Cross-Cloud Plots: Scalable Tools for Spatial and Multidimensional Data Mining

Agma Traina Caetano Traina Christos Faloutsos Spiros Papadimitriou

September 12, 2001

Abstract

We focus on the problem of finding patterns across two large, multidimensional datasets. For example, given feature vectors of healthy and of non-healthy patients, we want to answer the following questions: "Are the two clouds of points separable?", "What is the smallest/largest pair-wise distance across the two datasets?", "Which of the two clouds does a new point (feature vector) come from?".

We propose a new tool, the 'Cross-Cloud plot', which helps us answer the above questions, and many more. We present an algorithm to compute the Cross-Cloud plot, which requires only a single pass over the datasets, thus scaling up to arbitrarily large databases. More importantly, it scales linearly with the dimensionality, while most other spatial data mining algorithms explode exponentially. We show how to use our tool for classification, when traditional methods (nearest neighbor, classification trees) may fail. We also provide a set of rules on how to interpret a Cross-cloud plot, and we apply these rules on multiple, synthetic and real datasets.

1 Introdunction and motivation

The ability to automatically discover meaningful patterns and relationships that may lay hidden in vast repositories of raw information has become an issue of great importance. Multimedia systems that deal with satellite imaging, medical data, banking information are some examples of prolific sources of data. Many of these data are inherently multi-dimensional.

"Summarizing" a large number of attributes by extracting a few essential features is often difficult to do. Moreover, many proposed methods in the literature suffer from the so-called "dimensionality curse" and become impractical to apply directly. Thus, dealing efficiently with multi- and high-dimensional data has been a challenge for researchers in the database field [WSB98, BBK98]. This scenario worsens when more than one dataset is involved. This paper proposes a new method for exploring the relationship between two multi-dimensional datasets. Our method requires only a single pass on the data and scales linearly with the number of dimensions. In particular, we propose a method to summarize the information about the relative position of two datasets. Specifically, we focus on the following.

Problem definition: Given two large multi-dimensional datasets, find rules about their relative placement in space. For example, we want to answer questions such as:

- Q1: Do the datasets come from the same distribution?
- Q2: Do they "repel" each other?
- Q3: Are they close or far away?

- Q4: Are they separable?
- Q5: For a given, unlabelled point, which of the two sets does it come from (if any)?

The ideal method for dealing with the above problem should meet certain specifications:

- it should be linear on the number of dimensions
- it should perform a single pass on the whole data set

Our proposed method fulfills these goals.

The remainder of the paper is structured as follows. In the next section, we first give a brief discussion of the related work on data mining techniques and then a concise description of the datasets we used in our experiments and the motivation for this work. Section 3 introduces the Cross-Cloud plots as well as its 'anatomy'. Section 4 presents the rules which allow us to mine two clouds of points and the practical usage of the proposed method. Section 5 presents the algorithm which generated the graphs used in the analysis of the datasets and in speed considerations. Section 6 gives the conclusions of this paper.

2 Related work

There has been a tremendous amount of work on data mining during the past years. Many techniques have been developed that have allowed the discovery of various trends, relations and characteristics with large amounts of data [JAG99, Cha98]. Detailed surveys can be found in [CHY96] and [GGR99]. Also, [Fay98] contains an insightful discussion of the overall process of knowledge discovery in databases (KDD) as well as a comprehensive overview of methods, problems, and their inherent characteristics.

In the field of spatial data mining [EKS99] much recent work has focused on clustering and the discovery of local trends and characterizations. Scalable algorithms for extracting clusters from large collections of spatial data are presented in [NH94] and [KN96]. The authors also combine this with the extraction of characteristics based on non-spatial attributes by using both spatial dominant and non-spatial dominant approaches (depending on whether the cluster discovery is performed initially or on subsets derived using non- spatial attributes). A general framework for discovering trends and characterizations among neighborhoods of data-points is presented in [EFKS98]. This framework is built on top of a spatial DBMS and utilizes neighborhood-relationship graphs which are traversed to perform a number of operations. Additionally, scalable clustering algorithms are included [AGGR98, TZ96, SCZ98, FRB98].

Visualization techniques for large amounts of multidimensional data have also been developed. The work described in [KK94] presents a visualization method which utilizes views of the data around reference points and effectively reduces the amount of information to be displayed in a way that affects various characteristics of the data (eg. shape and location of clusters, etc.) in a controlled manner.

There has also been significant work on data mining in non-spatial, multidimensional databases. Recent work on a general framework that incorporates a number of algorithms is presented in [iHLN99]. The authors introduce a general query language and demonstrate its application on the discovery of a large variety of association rules which satisfy the anti-monotonicity property.

However, none of the above methods can answer all the questions Q1 to Q5, which we posed in the previous section. The method proposed in this paper can answer such questions as we will show in the next sections. To find a solution for the given problem we move away from association rules and focus on the relationship between two spatial datasets.

2.1 Description of the data sets

In order to test our method we used several synthetic and real datasets. We use synthetic datasets to help build intuition and real datasets to validate our techniques. The synthetic datasets are always normalized to a unit MBR (minimum bounding rectangle), and they may be translated, rotated and/or scaled in the experiments. For the synthetic datasets we used:

- 'Line' (points randomly chosen along a line segment)
- 'Circumference' (points randomly chosen along the periphery of a circle)
- 'Sierpinsky' (randomly generated points from the theoretical Sierpinsky triangle), Figure 1(a) shows an example of it.
- 'Square' (randomly generated points in a 2D manifold)
- 'Cube' (randomly generated points in a 3D manifold)
- 'Super-cluster' (256 uniformly distributed clusters, each of them with 7x7 points in a 2D manifold).

The number of points in the datasets are shown in the plots following their names. For example: 'Line10K' means a line with 10,000 points.

The real datasets used are the following:

- California - These are four two-dimensional sets of points that refer to geographical coordinates in California [oC89]. Each set corresponds to a feature: 'streets' (with 62,933 points), railways (with 31,059 points), political borders ('political' with 46,850 points), and natural 'water' systems (water with 72,066 points). Figure 1(b) shows the political and water datasets superimposed.
- Iris - This consists of three sets, each of which describes properties of the flower species of genus Iris. The points are 4-dimensional (sepal length, sepal width, petal length, petal width); the species are 'virginica', 'versicolor' and 'setosa', and there are 50 points from each species. This is a well-known dataset in the literature of machine learning and statistics, which we obtained from the UC-Irvine Repository. Figure 1(c) shows a 3D projection of this dataset.
- Galaxy - The Galaxy datasets come from the SLOAN telescope: (x,y) coordinates, plus the class label. There are 82,277 in the 'dev' class (deVaucouleurs), and 70,405 in the 'exp' class (exponential). Figure 1(d) shows both datasets superimposed.
- LC - Customer data from a large corporation, which requested confidentiality. There were 20,000 records, each with 19 numerical/boolean attributes. The records formed two classes, with populations of 1,716 and 18,284 respectively.
- Votes - These are two 16-dimensional datasets from the 1984 United States Congressional Voting Records Database: Democrat (267 entries) and Republican (168 entries).

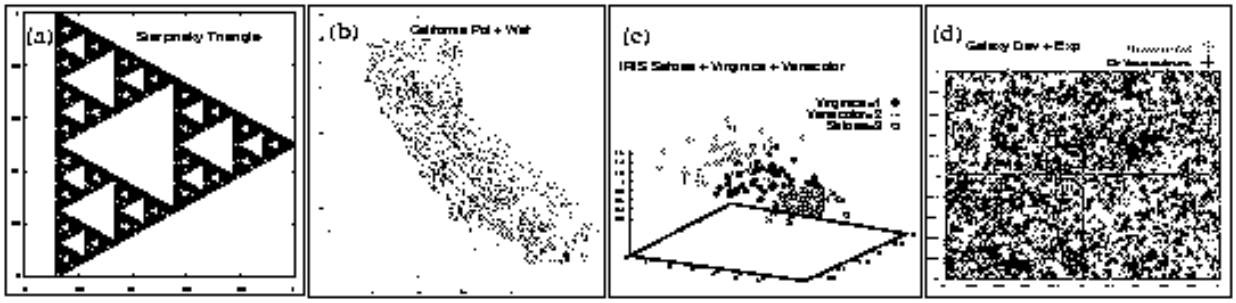

Figure 1: Some datasets used in the experiments: (a) Synthetic Sierpinski triangle, (b) California-politics and California-water superimposed, (c) the three Iris datasets, and (d) the two Galaxies datasets.

Symbol	Definition
N_A (or N_B)	Number of points in dataset A (or B)
$Cross()$	$Cross_{A,B}(r, 1, 1)$ plot between datasets A and B
$Self_A$	$Self_A(r, 1, 1)$ plot of dataset A
W_A	$Cross_{A,B}(r, 10, 1)$ cross-could plot weighted on dataset A
W_B	$Cross_{A,B}(r, 1, 10)$ cross-could plot weighted on dataset B
$C_{A,i}$ ($C_{B,i}$)	Count of type A (B) points in the i-th cell
n	Number of dimensions (embedding dimensionality)
$\hat{r}_{min}$	Estimated minimum distance between two points
$\hat{r}_{max}$	Estimated maximum distance between two points
cloud / point set	n-dimensional dataset

Table 1: Symbols and definitions

3 Proposed idea: Cross-cloud plots

The main intuition of our approach is to study how often members of the two sets A and B happen to be close to each other. This is captured by the Cross-function $Crossf_{A,B}(r, p, q)$ which we define below. This function aims to determine the spatial relationship between two clouds of n-dimensional points A and B. Table 1 presents the symbols used in this paper.

