

Incremental Clustering with special emphasis on Spatio Temporal Data

Submitted By

Dhruv Jain

15MCEC10



DEPARTMENT OF COMPUTER ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2017

Incremental Clustering with Special Emphasis on Spatio Temporal Data

Major Project

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology in Computer Science and Engineering

Submitted By

Dhruv Jain

(15MCEC10)

Guided By

Dr KP Agrawal



DEPARTMENT OF COMPUTER ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

December 2017

Certificate

This is to certify that the major project entitled ”**Incremental Clustering with Special Emphasis on Spatio Temporal Data**” submitted by **Dhruv Jain (Roll No: 15MCEC10)**, towards the partial fulfillment of the requirements for the award of degree of Master of Technology in Computer Science and Engineering of Nirma University, Ahmedabad, is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this thesis, to the best of my knowledge, haven’t been submitted to any other university or institution for award of any degree or diploma.

Dr. K.P Agrawal
Guide & Associate Professor,
CE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. Priyanka Sharma
Professor,
Coordinator M.Tech - CSE
Institute of Technology,
Nirma University, Ahmedabad

Dr. Sanjay Garg
Professor and Head,
CE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr Alka Mahajan
Director,
Institute of Technology,
Nirma University, Ahmedabad

Statement of Originality

I, **Dhruv Jain**, Roll. No. **15MCEC10**, give undertaking that the Thesis entitled **”Incremental Clustering With special emphasis on Spatio Temporal Data”** submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Computer Science & Engineering** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

Signature of Student

Date:

Place:

Endorsed by
Prof KP Agrawal
(Signature of Guide)

Acknowledgements

It gives me immense pleasure in expressing thanks and profound gratitude to **Dr K. P. Agrawal**, Associate Professor, Computer Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his valuable guidance and continual encouragement throughout this work. The appreciation and continual support he has imparted has been a great motivation to me in reaching a higher goal. His guidance has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

It gives me an immense pleasure to thank **Dr. Sanjay Garg**, Hon'ble Head of Computer Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his kind support and providing basic infrastructure and healthy research environment.

A special thank you is expressed wholeheartedly to **Dr Alka Mahajan**, Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad for the unmentionable motivation she has extended throughout course of this work.

I would also thank the Institution, all faculty members of Computer Engineering Department, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

- Dhruv Jain
15MCEC10

Abstract

Today in this age where we talk about terabytes and petabytes of data generation each and every day, there is a need for tools which help in the analysis and processing of this data. Clustering is a type of unsupervised learning in which clusters of different types can be created using various algorithms such as K-Means, DBSCAN, OPTICS, Nearest-neighbor chain and many others. It is a process of creating different partitions to a set of data (or objects) into a set of meaningful sub-classes, called clusters.

Density-Based Spatial Clustering of Applications with Noise(DBSCAN) is a density based clustering algorithm used for data clustering.

This algorithm will be used to a special kind of dataset ie spatio-temporal dataset. Here these datasets are used to form clusters and then incremental clustering is performed on them which eventually will add to reducing the time complexity.

Abbreviations

DBSCAN	Density-Based Spatial Clustering of Applications with Noise
STDBSCAN	Spatio Temporal DBSCAN
KNN	K Nearest Neighbors
NDVI	Normalized Difference Vegetation Index
OPTICS	Ordering Points To Identify the Clustering Structure

—

Contents

Certificate	iii
Statement of Originality	iv
Acknowledgements	v
Abstract	vi
Abbreviations	vii
List of Figures	x
1 Introduction	1
1.1 Clustering	1
1.2 DBSCAN Algorithm	2
1.3 Spatio-Temporal Datasets:	6
1.4 ST DBSCAN:	6
1.4.1 Distance Measure:	6
1.4.2 ST-DBSCAN Algorithm:	8
1.5 Incremental Clustering:	8
1.6 NDVI:	9
1.7 Our Approach	9
2 Literature Survey	11
2.1 A Density Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise:	11
2.2 ST-DBSCAN: An algorithm for clustering spatial-temporal data:	12
2.3 OPTICS: Ordering Points To Identify the Clustering Structure:	13
2.4 Analysis and Study of Incremental DBSCAN Clustering Algorithm:	14
2.5 Literature Review of Spatio-Temporal Database Models:	15
3 Implementation	16
3.1 Data-Set:	16
3.2 R:	16
3.3 Working:	16
3.3.1 Heuristics:	17
3.3.2 Proposed Approach:	18
4 Result And Validation	23

5	Conclusion	25
6	Future Enhancements	26
	Bibliography	27

List of Figures

1.1	Directly Density Reachable	3
1.2	Density Reachable	3
1.3	DBSCAN Algorithm	4
1.4	Points in 3d space	5
1.5	Clusters Formation by DBSCAN	5
1.6	ST-Dbscan	8
1.7	Clusters formed on Indian map	10
3.1	KNN Distance Plot	17
3.2	Input File	18
3.3	Dbscan Execution	19
3.4	Points on Google Map	19
3.5	Combined figure	20
3.6	NDVI Plot of Gujarat in Jan 08	20
3.7	Cluster Associated with their Range Values	21
3.8	Cluster Created From the Original Algorithm	21
3.9	Cluster Created from Incremental Clustering	22
4.1	Result	23

Chapter 1

Introduction

This section includes all the concepts and the pre-requisites that are needed to understand the working of the system. Here each sub-section introduces a new concept which together will provide an environment for the accurate and efficient working of the algorithm.

1.1 Clustering

Clustering is a method which is used in data mining which can help in creating groups without training the data. Clustering methods are used to form unknown classes from given data having similar properties. It is used for grouping data into classes on the basis of their similarity with each other. It is an unsupervised learning algorithm ie. training of data is not required.

