Chapter 4

Proximity of ball regions in metric spaces

4.1 Introduction

The approximate similarity search algorithm, whose details are presented in Section 6.9 of next chapter, is based on an estimation of the quantity of objects shared by the query region and data regions: data regions that are judged to share few objects with the query region are discarded. To this aim, this chapter deals with the problem of estimating, given two arbitrary ball regions defined in a metric space, the amount of objects contained in their intersection. Large overlap between regions not always implies that several objects are shared by them. In fact, the number of objects contained in the intersections depends on the distribution of objects in the space. There may be regions with a large intersection and few objects in common, but also regions with a small intersection and many objects in common, which happens when the intersection covers a dense area of the data space. In this chapter, this phenomenon, called the *proximity of regions*, is analyzed and techniques for its quantification are proposed.

The problem of region proximity in vector spaces was studied in [KF92] to decluster nodes of R-trees [Gut84] for parallelism. In this chapter proximity measures are developed for general metric spaces, which naturally subsumes the case of vector spaces. Techniques that satisfy the following criteria are proposed: (1) the proximity is measured with sufficient precision; (2) the computational cost is low; (3) it can be applied to different metrics and data sets; (4) storage overheads are moderate.

The problem of the proximity of regions is analyzed using a probabilistic approach. After a discussion about the computational difficulties of the proximity measurement, heuristics to compute it in an effective and efficient way are proposed. An extensive validation was performed to prove the quality of the proposed approaches.

4.2 Formal definition of proximity

In some existing applications, such as [CPZ97, TTSF00] where proximity was used to obtain a better organization of access methods' structure, a simplified measurement of the proximity between two ball regions was used. We refer to this simplified version as the *trivial proximity*. Specifically, the trivial proximity is computed through a function, linearly proportional to the overlap of the regions, which can be generalized as follows.

$$X^{trivial}(\mathcal{B}(O_x, r_x), \mathcal{B}(O_y, r_y)) = \begin{cases} 0 & \text{if } r_x + r_y < d(O_x, O_y) \\ \frac{r_x + r_y - d(O_x, O_y)}{2 \cdot d_m - d(O_x, O_y)} & \text{if } \max(r_x, r_y) \leq \min(r_x, r_y) + d(O_x, O_y) \\ \frac{2 \cdot \min(r_x, r_y)}{2 \cdot d_m - d(O_x, O_y)} & \text{otherwise} \end{cases} \tag{4.2.1}$$

Equation 4.2.1 sets the proximity to 0 when two ball regions do not overlap. Otherwise, the proximity is proportional to the regions' intersection. The values are

normalized to obtain proximity values in the range [0,1]. The proximity is 1 when both regions include all objects, that is when their radii are equal to the maximum distance d_m.

Although the trivial proximity measure is simple to compute, it is not accurate because it does not take into account the distribution of objects in the space.

The issue of proximity is far more complex. Intuitively, the proximity of two ball regions should be a value proportional to the amount of objects that simultaneously occur in both of the regions. Accordingly, using a probabilistic approach, we may define the proximity $X(\mathcal{B}(O_x, r_x), \mathcal{B}(O_y, r_y))$ of ball regions $\mathcal{B}(O_x, r_x), \mathcal{B}(O_y, r_y)$ as the probability that a randomly chosen object $\mathbf{O} \in \mathcal{D}$ appears in both regions, i.e.

$$X(\mathcal{B}(O_x, r_x), \mathcal{B}(O_y, r_y)) = \mathtt{Pr}\{d(\mathbf{O}, O_x) \leq r_x \wedge d(\mathbf{O}, O_y) \leq r_y\} \tag{4.2.2}$$

Note that the proximity cannot be quantified by the amount of space covered by the regions' intersection. Due to the lack of space coordinates in general metric spaces, such a quantity cannot be determined.

Our aim is to use proximity to design an approximate similarity search algorithm that discard data regions with small probability of sharing objects with the query region. As discussed and proved in Section 6.9, high accuracy of proximity measurement is fundamental for high precision of the approximate similarity search algorithm. In fact our experiments give evidence that results obtained using the trivial proximity are far less accurate than results obtained using the proximity measurements developed in this thesis.

4.3 Application considerations

Several practical applications may benefit from the measurement of the proximity between two regions, for example:

Region splitting Static regions are not typical and the evolution process in storage structures is regulated by specific *split* strategies. When a region $\mathcal{R}$ splits, two new regions, say $\mathcal{R}_1$ and $\mathcal{R}_2$, are created. One way of splitting a region may be more advantageous than another – the content of a region can typically be split in several ways. When a large number of objects is contained in the intersection of two new regions (i.e. their proximity is high), the probability of accessing both regions during query execution is also high, assuming query objects have the same distance distribution as the data objects. For example, consider Figure 4.1a where regions $\mathcal{R}_1$ and $\mathcal{R}_2$, resulting from a split, cover the shared area of a cluster of objects. Queries Q_1 and Q_2 access only one region, while Q_3 access both regions. However, respecting the assumption that query objects have the same distance distribution as the data objects, queries like Q_3 are far more frequent than Q_1 or Q_2, so this partitioning is not very beneficial. Proximity can be used to detect such situations and to determine a good split.