Consider a grid with cells of side r and let $C_{A,i}$ ($C_{B,i}$) be the number of points of type A (B) in the i-th cell, and exponents p and q . The grid partitions the minimum bounding box including both datasets. Then we have:

Definition 1 *Given two data sets A and B residing in the same n-dimensional space, we define the cross-function as*

$$Crossf_{A,B}(r, p, q) = \sum_i C_{A,i}^p \cdot C_{B,i}^q$$

Typically, we plot the cross-function in log-log scales, after some normalization, as we present in the next definition:

Definition 2 *Given two data sets A (with N_A points) and B (with N_B points) residing in the same n-dimensional space, we define the cross-cloud plot between the two datasets as the plot of*

$$Cross_{A,B}(r,p,q) = \frac{\log(N_A \cdot N_B)}{\log(N_A^p \cdot N_B^q)} \cdot \log\left(\sum_i C_{A,i}^p C_{B,i}^q\right) \text{ versus } \log(r)$$

Notice that the normalization factor scales the cross-cloud plot, maximizing the information presented. The proposed cross-function has several desirable properties:

Property 1 *For $p = q = 1$, the Cross-function is proportional to the count of A-B pairs within distance r. That is,*

$$Cross_{A,B}(r,1,1) \propto \textit{(number of pairs of points within distance} \leq r\textit{)}$$

Proof Using Schuster's lemma [Sch88].

This is a significant property. For $p = q = 1$ the cross-cloud plot gives the cumulative distribution function of the pair-wise distances between the two "clouds" A and B [FSJT00]. Because of its importance, we will use the case $p = q = 1$ as a reference. Thus, we will omit the p, q argument when both of them are equal to one. For the sake of simplicity we will omit the subscripts A, B from the cross-cloud plot when it is clear which datasets are involved. That is,

$$Cross(r) \triangleq Cross_{A,B}(r) \triangleq Cross_{A,B}(r,1,1)$$

Property 2 *The Cross-function includes the "correlation integral" as a special case when we apply it to the same dataset (i.e., $A \equiv B$).*

Proof From the definition of correlation integral [Sch91]. This is a very interesting property, because the correlation integral gives the correlation fractal dimension D_2 of a dataset A, if this dataset is self-similar.The above property is so important that we give a special name to the self cross-cloud plots:

Definition 3 *The self-plot of a given dataset A is the plot of*

$$Self_A(r) = \log\left(\frac{\sum_i C_{A,i} \cdot (C_{A,i} - 1)}{2}\right) \text{ versus } \log(r)$$

By definition, the self-plots exclude the counting of self pairs. This means that both $\langle p_1, p_2 \rangle$ and $\langle p_2, p_1 \rangle$ are counted only once. This is the reason we subtract '1' from the second count of points, so as to avoid artifacts that self-pairs generate.

Property 3 *If A is self similar, then (a) the Self-plot of A is linear and (b) its slope is its intrinsic dimensionality.*

Proof See [BF95]. We are now ready to define our two main proposed tools, the tri-plot and the pq-plot.

Definition 4 *The tri-plot of two datasets, A and B, is the graph which contains the cross-plot $Cross(r)$ and the normalized self-plots for both datasets ($Self_A(r) + \log(N_A/N_B)$ and $Self_B(r) + \log(N_B/N_A)$).*

Notice that the normalization factors, $\log(N_A/N_B)$ and $\log(N_B/N_A)$, only translate the self-plots, preserving the steepness of the graphs. In this paper, for every tri-plot we present the three graphs with the same color pattern: the cross-plot is presented in blue lines with diamonds, $Self_A$ in green lines with crosses and $Self_B$ in red lines with squares. We also report the slope or steepness of the fitting lines.

Definition 5 *The pq-plot of two datasets, A and B, is the graph which contains the three cross-cloud plots: $Cross_{A,B}(r)$, $Cross_{A,B}(r,1,k)$, and $Cross_{A,B}(r,k,1)$ for large values of k ($k \gg 1$).*

Figure 2 shows a tri-plot and a pq-plot of the Line and Sierpinsky datasets. Notice that, although the $Cross()$ is almost always linear (Figure 2(a)), this is not necessarily true for the $Cross(r,1,k)$ and $Cross(r,k,1)$ (in Figure 2(b), $k = 10$).

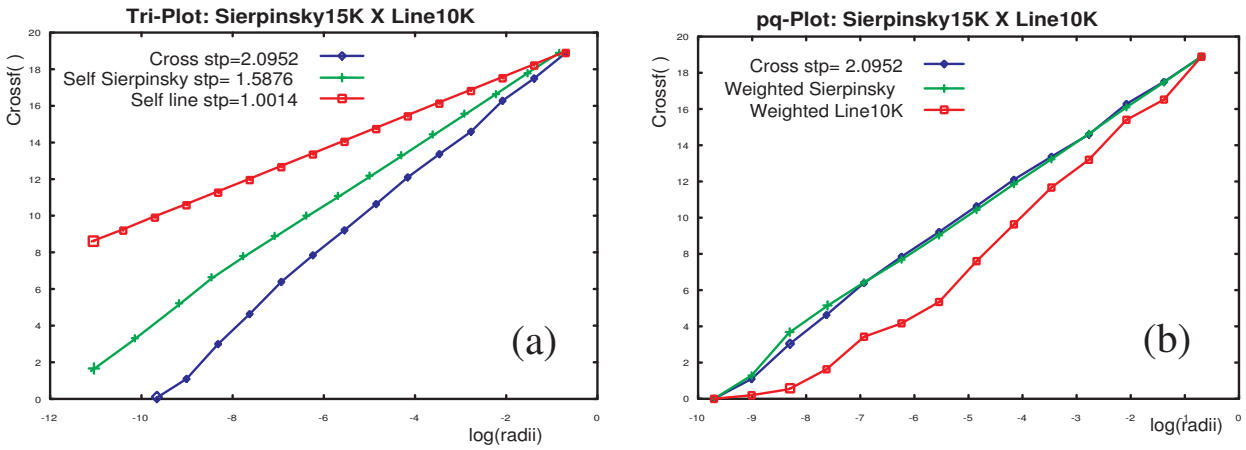

Figure 2: Sierpinsky triangle and Line datasets: (a) the tri-plot, (b) the pq-plot. The cross-plots are presented in blue with diamonds, the self- and weighted-Sierpinsky plots in green with crosses and the self- and weighted-Line in red with squares.

Definition 6 *The steepness of a plot is the slope of its regression line (that is, least squares fitting line).*

Definition 7 *The dissimilarity of two given plots is measured by the sum of the root mean square differences between them.*

The tri-plots will allow us to determine the relationship between the two datasets under analysis. If the datasets are self-similar, i.e, the self-plots of both datasets are linear for a meaningful range of radii, their slopes can be used in the comparisons that follow. However, the proposed analysis can be applied even to datasets which are not self-similar (do not have linear self-plots). Thus, we will refer to the "steepness" and "similarity" between the plots as measurements for comparison. The pq-plot is used in a further step of the analysis. Its use is more subtle and is discussed in section 4.3. Next we show some interesting properties of the datasets that can be derived from the tri- and pq-plots.

3.1 Anatomy of the proposed plots

This section shows how to "read" the cross-cloud plots and take advantage of the tri- and pq-plots without any extra calculation.

3.1.1 Properties of the self-plots

From a self-plot we derive valuable information:

Property 4 *The estimate value $\hat{r}_{min}$, which is the minimum distance between two points of a given dataset, ie, the first radius where the count-of-pairs is not zero is given by the self-plot.*

Property 5 *The estimate value $\hat{r}_{max}$, which is the maximum distance between two points of a given dataset (or the dataset diameter), i.e., the maximum value where the count-of-pairs increased is given by the self-plot. For bigger radii the counting remains the same.*