Types of Clustering Methods [\[1\]](#) :-

- Partitioning Method

In this type of clustering method, k partitions are constructed of n objects. These partitions will form clusters where $k \leq n$. The following requirements should be met:

1. At least one object exists in each cluster.
2. Each and every object should belong to exactly one group.

- Hierarchical Method

In this method, a the given set of data is decomposed hierarchically. There are two types of this cluster:

1. Agglomerative Approach: Here each object is designated with its own cluster

and then slowly merging of clusters takes place.

2. Divisive Approach: Here each and every object is in a single cluster and then division of this clusters takes place. It follows a top-down approach.

- Density based Clustering In this method, the basic idea is to form clusters on the notion of density. Each object has a mentioned radius and if some threshold number of points lie within this radius, a cluster is formed.
- Grid-Based Method A finite number of cells are formed in the object space which resembles a grid structure. The objects are quantized to belong to each of these cells and thus a pre-defined clusters are formed.
- Constraint-based Method User and application oriented constraints are incorporated in this type of clustering. The desired expectation of the result is referred as a constraint. They provide us an interactive way of communicating with the clustering process.

1.2 DBSCAN Algorithm

Density-Based Spatial Clustering of Applications with Noise [2] is a density based clustering algorithm. This algorithm forms clusters from the data-points that are closely related to each other in some space. It can discover arbitrary shaped clusters.

Here are some definitions which might help in understanding this algorithm.

- Eps neighbourhood of a point: It is the radius of a data object till where its effect lies and all the points that are interior to this cluster lies here.
- Minpts: It is the minimum number of points required to form a cluster.
- Core Object: An object is a core object if it has more than a specified number of objects ie.Minpts within a certain radius ie.Eps.
- Border Object: It is an object which has less number of objects in its eps than Minpts, but it lies in the neighborhood of a core point.

- Noise Object: An object which is neither a border object nor a core object.
- Directly Density Reachable: If an object q is in ϵ neighborhood of p and p is a core object then object q is known as directly density reachable object.

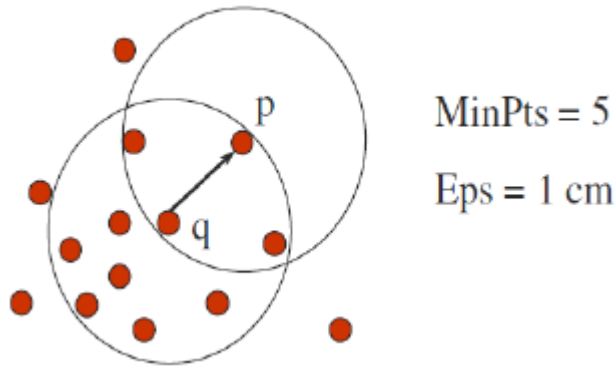


Figure 1.1: Directly Density Reachable

- Density Reachable: An object p is density-reachable from q w.r.t ϵ and MinPts if there is a chain of objects p_1, \dots, p_n , with $p_1 = q$, $p_n = p$ such that p_{i+1} is directly density-reachable from p_i w.r.t ϵ and MinPts for all $1 \leq i \leq n$.

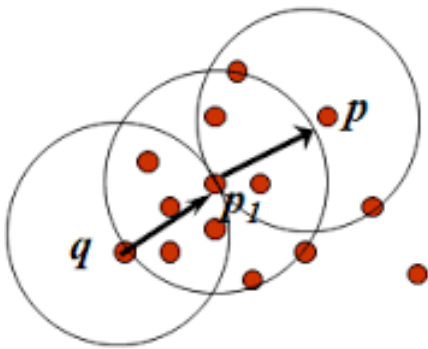


Figure 1.2: Density Reachable

- Density Connected: If there exists an object o such that both object p and q are density reachable from o then p and q are said to be density connected.

Here is the algorithm to perform DBSCAN clustering on data in fig. 1.3.

```
DBSCAN(D, eps, MinPts) {
    C = 0
    for each point P in dataset D {
        if P is visited
            continue next point
        mark P as visited
        NeighborPts = regionQuery(P, eps)
        if sizeof(NeighborPts) < MinPts
            mark P as NOISE
        else {
            C = next cluster
            expandCluster(P, NeighborPts, C, eps, MinPts)
        }
    }
}

expandCluster(P, NeighborPts, C, eps, MinPts) {
    add P to cluster C
    for each point P' in NeighborPts {
        if P' is not visited {
            mark P' as visited
            NeighborPts' = regionQuery(P', eps)
            if sizeof(NeighborPts') >= MinPts
                NeighborPts = NeighborPts joined with NeighborPts'
        }
        if P' is not yet member of any cluster
            add P' to cluster C
    }
}

regionQuery(P, eps)
    return all points within P's eps-neighborhood (including P)
```

Figure 1.3: DBSCAN Algorithm

DBSCAN algorithm takes 2 parameters as inputs(eps, MinPts) and it gives out the cluster each point belongs to.