Allocation When a new region is created, it must be placed in the storage system. In such a situation, region proximity measures can be used to determine the most suitable storage bucket for the new region. The strategy is different for single and multiple (independent) disk systems, where parallel processing can be supported. If parallel disks are available, buckets with a high probability of simultaneous access (that is buckets whose corresponding regions have a high

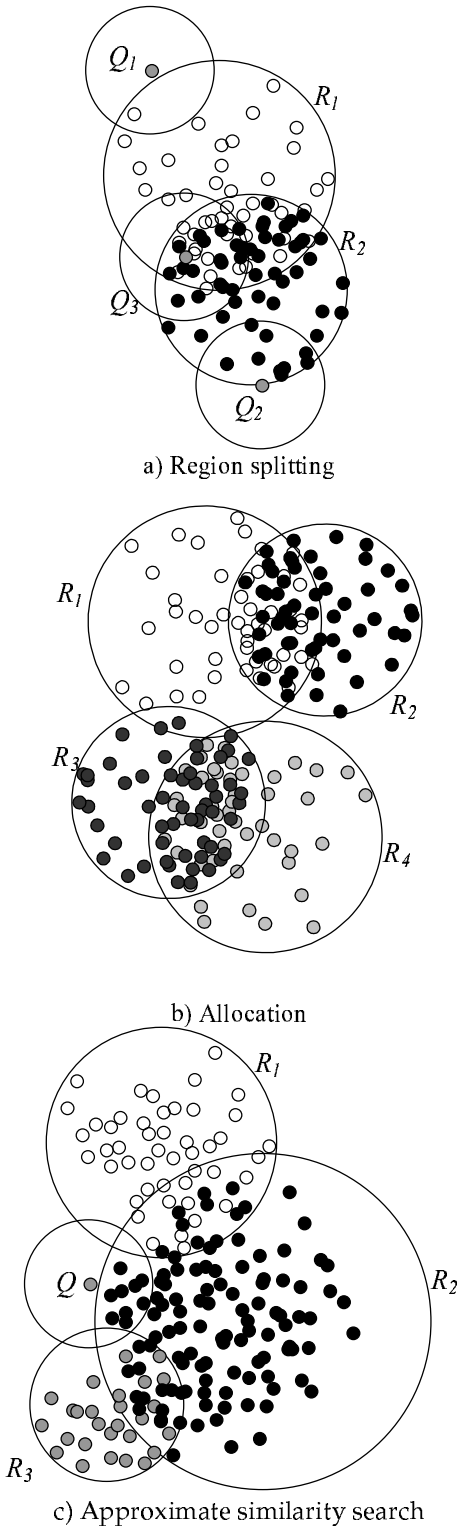


Figure 4.1: Use of the proximity measure for region splitting (a), the allocation of objects on disks (b), and approximate similarity search (c)

proximity) should not be put on the same disk, i.e. they should be *declustered* for parallel access. In a single disk environment, buckets with a high probability of being accessed together should be placed as close as possible, i.e. *clustered*. Consider the example in Figure 4.1b, where regions $\mathcal{R}_1$ and $\mathcal{R}_2$ ($\mathcal{R}_3$ and $\mathcal{R}_4$) have a high proximity. On the other hand, proximity between $\mathcal{R}_1$ and $\mathcal{R}_3$ ($\mathcal{R}_2$ and $\mathcal{R}_4$) is low. The consequence is that $\mathcal{R}_1$ and $\mathcal{R}_2$ ($\mathcal{R}_3$ and $\mathcal{R}_4$) should either be put on different disks, in the case of multiple disks, or they should be placed as close as possible, on a single disk system.

Approximate similarity search The proximity measure can also be useful for approximate similarity search algorithms to prune regions with a small probability of containing qualifying objects, that is data regions with a small proximity to the query region. Exact similarity search algorithms access all buckets whose bounding regions overlap the query region. However, even if the query region overlaps a data region, no or few data objects may appear in the intersection, i.e. the proximity between the query region and the data region is small. The result is that even though all data regions are accessed, few of them actually contain qualifying data objects (few regions have a non empty intersection with the query region). Many accesses are thus actually void and could be saved if the proximity is used as a condition for pruning. In Figure 4.1c, it can be seen that, although the query region Q intersects regions $\mathcal{R}_1$, $\mathcal{R}_2$ and $\mathcal{R}_3$, the intersection with $\mathcal{R}_1$ and $\mathcal{R}_3$ is empty and thus it is not necessary to access these regions.

One of the approximate similarity search methods developed in this thesis implements this idea. Results obtained with this method out-performs those obtained

with the other three methods. Full details of approximate similarity search by using proximity are given in Section 6.9.

4.4 Computational issues

To precisely compute proximity according to Definition 4.2.2, the knowledge of distance distributions with respect to the regions' centers is required. Since any object from $\mathcal{M}$ can become a region's center, such knowledge is not likely to be obtained. However, as discussed in Section 2.3.4 we can assume that, when the index of homogeneity of viewpoints is close to one, the distance distribution is (practically) independent of the centers themselves and depends on the distance between the regions' centers instead. This also implies that all pairs of regions with the same radii and constant distance between centers have on average the same proximity, no matter what their actual centers are. Consequently, we can approximate the proximity of two regions, whose distance between centers is d_{xy}, with the *overall proximity* $X_{d_{xy}}(r_x, r_y)$ of any pairs of regions having radii r_x and r_y, and whose distance between centers is d_{xy}. Specifically we define the overall proximity as the following conditional probability [HPS71]:

$$X_{d_{xy}}(r_x, r_y) = \mathtt{Pr}\{d(\mathbf{O}, \mathbf{O_x}) \leq r_x \wedge d(\mathbf{O}, \mathbf{O_y}) \leq r_y \mid d(\mathbf{O_x}, \mathbf{O_y}) = d_{xy}\}, \qquad (4.4.1)$$

where $\mathbf{O_x}$, $\mathbf{O_y}$, and $\mathbf{O}$ are random objects.

The overall proximity $X_{d_{xy}}(r_x, r_y)$ is defined as the probability that a random object $\mathbf{O}$ belongs to the regions with random centers $\mathbf{O_x}$ and $\mathbf{O_y}$, and radii r_x and r_y, given that the distance between centers $\mathbf{O_x}$ and $\mathbf{O_y}$ is d_{xy}.