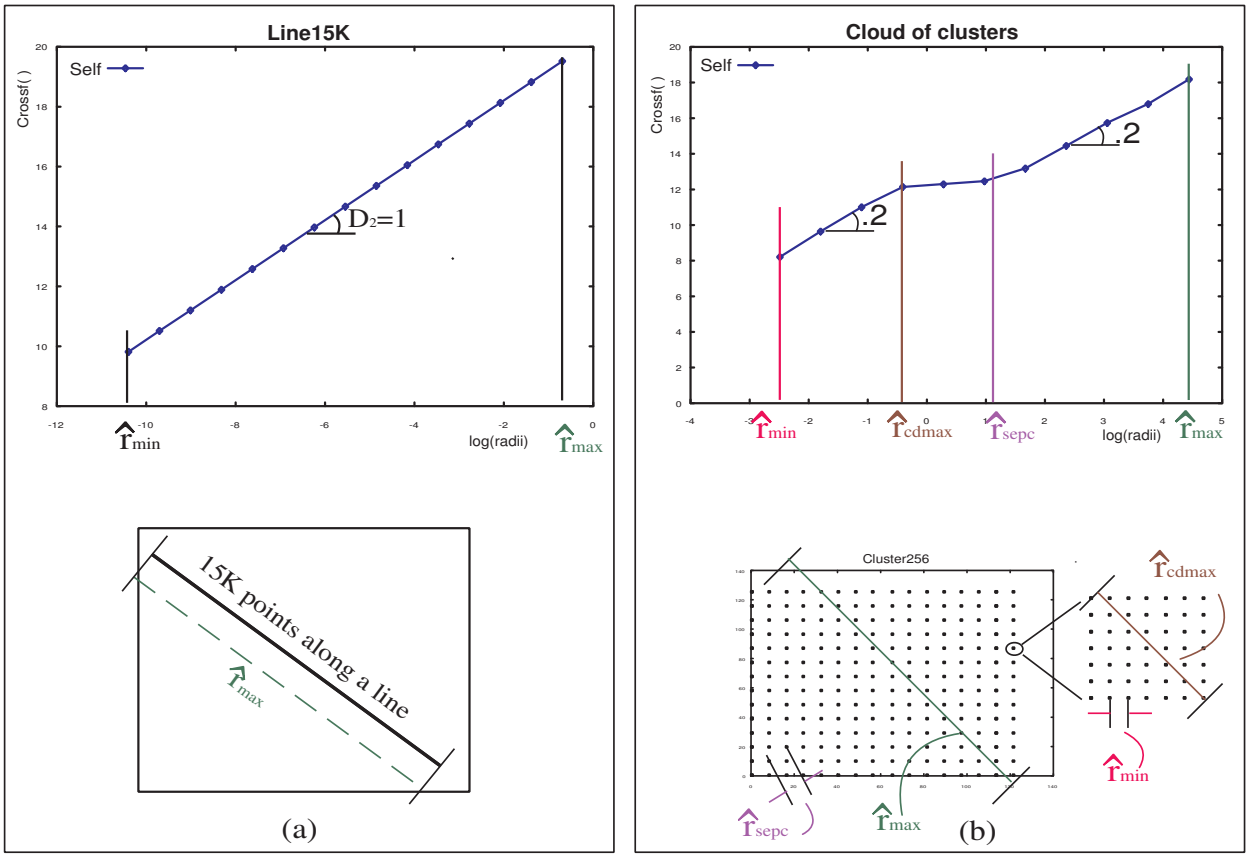

Figure 3: Measurements obtained from self-plots: (a) Line dataset, and (b) Super-cluster dataset.

Figure 3 illustrates the properties stated above. Figure 3(a) in the lower row presents a line with 15,000 points. Its Self-plot is linear, and its slope gives the value of D_2 equal to 1. This value is also the intrinsic dimensionality of a line. The estimate of minimum and maximum distances between two points $\hat{r}_{min}$ and $\hat{r}_{max}$ are also presented in Figure 3.

Property 6 *Whenever the self-plot has a plateau from radius $\hat{r}_{min}$ to $\hat{r}_{max}$, the dataset is probably consisting of clusters (see Figure 3).*

Notice that, whenever the self-plot is piecewise linear, then the dataset has characteristic scales. Especially interesting is the case of plateaus. This happens whenever the dataset is not homogeneous. Then, there is more information to be extracted from the self-plot. This occurs to the self-plot of the 'Super-cluster' dataset.

Whenever the self-plot presents flat areas in the middle of the graph, then it means that the dataset is composed by clusters, and the maximum diameter of the cluster can also be estimated. This is shown in Figure 3(b) as $\hat{r}_{cd\ max}$. The characteristic separation between the cluster scan also be estimated and is presented in Figure 3(b) as $\hat{r}_{sepc}$. Notice that the values of radius presented in the plots are actually the log of the radii. Next we show how to interpret a cross-cloud plot, to determine the relative positioning of two "clouds".

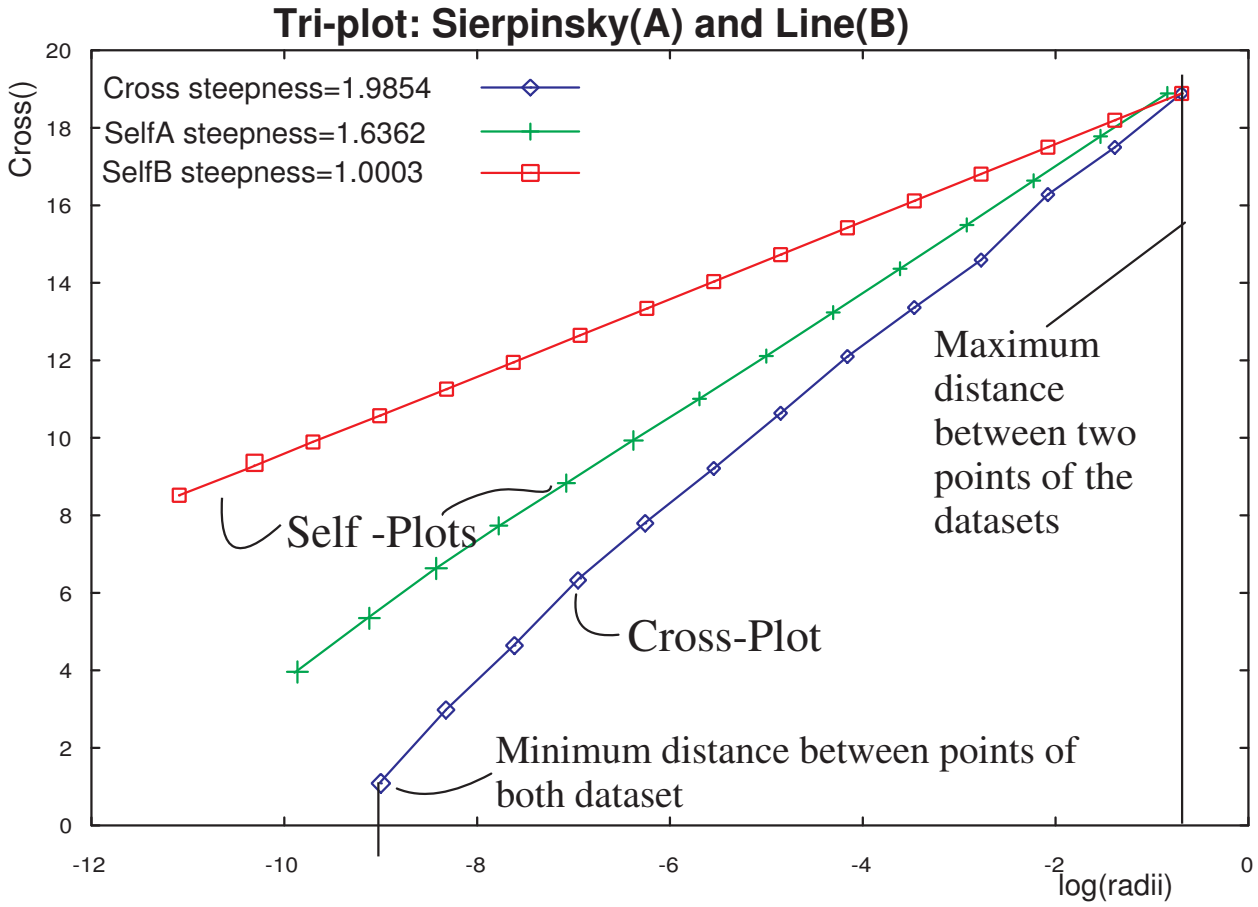

Figure 4: Example of a tri-plot indicating where to find meaningful information. The cross-plot is always in blue with diamonds, $Self_A$ in green with crosses and $Self_B$ in red with squares.

3.1.2 Properties of the cross-cloud plot

Figure 4 presents an example of a tri-plot, where dataset A is a randomly generated set of 6,000 points of a line ($y = x0/x, y \in [0, 1]$), and dataset B is a Sierpinsky triangle with 6,561 points. These two datasets where chosen to highlight some interesting properties of this kind of plot. These properties (depicted in Figure 4) are discussed in the following.

Property 7 *Estimate the minimum distance between the two datasets. This minimum distance is given by the smallest value of the* $\log(r)$ *axis, which has a non-zero value in the cross-cloud function.*

Property 8 *Estimate the maximum distance between the two datasets, that is, the maximum diameter surrounding both datasets. This estimate is given by the greatest value of the* $\log(r)$ *axis before the plot turns flat.*

Property 9 *If the cross-cloud plot is linear, its slope depends on the relative positioning of the two datasets.*

Property 10 *Whenever the cross-cloud plot has a flat part for very small radii, this means that are points in dataset* A *which coincide exactly with points in dataset* B *(duplicate points in both datasets).*

All the previous estimates can be obtained with a single processing pass over both datasets without effectively computing any distance. Recall that we count the occupation of each grid cell.

Property 11 *The steepness of the cross-cloud plot is always greater than or equal to the steepness of the steeper self-plot of the datasets.*

4 Practical usage / Data mining across two clouds

In order to present the analysis process through the cross-cloud plot method, it is necessary to define some terms used in this paper:

Definition 8 *The shape of a dataset means its formation law.*

Definition 9 *Two datasets are collocated if they have minimum bounding boxes that are highly overlapping.*

Definition 10 *The placement of a dataset stands for the position and orientation of it.*

We use these three terms when comparing two datasets. Two datasets can have the same shape but different placement (two non-colinear lines). If a dataset was manipulated by *affine transformations*, the resulting dataset will have the same shape but different placement. Indeed, two datasets with the same intrinsic dimensionality can have different shapes. This is the case of a line and a circumference as they have the same intrinsic dimensionality ('1'), but different shapes.

4.1 Rules for tri-plot analysis

In this section we present rules which will help analyze and classify the relationship between two datasets. Through the analysis of the tri-plots, we can get information about the intrinsic structure and the global relationship between both datasets.

Table 4.1 concisely presents the condition set used in the tri-plot analysis. The rules are presented below.

Rule 1 If all three plots of a tri-plot are similar, that is $Self_A \approx Self_B \approx Cross$, then both datasets are probably identical. In this case, the three graphs in the tri-plot will be on top of each other. This means that both datasets probably have the same intrinsic dimensionality, shape and placement. This is also the case when either one dataset is a subset of the other, or both datasets are samples from a bigger one. Figure 5 shows the tri-plots for (a) two lines with different number of objects, (b) two Sierpinsky triangles, and (c) two coplanar squares in 3D. All datasets in Figure 5 are in a 2D manifold. In all these examples both datasets have the same shape and placement but different number of points.