Dbscan is applied on the points in 3d space given in fig. 1.4. These are random points extended in space. Keeping the spatial attribute in consideration, clusters are formed

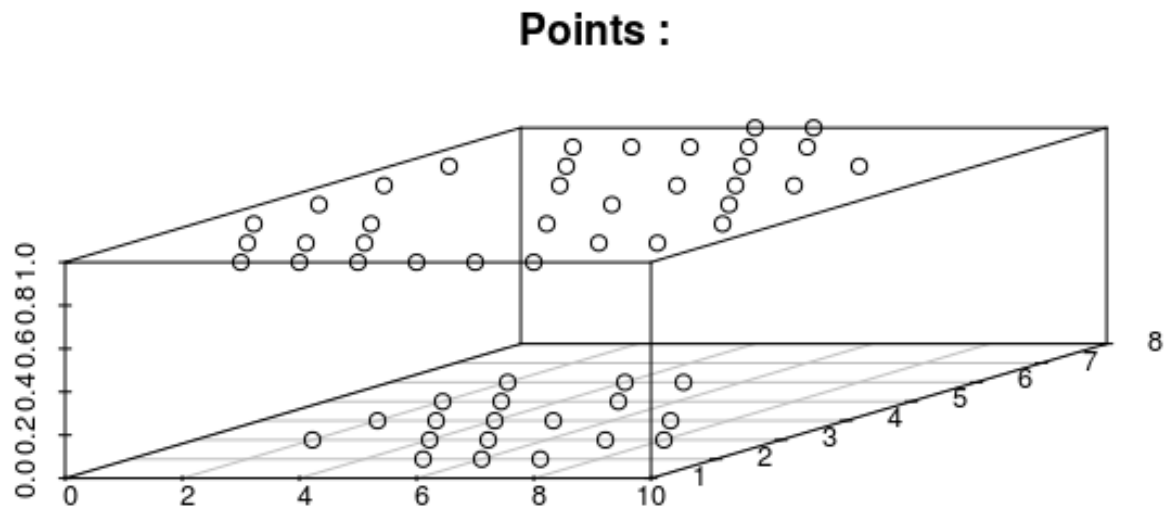


Figure 1.4: Points in 3d space

among the points. Each of the different colors represent a separate cluster. Since clusters are formed according the spatial property, points that lie close to each other forms a cluster.

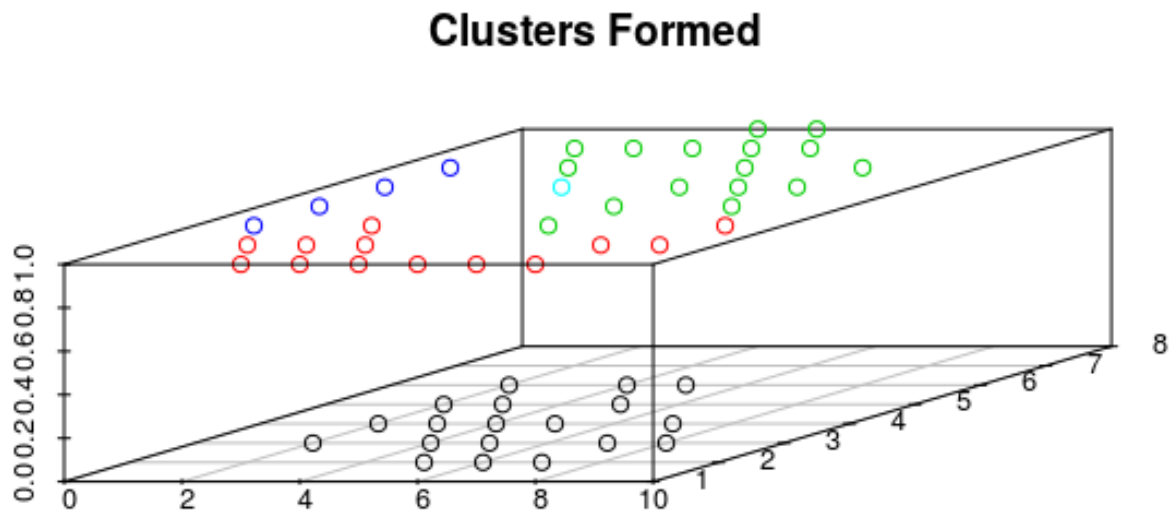


Figure 1.5: Clusters Formation by DBSCAN

1.3 Spatio-Temporal Datasets:

These are the datasets that are related with each other in time and space. It manages both the time as well as the space information in a single database. An example would be tracking a moving car which can at different places at different time. Many applications use such large datasets such as maps, repositories of remote-sensing images, virtual globes, Public safety, Public health and security, Mobile-commerce among many others. Classical Data Mining gives poor results if they are worked upon such datasets as the main reason is that classical data is often discrete while spatio-temporal data is continuous.

1.4 ST DBSCAN:

This is an algorithm which is an extension of the original DBSCAN where it takes into account the spatial and temporal properties of data into account while clustering them[3]. In other words they tend to work with spatio-temporal data. There were 3 improvements done on the original algorithm which are listed as follows:

- ST-DBSCAN can cluster the spatio-temporal data according to its non-spatial, spatial and temporal attributes.
- It can detect noise points when clusters of different densities exist. It does this by assigning a density factor with each cluster. This was not possible in the earlier version of the DBSCAN.
- Values of Opposite border objects in a cluster might be very different from each other. This algorithm solves this problem by comparing the average value of a cluster with new coming value.

1.4.1 Distance Measure:

To check the similarity between two objects that may or may not lie in a cluster, a characteristic value or a distance measure is to be evaluated[4]. The most used distance measures are Minkowski distance, Manhattan distance and Euclidean distance. Euclidean Distance can be evaluated as (1)

$$dist(i, j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \dots + (x_{in} - x_{jn})^2} \quad (1)$$

where i, j are two n -dimensional objects.

Here in STDBSCAN we use two distance parameters $Eps1$ which is used to measure the closeness of two points geographically (spatial attributes), and $Eps2$ which is used to measure the closeness of non-spatial values.

$$Eps1 = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (2)$$

$$Eps2 = \sqrt{(t_1 - t_3)^2 + (t_2 - t_4)^2} \quad (3)$$

where there are two points $A(x_1, y_1, t_1, t_2)$ and $B(x_2, y_2, t_3, t_4)$.

