Optimum Vdbscan(O-VDBSCAN) For Identifying Downtown Areas

Wei Wang  
International School, Beijing University of Posts and Telecommunications  
Beijing University of Posts and Telecommunications, BUPT  
Beijing, China  
Email: WW100ANG@hotmail.com

*Shuang Zhou, Qingqing Xiao  
School of Software Engineering, Beijing University of Posts and Telecommunications  
Beijing University of Posts and Telecommunications, BUPT  
Beijing, China  
Email: zhoushuang924@hotmail.com

ABSTRACT

Clustering is an important part of data mining techniques, and VDBSCAN is a well-known density-based one. VDBSCAN is robust against noise and can recognize arbitrary shapes of clusters. Besides, it works effectively when dealing with datasets with varying densities. A main drawback of VDBSCAN is that it still requires a user-specified parameter K. An inappropriate choice of K can seriously degrade the accuracy of results. So in this paper we propose a totally parameter-free algorithm, OVDBSCAN, to find the global optimum K automatically, using the concept of derivative. The basic idea of OVDBSCAN is regarding \( \Delta d_k \) as the derivative of k-dist, which means the distance between an object and the kth nearest object of it. Then it chooses the largest K on condition that \( \Delta d_k \) doesn't exceed the threshold we set. In OVDBSCAN, the determination of K is based on the distances among objects within a dataset, thus the generated K reflects the property of this dataset. We’ve applied OVDBSCAN to a two-dimensional sample dataset, and the result shows that it can identify dense areas of varying densities.

KEYWORDS

Data mining; Clustering algorithm; VDBSCAN; Downtown area; Derivative

1 INTRODUCTION

Data mining is becoming a very active field nowadays, due to the growing volume of data stored in various databases. Clustering is a major part of data mining. It can divide data objects into groups in a way that maximizes intragroup similarity and minimizes intergroup similarity. There are four basic clustering techniques, partitioning methods, hierarchical methods, density-based methods and grid-based methods. Clustering can be used to address a variety of real world problems, and different methods are appropriate to different ones. What we are addressing in this paper is problem of downtown area identification. The distribution of downtown areas can be of great business value, as it can be the basis of lots of business analysis. In electronic maps, any place worth notice is identified as a POI (point of interest). It can be a restaurant, a store, a railway station, a building in a university and so on. The areas in which POIs are very dense are more likely to be prosperous, thus they can be identified as downtown areas. So what we need to do is to recognize the dense areas, namely, clusters.

The rest of the paper is organized as follows. In section 2 we’ll present some previous work that helps understand the work proposed in this paper. The proposed work is discussed in Section 3. A simulated application of our work on a sample dataset is provided in Section 4. Finally a conclusion is given in Section 5.

2 PREVIOUS WORK

2.1 DBSCAN Algorithm
VDBSCAN was proposed on the basis of DBSCAN, so before we introduce VDBSCAN, we have to look into DBSCAN. DBSCAN is a density-based clustering algorithm which finds core objects and connects them to their neighborhoods iteratively to form dense regions, that is, clusters. The two user-specified parameters, Eps (short for Epsilon), which means the radius of a neighborhood we consider for every object, and MinPts, which means the density threshold, are crucial in determining the result’s accuracy.

DBSCAN works as follows.

- Finds the objects that have at least MinPts objects within their \( \varepsilon \)-neighborhood and marks them as core objects[1,2].
- The objects within at least one core object’s \( \varepsilon \)-neighborhood are regarded as border objects.
- Objects that are not within any core objects’ \( \varepsilon \)-neighborhood are marked as outliers and they are disposed of once recognized.
- Any core objects that share one or more objects within their \( \varepsilon \)-neighborhoods are called connected. Each group of connected core objects merge into a separate cluster.
- All border objects are assigned to the clusters that contain their associated core objects [2].

Although DBSCAN has been widely applied, there are two problems with it. The parameters affect the result significantly. If the threshold is too low, some unnatural clusters may be formed, if it's too high, some natural clusters would be left out[3]. More importantly, because of the fixed parameters MinPts and Eps, it can't deal with datasets with clusters that have varying densities.