Rule	Condition	Meaning	Example
A	A and B are similar ($Self_A$ and $Self_B$ have same steepness), **and**		
1	$Cross$ has the same steepness as $Self_A$ and $Self_B$	Datasets A and B are probably statistically identical	Figure 5
2	$Cross$ has steepness comparable to that of $Self_A$ and $Self_B$	Both datasets have the same intrinsic dimensionality	Figure 6
3	$Cross$ is much steeper than both $Self_A$ and $Self_B$	The datasets are disjoint	Figure 7
B	A and B are not similar ($Self_A$ and $Self_B$ have different steepness), **and**		
4	$Cross$ has the same steepnes as $Self_A$ or $Self_B$	The less steep dataset is probably a proper subset of the other	Figure 8
5	$Cross$ has steepness comparable to that of $Self_A$ and $Self_B$	Needs further analysis	Figure 9
3	$Cross$ is much steeper than both $Self_A$ and $Self_B$	The datasets are disjoint	Figure 7

Table 2: Conditions and rules used in tri-plot analysis.

Rule 2 If the steepness of both datasets are similar ($Self_A \approx Self_B$), but the steepness of $Cross$ is only moderately steeper than both, then both datasets probably have the same intrinsic dimensionality, but they come from different placements. Further analysis, as conducted by the pq-plot analysis, can indicate whether both datasets are separable or not, and if separable to what extent. These are the cases of intersecting lines in two or more dimensional spaces, intersecting planes, or two Sierpinsky datasets with one rotated over the other, as presented in Figures 6(a), 6(b) and 6(c) respectively. Notice that the figures show the datasets and the self-plots with the same color pattern.

Rule 3 If the $Cross$ is much steeper than $Self_A$ and $Self_B$ (does not matter whether they are similar or not), then the datasets are disjoint. For two intersecting datasets, the $Cross$ steepness will not be so far from the steepness of their self-plots. However, if the $Cross$ is much steeper than both $Self_A$ and $Self_B$, it means that the minimum distance between points from the datasets is bigger than the average distance of the nearest neighbors of points in both datasets, so the datasets are disjointed. In fact, this case leads to the conclusion that both datasets are well-defined clusters; hence they should be separable by traditional clustering techniques. Examples of this situation are non-intersecting lines, squares far apart, or a Sierpinsky triangle and a plane which is not coplanar with the Sierpinsky's supporting plane, as shown in Figure 7(a) to 7(c). All datasets are in 3D space. Notice that the self-plots have the expected slopes, but the cross-plots have very high steepness (18, 13 and 26 respectively).

Rule 4 Without loss of generality, let $Self_A$ be the steeper between $Self_A$ and $Self_B$. If the steepness of the self-plots are not similar ($Self_A \not\approx Self_B$) and the $Cross$ is equal to $Self_A$, then dataset B is a sub-manifold of dataset A. Remember that the steepness of the $Cross$ can not be smaller than the steepness of the self plots $Self_A$ or $Self_B$. Therefore, if the steepness of the $Cross$ is similar to one of the self steepnesses, e.g. $Cross \approx Self_A$, then the other graph (in this case $Self_B$) will be less steeper than $Cross$. This means that

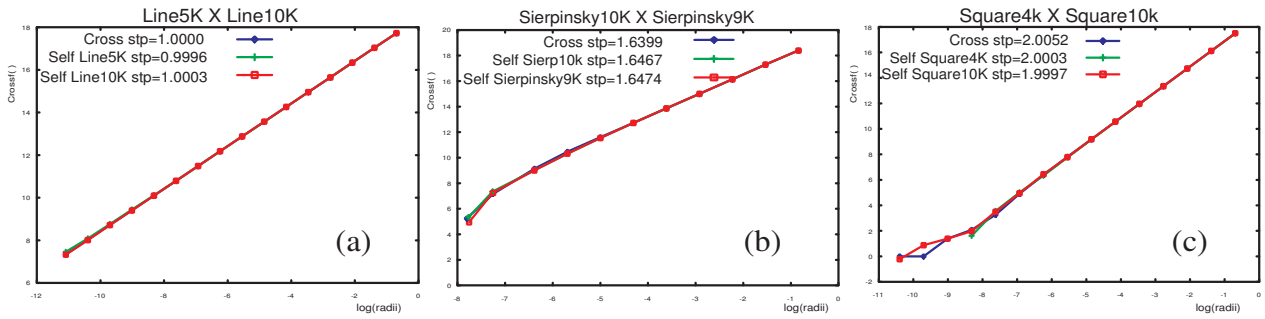

Figure 5: **Rule 1:** The two datasets have the same shape and placement: (a) Two superimposed lines (all plots have slope ≈ 1, (b) Two superimposed Sierpinsky triangles (all plots have slopes $\approx 1.64 \approx \log 3 / \log 2$, (c) Two superimposed squares (all plots have slopes ≈ 2. All datasets are in 2D space, and the axes of all tri-plots are in log-log scale.

the points in dataset B have a stronger correlation than the points in dataset A. Rule 1 deals with the situation where both datasets are subsets of a larger one, or one is a subset of another, but there is no rule to extract the subsets. Rule 4 deals with the same case of occurrence of subsets, but here there are rules to choose points that pertain to the dataset with a smoother self-plot. Examples of this case are a line embedded in a plane, a Sierpinsky dataset and its supporting plane, and a square embedded in a volume, as shown in Figures 8(a), 8(b) and 8(c) respectively.

Rule 5 If $Self_A \not\approx Self_B$ and the $Cross$ is only moderately steeper than $Self_A$ and $Self_B$, then both datasets come from different placement. In this case, both datasets have a different shape and intrinsic dimensionality, and they are not related to each other. They are, however, collocated, or at least intersecting. This means that although part of the datasets may be separable, this would not be true for the entire dataset, or for both datasets. Whenever this situation occurs, it should be further analyzed, for example, using the pq-plot. These are the cases of a line with a Sierpinsky triangle, a line piercing a square, and a Sierpinsky intersecting a square, as Figure 9 shows.

4.2 Applying the proposed rules to real datasets

In the previous section we presented the proposed rules and how such rules apply to synthetic datasets. In this way, we take advantage of the intuition that synthetic datasets provide. In this section we show how the tri-plots can be used to characterize the relationship between two datasets. In Figure 10 we present tri-plots for real datasets. This figure presents the four cases found in our real datasets.

Rule 1 (probably identical) There are four pairs of datasets in Figure 10 which conform to Rule 1: the two different subsets of the California-political datasets (Figure 10(a)), the two galaxy datasets (Figure 10(b)), the Iris-versicolor and Iris-virginica datasets (Figure 10(c)), and two different subsets of the California-water dataset (Figure 10(d)). As Figure 1 showed, all of these datasets, in fact, exhibit similar distribution.

Rule 3 (disjoint datasets) There are two pairs of datasets in Figure 10 which conform to Rule 3: the Iris-Versicolor and Iris-Setosa datasets (Figure 10(e)), and the Democrat and Republican vote datasets (Figure 10(f)). The steepness of their cross-plot is much bigger than the steepness of their self-plots. In Figure 1(c) we

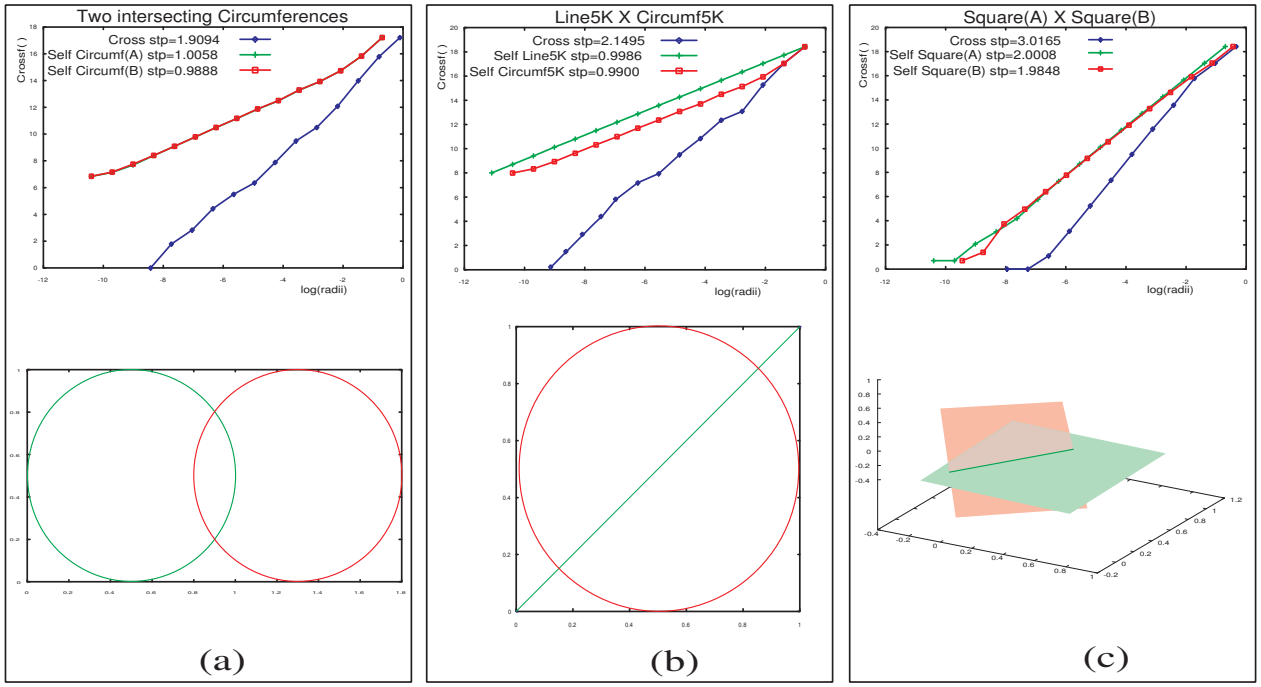

Figure 6: **Rule 2:** The two datasets have the same intrinsic dimensionality, but different placements: (a) Two intersecting circumferences in 2D space, (b) A line crossing a circumference in 2D space, (c) Two piercing planes in 3D space. The upper row shows the tri-plots with the axes in log-log scale. The lower row shows the corresponding datasets in their respective spaces.

saw that the Versicolor and Setosa datasets are indeed apart, and we know that the Democrat and Republican parties have distinct behavior which allows separation of their members.

