# Improved Affinity Propagation Clustering for Business Districts Mining


1st Zhi Chen
*Computer Science and Technology*
*Hangzhou DianZi University*
Hangzhou, China
162050110@hdu.edu.cn

2nd Jian Xu
*Computer Science and Technology*
*Hangzhou DianZi University*
Hangzhou, China
jian.xu@hdu.edu.cn

3rd Yang Wu
*Computer Science and Technology*
*Hangzhou DianZi University*
Hangzhou, China
161050040@hdu.edu.cn

4th Ning Zheng
*Computer Science and Technology*
*Hangzhou DianZi University*
Hangzhou, China
nzheng@hdu.edu.cn

5th Liming Tu
*Computer Science and Technology*
*Hangzhou DianZi University*
Hangzhou, China
tuliming@hdu.edu.cn

6th Ming Luo
*Computer Science and Technology*
*Hangzhou DianZi University*
Hangzhou, China
luom@hdu.edu.cn



***Abstract*—Business districts serve as basic structures for understanding the organization of real-world economic network. Discovering these business districts in cities establish new types of valuable applications that can benefit end users: Business investors can better identify the proximity of existing business districts and hence, can contribute a better future planning for investing. In this paper, we propose improved affinity propagation clustering for business districts mining. Given check-in data, whose geography information represents business venues' location, we introduce a affinity propagation clustering algorithm(AP), a basic solution, to cluster venues. This strategy requires that real-valued messages are exchanged among business venues until a set of centers and corresponding business districts gradually emerges. However, the computational complexity of AP is affected by the scale of input. And it's not adaptive for random distribution of venues when mining business districts. To conduct business districts mining efficiently, we introduce a pruning method, termed as PAP. And then present merging based mine approach, termed as MAP. We conduct experiments from Yelp data, and experimental results show that our proposed method outperforms the basic solutions and resolves the problem well.**



***Index Terms*—Business district, Affinity propagation, Fast algorithm, Exemplar-based clustering, PAP, MAP**


## I. Introduction

Many people migrate into cities in the process of urbanization, which has mostly changed beings' lives. New business shops are opened in the communities to meet the explosively increasing citizens. So it's important for business owners to mine business districts over entire city, which can help business owners to decide where business district is suitable for investing in the city. Many recommend works also based on area level [1] [2] [3] [4]. However, mining business districts is a cumbersome task for business owners, as we need to collect and analyze relevant data. To this end, business owners typically conduct ground surveys. But the expense is not affordable to most of business owners and investors, except some big chain retailers [5]. Fortunately, in the era of social media and mobile apps, we have amounts of LBSN data that capture both online activities of users and online activities at physical locations. As a result, the LBSN data provide an alternative way to understand the crowds instead of delivering surveys to them. The availability of online users, location, and other behavioral data makes it possible now to mine the business districts.

In data mining study field, greedy clustering is used to mine business districts [1]. Zhao et al. [1] think the most check-in location is the center of business districts. However, each person have their own habits of checking, so it is not always accurate to use number of check-in to find centers of business districts. For example, we satisfy the check-in of Charlotte city that is used to mine business district over entire city. We can see that there are value equal zero arrive 13.9% with check-in at some business venues in Charlotte from Fig.1. And they set distance threshold by experience. Yet, different distance threshold will get different number of business districts. Other classical exemplar-based clustering algorithms can also finish this task, such as $K$-means and $K$-centers. However, these methods work well only if the initial choice is close enough to a good solution. Recently, sparse subspace clustering(SSC) is a newly developed spectral clustering-based framework for data clustering [37] [38] [39]. Sparse subspace clustering methods pursue a sparse representation of high-dimensional data and use it to build the affinity matrix. The subspace clustering result of the data is finally obtained by means of spectral clustering. However, the representation of affinity matrix depend on regular terms. And many models need to choose different metrics for designing regular terms.

Facing these challenges, we propose improved affinity propagation clustering for business districts mining. Methods proposed in this paper are adaptive for clustering big-scale and random data. The following is a example.

$Example$ : As we can see in Fig.2, streets are usually surrounded by venues. Suppose that Mr Bob is a investor.

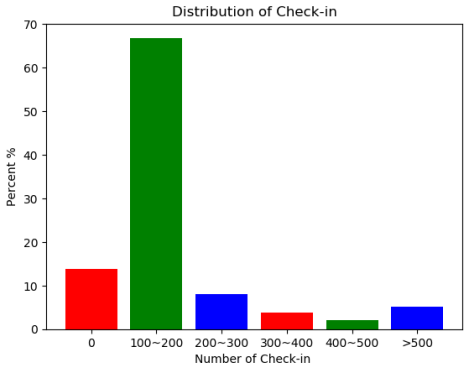


Fig. 1. Distribution of check-in in Charlotte

Fig. 2. Business districts

He wants to know where business districts are and then makes investment planning. So, to mine business districts, represented by red lines, over city for Bob, we conduct a improved affinity propagation clustering for business districts mining.