### 2.2 VDBSCAN Algorithm

To address the second problem, D. ZHOU and P. LIU proposed VDBSCAN (Varied DBSCAN) in 2009[4]. VDBSCAN provides a set of different Eps for a user-specified K (or MinPts) that can recognize clusters with varying density. Before we look into VDBSCAN, we have to explain the concept of k-dist: The k-dist of an object means the distance between this object and the kth nearest object of it. Now we briefly describe the way VDBSCAN works[4,5].

- Calculates and stores the k-dist of all objects.
- Determine the number of densities according to the k-dist plot.
- Choose Eps\(_i\) intuitively for each density.
- Apply DBSCAN with each Eps\(_i\) to get clusters of varying densities.

VDBSCAN can automatically choose Eps\(_i\) based on the dataset’s inherent property, hence it efficiently reduces the inaccuracy caused by inappropriate Eps selection.

### 3 OUR PROPOSED METHOD

#### 3.1 Motivation

VDBSCAN is a great choice when dealing with downtown area identification problems. In reality, downtown areas’ scales vary greatly, and typically it makes clustering much more difficult. But luckily, VDBSCAN can recognize clusters with different densities. Besides, it is robust against noise, so that areas with scarce POIs are neglected. However, it’s still not a perfect choice. Although VDBSCAN has successfully proposed a method for the Eps selection, it still requires users to specify K (or MinPts). And different selection of K almost certainly leads to different clustering result. So in OVDBSCAN, we propose a solution to the problem of K selection.

Firstly, take a look at the sample dataset shown in Fig.1. Let’s say object p is in the small cluster surrounded by the red circle. Now we consider p’s k-dist. When K=3, the K-dist is \( d_1 \), when K increases to 4, the K-dist becomes \( d_2 \). The difference \( \Delta d_1 = d_2 - d_1 \). But when K increases to 5, the K-dist of p suddenly rises to \( d_3 \), the difference becomes \( \Delta d_2 = d_3 - d_2 \), which is much larger than \( \Delta d_1 \). So for object p, 3 and 4 would be valid K selections, because they basically...
represent the density of that cluster. However, 5 is not a good K selection, because the corresponding K-dist cannot correctly indicate the density of that cluster. So we can conclude that sudden rises of $\Delta d_k$ are likely to suggest the edge of clusters. Note that here K=4 is better than K=3, because a larger K within the valid domain leads to a better clustering result[4].

![Fig.1 Sample of relation between K and K-dist when p is in a cluster](image)

Now let’s see what will happen if the object p is an outlier. In Fig.2 we show the object p and the nearest cluster of it. We can see that although $d_1$, $d_2$, $d_3$ and $d_4$ are all pretty large, yet they’re very close. This means that $\Delta d_k$ should be very smooth without any sudden rises. So, when we take all the objects into consideration, outliers won’t affect the general appearance of sudden rises.

![Fig.2 Sample of relation between K and K-dist when p is an outlier](image)

**3.2 Steps of OVDBSCAN**

First of all, we define an initial domain for possible Ks. The initial domain specifies the scope we consider. Also we assume that K only increases in the granularity of 1. Then we can use variable $\Delta d_k$ to represent the change rate of k-dist. We define that

$$\Delta d_{j,k} = d_{j,k+1} - d_{j,k}$$  \hspace{1cm} (1)

where $d_k$ represents the k-dist of object j.

We can summarize that when $\Delta d_{j,k}$ suddenly increases sharply as K increases by degrees, chances are that the Kth nearest neighborhood of object j is now on the edge of the cluster where object j is in, just as the sample in Fig.1 suggests. So this K-dist can indicate the density of the cluster most precisely. In other word, $\Delta d_k$ can indicate the size of natural clusters. OVDBSCAN makes perfect use of this property of $\Delta d_k$. It has four steps.

- Calculates and stores k-dists of all objects for every k in a given domain (initial domain).
- Calculates the overall change rate of k-dist (as outliers won’t affect the appearance, there’re no need to remove them because that would lower the efficiency):

$$\Delta d_k = \sum_{j=1}^{n} (d_{j,k+1} - d_{j,k})$$  \hspace{1cm} (2)

in which n stands for the number of objects in this dataset.
- Finds the first $\Delta d_k$ that exceeds the defined threshold. Here the threshold is usually determined by the calculated values of all $\Delta d_k$.
- Apply VDBSCAN with the k selected in (3) on the dataset.