Rule 4 (sub-manifold) Figure 10(g) presents the tri-plot of California-water and California-political datasets. Let us recall that the dataset with smaller steepness is probably a proper sub-manifold of the one with larger steepness (or from the set in which both are samples). Thus, Figure 10(g) indicates that the California-political dataset is probably a subset of the California-water one. This makes sense since many political divisions use water paths, and there are some criteria to choose the political divisions.

Rule 5 (different placement) There are two pairs of datasets in Figure 10 which conform to Rule 5. Figure 10(h) indicates that the California-railroad and the California-political datasets agree with Rule 5, which is a reasonable conclusion as both datasets hold geographical data built with distinct objectives. Figure 10(i) indicates that the LC datasets also agree with Rule 5 and require further analysis. The flat parts in Figure 10(i) and in the Self-political plot in Figure 10(h) indicate that these datasets possibly have duplicate points (or even very close points). Notice that we presented the case of the "Super-cluster" in real datasets (see Figure 10(b)). It shows that the Galaxy datasets have clusters at two characteristic distances. The datasets also repel each other for the radius range of the cluster diameter. After analyzing the relationship between two datasets by the Tri-plots, a more detailed analysis can be performed taking advantage of the pq-plots as we will explain in the next section.

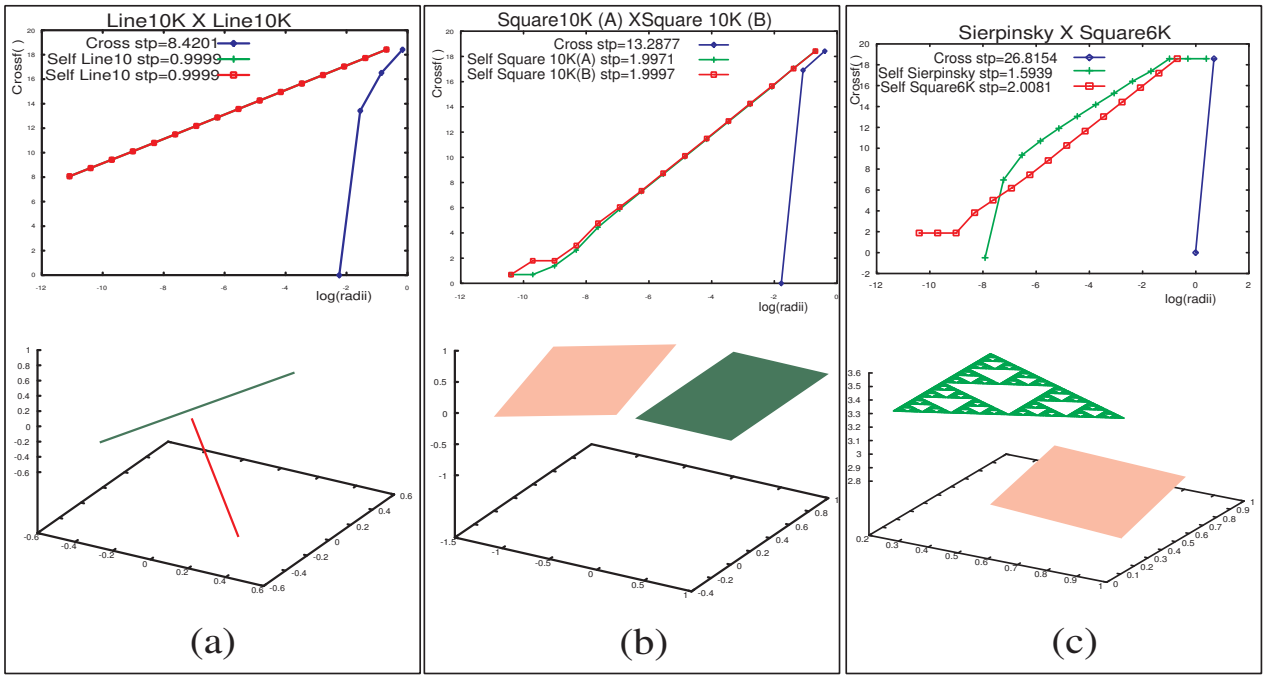

Figure 7: **Rule 3:** The two datasets are disjoint: (a) two non-intersecting lines, (b) two non-intersecting squares, (c) a square and a Sierpinsky triangle. The upper row shows the tri-plots with the axes in log-log scale. The lower row shows the corresponding datasets in 3D space.

4.3 Analysis of the pq-plot

The pq-plot allows emphasizing the relationship between two datasets, weighting the contribution of one dataset when comparing its distance distribution with the distance distribution of the other dataset.

The analysis of the pq-plots is directed to specific parts of the cross-cloud plots in contrast to the more global analysis of the tri-plots. Regarding the pq-plot, even if a $Cross_{A,B}(r, p, q)$ plot with $p \neq 1 \neq q$ happens to be a line, the slope is not meaningful; only its 'shape' has meaningful properties. Also, due to the normalization given by $\log(N_A \cdot N_B) / \log(N_A^p \cdot N_B^p)$, both the leftmost and rightmost points in all pq-plots are the same. Recalling Equation 1, if a particular $C_{A,i}$ (or $C_{B,i}$) used to calculate the $Cross_{A,B}(r, p, q)$ plots is null for a given radius r in a given region of the space, the corresponding $C_{B,i}$ (or $C_{A,i}$) will not contribute to the total number of counts for this particular radius, leading to a flattening region in this part of the curve. Otherwise, if there is a regular distribution of distances over a continuous part of the curve, the resulting curve will exhibit a linear shape. Sudden rises in a plot indicate a large growth of counts starting at that radius. Hence, the two shapes in the curves of the Cross-cloud plots that are worth looking for are: the linear parts, and the regions where the curves are flat.

The cross-cloud plots, $Cross_{A,B}(r, k, 1)$, and $CrossAB(r, 1, k)$ with $k \gg 1$ (which we have named W_A and W_B because they are 'weighted'), can be generated for any value of k. However, increasing k only increases the distortions on the plot, without giving any extra information. Thus, for the discussion that follows, we arbitrarily adopt the value of $k = 10$. Each conclusion is valid for the range of radii which presents specific behavior. Next we discuss two representative situations, using pairs of synthetic datasets and comparing the obtained tri-plots and pq-plots.

Figure 11 compares two pairs of datasets: circumference-circumference and line-circumference. This

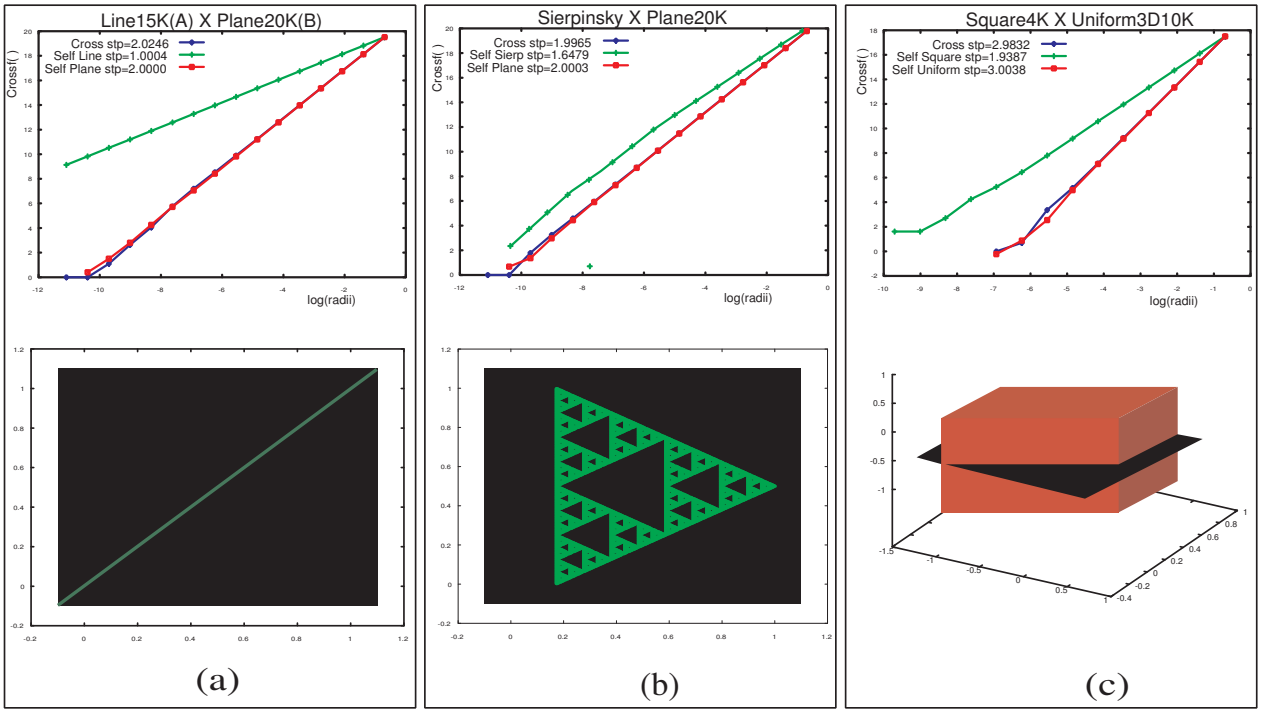

Figure 8: **Rule 4:** One dataset is a proper subset of the other dataset: (a) a square overlapping a line in 2D space, (b) a Sierpinsky triangle and its supporting plane in 2D space, (c) a volume travesed by a plane in 3D space. The upper row shows the tri-plots in log-log scale. The lower row shows the datasets in their respective spaces.

