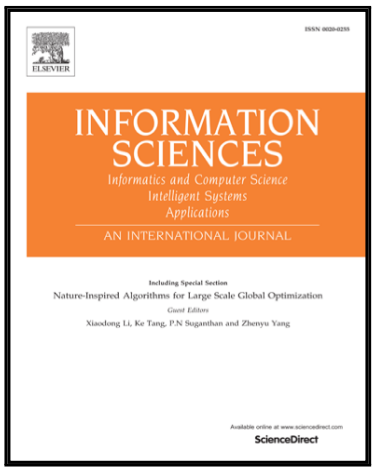


Improving ATM Coverage Area Using Density Based Clustering Algorithm and Voronoi Diagrams

N.Raghu Kisore, CH.B Koteswaraiah

PII: S0020-0255(16)31087-8
DOI: 10.1016/j.ins.2016.09.058
Reference: INS 12554

To appear in: *Information Sciences*

Received date: 8 February 2015
Revised date: 17 September 2016
Accepted date: 27 September 2016

Please cite this article as: N.Raghu Kisore, CH.B Koteswaraiah, Improving ATM Coverage Area Using Density Based Clustering Algorithm and Voronoi Diagrams, *Information Sciences* (2016), doi: 10.1016/j.ins.2016.09.058

Improving ATM Coverage Area Using Density Based Clustering Algorithm and Voronoi Diagrams

N. Raghu Kisore[a,*], CH. B Koteswaraiah[b]

[a] *Assistant Professor, IDRBT, Castle Hills, Road No.1, Masabtank, Hyderabad 500057*
[b] *M. Tech(Information Technology), University of Hyderabad, Hyderabad, Andhra Pradesh, India 500046*

Abstract

Facility location is a problem of paramount importance and optimizing business operations without affecting customer service is very challenging. In the case of banking services, the location of bank branches and ATMs must match the service demands (turn around time for service, reachability etc) of the customers' and the expected quality of service is determined by the socio-economic background of the customer. Therefore, it is necessary to formulate the optimization problem so as to reflect the customers' expectations and tolerance for quality of service in a given geographical region. The ability to do so requires clustering people living in the region into several smaller areas called service areas. An ideal clustering algorithm should consider the social behavior of people living in the service area and the uncertainty associated with their social behavior.

In this paper, we propose a modification to generalized density based clustering algorithm (GDBSCAN) to deal with fuzziness in the values describing the population demographics and the preferences for ATM location among customers utilizing the ATM services. The modified version of GDBSCAN clustering algorithm, which we call GFDBSCAN, is used to cluster people around key socio-economic parameters. GFDBSCAN can also be used to cluster geographical regions based on the requirement and preferences expressed by the customers for services like business outlets, ATMs, bank branch operations, public utilities, etc. We apply the proposed algorithm

*Corresponding author
Email addresses: nraghukisore@gmail.com (N. Raghu Kisore), balu12mcmb28@gmail.com (CH. B Koteswaraiah)

to cluster geo-spatial data based on personal traits of people living in the geographical area under study. We measure and compare the performance of GFDBSCAN with other popular clustering algorithms using Silhouette coefficient, Dunn index and Davies-Bouldin index. We plot the clustering results on Google maps for better visualization of results. We found that GFDBSCAN is better able to cope with fuzziness in the values of both spatial and non-spatial attributes. We finally use voronoi diagrams to identify the ideal locations to place ATMs so as to ensure that the customers' preferences are served and, at the same time, the service area of each ATM is optimized.

Keywords: density based, fuzzy, clustering algorithms, ATM location, voronoi diagram.

1. Introduction

Optimization of service location is a problem of paramount importance in many areas such as placement of sensors in a network [29], location of warehouses [38], public parks [19] and retail stores belonging to a brand [25]. An optimization technique typically involves variables and constraints and it's design typically involves several design parameters, of which some are highly sensitive to the proper working of the design. In order to reduce the complexity of the optimization algorithm the design variables are often represented by parameters. Finally, constraints are drawn to represent functional relationship among the design variables and design parameters. The constraints reflect certain physical phenomenon and resource limitations. The nature and number of constraints to be included in the formulation depends on the nature of business problem. Finally, the optimization process involves defining the objective function in terms of the design variables and problem parameters. The optimization algorithm aims to find values for the variables involved in the problem definition so as to either minimize or maximize the objective function, subject to the constraints. For example, this could be either the minimization of overall cost of manufacturing or overall weight of a component or the maximization of total life of a product.

Most of the variables in the case of facility management [2] can be precisely measured and quantified. Therefore, the constraints and objective function can be precisely expressed in mathematical form. But the same cannot be said about providing banking services through bank branches and ATMs as banking unlike other services is largely a social engineering problem

that requires provisioning for differential services catering to a wide variety of customers. Many of the variables associated with the definition of quality of service cannot be precisely measured due to the inability to quantify customer expectations and his tolerance for the failure to meet the desired quality of service. Therefore, it is beneficial to cluster users of banking services based on socio-economic factors and at the same time deal with fuzziness in these factors that define the consumer traits.

The objective function in case of establishing bank ATMs and branches, should maximize quality of service at an acceptable cost to the banks. The quality of service from customer's point of view is acceptable turnaround time [13] while availing banking services at either the branch or ATM and for banks it is meeting customer expectations for quality of service at the lowest operational cost. But the acceptable quality of service cannot be unanimously quantified across the entire population living in a geographical region since it is defined by the socio-economic background of the customers living in a given area. Quality of service is affected by the demand for service which in turn depends on the population density in a given region. Since banking services need to provide a quality of service that reflects the expectations of customers, an ideal solution requires designing a facility placement strategy (location of bank branches and ATMs) that reflects the socio-economic parameters of people in a given service area. The first step to provisioning a differential quality of service is clustering the geographical area (city) under study into several sub categories (or service areas) based on the socio-economic behavior of the people in the region. The clustering of the population is vital so that the parameters of the optimization model can subsequently be tweaked to reflect the customer behavior in the service area.

The process of identifying optimal site location and types of service to be delivered is an operational issue and typically involves the 3-stage process shown in figure 1. While today the decisions at each stage are based on heuristics, they can be greatly improved by tailoring the decision making process to the requirements of the population in the targeted service area. The customer requirements can be modeled using several social and economic variables associated with the customer. Each data point in the data set (of customer information) consists of several spatial and non-spatial attributes and involves fuzziness across these attributes. The fuzziness arises because of two factors. The first is the inability of the customer to accurately provide his preferences, i.e., there could be errors in the process of conducting surveys

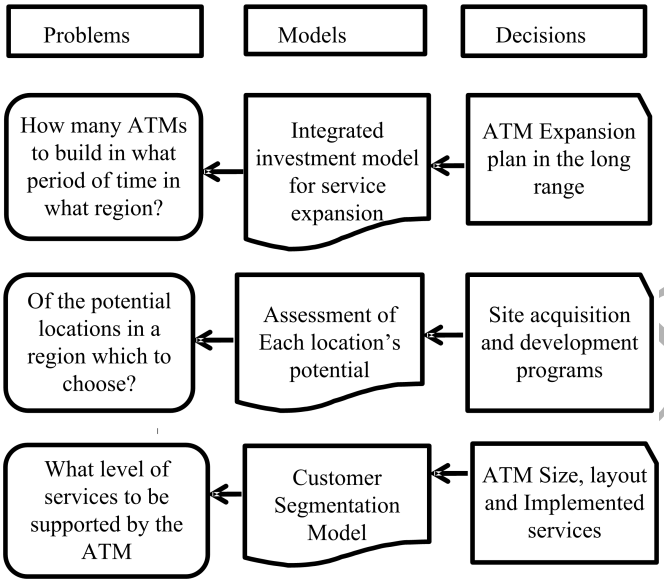


Figure 1: A typical decision making process undertaken by a bank.

and/or the inability of the design of survey form to accurately capture the individual's needs for banking services. The second factor is the fact that bank branches and ATMs (facilitators for banking services) cannot be placed to perfectly satisfy every single customer; hence the clustering algorithm has to deal with certain level of abstraction in the analysis phase.

In this paper, we propose a novel analytics-based approach to optimize ATM placement taking into consideration the financial and social behavior of the people living in a geographical area. The approach follows a three-stage process. The first stage involves identifying the key socio-economic factors that determine ATM placement. The second stage involves clustering the geographical area of ATM placement using a density-based clustering algorithm. The last stage involves using voronoi diagrams to study the relative merits of various ATM placement strategies, measured in terms of effective coverage area and turn around time incurred while availing services. For the first stage of the process we leveraged on our interaction with the banks and the existing work in literature to determine the relevant socio-economic factors. This paper examines the last two stages of the problem. For the clustering process we make use of a modified version of the Generalized DBSCAN clustering process which we call Generalized Fuzzy Density Based

Clustering (GFDBSCAN). GFDBSCAN takes into consideration the fuzziness in the values of the data during clustering process. In the last stage we apply voronoi diagrams to identify the best combination of ATM for a given cluster. Voronoi diagrams are useful to measure the service area of each ATM, and, through its dual graph, delauny triangulation, quantify the uniformity in the spread of ATM services. Ensuring the uniformity in the spread of ATM service requires optimizing the relative placement of ATMs. An ideal placement of ATM is one where the the service coverage (measured by turnaround time) is uniform while maintaining a high number of customer footfalls (at the bank ATM). We use the support weight metric to identify the ideal placement strategy. Support weight helps to identify the turnaround time for a customer (as he traverses across various paths in the region) in the worst case scenario.