In this paper, we model the problem of mining business districts from Location-based Social Networks. Firstly, we introduce the business districts mining problem and use the AP clustering algorithm to mine business districts. Specially, we construct similarity matrix between pairs of venues, that is taken as input of algorithm. Rleal-valued messages are exchanged between venues until a high-quality set of centers of business districts and corresponding business districts gradually emerges. Secondly, we also develop pruning based algorithm to efficiently discover business districts. Then, we propose a merged algorithm to find optimal business districts. To verify our method, we conduct experiments on Yelp data. Experimental results show that our proposed method resolves the problem well. We summarize the contributions as follows.

- To the best of our knowledge, this is first work mining the business districts for business owners based on AP clustering, which extend the usage of AP clustering.
- According to the properties of business districts, we proposed pruning strategies to reduce the computational complexity of original AP clustering.
- With merging approach, Improved AP is suitable for clustering data with non-spherical distribution, according to rules of merging.

## II. Problem Formulation

In urban cities, if they go outside, people would often choose an area(such as a business district) instead of a specific shop. For example, when you have a date with your friends, you may expect a series of dating activities and all that dating locations are geographically close. So you will want to search a prosperous region containing all that required venues. Different kinds of business venues are ensuing open to satisfy the commercial needs of these people, which naturally forms a business district for a specific function, for instance, business districts around a working place. From this phenomenon, we observe, 1)a center is surrounded by many business venues, which forms a business district and contains a specific function attracting people gathering; 2)there is a phenomenon that these venues in a district are geographically adjacent to the center location. In the process of business venues contraction, new business districts appear. To sum up, we aim to mine business districts and corresponding business venues. Before the formal problem definition, we introduce and define several basic terms as follows [1]:

**Definition 1 (Business Venue).**A business venue is a shop $v$, with geographical$< lat, lon >$, and a category set $C_v$ containing categories labeled to the venue $v$.

**Definition 2 (Business District).** A business district $V_d$ is an aggregation of different business venues, in which venues are geographically adjacent to each other.

**Definition 3 (Center of Business District).** A center of business venue also is a shop $v$. Centers are surrounded by neighbors with lower local density and that they are at a relatively large distance from any venues with a higher local density.

**Definition 4 (Multi-peak Cluster).** There are several points with large distance and local maximization of density in a cluster.

**Definition 5 (Business Districts Mining Problem).** The business districts mining problem aims to find centers of business districts, and cluster each business venue and its center into groups $V_d$.

## III. Basic Solution

Affinity propagation (AP) [11] is a clustering algorithm proposed by Frey and Dueck in Science, has received much attention in the recent past. It has been applied in tissue clustering [17], image categorization [18], subspace division [19]. By introducing AP clustering, the efficiency of experiments gets better. Guan et al. use it to achieve text mining [15]. Tang applicate the AP clustering to social influence analysis in large-scale networks [16]. Dueck et al. propose combining a novel clustering method, affinity propagation (AP), recently reported in the journal Science, with linear discriminant analysis (LDA) to form a new method, AP-LDA, for face recognition, and outperforms Fisher face in terms of recognition rate [21].

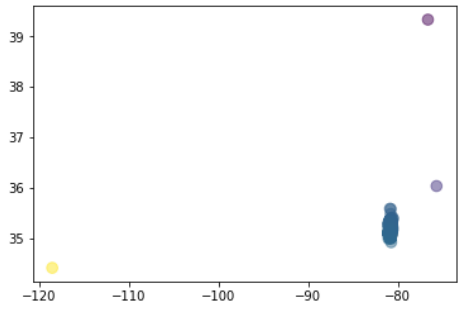


Fig. 3. Before filtering outliers in Charlotte

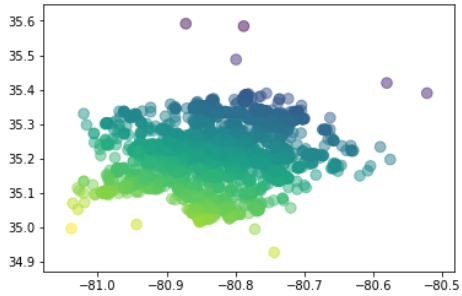


Fig. 4. After filtering outliers in Charlotte

In fact, treating each business district as a "cluster" and each business venue as an "object", the business districts mining problem can be formulated as a traditional clustering problem. According to apply the clustering technique, a set of data objects into multiple groups(or clusters), we can observe that objects in same cluster are more closer with others, so that a mount of clustering algorithms have been proposed [5] [6] [7]. However, in this paper, we aim to discover centers of business districts. We call center as "exemplar", from search results and simultaneously cluster the business venues into groups characterized by these exemplars. To cluster the business venues and find the exemplars, we are encouraged to employ AP clustering algorithm, that is an extremely successful one among these algorithms, to achieve this mining work.

### A. Data preprocessing

To get better clustering result, we firstly preprocess these check-in data. Due to some business venues far from other venues, they will not be a part of some business districts, regarded as outliers. For example, we can see there are three points far from the group with blue color in Charlotte from Fig.3. When we delete these outliers, we can get better points with closing to other. Fig.4 present details of group with blue color in Fig.3, the distances are not too large between each other.