illustrates the situation stated by Rule 2: the two datasets are similar ($Self_A \approx Self_B$ and $Cross$ steepness is less or equal than the steepness of $Self_A$ plus the steepness of $Self_B$). By looking only at the tri-plots in Figures 11(a) and 11(d), it is not possible to say anything else about the datasets. However, looking at Figure 11(b) we can see that the three graphs are on top of each other. This means that both datasets have the same behavior under weighted calculation ($Cross(r, 1, 10)$ and $Cross(r, 10, 1)$). Thus, both datasets have the same shape. On the other hand, the behavior of the pq-plot in Figure 11(e) shows that indeed both datasets have different shapes and it also shows how they are correlated within specific radii range (Region I and II on the plots).

In this section we proposed the rules to analyze the tri-plots and the pq-plots using well-understandable synthetic datasets in two- or three-dimensional spaces. However, the same conclusions should apply for real datasets in any multi-dimensional space. In fact, for real datasets it is usually difficult to know how to describe the relationship between the attributes and to know if they are correlated. Nonetheless, our proposed analysis can indicate not only the existence of correlations, but also how "tight" they are. This analysis can also provide evidence of how separable the datasets are, as well as if it is possible to classify points as belonging to one or to the other dataset.

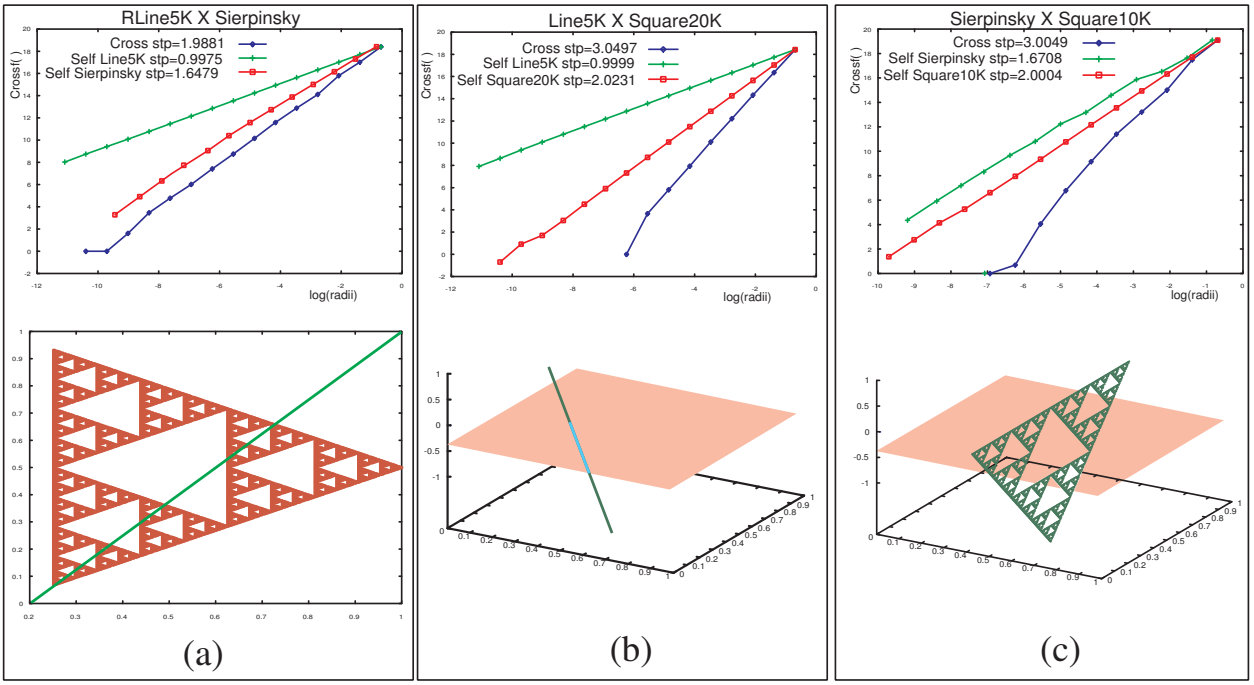

Figure 9: **Rule 5:** The datasets come from different placements: (a) a line and a Sierpinsky triangle in 2D space, (b) a line piercing a square in 3D space, (c) a plane and an intersecting Sierpinsky triangle in 3D space. The upper row shows the tri-plots in log-log scale. The lower row shows the corresponding datasets in their respective space.

4.4 Using pq-plots to analyze datasets

In Figure 12 we present pq-plots only for some cases of real datasets, due to space limitations. Figure 12(a) shows the pq-plot of the points from the Galaxy datasets. For the range indicated, there is a distinct separation between the datasets. Besides confirming that the two Galaxy types indeed repel each other, the pq-plots shows that there are few clusters consisting only of the 'exp' type (although there are clusters including points of both datasets also with only the 'dev' points). However, outside this range, they are almost identical. Figure 12(b), as expected, confirms that the Democrat and Republican datasets are separable. This is because the weighted plots have completely opposite behaviors.

Figure 12(c) shows the pq-plot of the California-water and California-political datasets. In this plot, there are four ranges with distinct behaviors. Range I corresponds to very small distances, so these distances are probably less than the resolution of the measurements; therefore they are not meaningful. Ranges II and III are where the real distances are meaningful. The sudden fall to the left of the wWater-plot in range II means that there are very few points in the political dataset at distances below this range from points in the water dataset. This indicates a kind of "repulsion" of points from both datasets for these small distances. In range III, both datasets have approximately the same behavior. Range IV is almost flat for all plots, meaning that there are almost no more pairs within this distance range. In fact, the "almost flat" part of the graph is due to a few outliers in the dataset.

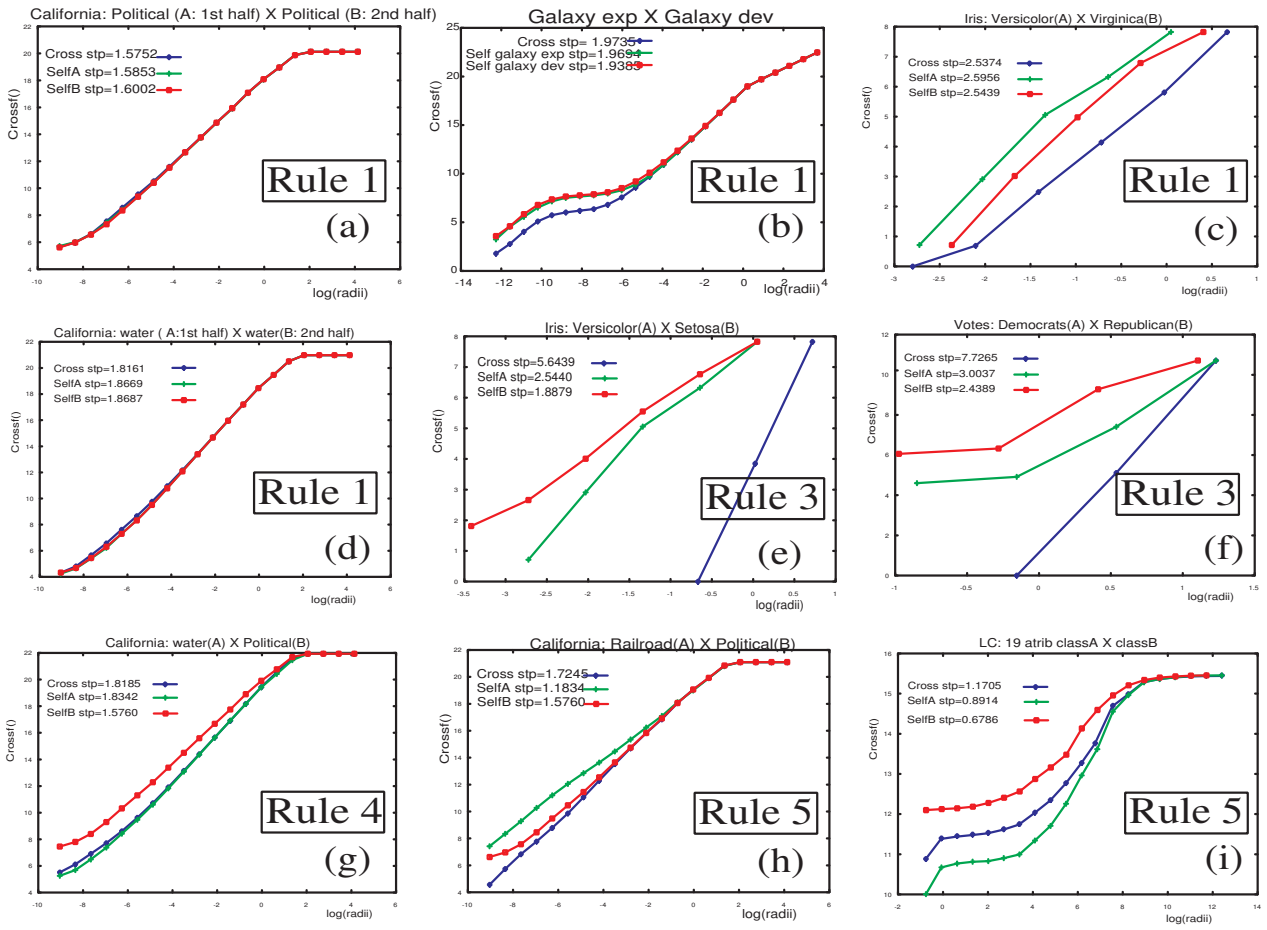

Figure 10: Tri-plots of real datasets and their classification as obtained from rules 1-5.