The rest of this paper is organized as follows. Section 2 provides an overview of work done to improve ATM placement, and Section 3 provides details on the strengths and weaknesses of various clustering algorithms proposed in literature. Section 4 provides implementation details of the proposed Generalized Fuzzy Density Based clustering Algorithm (GFDBSCAN). Section 5 discusses the results, and finally, Section 6 concludes our work.

2. Related Work

A great deal of research has been done to identify best possible locations for banking services by modeling the problem as an optimization problem. [26] proposes a model taking into account the hierarchical structure of banking and financial industry. The proposed model aims to map customer services to a three-tier banking hierarchy. The authors develop a stochastic model based on chance-constrained goal programming model. [36] proposes three heuristic models keeping in view the stochastic nature of customer demand for ATM services. The aim of the model is to minimize the average service time for delivering ATM services to the customer. The service time is calculated as sum of traveling time to reach the ATM and time spent waiting for service (queuing delay) at the ATM . The model assumes people consume ATM service as a standalone service. This is not true as in most cases people often use ATM services as part of their daily routine. In general, people prefer ATMs that are located along their daily routine over ATMs that are located off their travel plans. The model does not take into account spending habits and associated ATM withdrawal patterns associated with people

living in a geographical area. [36] regards the facility location problem as a multi-criteria decision making (MCDM) problem and provides a review of various solution methods. [28] and [31] provide details of several other facility location models. [27] proposes a mathematical model to aid banks in restructuring their branch locations by maintaining, closing, or opening branches. The restructuring process is modeled as a nonlinear problem and is formulated as a mixed binary, integer linear model. [28] extends the discussion on facility location to use of GIS software as a good data visualization tool. [32] aims to study the reasons for usage of ATMs among college students in Nigeria.

The published work discussed above assumes a homogenous usage pattern of ATM services by the customers. Such an assumption is not valid when multiple financial cultures coexist together in a small geographical region, and the regulatory policies are aimed at achieving financial inclusion across the country. None of the work discussed above takes into account the heterogeneity in service consumption by different individuals, and, because of the heterogeneity, a single unified strategy of service deployment does not work across the entire region. This makes it necessary to understand the customer requirements for ATM services and to develop a mathematical model around these requirements. To overcome these problems, we propose a density based clustering algorithm that clusters people living in a city based on their socio-economic parameters, and, for each cluster, we apply voronoi diagrams to identify the ideal number of ATMs required to meet the quality of service metrics (for ATM service) of people living within a cluster. The voronoi diagrams also assist us in identifying the best possible combination for deploying ATMs.

3. Background

Clustering is the process of grouping objects such that objects in a group are more identical to one another than to the objects of a different group. Several studies [37, 16] provide good classification of clustering algorithms. Based on the mathematical method employed and the structure of the clusters produced, clustering algorithms are typically classified as hierarchical, graph theory based, combinatorial search techniques based, fuzzy, neural networks based, kernel based and grid based. However, we limit our study to a smaller subset of clustering algorithms. As we are interested in the shape of the clusters produced and, as indicated in [35], we limit our classification to

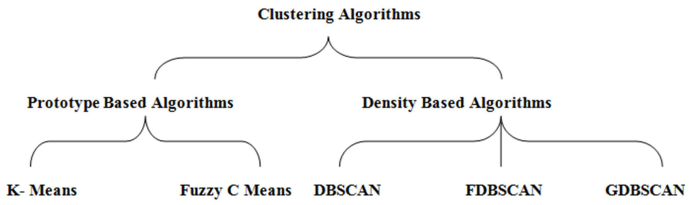


Figure 2: classification of clustering algorithms.

prototype-based and density-based clustering algorithms. Clustering algorithms under each category can in turn be classified into fuzzy and non-fuzzy algorithms.

- Prototype Based Clustering Algorithms [9, 17]
- Density Based Clustering Algorithms

Prototype-based clustering algorithms form clusters based on the similarity of objects with respect to randomly selected initial prototypes, whereas density-based clustering algorithms form clusters by considering variation in object densities present in the data set. Figure 2 shows the classification of some of the popular clustering algorithms proposed in literature.

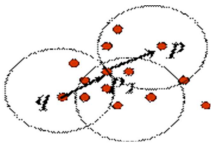
K-Means [22] is a partitional clustering technique that attempts to find a user-specified number of clusters (K). The data points in each of the identified clusters are more similar to respective cluster heads than to cluster heads of other clusters. Partitional clustering algorithms divide the objects into subsets such that the subsets do not overlap with one another, and therefore, each object is a member of exactly one subset. The K-means algorithm initially assigns objects into different clusters based on a similarity score, and the process is repeated until the value of optimization function falls below a threshold. The algorithm involves the following three steps,

1. Select K initial centroids.
2. Assign all data points to their closest centroids by measuring the similarity.
3. Recompute the centroids of each cluster and repeat steps 2 and 3 until objective function is optimized.

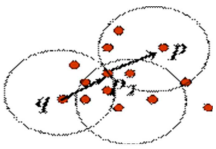
Drawbacks of the K-means algorithm include its inability to find clusters of arbitrary shape, its need for a priori specification of the number of clusters, its use of exclusive assignment, its inability to handle noisy data and outliers, and finally the dependency of clustering results on initial centroids (selected in Step 1 of the algorithm). While the time complexity of K-means algorithm is less in comparison to density-based clustering algorithms, it is recommended to repeatedly execute the algorithm with different centroids so as to overcome the bias in identifying the clusters caused by the initial selection of centroids.

Several techniques have been proposed to improve the problem of initial centroid selection. Two of the most common approaches are subtractive clustering [10] and K-Means++ [4]. Subtractive clustering is a one-pass algorithm for identifying centroids based on the *potential* of each data point. The algorithm defines a potential function to identify possible candidates for initial centroids. The highest potential data objects are chosen as initial centroids. K-means++ takes a more mathematical approach for selecting centroids. While the first center is chosen at random, each subsequent center is chosen using a weighted probability distribution as a metric, and the new point gets chosen as center with probability $D(x)^2$. $D(x)$ represents the distance between point x and the nearest point that has already been chosen as center. While this approach provides a 2-fold improvement over K-means in terms of computation required and error, it nevertheless still suffers from other weakness of K-means explained above.

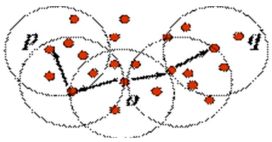
Fuzzy C-Means algorithm (FCM) [1, 7] is an extension of the K-means algorithm. It introduces fuzziness into the similarity measure between cluster heads and data points. FCM allows each data object to be a member of more than one cluster with varying probabilities, and the maximum value of this probability indicates the cluster to which the data object belongs to. The probabilistic relationship of each data point with respect to every cluster is represented by a membership matrix. The membership matrix is updated at the end of each iteration of algorithm. The updated matrix is then used in calculating new centroids for the next iteration. The additional parameter used in this algorithm is fuzziness metric 'm'. FCM deals with fuzziness in the measure of similarity [30, 34] between data objects and cluster head. However, often in real world scenarios there is fuzziness in the attribute values of the data object due to the inability to accurately measure it. The fuzziness is more often when dealing with qualitative metrics as customer behavior. In some cases fuzziness arises due to inability to obtain precise information



(a) Directly density reachable



(b) Density reachable



(c) Density connectivity

Figure 3: DBSCAN functions

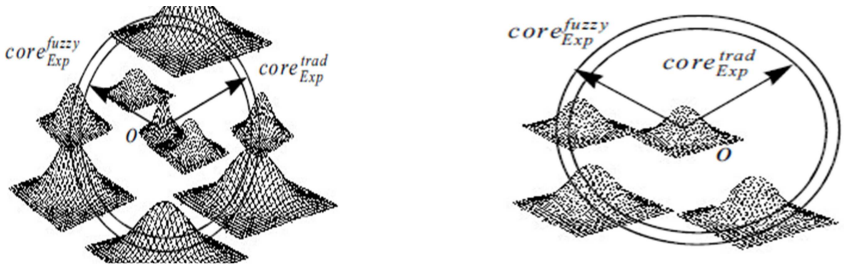
of personal data of a user.

The DBSCAN [3, 14] algorithm introduces a new approach to clustering, wherein clusters are formed based on the variations in densities of data points. The algorithm does not require a priori information about number of clusters, and further, it can identify clusters of arbitrary shapes. The control parameters used in this algorithm; *radius* and *min_samples*, aid in classifying data points into three different categories (shown in Figure 3) which, in turn, aids in identifying clusters. The three classifications for data points are *directly density reachable*, *density reachable* and *density connectivity*.

A point **p** is said to be directly density reachable from **q** if it satisfies the following two conditions

- **p** is within the radius of **q**.
- **q** has neighborhood of *min_samples* points within its radius.