### B. Input

In the AP clustering algorithm, we use and construct a similarity matrix $S$ where $s(i,j)$ represents the similarity between venue $i$ and venue $j$ as the input. And also any type of similarities is acceptable, e.g. negative Euclidean distance for real valued data and Jaccard coefficient for non-metric data, thus Affinity Propagation is widely applicable. In our works, we use negative Euclidean distance as similarity $s(i,j)$ between venue $i$ and venue $j$:

$$s(i,j) = -d_{i,j} = -||v_i - v_j||^2 \tag{1}$$

The similarity $s(i,j)$ in the matrix means how is suitable for venue with index $j$ as the center of business districts for venue with index $i$. Because Euclidean distance is positive value, we add negative sign. According to formulation, if two venues have shorter distance, the similarity would be higher. Specially, as Affinity Propagation takes as input a real number $s(k,k)$ for each venue $k$, venue with larger values of $s(k,k)$ are more likely to be chosen as center of business districts. These values are referred as "preferences". It's worth noting that the number of identified exemplar(number of clusters) is influenced by the values of the input preferences. Under normal condition, the common value could be set the median of the input matrix [11]. In our work, we don't know which venue would be center of business districts, so we give a common value to each venue as its the preferences. It makes all data points are equally suitable as center of business districts.

### C. Basic Algorithm

In Affinity Propagation, there are two kinds of message exchanged between venues. The message exchanged are defined as the "responsibility" $r(i,j)$ and the "availability" $a(i,j)$, and each takes into account a different kind of competition. Responsibility$r(i,j)$ is sent from venue $i$ to venue $j$, which indicates how strongly venue $i$ desires to choose venue $j$ as its center of business district.

The "availability" $a(i,j)$, sent from candidate center of business districts $j$ to venue $i$, reflects the accumulated evidence for how appropriate it would be for venue $i$ to choose venue $j$ as center of business districts, taking into account the support from other venues that venue $j$ should be an center of business districts. All responsibilities and availabilities are set to zero initially, and their values are iteratively updated as follows to compute convergence values:

$$r^{(t)}(i,j) = (1-\lambda)(s(i,j) - \max_{k \neq j}(a(i,k) + s(i,k))) + \lambda * r^{(t-1)}(i,j) \tag{2}$$

$$r^{(t)}(i,j) = (1-\lambda)(s(i,j) - \max_{k \neq j}(a(i,k) + s(i,k))) + \lambda * r^{(t-1)}(i,j) \tag{3}$$

$$a^{(t)}(i,j) = \begin{cases} (1-\lambda)(\min(0, r(j,j) \\ \quad + \sum_{k \neq \{i,j\}}^{n}(\max(0, r(k,j)))) \\ \quad + \lambda * a^{(t-1)}(i,j) & \text{if } i \neq j \\ (1-\lambda) * (\sum_{k \neq j}^{n}(\max(0, r(k,j))) \\ \quad + \lambda * a^{(t-1)}(i,j) & \text{if } i = j \end{cases} \tag{4}$$

When updating the messages, it is important that they should be damped to avoid numerical oscillations that arise in some circumstances. So we set the damping factor $\lambda$ is between

0 and 1. We used a default damping factor of $\lambda$ = 0.9 [1] in all of our experiments, because the value is bigger, the possibility of the iteration process oscillates will be lower[2]. The message-passing procedure may be terminated after a fixed number of iterations, after changes in the messages fall below a threshold $T$, or after the local decisions stay constant for some number of iterations. At any point during the iteration process, availabilities and responsibilities can be combined to identify centers of business districts. For any venue $i$, its center of business districts is:

$$j = \begin{cases} a(i,j) + r(i,j) & \text{if } i \neq j \\ argmax_j(a(i,j) + r(i,j)) & \text{if } i = j \end{cases} \quad (5)$$

If the venues have the same center of business districts, they should be grouped into the same business district. Now, we can get business districts $V_d$. Compared with classical exemplar-based clustering algorithms, a user does not assign the number of cluster and representative object. It's partly avoid a problem that bad performance appears, in which initial exemplars improper result, such as K-means and K-centers.

## IV. Improved Solution

### A. Disadvantages of Basic Solution

Basic solution is a functionally correct procedure, but we are unlikely to be satisfied with its performance, particularly with the large scale and random distribute of venues that now exist.

*1) Higher Computational Complexity:* The original AP clustering takes the full similarity matrix $S$ as input. In this case, the number of pairwise similarities is $N^2$. From basic solution, we can be known that $N^2$ responsibilities and availabilities need to be calculated in each iteration. Therefore, the time complexity of original AP clustering is $O(N^2T)$, where $T$ is the number of iterations. This does limit the running speed of AP clustering, especially if the number of venues is great.

*2) Bad for Random Distribution of Data:* Affinity Propagation clustering method work well only if the initial data distribution is close enough to near-spherical. However, real business venues always aren't near-spherical distribution, so one business distribute with random distribution may divide into several clusters. For example, there should be two clusters in Fig.5, when we apply AP clustering algorithm in objects in Fig.5. However, we get four clusters in Fig.6. As it makes the objects as adjacent as possible in the same cluster, so it can give a better result for objects with spherical distribution. Yet, there are lots of business venues, and they follow non-spherical distribution.

