low DSR value, which is much less than 1. This venue is a restaurant and intuitively there are more restaurants in this area than the actual needs.

index	venue	DSR	category	pattern
.....				
253	(40.765985, -73.789334)	1.0000000000299114	Bar	2
254	(40.762533, -73.974311)	19.044355059274523	Clothing Store	8
.....				

254	(40.762533, -73.974311)	19.044355059274523	Clothing Store	8

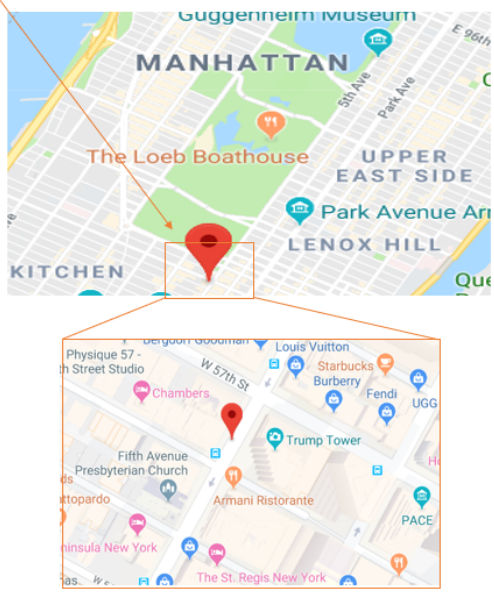

Fig. 13. An example venue with high DSR

The venue at point (40.762533, -73.974311) has a high DSR value, which is much more than 1. This venue is a clothing store and intuitively this area needs more clothing stores like this venue.

VIII. Future Work

The land area of New York City is around 302.64 square miles [11], therefore, there are a significant number of potential locations for which a venue could be planned. At the same time, determining the cost of constructing a venue involves multiple elements such that geographic locations, transportation fees, the nature of the venues, targeted customers etc. Also, it is very likely that the change of DSR of one venue could affect the DSR of other neighboring venues. These factors lead to a way more complicated assignment problem which could be hard (NP-hard) to solve. In fact, considering its size, it could computationally expensive to even approximate a solution. Plus, the real New York City network is dynamic such that old venues are being removed and new venues are being built constantly. On top of that, to come up with a realistic solution for the urban planning problem of New York City, a more recent data set with larger size and richer venues is needed. Due to our limited computational power and current time constraint, we will tackle these issues in the future. We will model the dynamic nature of New York City, acquire a larger data set and formulate the cost of planning. If possible, we will also explore possible opportunities to collaborate with the Department of Cultural Affairs of New York City.

IX. Conclusion

In this project, we have tackled the cultural pattern extraction and urban planning problem of New York City. By analyzing the NYC Check-ins data set, we have discovered meaningful cultural patterns, detailed active range of users, and supply-demand ratios of venues in New York City. On top that that, we identify mispositioned venues and formulate the urban planning problem into the linear assignment problem. At last, we suggest existing algorithms which can be used to solve the optimization problem.

Acknowledgment

We would like to thank Dr. Wang and Xiao Zhou [1] for their help on TLDA and POPTICS algorithms.

References

[1] X. Zhou, *et al.* "Discovering Latent Patterns of Urban Cultural Interactions in WeChat for Modern City Planning," Knowledge Discovery and Data (KDD), pp. 1069–1078, August 2018.

[2] D. Yang, *et al.* "Modeling User Activity Preference by Leveraging User Spatial Temporal Characteristics in LBSNs," IEEE Trans. on Systems, Man, and Cybernetics: Systems, (TSMC), 45(1), pp. 1069–1078, 2015.

[3] D. Blei, A. Ng, and M. Jordan, "Latent Dirichlet Allocation," Journal of Machine Learning Research, 3, pp. 993–1022, 2003.

[4] P. Geurts, "Pattern Extraction for Time Series Classification", Knowledge Discovery and Data (KDD), pp. 115–127, 2001.

[5] T. Kurashima, *et al.*, "Geo topic model: joint modeling of user's activity area and interests for location recommendation", Sixth ACM international conference on Web search and data mining, pp. 375–384, 2013.

[6] H. Yin, *et al.*, "LCARS: a location-content-aware recommender system", Knowledge Discovery and Data (KDD), pp. 221–229, 2013.

[7] M. Ester, H. Kriegel, J. Sander, and X.Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise", Knowledge Discovery and Data (KDD), pp. 226–231, 1996.

[8] M.Ankerst, M.Breunig, H.Kriegel, J.Sander, "OPTICS: Ordering Points To Identify the Clustering Structure", 1999.

[9] R. Campello, D.Moulavi, J.Sander, “Density-Based Clustering Based on Hierarchical Density Estimates”, PAKDD, 2013.
[10] J.ander, M.Ester, H.Kriegel, X.Xu, “Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications”, Data Mining and Knowledge Discovery, 2(2), pp 169-194, June 1998.
[11] The United States Census Bureau, U.S. Department of Commerce, “Facts of New York City”, 2018. [online]. Available: https://www.census.gov/quickfacts/fact/table/newyorkcitynewyork/PST045218 [Accessed: 01-Apr-2019].
[12] H. W. Kuhn, “The Hungarian method for the assignment problem”, Naval Research Logistics, 1955.
[13] V. Brummelen, G, Robert. “Heavenly Mathematics: The Forgotten Art of Spherical Trigonometry”, Princeton University Press, 2015.