The proximity of two ball regions $\mathcal{B}(O_x, r_x)$ and $\mathcal{B}(O_y, r_y)$ such that $d_{xy} = d(O_x, O_y)$

is therefore approximated as follows:

$$X(\mathcal{B}(O_x, r_x), \mathcal{B}(O_y, r_y)) \approx X_{d_{xy}}(r_x, r_y).$$

Now, let us consider the way how $X_{d_{xy}}(r_x, r_y)$ can be computed. Let X, Y and D_{XY} be continuous random variables corresponding, respectively, to distances $d(\mathbf{O}, \mathbf{O_x})$, $d(\mathbf{O}, \mathbf{O_y})$, and $d(\mathbf{O_x}, \mathbf{O_y})$. The *joint conditional density* $f_{X,Y|D_{XY}}(x, y|d_{xy})$ is the probability[1] that distances $d(\mathbf{O}, \mathbf{O_x})$ and $d(\mathbf{O}, \mathbf{O_y})$ are, respectively, x and y, given that $d(\mathbf{O_x}, \mathbf{O_y}) = d_{xy}$. Then, $X_{d_{xy}}(r_x, r_y)$ can be computed as

$$X_{d_{xy}}(r_x, r_y) = \int_0^{r_x} \int_0^{r_y} f_{X,Y|D_{XY}}(x, y|d_{xy}) dy dx \tag{4.4.2}$$

Unfortunately, an explicit form of $f_{X,Y|D_{XY}}(x, y|d_{xy})$ is unknown. In addition, computing and maintaining joint conditional densities as discrete functions would result in a very high number of values. The function depends on three arguments so that the required storage space is $O(n^3)$, provided n is the number of samples used for each argument. This makes the approach totally unacceptable.

We propose to compute the proximity measure by using an approximation of $f_{X,Y|D_{XY}}(x, y|d_{xy})$, designated as $f^{appr}_{X,Y|D_{XY}}(x, y|d_{xy})$, that is expressed in terms of the joint density $f_{XY}(x, y)$. Note that $f_{XY}(x, y)$ is simpler to determine than $f_{X,Y|D_{XY}}(x, y|d_{xy})$. We can observe that X and Y are independent – if we know the distance between $\mathbf{O}$ and $\mathbf{O_x}$, this does not affect the distance between $\mathbf{O}$ and $\mathbf{O_y}$, unless we add some additional information as for instance the distance between $\mathbf{O_x}$ and $\mathbf{O_y}$. Therefore, by definition, we have that

$$f_{XY}(x, y) = f_X(x) \cdot f_Y(y).$$

[1] As also stated in Section 2.3.4, we are using continuous random variables so, to be rigorous, the probability that they take a specific value is by definition 0. However, in order to simplify the explanation, we slightly abuse the terminology and use the term probability to give an intuitive idea of the behavior of the density function being defined.

Given the definition of the random variables X and Y, it is also easy to show that $f_X(d) = f_Y(d)$, so we can omit the name of the random variable and substitute them with the overall distance density $f(d)$ (see Section 2.3.4). Therefore the joint density is defined as:

$$f_{XY}(x, y) = f(x) \cdot f(y).$$

From a storage point of view, such an approach is feasible. The problem remains how to define an accurate transformation that produces the joint conditional density from the joint density. To achieve this we propose some heuristics and we prove through experimentation that they are very accurate. This is the topic of next sections.

4.5 Heuristics for an accurate measurement of the proximity

Given a metric space $\mathcal{M} = (\mathcal{D}, d)$ and two objects O_x and O_y of $\mathcal{D}$ with $d(O_x, O_y) = d_{xy}$, the space of possible distances $x = d(O, O_x)$ and $y = d(O, O_y)$, measured from an object $O \in \mathcal{D}$, is constrained by the triangular inequality, i.e. $x + y \geq d_{xy}$, $x + d_{xy} \geq y$, and $y + d_{xy} \geq x$ (see Section 2.3.1). Figure 4.2 helps to visually identify these constraints. In the gray area, called the *bounded area*, the triangular inequality is satisfied, while in the white area, called the *external area*, the triangular inequality is not satisfied, so an object O with such distances to O_x and O_y does not exist in $\mathcal{D}$.

In general, $f_{X,Y|D_{XY}}(x, y|d_{xy}) \neq f_{XY}(x, y)$, because the *joint density* $f_{XY}(x, y)$ gives the probability that the distances $d(\mathbf{O}, \mathbf{O_x})$ and $d(\mathbf{O}, \mathbf{O_y})$ are x and y, no matter what the distance is between $\mathbf{O_x}$ and $\mathbf{O_y}$. The difference between the two densities