### B. Method

Firstly, to solve this problem with higher computational complexity, we introduce a pruning method, termed as PAP. As known, the efficiency of AP clustering can be improved by eliminating unnecessary message exchanges. According to $Definition3$, we design a pruning method to find a representative potential centers of business districts set, which ensures us to find a potential center of business districts for whichever object that can become a final center of business districts. By doing this, the final center of business districts set is guaranteed to be similar with the real optimal center of business districts set. Then, We present merging based mine approach, termed as MAP, to solve problem with random distribution of venues. we can find that if two cluster are not too far away from each other, that can be considered as one cluster. The motivation of this paper is inspired by this fact. The challenge are to select the right potential center of business districts set and to select right threshold of distance. To this end, we develop an effective selection algorithm in IV.C and then develop merging algorithm IV.D.

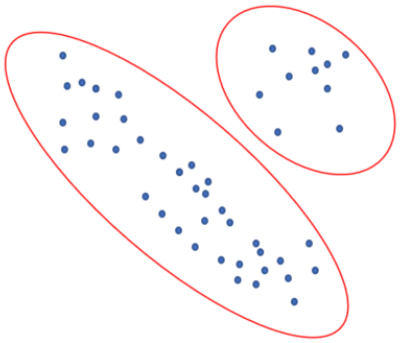

Fig. 5. Original data

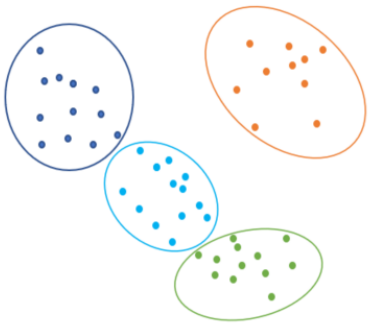

Fig. 6. After clustering

### C. PAP(Improved AP with Pruning)

Motivated by [22] and $Definition3$, for each business venue $i$, we will compute two quantities: local density $\rho_i$ and its distance $\delta_i$. We can know that these quantities depend only on the distances $d_{ij}$ between business venue $i$ and venue $j$, which are assumed to satisfy the triangular inequality. The local density $\rho_i$ of business venue $i$ can be measured as follows,

$$\rho_i = \sum_j \chi(d_{ij} - d_c) \quad (6)$$

where $\rho_i$ is the number of neighbor business venue that are closer than $d_c$ to venue $i$, $\chi(x)$ = 1 if $x$ <0 and $\chi(x)$ = 0 otherwise, $d_c$ is a cutoff distance. As a rule of thumb [22],

[1]http://www.psi.toronto.edu/affinitypropagation/faq.html.

[2]http://archive.ics.uci.edu/ml/

we can choose $d_c$ so that the average number of neighbors is around 1 to 2% of the total number of points in the data set. To achieve more potential centers of business districts, we set the value as 2%.

Distance $\delta_i$ can be measured by computing the minimum distance between the venue i and any other venue with higher density:

$$\delta_i = \min_{j:\rho_j > \rho_i} (d_{ij}) \tag{7}$$

For the venue with highest density, we conventionally take $\delta_i$= max($d_{ij}$). Note that $\delta_i$ is much larger than the typical nearest neighbor distance only for venues that are local or global maximum in the density. Thus, cluster centers of business districts are recognized as venues for which the value of $\delta_i$ is anomalously large.

To formulate these two impact factors, for each venue $i$, its score $y_i$ can be measured as follows,

$$y_i = \rho_i * \delta_i \tag{8}$$

where $y_i$ value is higher, it's more possible to become center of business districts, otherwise not. Due to two impact factors belong to different dimensions, we need to normalize them.

We come up with a strategy, mining potential centers of business districts, improve efficiency of AP in $Method$. We need to find suitable value $t$, by doing this, we can distinguish which venue is potential centers of business districts or common venues. According to $Definition3$, we can know potential center $v_i$ would have large $y_i$. Finding potential centers can follow three steps: First, for each venue $v_i$, we compute its' score $y_i$ by formulation (7). Secondly, we sort these $y_i$ values by ascending and draw these $y_i$ values. Thirdly, we choose these venues that value $y_i$ grater than $t$ as potential centers of business districts. In addition, because value $y$ of common venues are almost the same and smaller than potential centers, there is a jumping point from the potential venues to the common venues. The index value of jumping point is number of potential centers of business districts. In Fig.7, we can see that red dot will be the jumping point, before red dot appeared, the value of $y$ change obviously, the value isn't obviously changeable behind the red dot. We set red point corresponding value as $t$. Points on the left of red dot would be potential centers of business venues, the point on the right of red dot would be common venues. After this step, we can get a high-quality potential centers of business districts set. We construct similarity matrix $S$ between $v_i$ and potential centers and then execute AP algorithm on $S$.

