Mining Uncertain Data in Low-dimensional Subspace

Zhiwen Yu  
Department of Computer Science  
City University of Hong Kong

Hau-San Wong  
Department of Computer Science  
City University of Hong Kong

Abstract

Mining for clusters in a database with uncertain data is a hot topic in many application areas, such as sensor database, location database, face recognition system and so on. Since it is commonly assumed that most of the objects which are contained in a high-dimensional dataset are located in a low-dimensional subspace, mining clusters in a subspace in an uncertain database is a new task. In this paper, we adopt and combine fractal correlation dimension with fuzzy distance function to find out the clusters in a low-dimensional subspace in an uncertain database. We also propose the fuzzy kth NN algorithm to retrieve the kth nearest neighbor which can accelerate the process of mining. The experiments show that the new algorithm works well in an uncertain database.

1. Introduction

Recently, researchers are paying more attention to mining clusters in uncertain data since more and more applications, such as sensor database([1],[3]), location database([2],[4]) and face recognition system, produce vague and imprecise data. For example, a moving object continuously changes its locations, and the current coordinates will quickly become invalid. As a result, the moving object can only inform the server about its approximate location in the form of an uncertain region in which the object will appear.

The fuzzy objects in an uncertain database are expressed as a set of distribution functions. However, traditional distance metric cannot be directly applied to measure the similarity of the fuzzy objects. In addition, traditional similarity measure([5]) between the fuzzy objects is expressed by an aggregated distance value which leads to the loss of a lot of information. In order to exploit the information more fully, fuzzy distance function([6]) is proposed to measure the similarity between two fuzzy objects. According to the similarity information between the fuzzy objects, standard data mining algorithm, such as density-based clustering algorithm DBSCAN([7]), can assign the fuzzy objects to the different clusters. But these approaches focus on mining the uncertain data in the whole space, and none of them have addressed the problem of clustering the uncertain data in a low-dimensional subspace.

Mining uncertain data for patterns in a low-dimensional manifold and finding the useful patterns and regularities are the new challenges in an uncertain database. In this paper, we propose a new approach to address this problem. The new approach combine fuzzy distance function with fractal correlation dimension to find out the clusters in the subspace in an uncertain database. The remainder of the paper is organized as follows: Section 2 describes fuzzy distance function. Section 3 presents correlation dimension. Section 4 introduces fuzzy dimensional induced clustering algorithm. Section 5 evaluates the new algorithm experimentally. Section 6 is the conclusion and future work.

2. Fuzzy distance function

The fuzzy object \(o\) in an uncertain database \(D\) is represented by a probability distribution function, such as Gaussian distribution function or Poisson distribution function. Let \(o \in D \subseteq \mathbb{R}^m\) be an uncertain data item from \(D\) which can appear at any position \(x\) with respect to a probability density function \(p(x)\). Then, the following condition holds:

\[
\int_{\mathbb{R}^m} p(x) dx = 1
\]

where \(\mathbb{R}^m\) is the data space.

The similarity between traditional objects is measured by the distance between two objects which is represented as a numerical value. Unfortunately, the similarity of two fuzzy objects cannot be measured by a single value. As a result, Kriegel H.P.et.al ([6]) proposed a distance density function. Let \(\theta : D \times D \rightarrow R\) be a metric function (Here we select Euclidean distance as the distance metric \(\theta\)). \(P(\theta(o, o') \leq \epsilon)\) denotes the probability that \(\theta(o, o')\) is not greater than \(\epsilon\) (where \(o\) and \(o'\) denotes the object \(o\) and \(o'\), \(\epsilon\) is a distance threshold). The distance distribution function is defined as

\[
P(\theta(o, o') \leq \epsilon) = \int_{-\infty}^{\epsilon} p_{\theta(o, o')}(x) dx
\]

where \(p_{\theta(o, o')}(x)\) is a probability density function. If the distance \(\theta(o, o')\) between two objects is deterministic, the probability density function is equal to the Dirac-delta function \(\delta\).

\[
p_{\theta(o, o')}(x) = \delta(x - \theta(o, o'))
\]

The similarity between the fuzzy objects can be measured by the distance expectation value \(E_{\theta}\) which is the average distance between the fuzzy objects.

\[
E_{\theta}(o, o') = \int_{-\infty}^{\epsilon} x \cdot p_{\theta(o, o')}(x) dx
\]
3. Correlation dimension