Let us consider a chain of points p_1, p_2,....., p_n, and where $p_1 = \mathbf{p}$ and $p_n = \mathbf{q}$. Points **p** and **q** are said to be density reachable if p_{i+1} is *directly density reachable* from p_i. The two points **p**, **q** are said to be density connected with each other, if there exist a point **o** such that both **p** and **q** are density reachable from **o**. The explanation and the relative positions of the above three functions is visualized in Figure 3. The shortcoming of the DBSCAN algorithm is its inability to deal with fuzzy data [21]. The ability to deal with fuzzy data is vital in case of ATM placement as each point in the data



(a) Object with high density.

(b) Object with low density.

Figure 4: difference between DBSCAN and FDBSCAN.

set is not an individual data object but instead represents people staying in an area of 50m by 50m. A grid size of 50m by 50m was chosen due to the limitations of computational power available.

In [20], a fuzzy version of the DBSCAN (FDBSCAN) algorithm is introduced, where each data object is represented by a continuous probability density function rather than a single value. Unlike the FCM algorithm which handles fuzziness in the membership of objects to clusters, FDBSCAN handles inaccuracies in the values of the data objects. The algorithm introduces a new control parameter *core_probability* for identifying the core objects in the data set. The parameter *core_probability* represents the probability of an object becoming a core object. The results of FDBSCAN converge to DBSCAN when a value 0 is chosen for *core_probability*. A higher value for *core_probability* improves the ability of the algorithm to handle fuzziness in the data set. The distance function necessary to measure similarity between two objects is redefined to take into account the uncertainty in the values of data objects. FDBSCAN introduces *Distance Density function* and *Distance Distribution function* to capture the aggregated value of the similarity between fuzzy objects.

In many situations, clustering based on expected distance value is more appropriate than using the centroids of the fuzzy object representations. For example, in a hypothetical scenario it could be that the centroids are close to each other, but due to a high uncertainty in the values of the data objects, the distance expectation could indicate a high distance between the objects. In such a case, clustering based on euclidean distance as in the case of centroids approach would result in wrong clustering of data objects. Figure 4b provides

an example scenario of the such a scenario.

The major drawback of DBSCAN and FDBSCAN is their inability to deal with spatial attributes in addition to non-spatial attributes. In case of ATM placement, spatial attributes play a vital role since the customers would prefer not to walk/travel beyond a certain distance to use ATM services and hence the similarity metric used to cluster data points should consider the geographical distance between the data point and the possible location of ATM.

In [34], the authors propose and implement a generalized version of DBSCAN algorithm called GDBSCAN. GDBSCAN takes into account both spatial and non-spatial attributes of data object. GDBSCAN generalizes the notion of "density-based clusters". First, any notion of a neighborhood can be used instead of an ϵ-neighborhood (objects within a radius of ϵ from an object) if the definition of the neighborhood is based on a binary predicate that is both symmetric and reflexive. Second, instead of using the number of objects in the neighborhood of an object, better measures can be used to define the "*cardinality*" of that neighborhood. While DBSCAN uses simple euclidean distance to define the notion of neighborhood, GDBSCAN uses topological relations such as *intersects* or *meets* to cluster spatially extended objects such as a set of polygons of largely differing sizes.

The major drawback of GDBSCAN is its inability to handle fuzziness in the attribute values of data objects so GDBSCAN doesn't solve the current business problem. In the ATM placement problem, the data set consists of fuzzy objects. The fuzzy nature in the data set is due to the abstraction introduced so as to reduce the run-time complexity i.e., all data points in a given geographical region of 50m * 50m cell are represented by a single data point.

3.1. Voronoi Diagram

Given a set of S points (also called voronoi sites) $p_1, p_2, ..., p_n$ ($|S| = n$) in the plane, a voronoi diagram divides the plane into n voronoi regions such that each point p_i lies in exactly one region. Further, if a point $q \notin S$ lies in the same region as p_i, then the euclidean distance from p_i to q will be shorter than the euclidean distance from p_j to q, where p_j is any other point in the set S. The working principle of voronoi diagram is given in equation (1).

$$V(P_i) = \{\, x \mid \|x - x_i\| \leq \|x - x_j\| \; for \; i \neq j \; and \; j \in [1, n] \,\} \qquad (1)$$

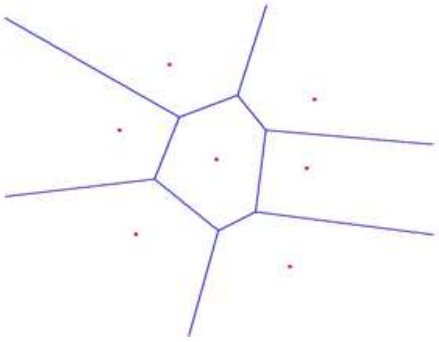


Figure 5: ordinary voronoi diagram.

Where V is the planar voronoi diagram associated with the set of points in S and is represented by equation (2).

$$V = V(P_1), V(P_2), ..., V(P_n) \tag{2}$$

The example of a simple voronoi diagram is shown in Figure 5. In this voronoi diagram, all points carry same weight. We can also assign weights to points which indirectly represent the priorities given to those areas. A weighted voronoi diagram is represented by the equation (3).

$$V(P_i) = \{\, x \mid \frac{1}{w_i}\|x - x_i\| \leq \frac{1}{w_j}\|x - x_j\| \;\; for\; i \neq j \;\; and \;\; j \in [1, n] \,\} \tag{3}$$

Where w_i represents the weight assigned data point. Weights on the edges of the voronoi diagram cause them to be arcs instead of straight lines. These arcs are formed by the bisector condition defined in equation (4).

$$\begin{aligned} b(P_i, P_j) = \{\, x \mid \; & \|x - \frac{w_i^2}{w_i^2 - w_j^2}x_j + \frac{w_j^2}{w_i^2 - w_j^2}x_i\| \\ & = \frac{w_i w_j}{w_i^2 - w_j^2}\|x_j - x_i\| \; for \; i \neq j \; and \; j \in [1, n] \,\} \end{aligned} \tag{4}$$

The voronoi diagram represented by the equation (3) is shown in Figure 6. For the ATM placement problem we apply voronoi diagram to determine the ideal number of ATMs and their location. Voronoi cell sites represent possible ATM locations and weights represent waiting time for availing service at an ATM. The waiting times are influenced by the population density and the nature of on going business activity in the geographical location under study. The voronoi region of each cell site indicates the ATM service areas.

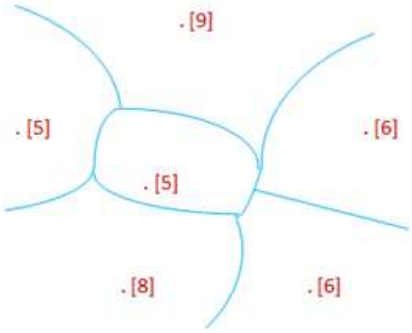


Figure 6: weighted voronoi diagram.

3.2. Details of Data Set

The data set was obtained from the 2011 population census of India. The features of the data set include *Age*, *Gender*, *Marital_status*, *Occupation*, *Salary*, and *Type_of_house*. Among these features Gender, Marital status, Occupation and Type_of_house are categorical variables and the remaining are numerical variables. We identified these features as important by conducting a survey of 1000 people. These are also the features that are collected by most banks when on boarding a new customer account. Principal component and exploratory analysis were performed using MATLAB to identify the level of dependency across various questions posed to customers of different banks. To conserve space we do not reproduce the results of that survey in this paper. Based on evaluation of survey results, we found that people's preference for ATM location differed greatly among customers of different banks in addition to the personal traits mentioned above. For ATM location preferences, people rated 23 different site locations ranging from entertainment centers, businesses, retail shopping centers, etc.

Age	**Gender**	**Marital status**	**Occupation**	**Salary**	**Type of house**
16	Male	Single	Student	0	Rented
25	Male	Single	Employee	20000	Rented
34	Female	Married	Employee	35000	Own
38	Male	Married	Business	50000	Rented
....	....	....	....	....	....
8	Female	Single	Student	0	Own

Table 1: Details of Data Set

The raw census data provides individual records creating a large number of data points for any given geographical region. This is primarily due to high population density in India. The data covers 10 administrative areas in Hyderabad city, India covering an area of 8Km x 5Km. An area of 8Km x 5Km translates to a population of about 1,500,000. Table 1 shows a sample representation of raw data. Given the large population, to reduce execution time for clustering process, we took an abstract view of the population in which each data point in our clustering algorithm represents the average and standard deviation of personality traits of people living in a 50m x 50m geographic area. Each data object was represented using an N-mixture Gaussian model with N set to 5. The model was obtained by using curve fitting functionality provided by MATLAB. The population in each grid of size 50m x 50m is not uniform due to the coexistence of high rise buildings with short and historical buildings. We normalize the data set and assume that each grid has on average a population of 168 records where in each record represents the attributes of people living in the 50m x 50m geographical grid. The administrative regions covered are shown in Figure 7 and include *Jubilee hills*, *Banjara hills*, *Punjagutta*, *Begumpet*, *KBR Park*, *Ameerpet*, *Husain sagar*, *Mehdipatnam*, *Lakdikapool* and *Himayat Nagar*. The spread of each municipal area and the associated latitude and longitude coordinates were obtained from Google maps. The boundaries of various regions shown in Figure 7 are only indicative and do not necessarily represent legal boundaries. Categorical variables such as marital status, occupation and gender were encoded as quantitative variables by assigning numerical codes. The variable salary was compartmentalized, and a numerical code was assigned to each.