Specially, there are difference with [22]. Although it has a good performance, it is not good at $Definition4$. When we use this method to choose potential centers, the number of result will be more than number of real centers. For example, we can see blue cluster should be one cluster from Fig.8. When we apply this method in blue cluster from Fig.8, the blue cluster will be divided into two clusters, respectively blue cluster and red cluster in Fig.9. We also can discover some distribution of venues is similar to multi-peak cluster. When we mine potential centers of business districts, its number of result is more than real. By doing this, it can confirm we can't lose real centers in result. Algorithm 1 presents a full description of PAP.

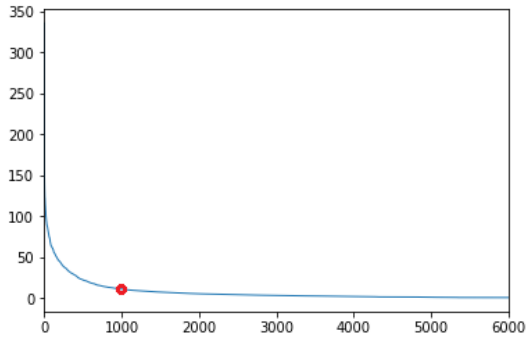


Fig. 7. The value of $y = \rho_i * \delta_i$ in decreasing order for the venues

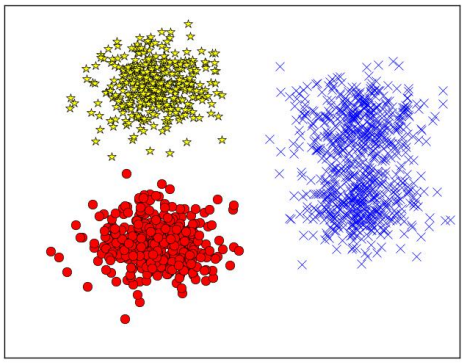

Fig. 8. Real cluster

### D. MAP(Improved AP with Merging)

Given set of clusters $\Omega$ =$\{\omega_1, \omega_2, ...\omega_k..., \omega_K\}$, which $\omega_k$ represent cluster of $k_{th}$, and $K$ is number of clusters. $X$=$\{x_1, x_2, ..., x_n\}$ is sets of data, and $n$ represents the total number of data, $N_k$ represent the number of data points in $k_{th}$ cluster. So we can merge clusters based on distance as steps:

Firstly, we compute distance with random two data $x_i$, $x_j$ in $k(k = 1, 2, 3, ..., K)$, and use $d_{i,j}$ to represent it.

$$d_{i,j} = ||x_i - x_j||^2 \tag{9}$$

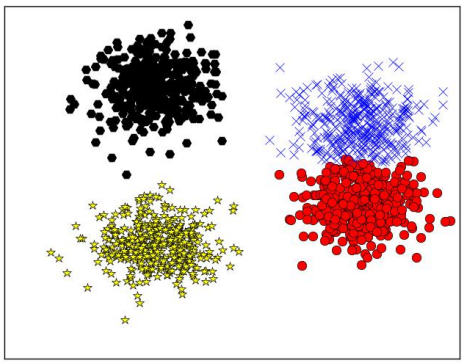

Fig. 9. Cluster result

**Algorithm 1** Improved AP with pruning

**Input:** Venues $v_i < longitude, latitude >$, $rate$, iteration $T$=1000, $\lambda$=0.9;

**Output:** Business districts $V_d$;

Given $rate$=2% and venues $i< longitude, latitude >$, finding cutoff $d_c$;

Compute $\delta$ and $\rho$ of each business venue $v_i$;

Generate a set of potential center of business district using formulation $y = \delta * \rho$;

sort $y_i$ and find potential centers;

**for all** $v_i$ **do**

Construct compression similarity matrix S between vi and potential centers;

**end for**

Implement message-passing on sparse factor graph;

**return** $V_d$

Then, we compute average distance $d_{\omega_k}$ between all pairs of data in $\omega_k (k = 1, 2, 3, ..., K)$

$$d_{\omega_k} = \sum_{i,j=1}^{N_k} d_{ij} \frac{1}{N_k(N_k - 1)} \tag{10}$$

Finally, for each cluster $\omega_k$, we compute min distance $d_c$ between $\omega_i$,$\omega_j$ . If it satisfy formulation 10, we will merge these two clusters.

$$d_c \leq max(d_{\omega_i}, d_{\omega_j}) \tag{11}$$

Here, aiming to avoid the influence of outliers that far from other objects in the same cluster, we set max average distance of two clusters as threshold.

## V. EXPERIMENT

In this section, we evaluate the proposed improved Affinity Propagation algorithm on real-world data. The evaluation criteria include $SS$, $S\_Dbw$ and an efficiency criterion(real CPU time). All algorithms are implemented in C++ and evaluated on a 64-bits machine with 2.5GHz CPU, 16GB RAM. We conduct experiments on the LBSN data from the Yelp dataset challenge 2015.