To incorporate temporal aspects of the data, the data should be modified such that there exists only the data which are temporal neighbours. Two objects are considered to be temporal neighbors if the values of these objects are observed in consecutive time units such as consecutive days in the same year or in the same day in consecutive years.

1.4.2 ST-DBSCAN Algorithm:

```
Algorithm ST_DBSCAN (D, Eps1, Eps2, MinPts, Δε)
// Inputs:
// D={o1, o2, ..., on} Set of objects
// Eps1 : Maximum geographical coordinate (spatial) distance value.
// Eps2 : Maximum non-spatial distance value.
// MinPts : Minimum number of points within Eps1 and Eps2 distance.
// Δε : Threshold value to be included in a cluster.
// Output:
// C={C1, C2, ..., Ck} Set of clusters

Cluster_Label = 0

For i=1 to n
    // (i)
    If oi is not in a cluster Then // (ii)
        X=Retrieve_Neighbors(oi, Eps1, Eps2) // (iii)

        If |X| < MinPts Then
            Mark oi as noise // (iv)
        Else //construct a new cluster (v)
            Cluster_Label = Cluster_Label + 1

            For j=1 to |X|
                Mark all objects in X with current Cluster_Label
            End For

            Push(all objects in X) // (vi)

            While not IsEmpty()
                CurrentObj = Pop()
                Y= Retrieve_Neighbors(CurrentObj, Eps1, Eps2)

                If |Y| >= MinPts Then
                    ForAll objects o in Y // (vii)
                        If (o is not marked as noise or it is not in a cluster) and
                            |Cluster_Avg() - o.Value| <= Δε Then
                            Mark o with current Cluster_Label
                            Push(o)
                        End If
                    End For
                End If
            End While
        End If
    End For
End For
End Algorithm
```

Figure 1.6: ST-Dbscan

1.5 Incremental Clustering:

Once the clusters are made, the next most important task is to predict the next object and make it fit in a cluster and declare it as a noise. For this one way to approach is to form the clusters again including the new dataset.[5] But this method is not very efficient to function. As the calculation will have to be done again with an increased dataset which will require lot of time and resources.

Second way to approach this is what is known as incremental clustering. The clusters need not be evaluated again, but a new method by which each new object can be identified to a cluster can be generated without the need of this tremendous calculation. Only a small number of objects are to be stored in the main memory and so this process is efficient. One way to do is by calculating the average value of each core point for a

cluster and only store them in the main memory and then to measure the new points with these average values.

1.6 NDVI:

The Normalized Difference Vegetation Index is a spatio-temporal dataset that is an indicator which analyses whether the target has a live green vegetation cover or not. It is calculated from the images that we get from the satellite of a target land and the type of visible rays that is reflected back from earth.

NDVI can be calculated as follows [6]:

$$NDVI = \frac{(NIR - Red)}{(NIR + Red)}$$

where Red and NIR corresponds to the spectral reflectance measurements of the red (visible) and near-infrared regions respectively

1.7 Our Approach

: The main objective over here is to devise a way of incremental clustering and comparing it with the conventional DBSCAN algorithm and to get its accuracy and efficiency. By using incremental methods there might be loss of accuracy in assigning the points to clusters then the regular way but still there will be a huge difference in the run-time of the algorithm. The NDVI dataset is used here, so a cluster is to be made out of these NDVI points and these are to be plotted on the map according to the spatial coordinates mentioned. An example of these clusters are shown below in fig. 1.7 [7]

Similar kind of clusters will be formed while working in the NDVI values of the state Gujarat. Once the clusters are made, the NDVI values of certain points will change with time. Some points are taken and their new NDVI values is used to build the new cluster incrementally.