There is no single correct way to deal with hybrid data set consisting of numerical and categorical values and arrive at a universal similarity metric

[18]. It is widely accepted that the context [8] of the data set greatly helps in identifying the right approach. For categorical variables gender, marital status, occupation and type of house a predefined set of possible values were identified and listed for the participants of the survey to choose from. The list of possible values in each case were identified after studying the data collection forms used for gathering population census of India in 2011 and the bank account opening forms obtained from different banks. This was done intentionally so that data requirements of our solution are in tune with the data available in the IT systems of banks. The current problem deals with similarity in socio-economic parameters of population living in two areas of size 50m x 50m, and hence, after numerically encoding the data, the attribute values were converted into probabilities. Finally, all attribute values were normalized to make sure that no one attribute dominates over others. By evaluating the data in the form of 50m x 50m grids, we greatly reduce the computational complexity of having to compare and analyze the similarity between 1,500,000 records (a challenge in the case of density-based clustering algorithms). Instead we sample a small subset of records (say 10) present in each grid and compute the similarity between these aggregated data points. Further, by doing so we randomize the fuzziness in the values of each of the 1,500,000 records (the census data has granularity issues) to a small subset of records chosen at random from a grid.

4. Proposed Solution

The proposed solution involves a 3-stage process:

1. Identify the socio-economic indicators that affect ATM usage.
2. Cluster the geographical region where the ATMs are to be deployed using the socio-economic parameters identified in step 1.
3. Apply voronoi diagrams to each cluster identified in step 2 to assess the number and location of ATMs to be placed.

The first of the above three steps was done by surveying people using ATMs in the identified geographic region in concurrence with historical ATM usage data obtained from banks. We do not reproduce the results here as they have been published earlier. To facilitate clustering based on spatial and non-spatial variables (representing the socio-economic indicators) and handle the fuzziness in the data values, the GDBSCAN clustering algorithm was modified. We name this algorithm Generalized Fuzzy Density Based

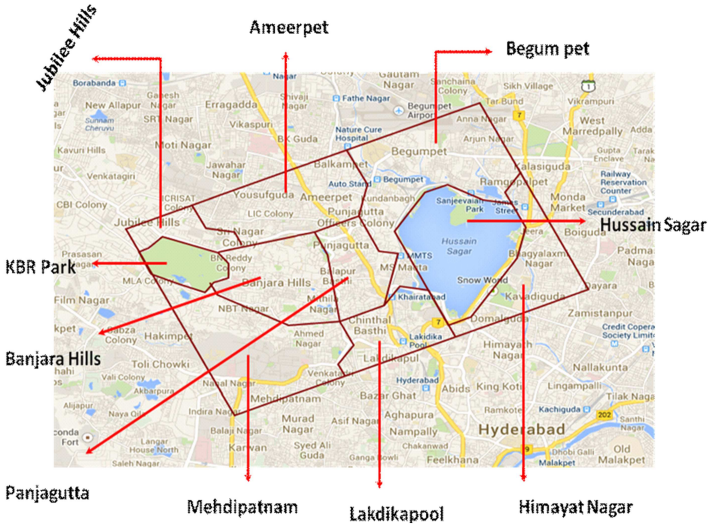


Figure 7: layout of 10 regions in Hyderabad city.

Clustering algorithm (GFDBSCAN). GFDBSCAN takes into account both geographical constraints as well as fuzziness in the data. Uncertainty or fuzziness in the data arises because of inaccuracies in the user's profile information captured either in the census data or in the bank's customer profile information recorded in the bank's database.

The input parameters *wArea* and *min_samples* of GFDBSCAN algorithm serve the same purpose as in GDBSCAN, while the additional control parameter *core_probability* handles the fuzziness present in the data. As already discussed, each data point in the data set represents the people residing in each individual 50m * 50m cell. The algorithm starts by selecting an object which satisfies the core point condition with respect to *wArea*, *min_samples* and *core_probability* The neighborhood of an object is identified using Lemma 1 as follows,

Lemma 1: The Neighborhood of an object 'o' is defined such that $\forall p$,

$$d(o, p) \leq wArea \tag{5}$$

In equation (5), *wArea* is a spatial constraint (in this work we set $wArea = 1km^2$). For the object to become a core object it should satisfy lemma 2.

Lemma 2: Let $P(core\_probability(o))$ denote the probability for an object 'o' to be considered a core point. An object 'o' is considered a core point, if it satisfies the following two conditions;

$$Condition\ 1 : Neighbors(o, wArea) \geq min\_samples, and \tag{6}$$

$$Condition\ 2 : \frac{\sum_{i=1}^{Neighbors.size} P(d(o, Neighbors(i)))}{Neighbors.size} \geq core\_probability \tag{7}$$

Parameter *min_samples* represents minimum samples required to satisfy the core point condition and the additional parameter *core_probability* represents the probability that the object becomes a core point. So, the threshold for *core_probability* also influences the number of clusters in addition to *wArea* and *min_samples*. In case of GDBSCAN, only *wArea* and *min_samples* are the control parameters, whereas in GFDBSCAN the additional condition for *core_probability* has to be satisfied in order to satisfy core object condition.

Algorithm 1 Initialization

Require: Given the M detected objects
Ensure: new clusters
  for (each object i) **do**
    Create new cluster C_i;
    Initialize the cluster feature;
    $clusterFeature(i) = (x_i, y_i, v_{x_i}, v_{y_i}, 1, 0, i)$;
  end for

Objects in the data set are iteratively checked to identify core points, and those that do not meet the criteria are labeled as *NOISE*. 'Id' indicates the cluster number to which the object belongs. The function *next_id (Id)* returns the successor of the object 'Id'. The function *Cluster Expansion* is used to expand the cluster that is formed by the core object p. The function *neighborhood (Object, min_samples)* returns the list of objects that are within the range of *wArea* of the object. In this algorithm for a point to become the core point it should satisfy the two conditions,

1. Size (Neighbors) > min_samples and
2. Probability (Neighbors, Point) > core_probability.

Algorithm 2 GFDBSCAN

Require: Dataset, wArea, min_samples, core_probability
Ensure: New clusters ▷ Dataset is UNCLASSIFIED

```
Id ← next_id(NOISE)
for i from 1 to Dataset.Size do
    Object ← Dataset.Get(i)
    if Object_Id ≡ UNCLASSIFIED then
        if Cluster_Expansion (Dataset, Object, Id, wArea, min_samples,
core_probability) then
            Id ← next_id(Id)
        end if
    end if
end for
```

The above two conditions ensure that all core points have *min_samples* within *wArea* around it and are similar to one another by as much as *core_probability*. The second condition is very important when the data is fuzzy because an object might have the minimum number of objects as its neighbors, but when we calculate the similarity metric between the neighbors and the point, it may not satisfy the core point condition as shown in Figure 4b.

The function *fuzzy_similarity* is calculated between the point identified as a core point and its neighbors. While calculating the similarity score, from each group (a grid of size 50m x 50m) of points, we draw 10 samples and then compute the similarity between all combinations. The value of *fuzzy_similarity* is calculated by averaging the similarity value of all the possible combinations. The number of samples chosen from a group while calculating the similarity score is a measure of fuzziness in the similarity relation of the two groups. If all the samples in the group are considered, then will have a more accurate indication of similarity but at the cost of increased computational cost. On the other hand, if only one sample is used, we would have high fuzziness in the results. Consequently, we choose an intermediate approach and consider 10 samples in a group to calculate similarity score.

Once a point is identified as core point, the same approach is used for the objects that are in the set $min\_samples - neighborhood$ of the core point. The *Id* of a point may change later from NOISE to core point because it may be in the $min\_samples - neighborhood$ set of another object and we cannot add these points into the seed-list because they are already classified

Algorithm 3 Cluster Expansion

```
Require: Dataset, Object, Id, wArea, min_samples, CoreProb
Ensure: Boolean
    if thenSize(Object) ≤ 0                          ▷ point not in selection
        Object. changeId(Object, UNCLASSIFIED);
        return False;
    end if
    Seeds := Dataset.neighborhood(Object, min_samples)
    if thenSize(seeds) < min_samples
        Object.Id (Object, NOISE);
        return False;
    end if                                  ▷ still here? Object is a core object
    if P thenrobability(seeds, Object) ≥ core_probability
        Object.changeIds (seeds, Id);
        Seeds.delete (Object);
        while seeds ≠ Empty do
            current_object: = seeds. first ();
            Result: = Dataset. neighborhood (current_object, min_samples);
            if Size (result) ≥ min_samples) && (Probability(result, current_object) ≥ core_probability) then
                for i from 1 TO result.size do
                    Object: = result. get (i);
                    if Size (Object > 0 ) && Object.id IN (UNCLASSIFIED, NOISE) then
                        if O thenbject.id = UNCLASSIFIED
                            Seeds.append (Object);
                        end if
                        Object. changed (Id);
                    end if
                end for
            end if
            Seeds.delete (current_object);
        end while
    end if
    return True;
```

Algorithm 4 Probability

Require: Neighbors, Object
Ensure: Float
Probability = 0
for i FROM 1 TO Neighbors.size **do** DO
Probability = Probability + Fuzzy_similarity (Object, Neighbors(i));
end for
Probability = Probability/Neighbors.size; **return** Probability;

Algorithm 5 Fuzzy_similarity

Require: Object, Neighbor_object
Ensure: Float
n_samples = 10
Similarity = 0
Draw n_samples for Object
Draw n_samples for Neighbor_object
for i FROM 1 TO n_samples **do**
for j FROM 1 TO n_samples **do**
Similarity = Similarity+distance (Object (i), Neighbor_object (j))
end for
end for
return $Similarity/n\_samples^2$

as NOISE due to the condition $wArea(Neighbors) < min\_samples$. The core point condition is checked for all the points which are newly added and labeled as either *UNCLASSIFIED* or *NOISE*, and this procedure is continued until no point satisfies a core point condition. During the next iteration the '*Id*' is set as the successor of the previous '*Id*' which is obtained using the function *next_id* (Id). The performance of this algorithm depends on the time complexity of the *neighborhood* function since it is performed for each object and also for finding the *core_probability*.