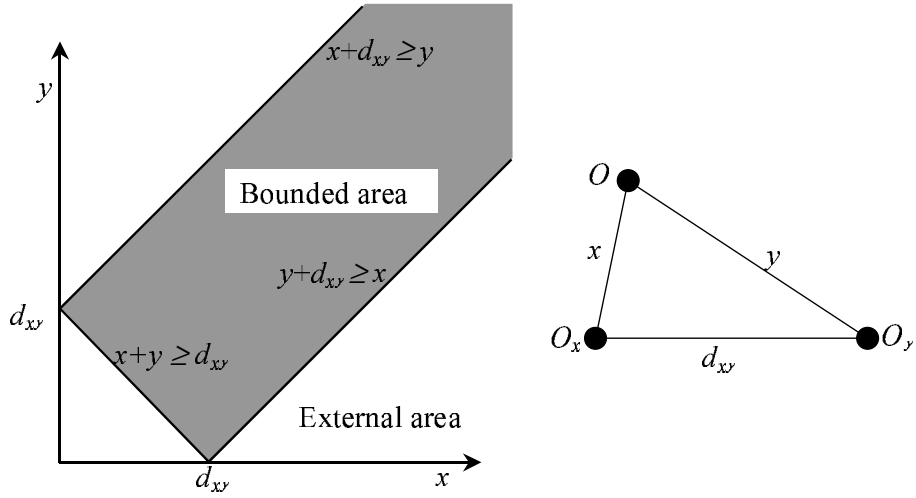
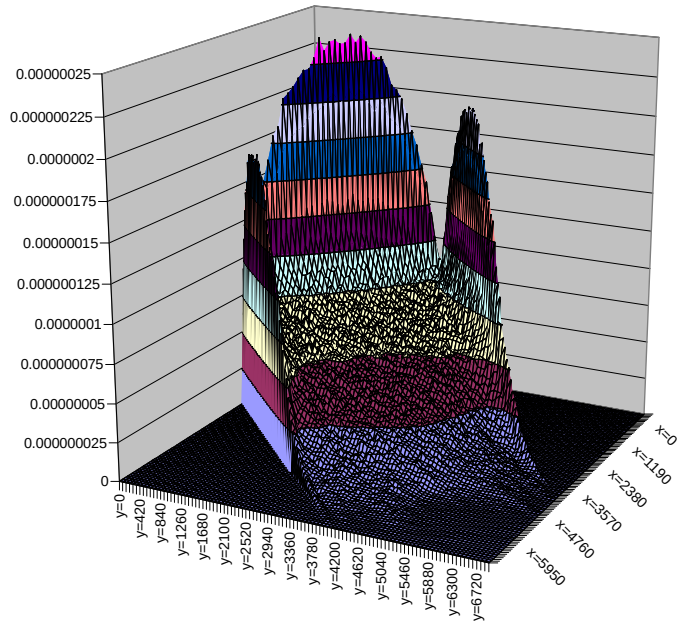


Figure 4.2: Area bounded by the triangular inequality

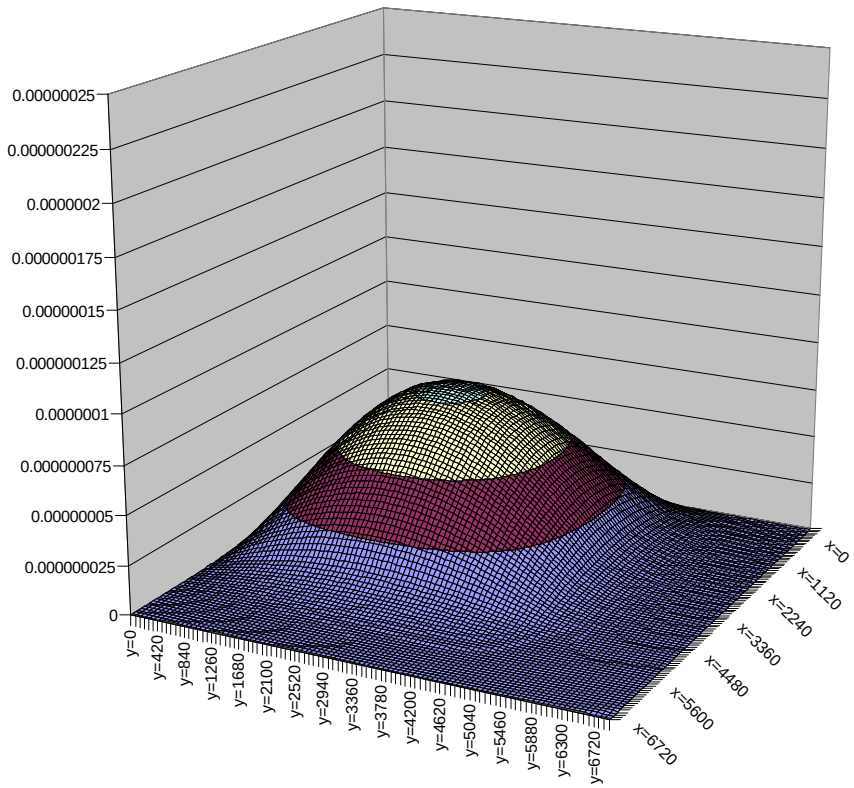
is immediately obvious when we consider the metric space postulates. Accordingly, $f_{X,Y|D_{XY}}(x, y|d_{xy})$ is 0 if x, y, and d_{xy} do not satisfy the triangular inequality, because such distances cannot simply exist. However, $f_{XY}(x, y)$ is not restricted by such a constraint, and any pair of distances $\leq d_m$ is possible. To illustrate this, Figure 4.3 shows the joint conditional density $f_{X,Y|D_{XY}}(x, y|d_{xy})$ for a fixed d_{xy} and the joint density $f_{XY}(x, y)$. Note that the graph of the joint conditional density has values greater than zero only in the bounded area, and that quite high values are located near the edges, while the joint density has values greater than zero also outside the bounded area. The graphs in Figure 4.3 are obtained using the two dimensional uniformly distributed data set UV described in Section 2.5.

4.5.1 Definition of the heuristics

Previous observations form the basis for the heuristics we propose to obtain the approximate joint conditional density $f^{appr}_{XY|D_{XY}}(x, y|d_{xy})$ by means of the joint density.



Joint conditional density



Joint density

Figure 4.3: Comparison between $f_{X,Y|D_{XY}}(x, y|d_{xy})$ and $f_{XY}(x, y)$

The intuitive idea can be outlined as follows:

> *Given d_{xy}, collect values of $f_{XY}(x, y)$ for x and y from the external area and put them inside the bounded area.*

When distances x, y, and d_{xy} satisfy the triangular inequality, the value of $f^{appr}_{XY|D_{XY}}(x, y|d_{xy})$ depends on the specific strategy used to implement the previous idea, otherwise, $f^{appr}_{XY|D_{XY}}(x, y|d_{xy}) = 0$. In this way, the integral over the bounded area is 1. This is the basic assumption of any probabilistic model that would be violated if the joint densities were simply trimmed out by the triangle inequality constraints.

We have tried four different implementations of this heuristic, varying the strategy applied to move density values. Figure 4.4 provides a visual representation of the methods, where the circles represent the joint density function, while the arrows indicate directions in which the necessary quantities are moved from the external area to the bounded area. The strategies can be briefly characterized as follows.