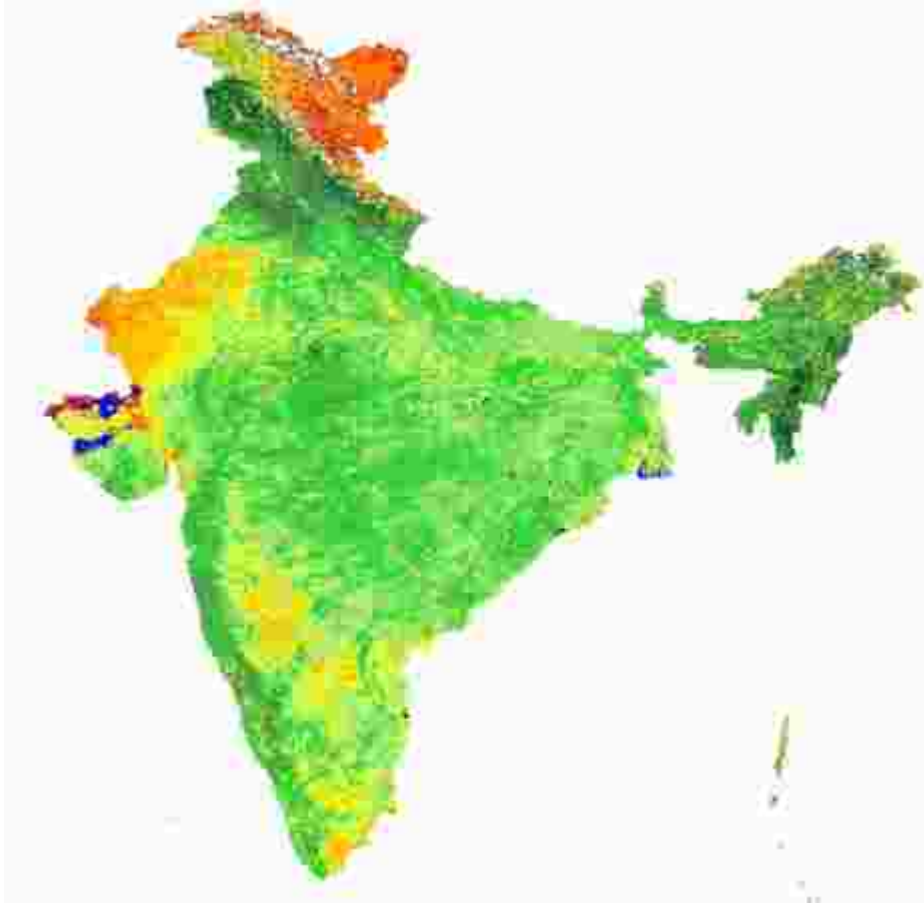


Figure 1.7: Clusters formed on Indian map

This cluster will be compared with the one which will be built directly from DBSCAN algorithm. The similarity in cluster points will give the accuracy and the ratio of execution time between the incremental and the original one will provide with the speed up.

Chapter 2

Literature Survey

This section provides an overview of the work done till now in this field and the different techniques and concepts which will be used in the implementation.

2.1 A Density Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise:

Author- Martin Ester, Hans-Peter Kriegel, Jiirg Sander, Xiaowei Xu

Publisher- Knowledge Discovery And Data Mining

Year- 1996

Datasets- Spatial Dataset, SEQUOIA 2000 benchmark data.

This is the base paper where the concepts of a Density Based Clustering-DBSCAN was first introduced. DBSCAN was compared with a well-known algorithm-CLARANS and it outperformed it by a factor of 100 in terms of efficiency. DBSCAN can identify clusters of arbitrary shape and so it is extensively used with spatio-temporal datasets.

Future Improvements-

- Spatial Datasets can contain objects that are polygons.
- Applications of DBSCAN to high dimensional feature spaces should be investigated.

2.2 ST-DBSCAN: An algorithm for clustering spatial-temporal data:

Author- Derya Birant, Alp Kut

Publisher- Data and Knowledge Engineering

Year- 2007

Datasets- Spatio-temporal Data, Topex/Poseidon Satellite provides sea surface height residual data.

This paper has introduced a new algorithm ST-DBSCAN which is very similar to DBSCAN but it can work with spatio-temporal data. It presents a spatio-temporal data warehouse system designed for storing and clustering a wide range of spatio-temporal data. The original DBSCAN algorithm was improved in three important aspects-

- This algorithm can cluster spatialtemporal data according to its non-spatial, spatial and temporal attributes.
- DBSCAN cannot detect some noise points when clusters of different densities exist. while ST-DBSCAN solves this problem by assigning to each cluster a density factor.
- The values of border objects in a cluster may be very different than the values of border objects in opposite side, this algorithm solves this problem by comparing the average value of a cluster with new coming value.

Future Improvements-

- It is intended to run the algorithm in parallel in order t improve the performance.
- More useful heuristics may be found to determine the input parameters Eps and MinPts.

2.3 OPTICS: Ordering Points To Identify the Clustering Structure:

Author- Mihael Ankerst, Markus M Breunig, Hans Peter Kriegel, Jorg Sander

Publisher- International Conference on Management of Data

Year- 1999

Datasets- 30000 records consisting of 16 attributes of fourier-transformed data describing contours of industrial parts and the reachability attribute are visualized by setting the discretization to just three different colors.

This paper has introduced another Density based clustering algorithm, OPTICS. This algorithm does not produce a clustering of the data set explicitly but instead creates an augmented ordering of the database representing its density-based clustering structure and it not only extracts the traditional clustering information but also the intrinsic clustering structure. Since it needs less input parameters then DBSCAN algorithm, its complexity is reduced as the output depends majorly on the input parameters.

Future Improvements-

- For very high-dimensional spaces, no index structure exist to efficiently support the hypersphere range queries needed by the OPTICS algorithm.
- There can be a trade-off between a limited amount of accuracy for a large gain in efficiency by using an incremental approach.
- Incrementally managing a cluster-ordering when updates on the database occur is another interesting challenge.

2.4 Analysis and Study of Incremental DBSCAN Clustering Algorithm:

Author- Sanjay Chakraborty, Professor NK Nagwani

Publisher- International Journal of Enterprise Computing and Business Systems.

Year- 2011

Datasets- Set of two dimensional data objects are used to explain Incremental clustering in DBSCAN.

This paper describes the incremental behavior of Density based clustering. Here DBSCAN algorithm is used and a way to add incremental clusters is discussed. It describes at what percent of delta change in the original database the actual and incremental DBSCAN algorithms behave like same. Thus with slight effect in accuracy, efficiency can be improved. With loading only a few datapoints instead of the entire dataset to identify the clusters of new emerging data, efficiency can be improved.

Future Improvements-

- Analysing the other popular clustering techniques in incremental fashion.
- Spatio-Temporal Datasets can be used.

2.5 Literature Review of Spatio-Temporal Database

Models:

Author- Nikos Pelekis, Babis Theodoulidis, Ioannis Kopanakis, Yannis Theodoridis

Publisher- The Knowledge Engineering Review.

Year- 2004

Datasets- Spatio-Temporal datasets used here to describe the storage of data.

This paper reviews the different types of spatio-temporal datamodels that have been proposed in the literature as well as new theories and concepts that have emerged. It provides an overview of previous achievements within the domain and critically evaluates the various approaches through the use of a case study and the construction of a comparison framework.