4.5 Membership testing and classification

In this section we illustrate the power of our cross-cloud plots in another setting: namely membership testing and classification. Figure 13 illustrates the following situation. Given two datasets A and B, where dataset A consists of 20 points along a line, and dataset B of 900 points in a 'tight' square, a new point, indicated by the question mark '?' arrives. Which set, if any, does it belong to?

Visually, the new point '?' should belong to the Line20 set. However, nearest neighbors or decision- tree-based classifiers would put it into the square: the new point has 900 neighbors of type square, before its first neighbor of type Line20 shows up!

We propose a method that exploits Cross-cloud plots to achieve the correct classification of the incoming point '?'. Our method processes the new point as a singleton dataset and compares its cross-plots with the self-plots of each target dataset.The classification process we propose compares the steepness of CrossLine, Point and CrossSquare, Point with the steepness of SelfLine and SelfSquare. If CrossLine, Point is similar to SelfLine, we classify the new point into the Line20 dataset. Conceivably, our new point could be classified under one, two, or more of the target datasets.

The full details of the classification method are the topic of ongoing research. We have just shown that this is another benefit that can be derived from the presented Cross-cloud technique.

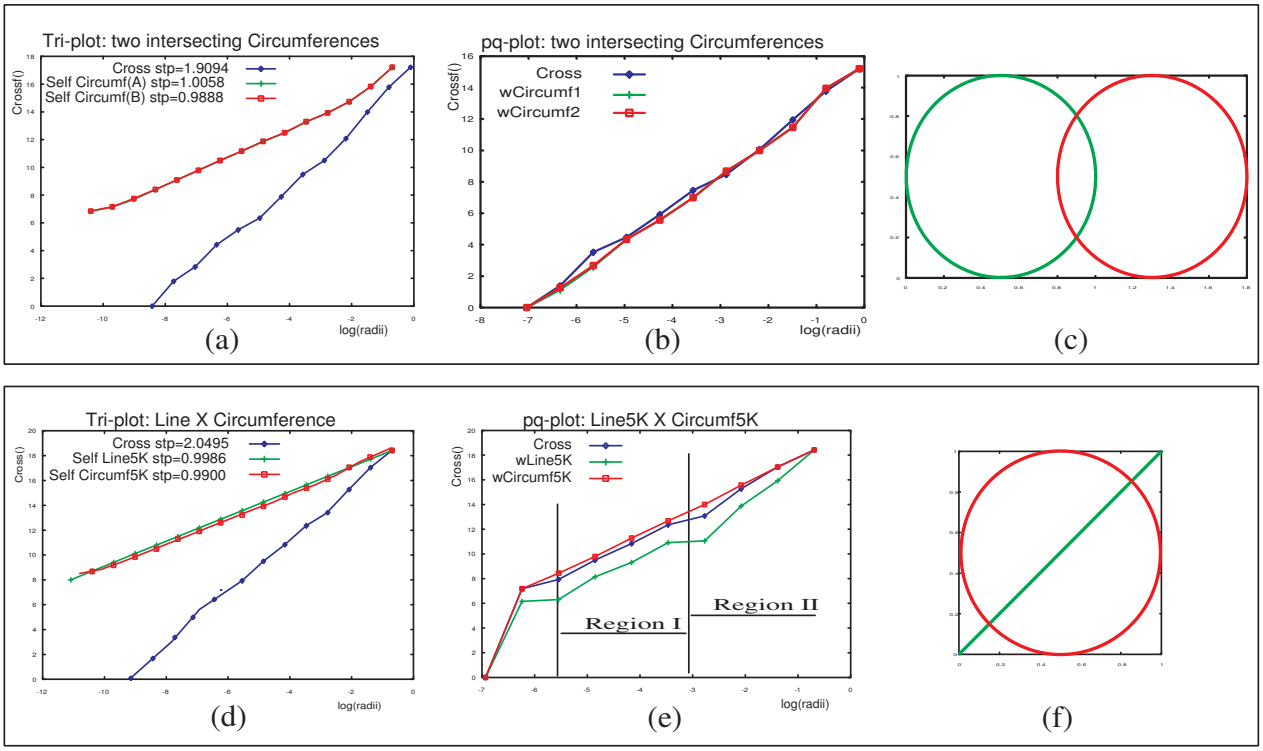

Figure 11: Cross-cloud plots for two pairs of datasets: (a) the tri-plot of two intersecting circumferences (as shown in (c)), (b) the pq-plot of the two circumferences, (d) the tri-plot of a line intersecting a circumference (as shown in (f)), and (e) the pq-plot of the line and the circumference.

5 Speed considerations

In this section we present the algorithm designed to generate the tri-plots as well as its performance for multi-dimensional large datasets.

5.1 Algorithm to generate the Tri-plots

We developed a single-pass algorithm over both datasets to obtain the required tri-plots, which is presented in Figure 14. It is defined as follows.

Given two datasets A and B with cardinalities N_A and N_B in a n-dimensional space,

Generate the tri-plot of the datasets, that is, the cross-plot, $Self_A$ plot and $Self_B$ plot.

Calling F the number of non-empty cells in each grid, clearly $1 \leq F \leq N_A + N_B$, and it does not depend on the dimensionality n. Thus, our algorithm scales up for arbitrarily large datasets, and arbitrarily high dimensionality. This is rarely true for any other spatial data mining method in the literature. The algorithm to generate the pq-plots is similar to the algorithm depicted in Figure 14, where instead of $Self_A$ and $Self_B$ plots W_A and W_B plots are drawn. Due to space limitations we do not show it here.

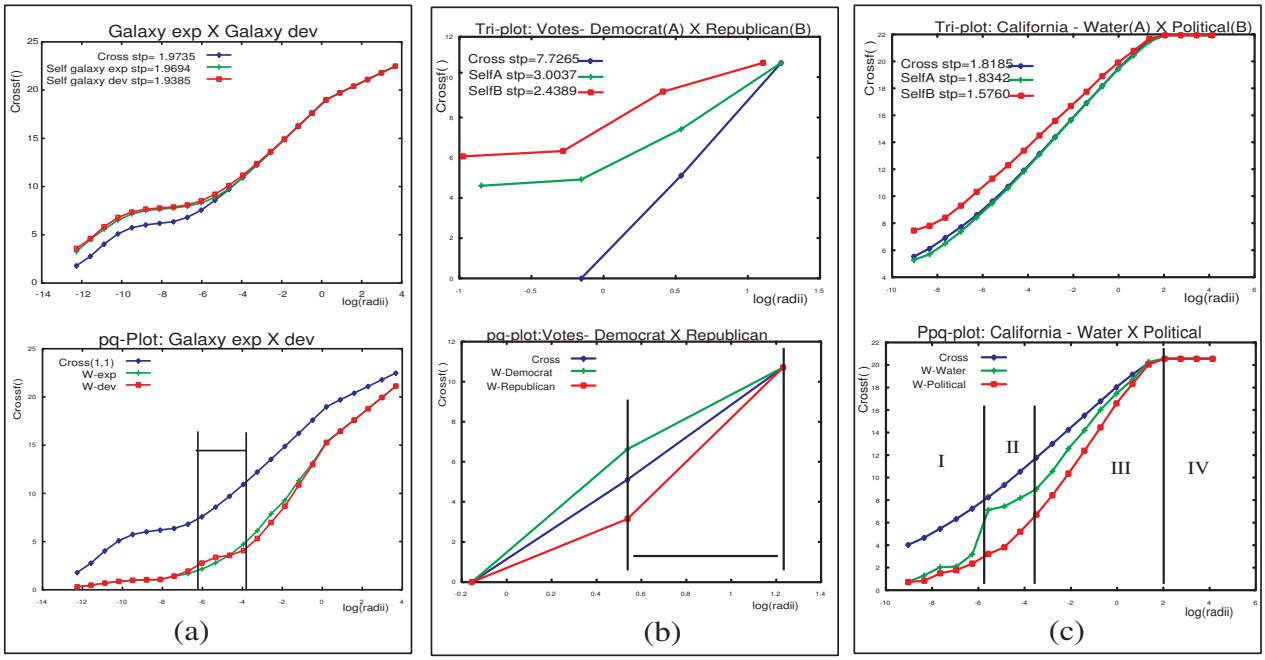

Figure 12: pq-plots for real datasets: (a) Galaxy, (b) Democrat and Republican, (c) California-water and California-political. The upper row shows the tri-plots and the lower row the corresponding pq-plots. The axes are in log-log scale

5.2 Scalability

The algorithm developed is linear over the total number of points in both datasets, i.e., $O(N_A + N_B)$. If l is the number of points that we want in each Cross-cloud plot (i.e., the number of grid sizes), then the complexity of our algorithm is $O((N_B + N_A) \cdot l \cdot n)$, where n is the embedding dimensionality of the input datasets. Figure 14 presents a brief algorithm to generate the required Cross-cloud plots. Figure 15 shows the wall-clock time required to process datasets on a Pentium II machine under NT4.0. The datasets have varying numbers of points in 2, 8 and 16-dimensional spaces, and we generated 20 grid sizes for each dataset. The number of points shown in Figure 15 is the sum of the number of points in both datasets. In this figure we can see that the execution time of this algorithm is indeed linear on the number of points in the datasets. Figure 16 corroborates that the our algorithm is also linear on the dimensionality of the datasets. In this case we used datasets with 100,000, 200,000 and 300,000 with dimensions 2 to 40. This shows that the proposed algorithm is safe of the so-called 'dimensionality curse'.