5. Results and Discussion

5.1. Experimental setup

The proposed GFDBSCAN clustering algorithm was compared with K-Means++, FCM, DBSCAN, FDBSCAN and GDBSCAN. The problem we aimed to solve was clustering the people of Hyderabad city based on attributes discussed in 3.2. The values for the attributes were obtained from the 2011 population census of India. Scikit-learn [33] was used for our analysis. Scikit-learn is an open source tool that provides a Python API for a wide range of state-of-the-art machine learning algorithms. For better visualization of clustering results, the results were plotted on Google maps using the Google Map API. Each pin in Figures 8 to 13 represents a geographical region of 50m x 50m . Figure 7 shows the boundaries of various municipal regions of Hyderabad city, India. In each of the visualizations shown, black color pin indicates noise while a random spread of varied colored pins among a dominating colored pins indicates wrong clustering of data objects. For further discussion we classify these as fuzzy pins.

5.2. K-Means++ clustering

Figure 8 shows the clustering results when the K-means++ algorithm is applied. K-means++ defines optimum clustering based on the value of sum of squared error of the distances between the centroid and the cluster members. For our dataset, the algorithm reaches the global minimum of the objective function when the number of centroids was chosen to be 4 (K = 4). From our data set, we observed that the personality attributes of people in Ameerpet and Lakdi-ka-pool areas vary widely (based on chi squared analysis).

K-means++ clustering identifies *KBR park* and *Husain Sagar* lake regions (blue color) correctly. It also correctly identifies the Jubilee hills and *Banjara Hills* regions (green color) and *Punjagutta* region (yellow color).

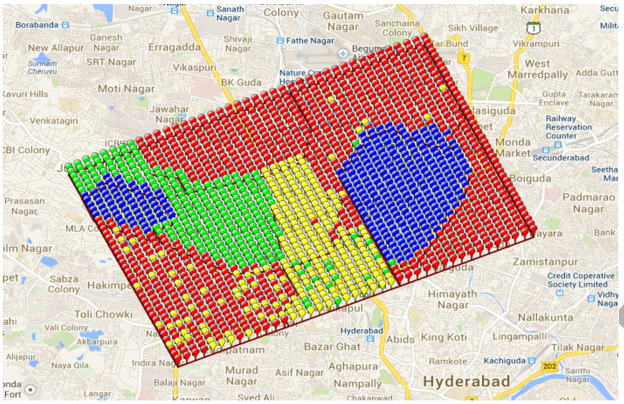


Figure 8: Clustering of people using K-Means.

From Figure 8 we can observe that K-means++ produces only 4 clusters,but in reality the data set had 10 different geographic regions. This is because while each data object is represented using N-mixture Gaussian curve, the K-means algorithm needs a deterministic value. Hence, in our implementation for K-means++,a simple average was taken to represent each data object instead of N-mixture Gaussian curve.

5.3. Fuzzy C Means (FCM) clustering

The K-Means++ algorithm assigns people to one of the 4 clusters. K-Means++ provides an indicative measure of different classes of people living in the geographical region under study. As a result, we set the number of clusters to 4 for Fuzzy C Means algorithm. The drawback of K-Means++ algorithm is its inability to handle fuzziness present in the data set. But in reality, people living in a given area often demonstrate properties of more than one region. We apply FCM algorithm to the data set with the input parameters as $m = 2$ and termination criteria = 0.00001. It identifies the fuzzy data (mixed data present in *Ameerpet*) correctly and also identifies the boundaries of KBR park and *Husain Sagar* regions. However, the clustering of the remaining regions is not correct as seen by the presence of a number of fuzzy pins. This can be noticed especially in the clustering of *Ameerpet* region. The result of the Fuzzy C Means algorithm is shown in Figure 9. It is well understood that it is very difficult to identify the optimum number of centroids that would result in the best possible clustering of data objects, so we also experimented with density based clustering algorithms.

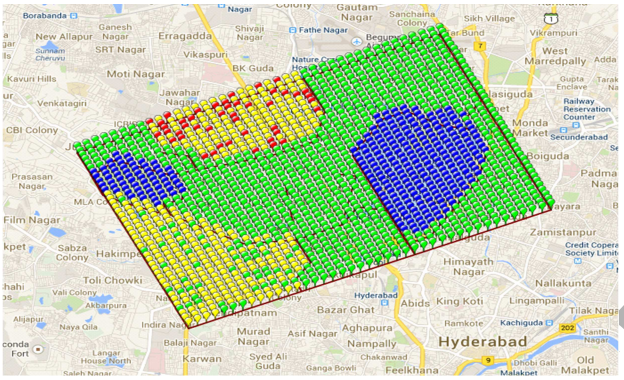


Figure 9: Clustering of people using Fuzzy C Means.

5.4. Density-based spatial (DBSCAN) clustering

As prototype-based algorithms are not good enough to solve the current business problem, we experimented with density based clustering algorithms proposed in literature. Figure 10 shows the results for the DBSCAN clustering algorithm. We obtained varying results based on values chosen for the input parameters of *eps* and *min_samples*. But the optimum results (based on Silhouette coefficient) was obtained for $eps = 0.3$ and $min\_samples = 35$ for identifying the 4 clusters with good representation of people. DBSCAN clusters people staying in different regions properly except for regions where there is a blend of people from different regions. The reason for this is that DBSCAN considers only the mean of each data object rather than the distribution curve. As mentioned before, each data object represents an abstract view of people living in each 50m x 50m cell, so the mean value of each record does not capture the true picture. To deal with this kind of data, the simple DBSCAN algorithm is not ideal.

5.5. Fuzzy DBSCAN (FDBSCAN) clustering

In FDBSCAN we use fuzzy distance instead of normal distance. The fuzzy distance is calculated by drawing 10 values from the Gaussian mixture model associated with each pair of data objects. In addition to the value of the fuzzy distance between the two data points, we also consider the values of *core_probability*, *eps* and *min_samples*. The result of clustering algorithm depends on the input values. We achieve the optimum results by choosing parameter values as $eps = 1.0$, $min\_samples = 25$ and $core\_probability = 0.5$

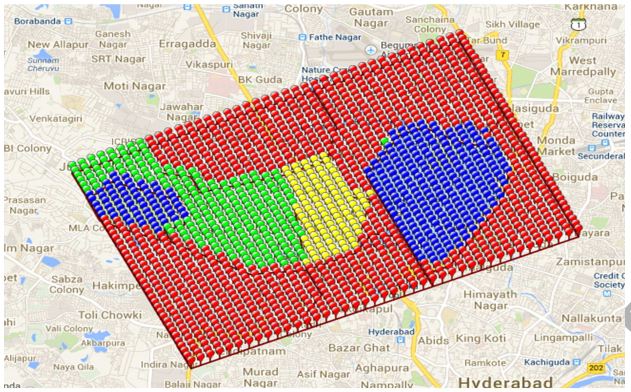


Figure 10: Clustering of people using DBSCAN.

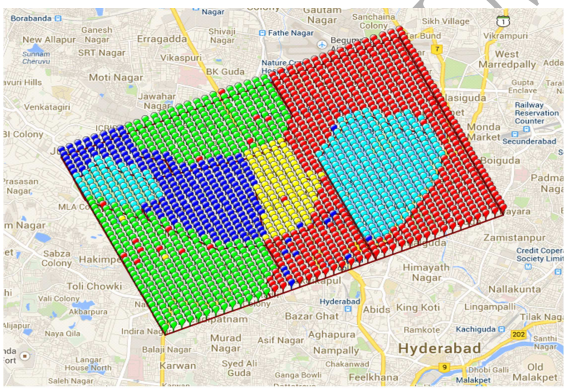


Figure 11: Clustering of people using FDBSCAN.

(50%). We reduce the overall execution time of the algorithm by choosing high value for *eps* and minimum value for *min_samples* when compared to DBSCAN. By considering the additional control parameter *core_probability*, FDBSCAN identifies the fuzzy pins present in the data. The output of the FDBSCAN algorithm is shown in figure 11.