Future Improvements-

- Continuous motion modeling
- Multiple time line representation
- Complex behavioral spatio-temporal queries
- Very large data manipulation

Chapter 3

Implementation

In this section the working and the methods used in the algorithm are explained in detail. First, the tools that were used are mentioned below.

3.1 Data-Set:

The dataset consists of the NDVI values of Gujarat of the year 2001. This dataset consists of Latitude, Longitude and 23 different NDVI readings throughout the year. There are 7028 tuples each containing the above columns.

3.2 R:

It is a free software environment which is used for graphics and statistical computing. It provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering) and graphical techniques, and is highly extensible. There are various packages which can be imported and used such as dbscan and spacetime for spatio-temporal datasets.[\[8\]](#). R has been used mainly for analysis and the formation of incremental clusters.

3.3 Working:

First the training data is imported in the R workspace and it will be declared in a variable 'mydata'.

```
mydata = read.csv("gujdata.csv")
```

Feature or Attribute selection is performed on this data and only the required columns are kept.

3.3.1 Heuristics:

The 2 additional inputs(eps and Minpts) are to be determined, before performing DB-SCAN algorithm on this data[2].

- For calculating the appropriate Minpts value the logarithm of the total tuples in the data that exist have been used .The value of minPts found was 9.

$$minPts = ceiling(log(length(mydata)))$$

- Then to find eps value, a knn distance plot is generated where k=Minpts. The knee of the curve will give the appropriate eps value. The value of eps came around 0.00035.

$$dbscan :: kNNdistplot(mydata, k = minPts)$$

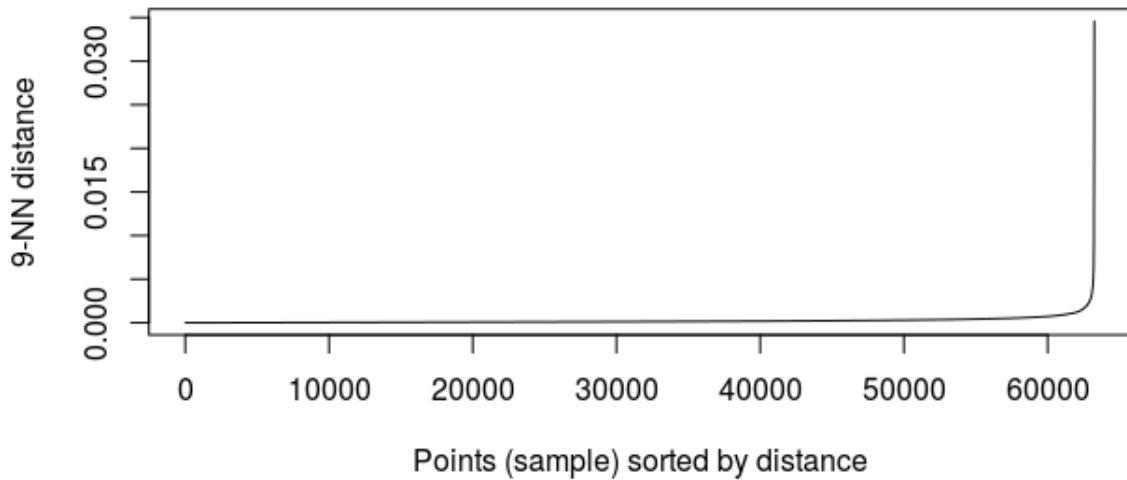


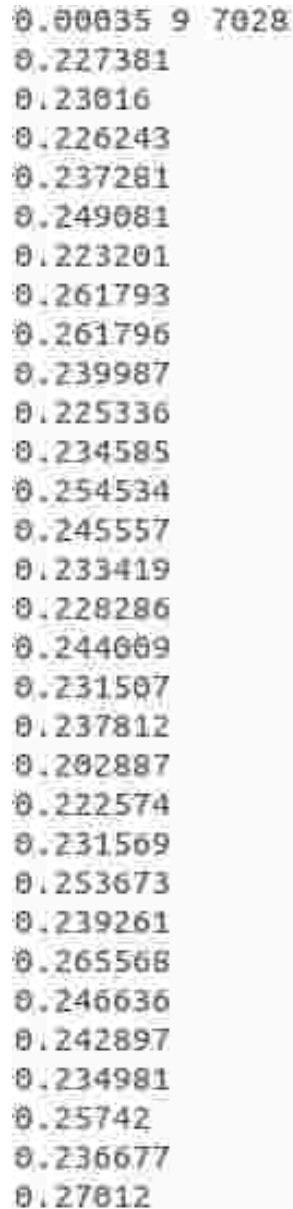
Figure 3.1: KNN Distance Plot

3.3.2 Proposed Approach:

After determining the input parameters, DBSCAN algorithm is to be applied on the NDVI data, and clusters are to be formed accordingly. DBSCAN Algorithm is implemented in C Language. To compile the code, gcc compiler is used.

gcc dbscan.c -lm

This code takes in a .dat file as an input. A sample of that file is shown below in fig. 3.2.



```
0.00035 9 7028
0.227381
0.23016
0.226243
0.237281
0.249081
0.223201
0.261793
0.261796
0.239987
0.225336
0.234585
0.254534
0.245557
0.233419
0.228286
0.244009
0.231507
0.237812
0.202887
0.222574
0.231509
0.253673
0.239261
0.265568
0.246636
0.242897
0.234981
0.25742
0.236677
0.27012
```