**4 APPLICATION OF OVDBSCAN**

In order to corroborate that OVDBSCAN works, we need to simulate the whole process. We choose an area in which there are 20 POIs as our sample dataset, and each POI will be treated as an object. This experiment is implemented with JAVA language.

The dataset objects are shown in Fig.3 (The coordinates have been normalized). According to the appearance of objects, we can easily tell that a good algorithm should recognize the upper-left cluster and the upper-right cluster, and recognize the three objects in the right bottom as noise.
4.1 Obtaining The Value Of K

For this dataset, we define the initial domain of K as [1, 10], that is, K=1, 2 ..., 10. Next, we calculate and store the k-dist of all objects each time k increases from 1 to 10, then we calculate Δd_k for each increase. Thus we get Fig.4.

We can see that the Δd_k varies from 0.2396 to 2.9560, so now we need to define the threshold of Δd_k. We recommend a “percent below” method for setting the threshold. That is, set a threshold so that a certain percent of Δd_k are smaller than it. For this dataset, we define a “80 percent below” threshold. There are 10 values of Δd_k, so we just have to choose the 8th biggest one. (Shown in Fig.4) Apparently, when k=5, the Δd_k exceeds the threshold for the first time. Thus we choose 5 as our parameter.

4.2 Using K In VDBSCAN

Now we have obtained the global optimum k, all we have to do next is simply applying VDBSCAN on the dataset with the K selected to generate Eps(s). The k-dist plot of this dataset when k=5 is given in Fig.5. The two sharp rises, 12 to 14, and 17 to 18, indicate[3] that two densities can be intuitively recognized (the one in the end is caused by an outlier). So 0.1178 and 0.2995 are selected as Eps[4].

4.3 Applying DBSCAN With Generated K And Eps(S)

The two parameters for DBSCAN are both determined, so now we apply DBSCAN on this dataset twice with these two pairs of parameters. When k=5, Eps=0.1178, only one cluster is generated, as shown in Fig.6.

And when k=5, Eps=0.2995, two clusters are generated, as shown in Fig.7.
So, finally, two clusters are recognized, and three outliers are identified. Now the whole process of OVDBSCAN is completed.

4.4 Matched Group With Other K Selections

To prove that K=5 is the optimum selection, we’ve also carried out several matched group with other K values for comparison. Established that when K=5, the clustering result is natural and reflects the dataset’s inherent property, we use it as the criterion to evaluate other groups’ accuracy. And the accuracy is calculated with the following rules:

1. If an object within a cluster is recognized as an outlier, count it as inaccurate, and vice versa.
2. If the number of clusters exceeds two (according to the criterion), find two clusters that share the most objects with the two of our criterion, respectively, and treat other clusters as inaccurate (all objects in them).
3. Let’s say there are N inaccurate objects, so the accuracy of this experiment is (20-N)/20.

We consider K to vary from 1 to 10, as it’s the initial domain of our experiment, and apply each K to VDBSCAN on our dataset. The accuracy of all results are shown with a line chart, Fig.8.

As we can see, the accuracy keeps rising before K=5, and suddenly drops afterwards. In this case, the limited number of objects leads to similar results for K=4 and K=5. Nevertheless, the matched group has proved that the accuracy reaches the maximal value at, or at least around, 5. Hence, it in turn proves that OVDBSCAN can obtain the optimum K.

5 CONCLUSION

Identifying downtown areas is of high business value, for it can support a lot of business analysis. To name one, each downtown area can be tagged according to the types of POIs in it, then each area can have a characteristic on the whole. So if we want to analyze someone’s pattern based on the coordinates it has been on, we can just try to find which area the “points” are in, and then we can deduce what did it do and when did it like to do it. Density-based clustering methods are suitable for this kind of problem. Compared to traditional clustering algorithms, VDBSCAN is very effective dealing with datasets of varying density. But requiring K as an input from users deteriorates the accuracy of this algorithm. In OVDBSCAN, the selection of K is determined by examining the k-dists of all objects in datasets as K changes. In this way, the selected K can well reflect the inherent property of the datasets. So, OVDBSCAN is a totally human interference-free algorithm, thus it objectively reveals the characteristics of data. However, perfection of this method calls for future research. A systematic determination of when K is too large in OVDBSCAN may be an interesting topic. Minimizing the initial domain of K to raise
the efficiency of OVDBSCAN is also a big challenge.

6 REFERENCES