Notice that steps 1 and 2 of the algorithm read the datasets and maintain counts of each non-empty grid cell. These counts can be kept in any data structure (hash tables, quadtrees, etc.).

6 Conclusions

We have proposed the cross-cloud plot, a new tool for spatial data mining across two n-dimensional datasets. We have shown that our tool has all the desirable properties we asked for.

- It can spot whether two clouds are disjointed (separable), statistically identical, repelling, or in-between.

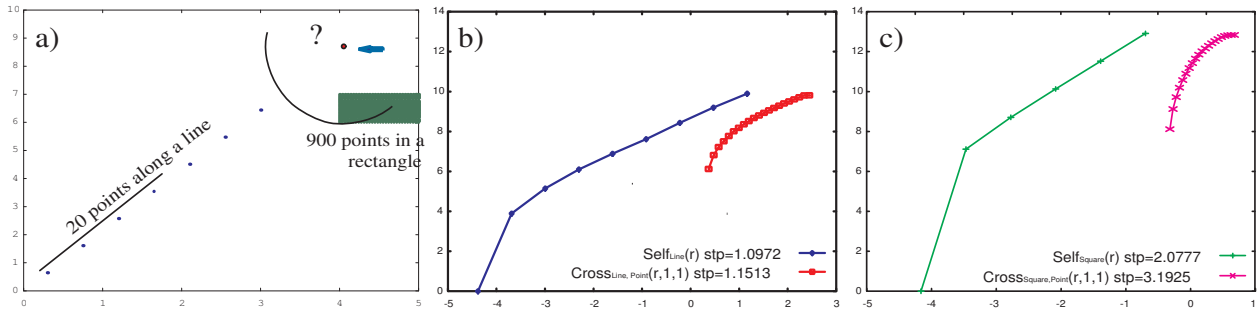

Figure 13: Classifying a point as either belonging to a sparse line or to a dense square, using the cross-cloud method: (a) spatial placement of the incoming point and the datasets, (b) $Self_{\text{Line}}$ and $Cross_{\text{Line,Point}}$ plot, (c) $Self_{\text{Square}}$ and $Cross_{\text{Square,Point}}$ plot.
q

That is, it can answer questions Q1 to Q4 presented in section 1.

- It can be used for classification and is capable of "learning" a shape/cloud, where traditional classifiers fail to do so (that is, question Q5 presented in section 1).
- It is very fast and scalable: Our algorithm requires a single pass over each dataset, and the memory requirement is proportional to the number F of non-empty grid cells and to the number l of grid sizes requested ($1 \leq F \leq N_A + N_B$, and clearly not exploding exponentially).
- It can be applied to high-dimensional datasets as well because the dimensionality of the dataset for the counting of points inside the grid cells does not matter.

The experiments on real datasets show that our tool finds patterns that no other known method can. We believe that our cross-cloud plot is a powerful tool for spatial data mining and that we have just seen only the beginning of its potential uses.

References

[AGGR98] Rakesh Agrawal, Johannes Gherke, Dimitrios Gunopoulos, and Prabhakar Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proc. of ACM SIGMOD Conf. on Management of Data*, pages 94–105, 1998.

[BBK98] Stefan Berchtold, Christian Böhm, and Hans-Peter Kriegel. The pyramid-tree: Breaking the curse of dimensionality. In *Proc. ACM SIGMOD Conf. on Management of Data*, pages 142–153, 1998.

[BF95] A Belussi and Christos Faloutsos. Estimating the selectivity of spatial queries using the 'correlation' fractal dimension. In *Proc. of VLDB - Very Large Data Bases*, pages 299–310, 1995.

[Cha98] Surajit Chaudhuri. Data mining and database systems: Where is the intersection? *Data Engineering Bulletin*, 21(1):4–8, 1998.

[CHY96] Ming-Syan Chen, Jiawei Han, and Philip S. Yu. Data mining - an overview from a database perspective. *IEEE Transactions on Knowledge and Data Engineering*, 8(6):866–883, 1996.

Algorithm: Fast Tri-plot
Inputs: Two datasets, A and B, with N_A and N_B points respectively, normalized to the unit hyper-cube, and the number l of desired points in each plot.
Output: Tri-plot

Begin
1 - For each point a of dataset A
For each desirable grid-size $r = 1/2^j$, $j = 1, 2, \ldots, l$;
Decide which grid cell it falls in (say, in the i-th cell);
Increment the count $C_{A,i}$;

2 - For each point b of dataset B
For each desirable grid-size $r = 1/2^j$, $j = 1, 2, \ldots, l$;
Decide which grid cell it falls in (say, in the i-th cell);
Increment the count $C_{B,i}$;

3 - Compute the sum of product occupancies for the functions:
$Self_A(r) = \log\left(\frac{1}{2}\sum_i C_{A,i} \cdot (C_{A,i} - 1)\right)$,
$Self_B(r) = \log\left(\frac{1}{2}\sum_i C_{B,i} \cdot (C_{B,i} - 1)\right)$,
$Cross_{A,B}(r, 1, 1) = \log\left(\sum_i C_{A,i} \cdot C_{B,i}\right)$

4 - Print the tri-plot:
for $r = 1/2^j$, $j = 1, 2, \ldots, l$;
Print $Cross_{A,B}(r, 1, 1)$
Print $Self_A$ normalized, that is: $Self_A(r) + \log(N_B/N_A)$
Print $Self_B$ normalized, that is: $Self_B(r) + \log(N_A/N_B)$
End

Figure 14: Figure 14

[EFKS98] Martin Ester, Alexander Frommelt, Hans-Peter Kriegel, and Jörg Sander. Algorithms for characterization and trend detection in spatial databases. In *roc. of the 4th International Conference on Knowledge Discovery and Data Mining (KDD-98)*, pages 44–50, 1998.

[EKS99] Martin Ester, Hans-Peter Kriegel, and Jörg Sander. Spatial data mining: A database approach. In *LNCS 1262: Proc. of 5th Intl. Symposium on Spatial Databases (SSD'97)*, pages 47–66, 1999.

[Fay98] Usama M. Fayyad. Mining databases - towards algorithms for knowledge discovery. *Data Engineering Bulletin*, 21(1):39–48, 1998.

[FRB98] Usama M. Fayyad, Cory Reina, and P. S. Bradley. Initialization of iterative refinement clustering algorithms. In *Proc. of the 4th International Conference on Knowledge Discovery and Data Mining (KDD-98)*, pages 194–198, 1998.

[FSJT00] Christos Faloutsos, Bernhard Seeger, Caetano Traina Jr., and Agma Traina. Spatial join selectivity using power laws. In *Proc. ACM SIGMOD 2000 Conf. on Management of Data*, pages 177–188, 2000.

[GGR99] Venkatesh Ganti, Johannes Gehrke, and Raghu Ramakrishnan. Mining very large databases. *IEEE Computer*, 32(8):38–45, 1999.

[iHLN99] iawei Han, Laks V. S. Lakshmanan, and Raymond T. Ng. Constraint-based multidimensional data mining. *IEEE Computer*, 32(8):46–50, 1999.

[JAG99] Roberto J. Bayardo Jr., Rakesh Agrawal, and Dimitrios Gunopulos. Constraint-based rule mining in large, dense databases. In *Proc. of IEEE Intl. Conference on Data Engineering*, pages 188–197, 1999.

[KK94] Daniel A. Keim and Hans-Peter Kriegel. Possibilities and limits in visualizing large amounts of multidimensional data. In *Perceptual Issues in Visualization*. Springer, 1994.

[KN96] Edwin M. Knorr and Raymond T. Ng. Finding aggregate proximity relationships and commonalities in spatial data mining. *IEEE Transactions on Knowledge and Data Engineering*, 8(6):884–897, 1996.

[NH94] Raymond T. Ng and Jiawei Han. Efficient and effective clustering methods for spatial data mining. In *Proc. of VLDB - Very Large Data Bases*, pages 144–155, 1994.

[oC89] Bureau of Census. Tiger/line precensus files: 1990 technical documentation. Bureau of the Census. Washington, DC, 1989.

[Sch88] Heinz Georg Schuster. *Deterministic Chaos*. VCH Publisher, Weinheim, Basel, Cambridge, New York, 1988.

[Sch91] Manfred Schroeder. *Fractals, Chaos, Power Laws*. W.H. Freeman and Company, New York, 1991.

[SCZ98] Gholamhosein Sheikholeslami, Surojit Chatterjee, and Aidong Zhang. Wavecluster: A multi-resolution clustering approach for very large spatial databases. In *Proc. of VLDB - Very Large Data Bases*, pages 428–439, 1998.

[TZ96] Miron Livny Tian Zhang, Raghu Ramakrishnan. Birch: An efficient data clustering method for very large databases. In *Proc. of ACM SIGMOD Conf. on Management of Data*, pages 103–114, 1996.

[WSB98] Roger Weber, Hans-Jörg Schek, and Stephen Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *Proc. of VLDB - Very Large Data Bases*, pages 194–205, 1998.