Figure 3.2: Input File

The first line here mentions the eps value, MinPts and the total number of points respectively. The execution of DBSCAN algorithm is shown below in fig. 3.3

```

dhruv@dhruv-VPCEB44EN:~/Desktop$ gcc db.c -lm
dhruv@dhruv-VPCEB44EN:~/Desktop$ cat example.dat|./a.out
Epsilon: 0.000350
Minimum points: 9
Core Points: 97
Number of noise points:1926
Number of zero points:1595

```

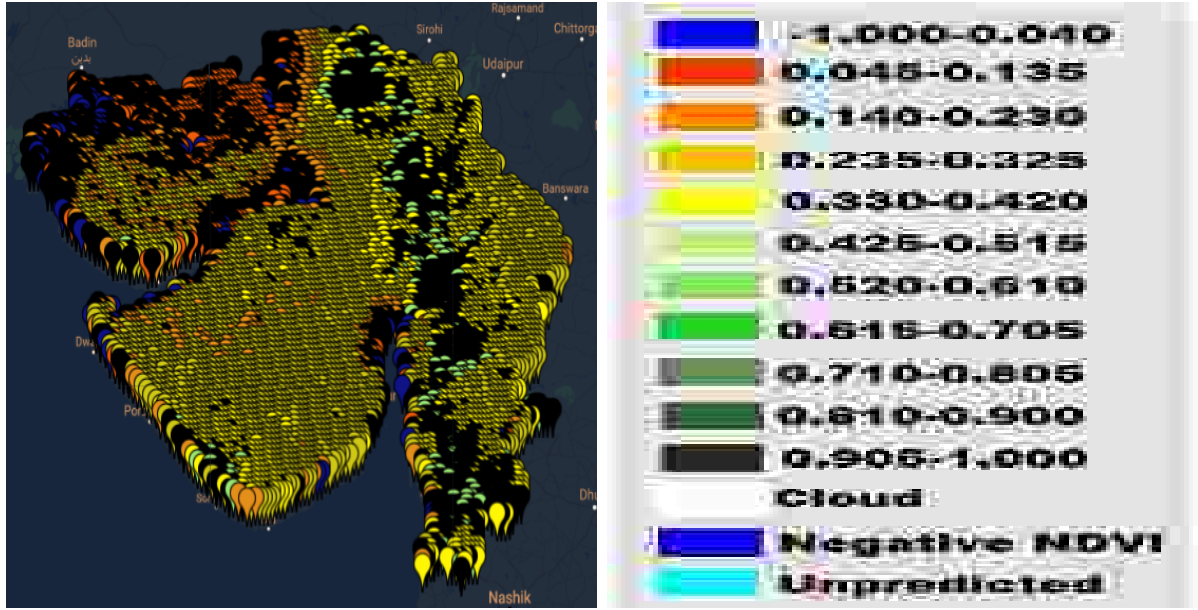
Figure 3.3: Dbscan Execution

With the help of GoogleMap Javascript API, all the 7028 points with latitude and longitude can be located on the map as shown in fig. 3.4.



Figure 3.4: Points on Google Map

Once the clusters are created and the map is ready, these cluster points will be plotted on the map. But for that these points are to be stored in a database from where Javascript and PHP could access. MySQL database is used to store these points. Total number of clusters formed by DBSCAN are 96. Average is taken of all the points lying in that cluster and the clusters are made to lie in a range value of specified NDVI values. Points are added to the map for visualization of clusters according to the NDVI cluster range as shown in fig. 3.5a.



(a) Cluster Formation of Points on Google Map

(b) Color Range

Figure 3.5: Combined figure

Here the black points are the noise which is formed when these points cannot form a cluster. The similarity of the clusters with the original pattern of Gujarat NDVI values can be easily observed by comparing fig. 3.5a and fig. 3.6.



Figure 3.6: NDVI Plot of Gujarat in Jan 08

Since they are quite similar, we can say that the clusters have been formed appropriately.

Now 200 random points have been randomly chosen to incorporate and compare incremental clustering with the original DBSCAN algorithm. These points have the new reading of NDVI values and they will replace the previous ones and thus cluster is to be formed from these points again. There are 98 clusters formed of 7028 points with new 200 readings which are placed in the specific range according to their average values. It is shown in fig. 3.7

```
(-1 - 0.040) 44,
(0.045 - 0.135) 65,66,67,71,74,76,78,80,81,83,85,86,87,89,90,
(0.140 - 0.230) 1,5,6,7,8,9,10,19,23,27,50,52,53,72,73,75,77,79,82,84,88,95,
(0.235 - 0.325) 0,2,3,4,17,20,41,
(0.330 - 0.420)
11,12,13,14,16,18,19,21,22,24,25,26,28,29,30,31,32,37,39,40,42,43,45,47,48,49,51,54,55,56,57,59,60,61,62,63,64,68,70,91,98,
(0.425 - 0.515) 33,34,35,36,38,46,58,69,92,93,94,96,97,
(0.520 - 0.610)
(0.615 - 0.705)
(0.710 - 0.805)
(0.810 - 0.900)
(0.905 - 1.000)
```

Figure 3.7: Cluster Associated with their Range Values

After this the 200 points are plotted on the map with its new plotted points as shown in fig. 3.8