Orthogonal approximation Collect values of $f_{XY}(x, y)$ outside the bounded area and accumulate them on top of the corresponding constraint following a direction that is orthogonal to the constraint.

Parallel approximation Collect values of $f_{XY}(x, y)$ outside the bounded area and accumulate them on top of the corresponding constraint following a direction that is parallel to the axis.

Diagonal approximation Collect values of $f_{XY}(x, y)$ outside the bounded area and accumulate them on top of the corresponding constraint following a direction that always passes through d_m.

Normalized approximation Collect values of $f_{XY}(x,y)$ outside the bounded area to obtain a linear coefficient that modifies (increases) densities inside the bounded area.

Note that proximity, measured by using the $f^{appr}_{XY|D_{XY}}(x,y|d_{xy})$ defined according to the first three methods, Orthogonal, Parallel, and Diagonal, can also be obtained directly from the joint density $f_{X,Y}(x,y)$. In fact, instead of computing $f^{appr}_{XY|D_{XY}}(x,y|d_{xy})$ and integrating it, the same result can be obtained by integrating directly $f_{X,Y}(x,y)$ in the gray marked area, as illustrated in Figure 4.4: to simulate the gathering of values of $f_{XY}(x,y)$ outside the bounded area, required to obtain $f^{appr}_{XY|D_{XY}}(x,y|d_{xy})$, the integration is performed in the external area, covered by the gray marked areas, as well. Specifically we have the following,

$$X^{appr}_{d_{xy}}(r_x,r_y) = \int_0^{r_x}\int_0^{r_y} f^{appr}_{X,Y|D_{XY}}(x,y|d_{xy})dydx =$$

$$= \int_0^{b_x(d_{xy},r_x,r_y)} \int_{b_y^1(x,d_{xy},r_x,r_y)}^{b_y^2(x,d_{xy},r_x,r_y)} f_{X,Y}(x,y)dydx \qquad (4.5.1)$$

In the following, we simplify the terminology by omitting the d_{xy}, r_x, r_y parameters in the integration bounds and use only the symbols $b_x()$, $b_y^1(x)$ and $b_y^2(x)$. The integration bounds $b_x()$, $b_y^1(x)$, and $b_y^2(x)$ are functions that are specific for each approximation method. In particular, $b_x()$ gives the integration range along the x axis, while $b_y^1(x)$ and $b_y^2(x)$ form the lower and upper bounds of the gray area along the y axis for a specific x. A detailed definition of these integration bounds is given in next subsection.

The Normalized technique is even more simple because we only integrate in the bounded area restricted by the region radii (see the gray marked area in Figure 4.4) and we multiply the result by the normalization coefficient