Correlation dimension is also called fractal dimension[8]). We first assume that (i) the dataset P which consists of \{p_1, p_2, ..., p_n\} is a subset of \( \mathbb{R}^d \). (ii) \( \theta(p_i, p_j) : P \times P \rightarrow \mathbb{R} \) is a distance metric between the pair of points \((p_i, p_j)\). (iii) \( F(\epsilon) \) is the average fraction of pairs of points within the distance threshold \( \epsilon \).

\[
F(\epsilon) = \lim_{n \to \infty} \frac{1}{n^2} \sum_{p_i, p_j \in P} |f(p_i, p_j, \epsilon)|
\]

(5)

\[
f(p_i, p_j, \epsilon) = \{p_j | p_j \in P \land \theta(p_i, p_j) \leq \epsilon\}
\]

(6)

where \( f(p_i, p_j, \epsilon) \) retrieves a subset of points which are contained in the sphere whose center is the point \( p_i \) and whose radius is the distance threshold \( \epsilon \). \(|\cdot|\) denotes the cardinality of the point set.

Then, the correlation dimension \( d \) is defined as

\[
d = \lim_{\epsilon \to 0} \frac{\log[F(\epsilon)/F(\epsilon')]\log[\epsilon/\epsilon']}{\log[\epsilon/\epsilon']}
\]

(7)

But in practice, the dataset P contains a finite number of data points which conflicts with the fraction function \( F(\epsilon) \). As a result, Aristides, et al[9]) proposed local correlation dimension for each point \( p_i \). The local-growth curve is first defined as the function \( F_{p_i}(\epsilon) \).

\[
F_{p_i}(\epsilon) = \frac{1}{n} \sum_{p_j \in P} |f(p_i, p_j, \epsilon)|
\]

(8)

The local-correlation dimension \( d_i \) of the point \( p_i \) is defined as

\[
d_i = \frac{\log[F_{p_i}(\epsilon)] - \log[F_{p_i}(\epsilon')]}{\log[\epsilon/\epsilon']}
\]

(9)

Finally, they (i) compute the local growth curve \( F_{p_i}(\epsilon) \) for every point \( p_i \), and (ii) take the \( F_{p_i}(\epsilon) \) curve in log-log scale and find the linear growth model \( L_i \) which provides a best fit to the data.

\[
L_i(\log \epsilon) = d_i \log \epsilon + b_i
\]

(10)

The pair \((d_i, L_i)\) is used as the local representation of the point \( p_i \). According to the representation \((d_i, L_i)\) of the point \( x_i \), the standard clustering algorithm is performed on the 2D representation space to detect the interesting clusters.

Figure 1 illustrates an example of a linear growth model. Figure 1(a) is a 2D dataset, while Figure 1(b) shows the log-log neighbor count for one data point and its corresponding linear growth model.

4. Fuzzy dimensional induced clustering algorithm

By combining the fuzzy distance function and the local correlation dimension, we first define the local-growth curve for a fuzzy object \( o_i \) in the uncertain database \( D \) as a fuzzy function \( U_{o_i}(\epsilon) \).

\[
U_{o_i}(\epsilon) = \frac{1}{n} \sum_{o_j \in D} |u(o_i, o_j, \epsilon)|
\]

(11)

Figure 2 provides an overview of the fuzzy dimension induced clustering(FDIC) algorithm. The algorithm considers the fuzzy objects one by one. It first calculates the distance from the fuzzy object \( o_i \) to its \( k \)th nearest neighbor based on \( E_0 \). If \( \epsilon \) is set to the distance from \( o_i \) to its \( k \)th nearest neighbor, \( U_{o_i}(\epsilon) = k \cdot \log[U_o] \) and \( \log(\epsilon) \) then corresponds to the points in Figure 1(b). The algorithm varies the \( k \) value from \( k_{min} \) to \( k_{max} \). All the points in Figure 1(b) are obtained in this way. Then, the algorithm computes the local representation \((\phi_i, \psi_i)\) of \( o_i \). After obtaining the 2D data
set which consists of the local representation of the fuzzy objects, the algorithm performs EM to cluster the fuzzy objects in the 2D data set.

Unfortunately, the $k$th nearest neighbor of $o_1$ is difficult to compute since calculating $E_{o_1}$ is time consuming. In order to make FDIC algorithm more efficient, we substitute $s$ samples for the fuzzy object $o_1$. The samples are selected according to the probability density function $p(x)$ of $o_1$. A minimum bounding box $o_1.MBR$ is applied to contain the $s$ samples. Then, the distance between the fuzzy object $o_1$ and $o_2$ is redefined as the average distance $d(o_1, o_2)$ between the samples of the fuzzy object $o_1$ and $o_2$.

$$d(o_1, o_2) = \frac{1}{s} \sum_{h=1}^{s} \sum_{j=1}^{m}(o_1.s_h - o_2.s_j)^2 \quad (16)$$

According to the property of the minimum distance and maximum distance between the MBRs of the fuzzy objects, the following two lemmas are obtained.