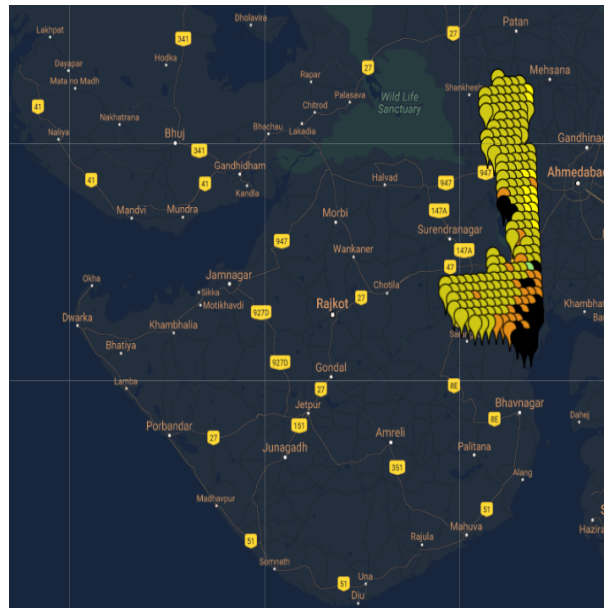


Figure 3.8: Cluster Created From the Original Algorithm

These are the points originated by the DBSCAN algorithm. All the points have been constructed here but for simplicity only the new 200 points have been shown.

The next step is to get the cluster of these points incrementally. Instead of loading the entire clusters again, only the core points of each cluster are loaded and then each point in the 200 points are compared with it and assigned a cluster. The Cluster formation by incremental clustering is shown in fig. 3.9

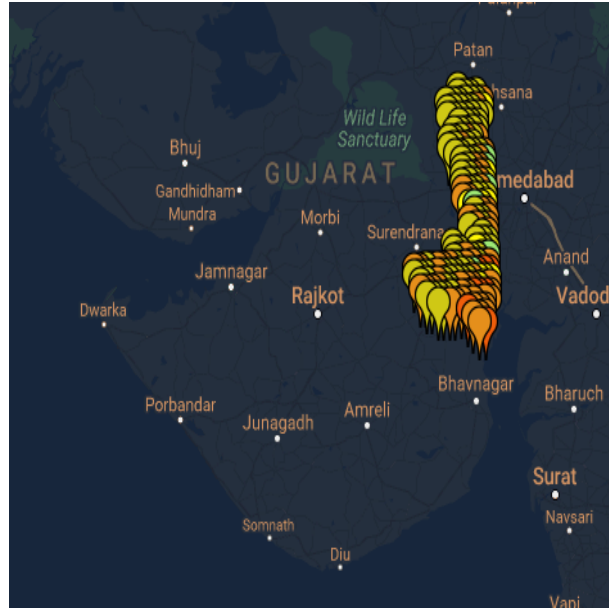


Figure 3.9: Cluster Created from Incremental Clustering

The two cluster formations can be compared and observations can be drawn about the accuracy and the efficiency of incremental clustering to the original method of adding new points by complete DBSCAN method.

Chapter 4

Result And Validation

In this section the outcome and results of the implementation has been discussed and validated.

Each point out of the 200 points can be compared with each other. Noise Points comes out to be 19 and the points matching are 159 and not matching are 22. Thus by not considering the noise points we get an accuracy of 87.84%.

87.84% Accuracy	
Full Algo(200 rows)	Incremental method (200 rows)
1.200444	0.000480
1.220973	0.000438
1.204831	0.000441
1.138008	0.000439
1.223003	0.000439
1.140619	0.000438
1.144150	0.000437
1.145525	0.000479
1.201012	0.000442
1.148015	0.000440

Figure 4.1: Result

The execution time of 10 trials of the original algorithm and the incremental method has been shown in fig. 4.1. It can be clearly seen that incremental approach is a more

efficient approach to find clusters of new points. The efficiency of the algorithm improves far better than the small amount of accuracy which is affected. While changing the NDVI values or using new spatial values for clustering, incremental clustering is the approach which should be used. There might be small errors in clusters but there will be a huge difference in the execution time.

Chapter 5

Conclusion

Accurate Clusters were formed by DBSCAN method. Incremental Clustering was successfully implemented and compared with the original method. Accuracy was coming out to be 87.84% for 200 points. Thus incremental clustering is the right way forward to predict new instances as they arrive on the basis of certain previous values(core points) instead of all of them. It has another advantage over the original method that the concerned dataset can be stored in the main memory which might not be the case in the use of complete algorithm.

Chapter 6

Future Enhancements

There are many fields where work can be done in the near future such as:

- A combination of spatio-temporal dataset can be used to understand their relation. For example NDVI and Rainfall.
- Different methods for incremental clustering can be used.
- Storage of these type of datasets can be worked upon.
- Different applications of incremental clustering can be found where this type of data analysis method might be helpful.
- Different heuristics can be found for the successful generation of clusters.
- Clustering algorithms can be used on a number of problems arising in day to day life.
- Different clustering algorithms can be used to cluster the data like OPTICS.
- Instead of creating new temporal data, new spatial data can be incorporated for incremental clustering.

Bibliography

- [1] M. LLC, “Types of cluster.” [Link](#).
- [2] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” 1996. [Paper](#).
- [3] D. Birant and A. Kut, “St-dbscan: An algorithm for clustering spatialtemporal data,” *Expert Systems with Applications*, vol. 60, no. 1, pp. 208–221, 2006. [Paper](#).
- [4] C. C. Aggarwal, A. Hinneburg, and D. A. Keim, “On the surprising behavior of distance metrics in high dimensional space,” *Database Theory*, 1973. [Paper](#).
- [5] M. Ackerman and S. Dasgupta, “Incremental clustering: The case for extra clusters,” *Expert Systems with Applications*, 2006. [Paper](#).
- [6] T. N. Carlson and D. A. Riziley, “On the relation between ndvi, fractional vegetation cover, and leaf area index,” *Remote Sensing of Environment*, 1997. [Paper](#).
- [7] “Bhuvan map.” [NDVI Bhuvan](#).
- [8] “R programming.” [R Homepage](#).