5.6. Generalized DBSCAN (GDBSCAN) clustering

The current business problem not only deals with fuzzy data, but also geographic constraints as a person may not be willing to go beyond a certain distance to use ATM services. We implemented Generalized DBSCAN [14]

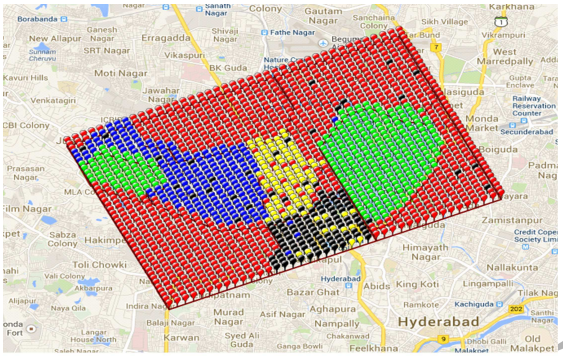


Figure 12: Clustering of people using GDBSCAN.

with two control parameters: one is spatial attribute, and the other a non-spatial attribute. The value we considered for spatial attribute is *wArea*. The parameter *wArea* in was set to $1Km^2$. The constraint for non-spatial attribute is minimum number i.e. *min_samples* of objects within an area of $1km^2$. The value for *min_samples* was chosen to be 25 as the number of data objects (each data object represents a geographical region of 50m x 50m) per 1 km^2 cannot be more than 400. The placement of an ATM is heavily influenced by the attributes of the majority of the people living in a given area and hence we consider 25 as appropriate value for *min_samples*. Because of the mixture of data objects present in *Lakdi-ka-pul* area, the algorithm classifies them as noise. Generalized DBSCAN produced the result shown in Figure 12.

5.7. Generalized Fuzzy DBSCAN (GFDBSCAN)

Figure 12 shows that GDBSCAN algorithm does not perform well if the data contains fuzzy properties. Hence, GDBSCAN classifies the people staying in *Lakdi-ka-pul* as noise. The proposed GFDBSCAN clustering algorithm can handle both spatial attributes and non-spatial attributes and also deal with fuzziness in the values of the data object. We implemented GFDBSCAN algorithm by setting *wArea* to $1Km^2$ and *min_samples* to 25. In addition, the fuzzy constraint *core_probability* was chosen to be 0.05. For a point to be considered as core point, it should satisfy all the above three conditions (line numbers 11 & 17 in algorithm 2 (function *Cluster_Expansion*). GFDBSCAN accurately identifies the fuzzy pins present in *Ameerpet* and *Lakdi-ka-pul* areas whereas the other algorithms failed to identify these fuzzy data objects.

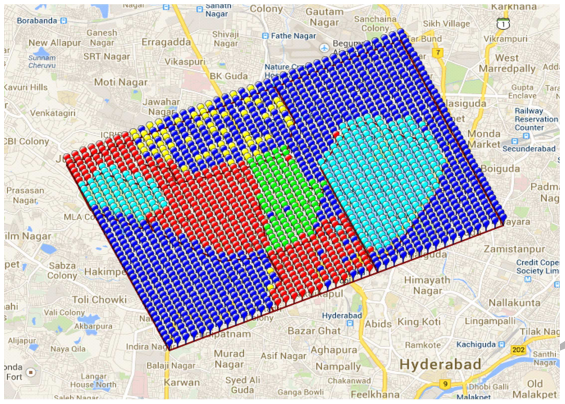


Figure 13: Clustering of people using GFDBSCAN.

The output of the proposed algorithm is shown in Figure 13.

5.8. Validation of Clustering Algorithms

Cluster validity measures are used to measure the performance of clustering algorithms. Several cluster validity indexes [6, 15, 5] have been proposed and Silhouette Coefficient is a prominent measure among them. Silhouette Coefficient combines the ideas of cohesion and separation, both for individual points, as well as clusters and clusterings. A higher Silhouette Coefficient score correlates to a model with better defined clusters. We evaluate the clustering algorithms in this paper both in terms of running time of the algorithm as well as on the basis of Silhouette coefficient. However, it must be observed that no cluster validity measure is agnostic to clustering algorithm. As statistical metrics, the validity measures tend to be more appropriate to one group of algorithms over the other. This is an issue when comparing fuzzy clustering algorithms with non-fuzzy clustering algorithms. In addition to Silhouette Coefficient we also chose Dunn Index and Davies-Bouldin Index for validating clusters.

5.9. Silhouette Coefficient

$$S(i) = \frac{b(i) - a(i)}{max(b(i), a(i))} \tag{8a}$$

$$SI = \frac{1}{n}\sum_{i=1}^{n} S(i) \tag{8b}$$

Silhouette coefficient for an individual point is provided in equation (8a) and the silhouette statistic is given by equation (8b). In equation (8a), $a(i)$ represents the average dissimilarity of object 'i' with respect to all the other objects in its cluster and $b(i)$ represents the smallest average dissimilarity of 'i' to any cluster in which 'i' is not a member. $S(i)$ is the silhouette coefficient for i^{th} object. $a(i)$ indicates how well an object 'i' is assigned to its cluster and $b(i)$ represents the similarity with the second nearest cluster. In general, Silhouette Coefficients lie between 0 and 1. A higher value for silhouette coefficient indicates well-clustered data.

5.10. Dunn index

Dunn Index [12] identifies clusters which are well separated and compact. ence an ideal clustering algorithm is one that maximizes the inter-cluster distance while minimizing the intra-cluster distance. The Dunn index for k clusters is defined by equation (9). A larger Dunn index indicates that compact and well separated clusters exist.

$$DU_k = \min_{i=1,\ldots,k}\left\{\min_{j=2,\ldots,k}\left(\frac{diss(c_i, c_j)}{\max_{m=1,\ldots,k} diam(c_m)}\right)\right\} \tag{9}$$

where,

- $diss(c_i, c_j) = \min_{x\in c_i, y\in c_j}\|x - y\|$ is the dissimilarity between clusters c_i and c_j and
- $diam(C) = \max_{x,y\in C}\|x - y\|$ is the intra-cluster function of the cluster.

5.11. Davies-Bouldin index

Davies-Bouldin [11] index identifies clusters which are far from each other and compact, and is defined as in equation (10a).

$$DB_k = \frac{1}{k}\sum_{i=1}^{k} \max_{j=1,\ldots,k,i\neq j} \left\{ \frac{diam(c_i) + diam(c_j)}{\|c_i - c_j\|} \right\} \tag{10a}$$

$$diam(c_i) = \left(\frac{1}{n_i} \sum_{x \in c_i} \|x - z_i\|^2 \right)^{1/2} \tag{10b}$$

In equation (10b), n_i represents the number of points and z_i is the centroid of cluster c_i. Since the objective is to obtain clusters with minimum intra-cluster distances, smaller values indicate better clustering.

Table 2 and table 3 provide performance results of fuzzy and non-fuzzy clustering algorithms respectively. As the dataset has only the aggregate information, considering direct values for computing the silhouette coefficient is not a good approach. Even though non-fuzzy clustering algorithms give higher value for this coefficient we must remember that non-fuzzy clustering algorithms are not good at dealing with fuzzy data as discussed earlier in this section.

The result shows that FCM clusters the people badly because it takes into account only the fuzziness in the relationship between the cluster centers and its members, but not the fuzziness in the values of the data object. In FDBSCAN each object is represented as a fuzzy object (10 samples from each object) and then we compute the similarity between the two fuzzy objects (10 * 10 computations) to find the similarity between the two objects. This is why FDBSCAN and the proposed GFDBSCAN algorithm take more time for execution. In terms of Silhouette Coefficient, the proposed GFDBSCAN algorithm gives the best value among non fuzzy algorithms. On the other hand GFDBSCAN algorithm performs better than other algorithms when compared using Dunn Index and Davies-Bouldin Index.

Performance Metric	Fuzzy Clustering Algorithms		
	FCM	FDBSCAN	GFDBSCAN
Time(Sec)	8.65	555.28	837.85
Silhouette Coefficient	0.358	0.614	0.619
Dunn Index	0.167	0.361	0.391
Davies-Bouldin Index	0.414	0.731	0.871

Table 2: Validating Clusters produced by Fuzzy Algorithms

Performance Metric	Non-Fuzzy Clustering Algorithm		
	K-Means++	DBSCAN	GDBSCAN
Time(Sec)	0.08	0.74	100.84
Silhouette Coefficient	0.675	0.645	0.143
Dunn Index	0.037	0.325	0.312
Davies-Bouldin Index	0.406	0.712	0.736

Table 3: Validating Clusters produced by Non-Fuzzy Algorithms

5.12. Quality of Service

Banking services provided through ATMs are more effective if they are placed at optimum locations so as to reduce the turnaround time for the customer. We define the turnaround time to include the time to reach an ATM and the queuing time at the ATM. The turnaround time serves as a key metric to quantify ATM deployment strategy. The quality of service expected by the customers depends on their socio-economic background. This is because banks need to invest manpower and IT systems to improve their customer service, and this cost is usually recovered from the customers through service fees. Our clustering algorithm classifies customers using socio-economic patterns. The cost incurred by the bank in providing ATM services is the cost of setting up and operating the ATM,so banks would prefer to place ATMs at a place where there is more foot traffic.

After successfully clustering the people living in the geographical region under study, we try to identify the best ATM placement strategy reflecting

the preferences provided by the people living in the area. In our survey of 1000 people, participants were also asked to indicate their preferences for ATM location and their primary reason for using ATM services. All together, people identified 23 points of interest, but using scree analysis these were reduced to 10 components. Further analysis using principal component analysis (PCA) helped us to cluster the 23 points of interest into the 10 components. We do not discuss these results here as they have been published in another paper. By applying voronoi diagrams we intend to answer the following questions:

1. What is the best combination of deploying N ATMs in a chosen geographical region?
2. Having decided to deploy N ATMs what is the best possible combination to achieve the least possible turnaround time?
3. What is the minimum number of ATMs necessary to achieve a predetermined turnaround time from the customer perspective?