**Lemma 1.** If $d_{\min}(o_1.MBR, o_j.MBR) > d_{k-\max}, o_j$ cannot be the $k$th nearest neighbor of $o_1$.

**Lemma 2.** If $d_{\max}(o_1.MBR, o_j.MBR) < d_{k-\min}, o_j$ cannot be the $k$th nearest neighbor of $o_1$.

where $d_{\min}(o_1.MBR, o_j.MBR)$ and $d_{\max}(o_1.MBR, o_j.MBR)$ denote the minimum distance and the maximum distance between the MBRs of the fuzzy objects $o_1$ and $o_j$ respectively. $d_{k-\max}$ is the $k$th maximum distance among all the maximum distances between the MBR of $o_1$ and other fuzzy objects. $d_{k-\min}$ is the $k$th minimum distance among all the minimum distances between the MBR of $o_1$ and other fuzzy objects.

**Proof:** We first prove Lemma 1. We assume (i) the fuzzy object $o'$ is the $k$th nearest neighbor of $o_1$ and (ii) $d_{\min}(o_1.MBR, o'.MBR) > d_{k-\max}$. Based on the property of the average distance $d(o_1, o_2), d_{\min}(o_1.MBR, o_2.MBR) < d(o_1, o_2) < d_{\max}(o_1.MBR, o_2.MBR)$, there exist at least $k$ minimum distances which are not greater than $d_{k-\max}$. We can deduce that there exist at least $k$ average distances which are not greater than $d_{k-\max}$. But $d(o_1, o') > d_{\min}(o_1.MBR, o'.MBR) > d_{k-\max}$. It means that the fuzzy object $o'$ is not the $k$th nearest neighbor of $o_1$, which conflicts with the assumption. So, Lemma 1 is correct. Lemma 2 can be proved in a similar way.

Figure 3 demonstrates an example of finding the 2nd nearest neighbor of the fuzzy object $o$ by lemma 1 and lemma 2. The fuzzy object $o_2$ in Figure 3(a) cannot be the 2nd nearest neighbor of $o$ according to lemma 1 since $d_{\min}(o.MBR, o_2.MBR) > d_{2-\max} = d_{\max}(o.MBR, o_2.MBR)$, while the fuzzy object $o_3$ in Figure 3(b) cannot be the 2nd nearest neighbor of $o$ according to lemma 2 because $d_{\min}(o.MBR, o_3.MBR) < d_{2-\min} = d_{\min}(o.MBR, o_2.MBR)$.

Figure 4 provides an overview of Fuzzy $k$th nearest neighbor algorithm(FKNN). The samples and MBR of the fuzzy objects are indexed by R*-tree([12]) or X-tree([13]). The algorithm first prunes the false candidates by lemma 1 and lemma 2. Then, it calculates the average distance between $o_1$ and the remaining. **Note that**, the candidates which are pruned by lemma 2 are within the range of $k$ nearest neighbors of $o_1$. So, the algorithm only need to find out the $(k-c)$ th average distance between $o_1$ and the remaining. Finally, the algorithm obtains the $k$th nearest neighbor.

**Algorithm FKNN(Uncertain dataset D, the fuzzy object o_i)**

1. Prune the false candidates by lemma 1;
2. Prune the false candidates by lemma 2;
3. Compute the average distance according to the samples;
4. Find the $(k-c)$ th average distance between $o_1$ and the remaining;
5. Return the $k$th nearest neighbor of $o_i$;

**Figure 4. Fuzzy $k$th NN algorithm**

5. **Experiment**

5.1. **Experimental Setting and Data Set**

All the experiments presented are executed with a Pentium 3.2 GHz CPU with 1 GByte memory. We perform fuzzy dimensional induced clustering algorithm(FDIC) on a synthetic dataset and a sensor database. All the samples and the MBR of the objects are indexed by R*-tree. The time complexity of FDIC is $O(k_{\max} \cdot n \cdot \log n)$ where $k_{\max}$ is the maximum value of $k$, $n$ is the number of the objects.