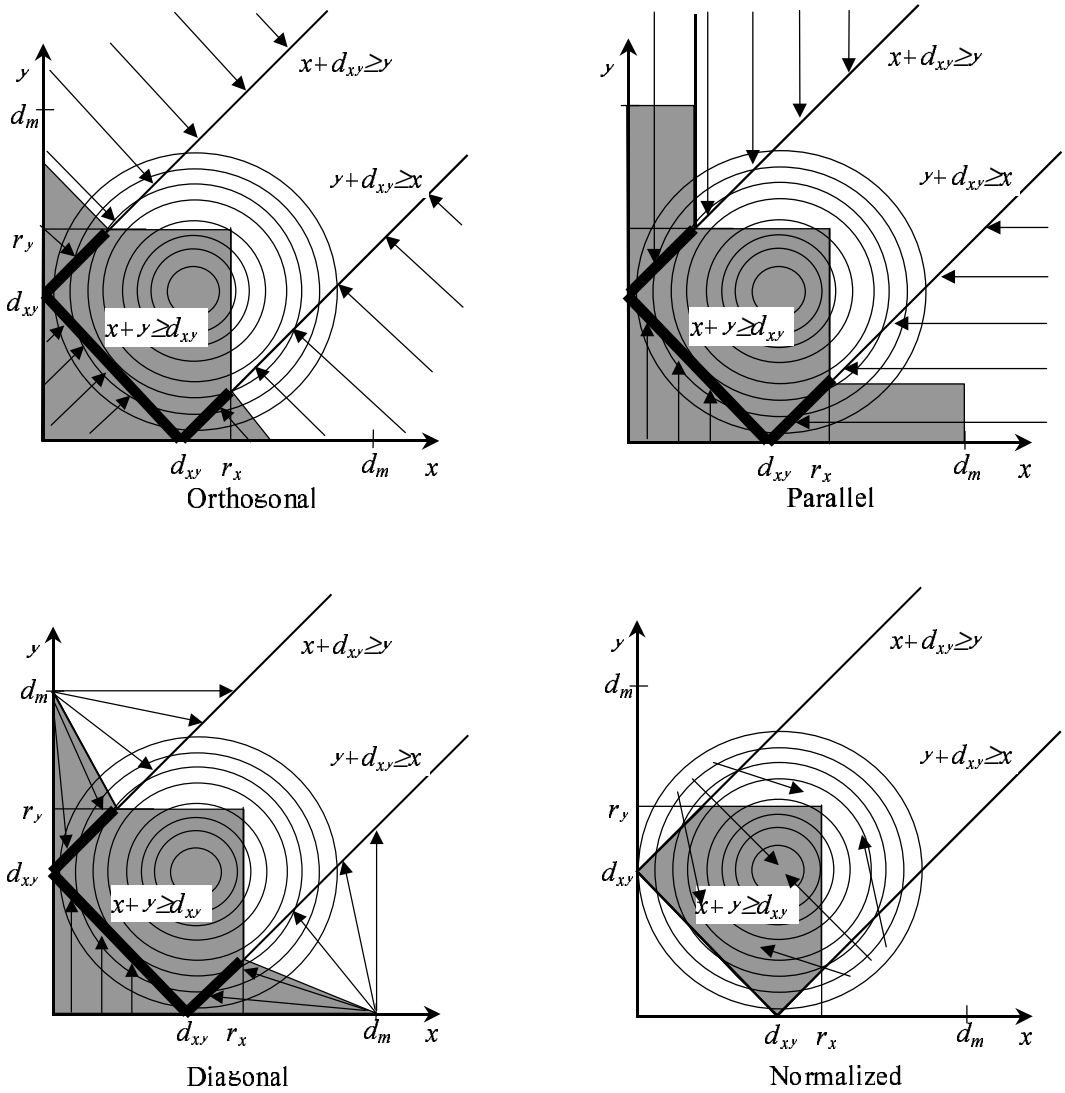


Figure 4.4: The four heuristics proposed to compute region proximity

$$NC(d_{xy}) = 1/(1 - E(d_{xy})), \tag{4.5.2}$$

where $E(d_{xy})$ is the integral of $f_{XY}(x, y)$ over the external area.

Integration bounds

In this section we formally define the bounding functions b_x, $b_y^1(x)$ and $b_y^2(x)$ of the four approximation methods described above. Even though the graphical representation of the integration areas seems to be easy and clear, its formalization is not straightforward, because several special cases should be taken into account to obtain the correct behaviour. Notice that in our simplified formalisation of the problem, the function $f_{XY}(x, y)$ can assume arguments outside the range $[0, d_m]$. In those cases we suppose that the returned value is 0.

We decompose the problem in subcases that can be considered separately – see Figure 4.5 as a convenient graphical reference. First, we distinguish two different cases: (i) $r_y < d_{xy}$ and (ii) $r_y \geq d_{xy}$. In these two situations, we identify some intervals along the x axis.

In case (i), we identify these three intervals:

1. $I'_1 = [0, d_{xy} - r_y)$,

2. $I'_2 = [d_{xy} - r_y, min(d_{xy} + r_y, r_x))$, and

3. $I'_3 = [min(d_{xy} + r_y, r_x), d_m]$.

In case (ii), we identify other three intervals:

1. $I''_1 = [0, min(r_y - d_{xy}, r_x))$,

2. $I_2'' = [min(r_y - d_{xy}, r_x), min(d_{xy} + r_y, r_x))$, and

3. $I_3'' = [min(d_{xy} + r_y, r_x), d_m]$.

Using the intervals defined above, we define the upper bound $b_y^2(x)$. First consider case (i) $r_y < d_{xy}$. When $x \in I_1'$, since the regions do not intersect (that is $d_{xy} - r_x > r_y$), the proximity is 0 (see Figure 4.5-a_3) so the upper bound $b_y^2(x)$ is 0 too. Otherwise, the upper bound is the straight line, which is specific for a method used, passing by point A shown in Figure 4.5-(a_1 and a_2). We call $a(x)$ that straight line. When $x \in I_2'$, the upper bound is always equal to r_y. When $x \in I_3'$, the upper bound is the minimum between the two, method specific, straight lines passing respectively, by point B_1 and B_2. We call $b(x)$ the straight line passing by B_1 and $c(x)$ the one passing by B_2. Figures 4.5-(a_1 and a_2) show two different situations. In the first case, the minimum is $b(x)$, in the other case, the minimum is $c(x)$.

Now consider case (ii) $r_y \geq d_{xy}$, that is there is always an intersection between the two regions. When $x \in I_1''$, the upper bound is the minimum between the two, method dependent, straight lines passing, respectively, by point A_1 and A_2. We call $d(x)$ the line passing by A_1 and $e(x)$ the one passing by A_2. Figures 4.5-(b_2 and b_3) show two different situations. In the first case the minimum is $e(x)$, while in the other case the minimum is $d(x)$. When $x \in I_2''$, the upper bound is always equal to r_y. Since I_3'' is defined exactly as I_3', the same arguments apply as can be seen in Figures 4.5-(b_1 and b_2).

More formally, $b_y^2(x)$ can be defined as follows:

$$b_y^2(x) = \begin{cases} \begin{cases} \begin{cases} a(x) & \text{if } d_{xy} - r_x \leq r_y \\ 0 & \text{elsewhere} \end{cases} & \text{if } x \in I_1' \\ r_y & \text{if } x \in I_2' \\ min(b(x), c(x)) & \text{if } x \in I_3' \end{cases} & \text{if } r_y < d_{x,y} \\ \begin{cases} min(d(x), e(x)) & \text{if } x \in I_1'' \\ r_y & \text{if } x \in I_2'' \\ min(b(x), c(x)) & \text{if } x \in I_3'' \end{cases} & \text{elsewhere} \end{cases} \tag{4.5.3}$$

where $a(x)$, $b(x)$, $c(x)$, $d(x)$, and $e(x)$ are defined for the individual approximation methods as follows.

Let us now specifically define the integration bounds $b_x()$, $b_y^1(x)$, and $b_y^2(x)$ for each specific approximation method, using previous observations.

In the Orthogonal method, to define $b_y^2(x)$, that is the upper bound of the integration area, the required straight lines $a(x)$, $b(x)$, $c(x)$, $d(x)$, and $e(x)$ are the following:

$$\begin{aligned} a(x) &= 2 \cdot r_y - d_{xy} + x \\ b(x) &= 2 \cdot r_y + d_{xy} - x \\ c(x) &= 2 \cdot r_x - d_{xy} - x \\ d(x) &= 2 \cdot r_x + d_{xy} - x \\ e(x) &= 2 \cdot r_y - d_{xy} - x \end{aligned}$$

The lower bound $b_y^1(x)$ of the integration area is defined as

$$b_y^1(x) = \begin{cases} \begin{cases} d_{xy} - 2 \cdot r_x + x & \text{if } d_{x,y} - r_x \leq r_y \\ 0 & \text{elsewhere} \end{cases} & \text{if } r_x < d_{x,y} \\ 0 & \text{elsewhere} \end{cases}$$