### A. Data sets

We used two business venues datasets of two cities available at Yelp[3]. And we use geographic coordinates of check-in, location of venues, to mine business districts. For each dataset, we only reserve one, if there are several same geographic coordinates of check-in, location of venues, to mine business districts. of business venues, that is, 6194 venues in Charlotte, 5297 venues in Pittsburgh. we set the $s(i, i)$ is default value that equal median of distance between business venues. Parameters of AP in this work focused on the scenario when $T$ = 1000, $\lambda$ = 0.9.

[3]https://www.yelp.com/dataset

### B. Evaluation Criteria

*1) $SS$:* The goal of exemplar-based clustering is to find an exemplar set that the sum of similarities between each object and its exemplar is maximized. Therefore, the Sum of Similarities(SS) is the most important criterion of exemplar-based clustering algorithm. We use $X_{c_i}$ to denote the center of business venue $X_i$, and $c_i$ is the index number of business venue $X_{c_i}$ So it is defined as

$$SS = \sum_{i=1}^{N} s(i, c_i) \tag{12}$$

where $s(i, c_i)$ denotes the similarity between $X_i$ and its exemplar $X_{c_i}$. A larger SS indicates a better clustering performance.

*2) $S\_Dbw$:* This validity index has been proposed in [30]. Similarly to SD index its definition is based on cluster compactness and separation but it also takes into consideration the density of the clusters. Formally the $S\_Dbw$ index measures the intra-cluster variance and the inter-cluster variance. The intra cluster variance measures the average scattering of clusters and it is described by Equation 7. The inter–cluster density is defined as follows:

$$density_{ij} = \frac{density(u_{ij})}{\max(density(v_i), density(v_j))} \tag{13}$$

$$Dens\_bw = \frac{1}{n_c(n_c - 1)} \sum_{i=1}^{n_c} \sum_{i=1, i \neq j}^{n_c} density_{ij} \tag{14}$$

where $u_{ij}$ is the middle point of the line segment that is defined by the $v_i$ and $v_j$ clusters centers. The density function around a point is defined as follows: it counts the number of points in a hyper-sphere whose radius is equal to the average standard deviation of clusters. The average standard deviation of clusters is defined as:

$$stdev = \frac{1}{n_c} \sqrt{\sum_{i=1}^{n_c} \left\| \sigma(v_i) \right\|} \tag{15}$$

The $S\_Dbw$ index is defined in the following way:

$$S\_Dbw = Scatt + Dens\_bw \tag{16}$$

The definition of $S\_Dbw$ indicates that both criteria of "good" clustering are properly combined and it enables reliable evaluation of clustering results. Lower index value indicates better clustering schema.

### C. Experimental Settings

We complete the business districts mining task with three steps: computing similarity matrix, pruning the similarity matrix and merge clusters. To demonstrate the effectiveness of our method, we compare the proposed model with following criteria, SS ,computational time and $S\_Dbw$. We want to use criteria SS and computational time to present performance of pruning Affinity Propagation and original Affinity Propagation. Then we use criteria $S\_Dbw$ to present performance of improved Affinity Propagation and greedy algorithm that was used to solve the same problem.

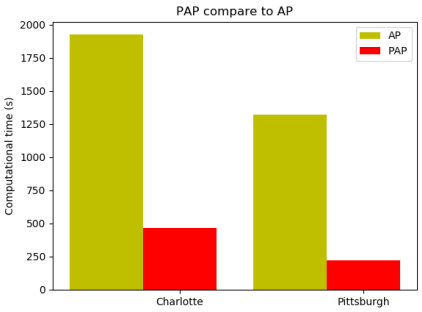


Fig. 10. PAP and AP in computational time

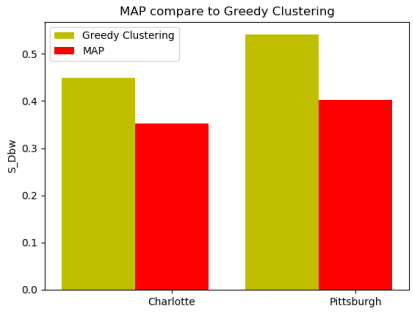


Fig. 11. MAP and greedy clustering in $S\_Dbw$

### D. Experimental results

We now present the results of all algorithms described above. We measure the performance of the algorithms on different cities. Table 1 compared PAP with the original AP. The ordinate is the ratio of SS achieved by PAP divided to that achieved by the original AP. As larger SS indicates better clustering performance, we can see SS is not more than $1\%$, almost the points between 0.9 and 1, from table 1, which can represent our PAP don't decline the performance of AP.

TABLE I
COMPUTING SS ON TWO CITIES

| Key Value | Name of City | |
|---|---|---|
| | ***Charlotte*** | ***Pittsburgh*** |
| SS(PAP)/SS(AP) | 0.996 | 0.981 |
| Number of venues | 6194 | 5297 |

The comparison of PAP and AP in computational time is shown in Fig.10. The time cost of AP far beyond the time of PAP, which can indicate our work is effective in improving the efficiency. Full similarity matrix is inputted in original AP, which result in $N^2$ responsibilities and availabilities are computed in each iteration. Therefore, the time complexity of original AP is $O(N^2T)$. In PAP, because we reduced the scale of similarity matrix before implementing message-passing, the number of left pairwise similarities is only $Nq$, $q$ is the number of potential centers and it is much less than $N$. So we only need to compute $Nq$ responsibilities and availabilities in each round. By reducing the scale of similarity matrix, the complexity of AP clustering has been dramatically reduced.