In the current business problem we chose to apply voronoi diagrams to answer question 3 above. We arbitrarily chose the *Punjagutta* region of the city and chose an arbitrary turnaround time of 10 minutes as the expectation from customers living in the same region. The turnaround time is calculated using equation 11. We now rephrase question 3 from above as "Having identified that m is the minimum number of ATMs required to meet the expected turnaround time, which of the ${}^{10}C_m$ provides a more even spread of ATM services?"

$$Time = d * T_a + T_b \tag{11}$$

Where,

- 'd' depicts the distance to the nearest ATM
- T_a represents time needed to reach the nearest ATM for every 1Km, and
- T_b indicates waiting time at ATM location to do a transaction.

Figure 14, shows ATM locations marked with colored star symbols. The results are not overlayed with map information of the same region.

Clearly with 4 ATMs the service time was 16 minutes. So, we tried with a combination of 5 ATMs and in this case we achieved the objective of 15 minutes except for few blocks (50m * 50m blocks) of people where it was

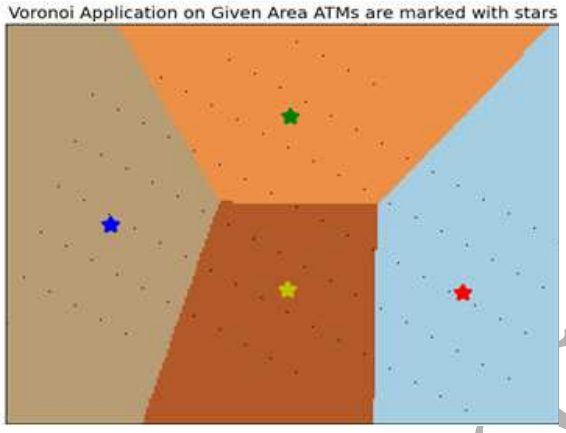


Figure 14: Installation of 4 ATMs in Punjagutta.

taking 16 minutes. The output with installation of 5 ATMs is shown in figure 15. We next tried with 6 ATMs and the turn around time was 12 minutes. Figure 16 represents the ATM locations and the regions covered by ATMs.

Figure 17 provides the best possible travel time from the nearest ATM to each pin (represents the center of a 50m*50m cell) possible as the number of ATMs are increased. The travel time is calculated using equation 11. The distance is calculated using the L_p norm for distance calculation in vector space with the p value set to 1.56. The value for p was arrived at by measuring 1000 sample distances from Google maps for the geographical region under study. From figure 17 it can be seen that the reach-ability (measured in minutes) begins to level off at 6 ATMs. In a practical scenario the number of ATMs can be set based on the expectation of the banking customers in a given geographical area.

5.13. Application of Voronoi Diagrams

Figure 18 shows the deployment of 4 ATMs in the *Punjagutta* region. The solution was arrived at after evaluating ${}^{10}C_4$ combinations of locations for ATM placement. The value of 10 represents the points of interest (as indicated by the customers in the survey) that are available in the region and 4 is the number of ATMs to be placed. Though, placing 7 ATMs is idle, we nevertheless opted for placing only 4 ATMS as shown in Figure 18. The primary reason is that with 4 ATMs, reachability is 15 mins (Figure 17) and with 6 ATMs it is 11 mins. We chose 15 mins as acceptable QoS. Similarly,

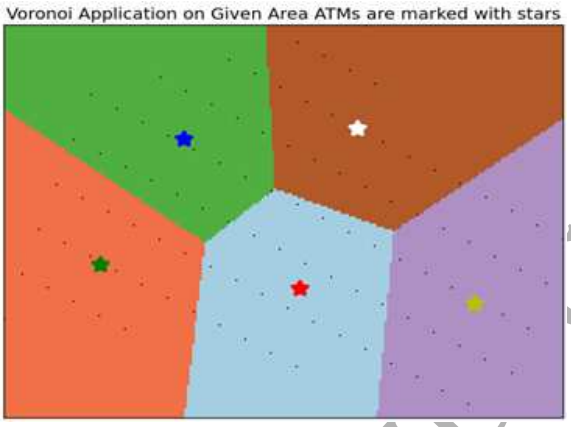


Figure 15: Installation of 5 ATMs in Punjagutta.

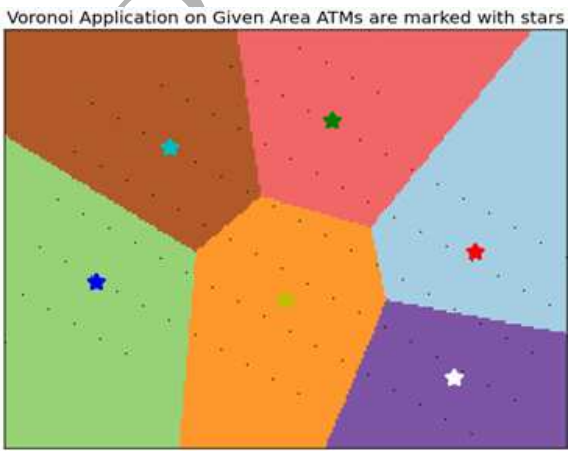


Figure 16: Installation of 6 ATMs in Punjagutta.

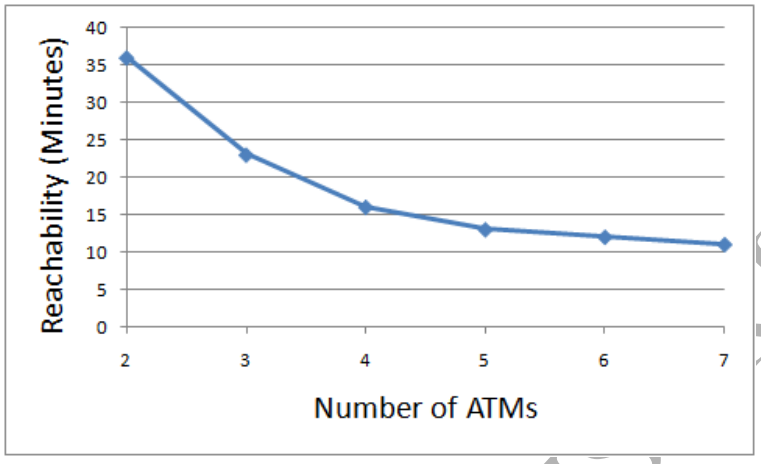


Figure 17: Quality of Service.

evaluation was done by placing 4 ATMs in the neighboring region of *Banjara Hills* and we found that reachability of an ATM was with in 15 mins. The orange bubbles in Figure 18 represent the ATM locations, and the arcs in red color provides the boundaries covered (Voronoi cell or region of coverage) by the respective ATMs.

An ideal placement of ATM is one where the the service coverage (measured by turnaround time) is uniform while at the same time the customer footfalls are high. Therefore, having decided to place 4 ATMs, the question to be addressed is "which is the best ATM placement strategy out of ${}^{10}C_4$ possible combinations?" We use the support weight metric to identify the ideal placement strategy. Support weight helps to identify the turnaround time for a customer (as he traverses across various paths in the region) in the worst case scenario. The support weight [24] of a path is an upper-bound on the distance of any point on the path to any ATM and is solved using delauny triangulation representation of the ATM space. Delauny triangulation corresponds to the dual graph of the voronoi diagram of the ATM space. Support weight for each possible ATM placement strategy was determined and 17 shows the placement strategy for the lowest possible support weight [23]. A similar analysis was done for the neighboring region of *Banjara Hills*.

5.14. Validation

It is challenging to validate the results without actually deploying the solution in the identified region and studying the cash withdrawal pattern by

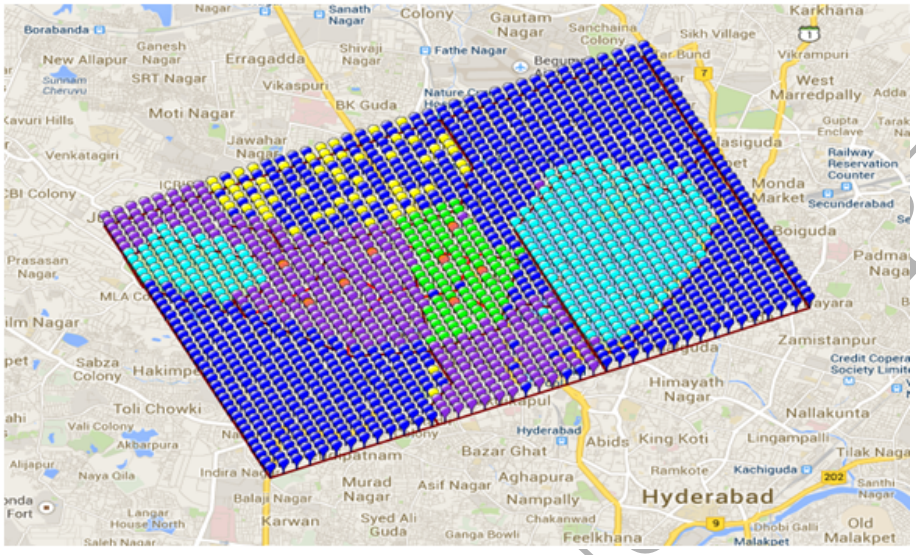


Figure 18: ATM locations after applying GFDBSCAN with Voronoi Diagram.

the customers. Since validating through a real world deployment is difficult, we divided the validation step into two stages.