The process of generating the synthetic dataset consists of two stages. In the first stage, we generate a deterministic dataset according to the setting in [6]. First, the dataset D which consists of $n$ objects in $\mathbb{R}^m$ are decomposed into two subsets $A, B$ with size $a, b$ respectively. The objects in A are uniformly distributed in $[0,1]^m$. The values of the objects of B in the first $l$ dimensions are normally distributed around 0.5 with variance 0.02, while the coordinates of the objects in the last $(m-l)$ dimensions are uniformly distributed in $[0,1]$. As a result, the correlation dimension of A, B are close to $m$ and $l$ respectively. B is called an ID-flat. In general, the dataset D can contain more than one ID-flat. It means that the dataset D can be decomposed into subsets $\{A, B_1, B_2, ..., B_k\}$ with correlation dimensions $\{m, l_1, l_2, ..., l_k\}$ respectively where $m > l_1 > l_2 > ... > l_k$.

In the second stage, each object is randomly surrounded by a box with side length of $\delta < 1$ in each dimension $[o.x_i - \frac{\delta}{2}, o.x_i + \frac{\delta}{2}]$, where $o.x_i$ denotes the value of the $i$th dimension of the object $o$ generated in the first stage. We assume that the pdf of the object is uniform. It means that the object will appear in any position of the box with the same probability.

0-7695-2521-0/06/$20.00$ (c) 2006 IEEE
We generate 2 synthetic datasets. The first one which consists of 2000 objects in $\mathbb{R}^8$ are decomposed into two clusters $A$ (1000 objects) and $B$ (1000 objects) with correlation dimensions 4 and 3 respectively. The second one which consists of 2000 objects in $\mathbb{R}^6$ are decomposed into three clusters $A$ (1000 objects), $B_1$ (500 objects) and $B_2$ (500 objects) with correlation dimensions 8, 4 and 2 respectively.

In the second experiment, the data set consists of 3000 objects which originate from a sensor database. Each object is represented by a 10-dimensional feature vector. All the values of the objects are normalized within $[0, 1]$.

Total classification error $E$ in [6] is used as the error measure:

$$ E = 1 - \frac{1}{n} \sum_{ij} M_{ij} $$

(17)

where $M$ is the confusion matrix. $M_{ij}$ is an entry of $M$ which represents the number of overlapping objects between the i-th cluster of the ground truth and j-th cluster of the clustering found by the algorithm.

In the experiments, we use $k_{min} = 10$, $k_{max} = 50$, and the side length of the box $\delta = 0.01$, the number of the sample $s = 8$ as the default setting. We vary a single parameter each time, while setting the remaining ones to their default values.

5.2. Experiment results

Figure 5 and Figure 6 illustrates the effectiveness of FDIC algorithm. Figure 5(a) discovers 3D-flat in 4D space, while Figure 5(b) finds out the clusters in 2D and 4D subspace which are within 8D full space. Figure 6 mines the uncertain data in the sensor database. There are three clusters discovered in the database.

Figure 1 shows the classification error of discovering $l$ - flat clusters by FDIC.

<table>
<thead>
<tr>
<th>subspace</th>
<th>3D in</th>
<th>2D in</th>
<th>4D in</th>
<th>3D in</th>
<th>8D in</th>
<th>10D in</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E$</td>
<td>0.055</td>
<td>0.013</td>
<td>0.042</td>
<td>0.018</td>
<td>0.018</td>
<td>0.061</td>
</tr>
</tbody>
</table>

Table 1. Total classification error

Figure 2 compares the running time of FDIC combined with FkNN and FDIC without FkNN in the first synthetic dataset. By combining with FkNN, FDIC is faster than original FDIC.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>FDIC without FkNN</th>
<th>FDIC with FkNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Running time (s)</td>
<td>10.6</td>
<td>7.4</td>
</tr>
</tbody>
</table>

Table 2. Running time

6. Conclusion and future work

The paper investigates the problem of mining uncertain data in a low-dimensional subspace which is addressed for the first time. Our major contribution is a fuzzy dimensional induced clustering(FDIC) algorithm which combines fuzzy distance function and fractal correlation dimension to find out the interesting clusters in the low dimensional manifolds in an uncertain database. We also propose fuzzy kth nearest neighbor algorithm(FkNN) to retrieve the kth nearest neighbor efficiently. The experiments illustrate that FDIC and FkNN works well in an uncertain database.

Acknowledgments

The work described in this paper was partially supported by a grant from the Research Grants Council of Hong Kong Special Administrative Region, China. [Project No. CityU 1197/03E] and a grant from City University of Hong Kong [Project No. 936091].

References