It shows performance of MAP and the method with greedy clustering in Fig.11, we can see value with MAP always lower than value with greedy clustering, which can indicate that our method has a better performance than greedy clustering.

## VI. RELATED WORK

### A. LBSN applications.

To improve user experience and prosper the businesses in LBSNs, a variety of new applications come out, e.g., point-of-interest (POI) recommendation and retail allocation system. some approaches leveraged Gaussian mixture model to characterize user's check-in activities [24] [25]; While some approaches utilized the kernel density estimation (KDE) to study user's check-in behavior and avoid employing a specific distribution [26] [27]. [28] [29] proposed user-based collaborative filtering to estimate the unobserved rating by directly using the check-in information of friends. There are some recommendation works based on content, sentiments and temporal effect [31] [32] [33] [34] [35] [36].

### B. Connection with prior work

In[1], author employ the breadth-first search (BFS) to visit the geographical venue graph G. And select the venues whose distances to venue with the most checked-in location are less than the threshold $d_t$. As a result, venue with the most checked-in location and the selected venues constitute a business district $V_d$. Because check-in is subjectivity, what use it to choose centers of business districts be improper. In [23], author use AP to cluster cluster all the check-in locations into groups, the business circles can be obtained with each group representing one business circle and the representative location being the centroid of the business circle. Although [23] also use AP algorithm to mine business districts, they can't consider AP is improper for non-spherical distribution of location. Frey and Dueck [11] pointed out that if we use the sparse similarity matrix instead of complete similarities matrix, the computational complexity of AP clustering could be reduced. Based on this idea, many fast AP clustering algorithms were proposed [9] [13] [14]. We also propose pruning algorithm based on the idea.

## VII. CONCLUSIONS

In this paper, we modeling the problem of mining business districts from Location-based Social Networks. In particular, we use AP clustering algorithm based on passing-message to finish mining task. Furthermore, we reduce the similarity matrix by selecting potential centers of business districts, and propose a merged algorithm. The two steps improve the computational efficiency of AP and get optimal result. Extensive experiments over LBSN data verify the effectiveness and efficiency of our algorithms.

## VIII. ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China (No. 61572165), the Natural Science Foundation of Zhejiang Province (No. LZ15F 020003).

## REFERENCES


[1] Zhao, S., King, I., Lyu, M. R., Zeng, J., & Yuan, M. (2017, August). Mining business opportunities from location-based social networks. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 1037-1040). ACM.

[2] Lin, J., Oentaryo, R. J., Lim, E. P., Vu, C., Vu, A., Kwee, A. T., & Prasetyo, P. K. (2016, March). A business zone recommender system based on Facebook and urban planning data. In European Conference on Information Retrieval (pp. 641-647). Springer, Cham.

[3] Li, M. R., Huang, L., & Wang, C. D. (2017, September). Geographical and Overlapping Community Modeling Based on Business Circles for POI Recommendation. In International Conference on Intelligent Science and Big Data Engineering (pp. 665-675). Springer, Cham.

[4] Liu, Y., Liu, C., Lu, X., Teng, M., Zhu, H., & Xiong, H. (2017, August). Point-of-Interest Demand Modeling with Human Mobility Patterns. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 947-955). ACM.

[5] Thau, B. (2015). How big data helps chains like starbucks pick store locations—an (unsung) key to retail success.

[6] Jain, A. K. (2010). Data clustering: 50 years beyond K-means. Pattern recognition letters, 31(8), 651-666.

[7] Xu, R., & Wunsch, D. (2005). Survey of clustering algorithms. IEEE Transactions on neural networks, 16(3), 645-678.

[8] Barbakh, W. A., Wu, Y., & Fyfe, C. (2009). Non-standard parameter adaptation for exploratory data analysis (Vol. 249). Berlin: Springer.

[9] Fujiwara, Y., Irie, G., & Kitahara, T. (2011, July). Fast algorithm for affinity propagation. In IJCAI Proceedings-International Joint Conference on Artificial Intelligence (Vol. 22, No. 3, p. 2238).

[10] Sun, L., Guo, C., Liu, C., & Xiong, H. (2017). Fast affinity propagation clustering based on incomplete similarity matrix. Knowledge and Information Systems, 51(3), 941-963.

[11] Frey, B. J., & Dueck, D. (2007). Clustering by passing messages between data points. science, 315(5814), 972-976.

[12] Givoni, I., Chung, C., & Frey, B. J. (2012). Hierarchical affinity propagation. arXiv preprint arXiv:1202.3722.

[13] Jia, Y., Wang, J., Zhang, C., & Hua, X. S. (2008, October). Finding image exemplars using fast sparse affinity propagation. In Proceedings of the 16th ACM international conference on Multimedia (pp. 639-642). ACM.