1. Calculate the turnaround time for the ATM placement strategy currently in place.
2. Evaluate the cash withdrawal pattern for ATMs that are placed in either the same or nearby locations of the ATMs shown in 18

The bank currently has 6 ATMs placed in the region and support weight for the specific configuration of ATMs was 25 minutes. The location of ATMs was unevenly distributed and hence do not have an even coverage area. To validate the second stage we used the cash withdrawal information from 3 different banks and ranked them in the order of cash withdrawals. The 4 ATM positions indicated in 18 ranked consistently among the top 6 ATMs in terms of usage for the geographical region under study.

6. Conclusion

The selection of site locations for ATM placement has great impact on the banking business, and these locations are dependent on the financial needs of the people living in the service area. The proposed solution involves

clustering people living in a city based on their socio-economic background. For the process of clustering we modified generalized density based clustering algorithm so as to better handle fuzziness in the perceived values of the socio-economic parameters of people living in a specific geographical area. Our proposed algorithm, which we call generalized fuzzy density based clustering algorithm (GFDBSCAN), effectively clusters people staying in all areas ,and we compare the results of various clustering algorithms using Silhouette Coefficient, Dunn index and Davies-Bouldin index. The clustering results achieved by the proposed GFDBSCAN were better than other clustering algorithms proposed in the literature. GFDBSCAN was able to cluster the city in accordance with the widely understood perception of the geographical and financial boundaries of municipal areas in Hyderabad city, India. Finally we applied voronoi diagrams to identify the best locations to place ATMs to ensure that the people's preferences are served and at the same time ensuring optimum service area under each ATM. The uniformity of ATM coverage area is measured using the support weight of the delauny dual of voronoi diagram.

In this paper we solve the issue of identifying the best locations for establishing the ATMs in a green field area (no ATMs present in the area to begin with), and in the future we want to extend this problem to the areas where few ATMs already exist. We propose to use a multi-player game theoretic-model (for each cluster) to understand the tradeoffs between setting up a new ATM versus purchasing services for another bank.

References

[1] Aggarwal, C., Yu, P., May 2009. A survey of uncertain data algorithms and applications. Knowledge and Data Engineering, IEEE Transactions on 21 (5), 609–623.

[2] Aldajani, M. A., Alfares, H. K., Nov. 2009. Location of banking automatic teller machines based on convolution. Comput. Ind. Eng. 57 (4), 1194–1201.

[3] Ankerst, M., Breunig, M. M., Kriegel, H.-P., Sander, J., 1999. Optics: Ordering points to identify the clustering structure. In: Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data. SIGMOD '99. ACM, New York, NY, USA, pp. 49–60.

[4] Arthur, D., Vassilvitskii, S., 2007. K-means++: The advantages of careful seeding. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms. SODA '07. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, pp. 1027–1035.

[5] Berkhin, P., 2006. A survey of clustering data mining techniques. Grouping Multidimensional Data, 25–71.

[6] Bezdek, J., Pal, N., Jun 1998. Some new indexes of cluster validity. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 28 (3), 301–315.

[7] Bezdek, J. C., Ehrlich, R., Full, W., 1984. Fcm: The fuzzy c-means clustering algorithm. Computers & Geosciences 10 (2), 191–203.

[8] Boriah, S., Chandola, V., Kumar, V., 2008. Similarity measures for categorical data: A comparative evaluation. In: SDM. SIAM, pp. 243–254.

[9] Brandes, U., Gaertler, M., Wagner, D., 2003. Experiments on graph clustering algorithms. In: Di Battista, G., Zwick, U. (Eds.), Algorithms - ESA 2003. Vol. 2832 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 568–579.

[10] Chiu, S. L., May 1994. Fuzzy model identification based on cluster estimation. J. Intell. Fuzzy Syst. 2 (3), 267–278.

[11] Davies, D. L., Bouldin, D. W., Feb. 1979. A cluster separation measure. IEEE Trans. Pattern Anal. Mach. Intell. 1 (2), 224–227.

[12] Dunn, J. C., 1974. Well separated clusters and optimal fuzzy-partitions. Journal of Cybernetics 4, 95–104.

[13] Duttagupta, A., Bishnu, A., Sengupta, I., 2008. Maximal breach in wireless sensor networks: Geometric characterization and algorithms. In: Kutylowski, M., Cichol, J., Kubiak, P. (Eds.), Algorithmic Aspects of Wireless Sensor Networks. Vol. 4837 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 126–137.

[14] Ester, M., Kriegel, H.-P., Sander, J., Xu, X., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In:

Proc. of 2nd International Conference on Knowledge Discovery. AAAI Press, pp. 226–231.

[15] Halkidi, M., Batistakis, Y., Vazirgiannis, M., Sep. 2002. Clustering validity checking methods: Part ii. SIGMOD Rec. 31 (3), 19–27.

[16] Han, J., 2005. Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[17] Hartuv, E., Shamir, R., Dec. 2000. A clustering algorithm based on graph connectivity. Inf. Process. Lett. 76 (4-6), 175–181.

[18] Huang, Z., Sep. 1998. Extensions to the k-means algorithm for clustering large data sets with categorical values. Data Min. Knowl. Discov. 2 (3), 283–304.

[19] Iamtrakul, P., Teknomo, K., Hokao, K., 2003. Evaluation of public park location using voronoi diagram. In: 9 th International Student Seminar on Transport Research (ISSOT 2003). pp. 16–18.

[20] Kriegel, H.-P., Pfeifle, M., 2005. Density-based clustering of uncertain data. In: Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining. KDD '05. ACM, New York, NY, USA, pp. 672–677.

[21] Kriegel, H.-P., Pfeifle, M., 2005. Hierarchical density-based clustering of uncertain data. In: Proceedings of the Fifth IEEE International Conference on Data Mining. ICDM '05. IEEE Computer Society, Washington, DC, USA, pp. 689–692.

[22] MacQueen, J. B., 1967. Some methods for classification and analysis of multivariate observations. In: Cam, L. M. L., Neyman, J. (Eds.), Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability. Vol. 1. University of California Press, pp. 281–297.

[23] Megerian, S., Koushanfar, F., Potkonjak, M., Srivastava, M. B., Member, S., 2005. Worst and best-case coverage in sensor networks. IEEE Transactions On Mobile Computing 4, 84–92.

[24] Meguerdichian, S., Koushanfar, F., Potkonjak, M., Srivastava, M. B., 2001. Coverage problems in wireless ad-hoc sensor networks. In: IEEE INFOCOM. pp. 1380–1387.

[25] Mendes, A. B., Themido, I. H., 2004. Multi-outlet retail site location assessment. In: International Transactions in Operational Research. Vol. 11. Wiley, pp. 1–18.

[26] Min, H., Melachrinoudis, E., 2001. The three-hierarchical location-allocation of banking facilities with risk and uncertainty. International Transactions in Operational Research 8 (4), 381–401.

[27] Monteiro, M., Fontes, B. M. M. D., 2005. Locating and sizing bank-branches by opening, closing or maintaining facilities. In: Haasis, H.-D., Kopfer, H., Schnberger, J. (Eds.), OR. pp. 303–308.

[28] Murray, A. T., 2009. Business Site Selection, Location Analysis and GIS. Wiley, New York.

[29] Neelisetti, R. K., 12 2009. Improving reliability of wireless sensor networks for target tracking using wireless acoustic sensors. Ph.D. thesis, Auburn University, Auburn, AL, USA.

[30] Nefti, S., Djouani, K., 2003. Extended fuzzy clustering algorithm based on an inclusion concept. In: The 12th IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2003, St. Louis, Missouri, USA, 25-28 May 2003. pp. 869–874.

[31] Nickel, S., Puerto, J., 2005. Location theory: A unified approach, springer. European Journal of Operational Research 181 (1), 523–525.

[32] Olowookere, E., Olowookere, A., 2014. Determinants of atm usage among students of tertiary institutions in nigeria. Journal of Economic Theory 8 (1), 5–13.

[33] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830.

[34] Sander, J., Ester, M., Kriegel, H.-P., Xu, X., Jun. 1998. Density-based clustering in spatial databases: The algorithm gdbscan and its applications. Data Min. Knowl. Discov. 2 (2), 169–194.

[35] Tan, P.-N., Steinbach, M., Kumar, V., May 2005. Introduction to Data Mining, us ed Edition. Addison Wesley.

[36] Wang, Q., Batta, R., Rump, C. M., 2002. Algorithms for a facility location problem with stochastic customer demand and immobile servers. Annals OR 111 (1-4), 17–34.

[37] Xu, R., Wunsch, II, D., May 2005. Survey of clustering algorithms. Trans. Neur. Netw. 16 (3), 645–678.

[38] Yushimito, W., Jaller, M., Ukkusuri, S., 2012. A voronoi-based heuristic algorithm for locating distribution centers in disasters. Networks and Spatial Economics 12 (1), 21–39.