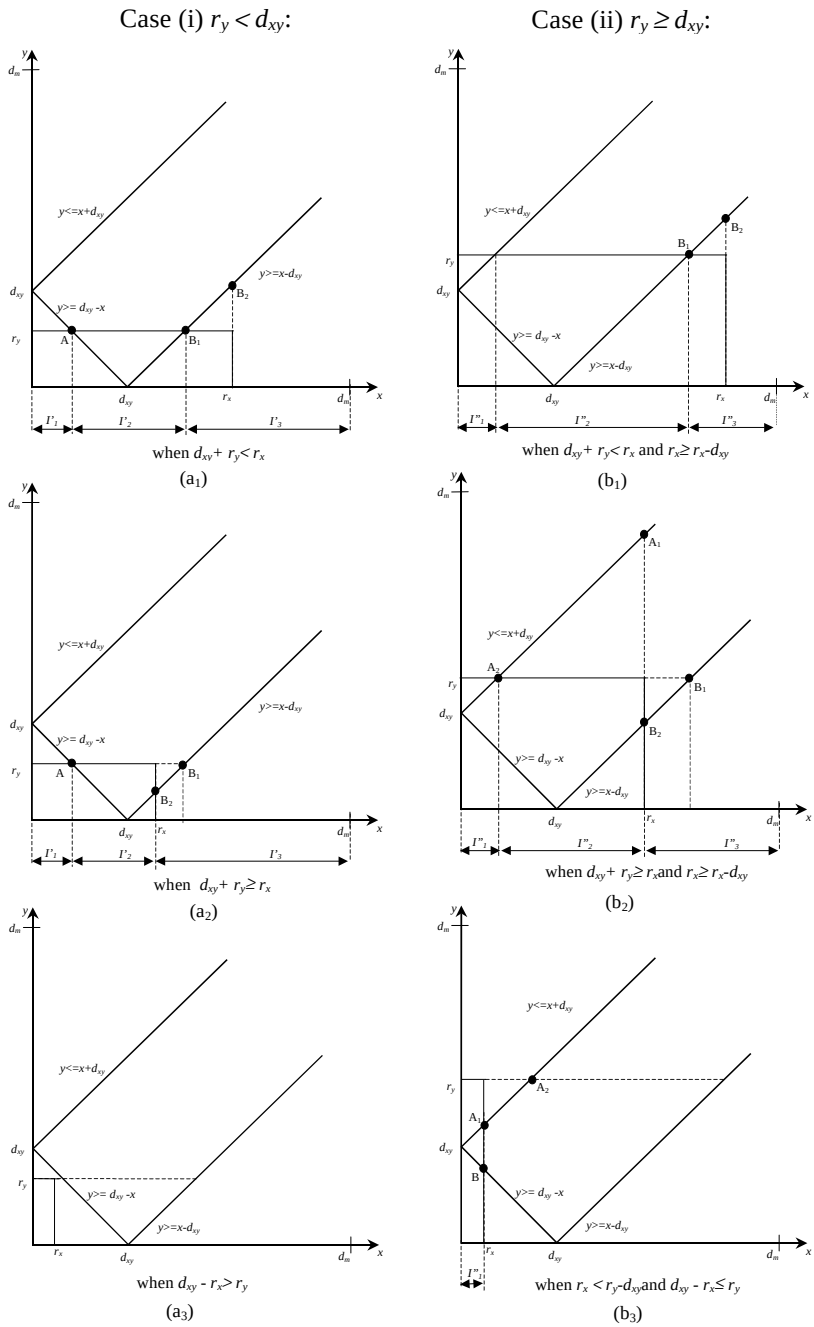


Figure 4.5: Cases to be taken into account when defining bounding functions

That is, when r_x is smaller than d_{xy}, and r_x is such that $d_{xy} - r_x \leq r_y$, the lower bound is the straight line $y = x + d_{xy} - 2 \cdot r_x$ passing by point B – see Figure 4.5-b$_3$. In all other cases the lower bound is 0.

Last, we need to define b_x, that is the range of the integration. If r_x is smaller than d_{xy} the integration is made in the interval $[0, r_x]$, otherwise in the interval $[0, min(r_1, r_2)]$, where r_1 and r_2 are such that, respectively, $b(r_1) = 0$ and $c(r_2) = 0$. We can make it explicit as follows:

$$b_x = \begin{cases} r_x & \text{if } r_x < d_{x,y} \\ min(2 \cdot r_y + d_{xy}, 2 \cdot r_x - d_{xy}) & \text{if } r_x \geq d_{x,y} \end{cases}$$

To define $b_y^2(x)$ for the Parallel method, we use again the outline of equation 4.5.3 and define $a(x)$, $b(x)$, $c(x)$, $d(x)$ and $e(x)$ as follows:

$$\begin{aligned} a(x) = b(x) &= r_y \\ c(x) &= r_x - d_{xy} \\ d(x) = e(x) &= d_m \end{aligned}$$

$b_y^1(x)$ is always 0 so:

$$b_y^1(x) = 0$$

Last, b_x is defined as follows:

$$b_x = \begin{cases} r_x & \text{if } r_x < d_{x,y} \\ d_m & \text{if } r_x \geq d_{x,y} \end{cases}$$

That is, if r_x is smaller than d_{xy}, then the integral is made in the interval $[0, r_x]$, otherwise in the interval $[0, d_m]$.