[14] Xiao, J., Wang, J., Tan, P., & Quan, L. (2007, October). Joint affinity propagation for multiple view segmentation. In Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on (pp. 1-7). IEEE.

[15] Guan, R., Shi, X., Marchese, M., Yang, C., & Liang, Y. (2011). Text clustering with seeds affinity propagation. IEEE Transactions on Knowledge and Data Engineering, 23(4), 627-637.

[16] Tang, J., Sun, J., Wang, C., & Yang, Z. (2009, June). Social influence analysis in large-scale networks. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 807-816). ACM.

[17] Verma, R., & Wang, P. (2007, October). On detecting subtle pathology via tissue clustering of multi-parametric data using affinity propagation. In Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on (pp. 1-8). IEEE.

[18] Dueck, D., & Frey, B. J. (2007, October). Non-metric affinity propagation for unsupervised image categorization. In Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on (pp. 1-8). IEEE.

[19] Zhao, Z. Q., Gao, J., Glotin, H., & Wu, X. (2010). A matrix modular neural network based on task decomposition with subspace division by adaptive affinity propagation clustering. Applied Mathematical Modelling, 34(12), 3884-3895.

[20] Leone, M., & Weigt, M. (2007). Clustering by soft-constraint affinity propagation: applications to gene-expression data. Bioinformatics, 23(20), 2708-2715.

[21] Du, C., Yang, J., Wu, Q., & Li, F. (2007). Integrating affinity propagation clustering method with linear discriminant analysis for face recognition. Optical Engineering, 46(11), 110501.

[22] Rodriguez, A., & Laio, A. (2014). Clustering by fast search and find of density peaks. Science, 344(6191), 1492-1496.

[23] Li, M. R., Huang, L., & Wang, C. D. (2017, September). Geographical and Overlapping Community Modeling Based on Business Circles for POI Recommendation. In International Conference on Intelligent Science and Big Data Engineering (pp. 665-675). Springer, Cham.

[24] Cho, E., Myers, S. A., & Leskovec, J. (2011, August). Friendship and mobility: user movement in location-based social networks. In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 1082-1090). ACM.

[25] Cheng, C., Yang, H., King, I., & Lyu, M. R. (2012, July). Fused Matrix Factorization with Geographical and Social Influence in Location-Based Social Networks. In Aaai (Vol. 12, pp. 17-23).

[26] Zhang, J. D., & Chow, C. Y. (2013, November). iGSLR: personalized geo-social location recommendation: a kernel density estimation approach. In Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (pp. 334-343). ACM.

[27] Lichman, M., & Smyth, P. (2014, August). Modeling human location data with mixtures of kernel densities. In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 35-44). ACM.

[28] Ye, M., Yin, P., & Lee, W. C. (2010, November). Location recommendation for location-based social networks. In Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems (pp. 458-461). ACM.

[29] Ye, M., Yin, P., Lee, W. C., & Lee, D. L. (2011, July). Exploiting geographical influence for collaborative point-of-interest recommendation. In Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval (pp. 325-334). ACM.

[30] Halkidi, M., & Vazirgiannis, M. (2001). Clustering validity assessment: Finding the optimal partitioning of a data set. In Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on (pp. 187-194). IEEE.

[31] Li, H., Hong, R., Wu, Z., & Ge, Y. (2016, June). A spatial-temporal probabilistic matrix factorization model for point-of-interest recommendation. In Proceedings of the 2016 SIAM International Conference on Data Mining (pp. 117-125). Society for Industrial and Applied Mathematics.

[32] Liu, B., Fu, Y., Yao, Z., & Xiong, H. (2013, August). Learning geographical preferences for point-of-interest recommendation. In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 1043-1051). ACM.

[33] Yang, D., Zhang, D., Yu, Z., & Wang, Z. (2013, May). A sentiment-enhanced personalized location recommendation system. In Proceedings of the 24th ACM Conference on Hypertext and Social Media (pp. 119-128). ACM.

[34] Yuan, Q., Cong, G., & Sun, A. (2014, November). Graph-based point-of-interest recommendation with geographical and temporal influences. In Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management (pp. 659-668). ACM.

[35] Yu, Z., Xu, H., Yang, Z., & Guo, B. (2016). Personalized travel package with multi-point-of-interest recommendation based on crowdsourced user footprints. IEEE Transactions on Human-Machine Systems, 46(1), 151-158.

[36] Zhu, H., Chen, E., Xiong, H., Yu, K., Cao, H., & Tian, J. (2015). Mining mobile user preferences for personalized context-aware recommendation. ACM Transactions on Intelligent Systems and Technology (TIST), 5(4), 58.

[37] Peng, X., Zhang, L., & Yi, Z. (2013). Scalable sparse subspace clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 430-437).

[38] Peng, X., Feng, J., Xiao, S., Yau, W. Y., Zhou, J. T., & Yang, S. (2018). Structured AutoEncoders for Subspace Clustering. IEEE Transactions on Image Processing.

[39] Zhen, L., Yi, Z., Peng, X., & Peng, D. (2014). Locally linear representation for image clustering. Electronics Letters, 50(13), 942-943.