To define $b_y^2(x)$ for the Diagonal method, we use again the layout of equation 4.5.3 and define $a(x)$, $b(x)$, $c(x)$, $d(x)$ and $e(x)$ as follows:

$$\begin{aligned}
a(x) &= r_y \\
b(x) &= -\frac{r_y}{d_m - r_y - d_{xy}} \cdot x + \frac{r_y}{d_m - r_y - d_{xy}} \cdot d_m \\
c(x) &= -\frac{r_x - d_{xy}}{d_m - r_x} \cdot x + \frac{r_x - d_{xy}}{d_m - r_x} \cdot d_m \\
d(x) &= -\frac{d_m - d_{xy} - r_x}{r_x} \cdot x + d_m \\
e(x) &= -\frac{d_m - r_y}{r_y - d_{xy}} \cdot x + d_m
\end{aligned}$$

Bounding functions $b_y^1(x)$ and b_x for the Diagonal method are defined exactly the same as for the Parallel method so we make no further discussion on them.

Last, we have to consider the Normalized method. Because of the different nature of this method, also the definitions of the bounding functions have a different layout. In particular, there is not need for extending the integration area, so just the intersection between the original constraints ($x \leq r_x$ and $y \leq r_y$) of the integration area and the triangular inequality constraint are needed. The result is the following:

$$b_x = r_x$$

$$b_y^1(x) = |x - d_{xy}|$$

$$b_y^2(x) = x + d_{xy}$$

4.5.2 Computational complexity of the heuristics

The computational cost of Equation 4.5.1 is clearly $O(n^2)$, where n is the number of samples needed for one integration. Since one of our major objectives is efficiency, such a cost is still high. However, we can transform the formula as follows:

$$\int_0^{b_x()} \int_{b_y^1(x)}^{b_y^2(x)} f_{XY}(x,y)dydx = \int_0^{b_x()} \int_{b_y^1(x)}^{b_y^2(x)} f(x)f(y)dydx = \\ = \int_0^{b_x()} f(x) \cdot (F(b_y^2(x)) - F(b_y^1(x)))dx \qquad (4.5.4)$$

Provided that density $f(d)$ and distribution $F(d)$ functions are explicitly maintained in the main memory, Equation 4.5.4 can be computed with complexity $O(n)$. This assumption is realistic even for quite high values on n, so the computational complexities of the Orthogonal, Parallel and Diagonal methods are linear. As far as the Normalized method is concerned, we can see that the normalization coefficient, defined by Equation 4.5.2, is not restricted by specific region radii, thus it only depends on d_{xy}. Such information can also be maintained in the main memory. Consequently, the computational complexity of the Normalized method is also $O(n)$.

4.6 Validating the approaches to the proximity measure

In this section, we investigate the accuracy of the proposed approaches to computing the proximity. Before presenting the simulation results, we first describe the evaluation process and define comparison measures.

4.6.1 Experiments and comparison measures

We computed the *actual proximity* $X_{d_{xy}}^{actual}(r_x, r_y)$ for all data sets described in Section 2.5 as follows. We uniformly chose 100 values of d_{xy}, r_x, and r_y, in the range of possible distances. The proximity $X_{d_{xy}}^{actual}(r_x, r_y)$ was computed for all possible combinations of the chosen values. To accomplish this task, we found for each d_{xy} 400 pairs of objects (O_x, O_y), i.e. the centers of the balls, such that $|d(O_x, O_y) - d_{xy}| \approx 0$. For each pair of objects, we used the predefined values of r_x and r_y to generate ball regions. Then, we only considered pairs of intersecting regions, because non intersecting balls have 0 proximity and no verification is needed. For each pair of balls, we counted the number of objects in their intersection. The actual proximity was finally obtained by computing the average number of objects in the intersection for each generated configuration of d_{xy}, r_x, and r_y and by normalizing (dividing by the total number of objects in the data set) such values to obtain the probability.

We did not consider distances d_{xy} of very low densities, because it was not possible to compute the actual proximity with sufficient precision – the data sets contained only very few objects at such distances.

Once the actual proximity was determined, we computed the approximate proximity for the same values of d_{xy}, r_x, and r_y. The comparison between the actual and the approximate proximity was quantified for each possible configuration as the *absolute error*

$$\epsilon(r_x, r_y, d_{xy}) = |X_{d_{xy}}^{actual}(r_x, r_y) - X_{d_{xy}}^{appr}(r_x, r_y)|$$

An alternative way to evaluate our approaches would be to use the *relative error*, defined as the ratio of the absolute error and the actual proximity. However, our choice for the absolute error can be justified as follows. Suppose that the actual

proximity is almost 0 (e.g. 10^{-5}), while the approximate proximity is exactly 0. In this case, the relative error is 1, i.e. we have a high error. Consider now the opposite case where the actual proximity is zero, and our approximation is 10^{-5}. In this case, the relative error is ∞. However, given the meaning of proximity (see Section 4.1 and Section 4.3), and considering previous examples, we can say that such an approximation is good, because it almost produces the correct results. For example, when it is applied for approximate similarity search, as discussed in section 6.9, regions can be safely pruned if the proximity is 10^{-5} because, statistically, only one object in 100,000 can be lost. This means that the absolute error is a more objective measure, i.e. more suitable for our purposes.

Given the large number of results, we summarized them by computing the average error $\epsilon'_\mu(d_{xy})$ for all pairs of radii at a given distance between the centers, and the average error $\epsilon''_\mu(r_x, r_y)$ for all distances between the ball centers at a given pair of radii, specifically:

$$\epsilon'_\mu(d_{xy}) = Avg_{\mathbf{r_x r_y}}(\epsilon(\mathbf{r_x}, \mathbf{r_y}, d_{xy}))$$

and

$$\epsilon''_\mu(r_x, r_y) = Avg_{\mathbf{d_{xy}}}(\epsilon(r_x, r_y, \mathbf{d_{xy}})).$$

In a similar way, we computed the variance of the error for a given distance d_{xy}:

$$\epsilon'_\sigma(d_{xy}) = Var_{\mathbf{r_x r_y}}(\epsilon(\mathbf{r_x}, \mathbf{r_y}, d_{xy}))$$

The evaluation of ϵ'_μ is used to measure the average error of approximations for specific distances between the ball centers. However, ϵ'_μ alone is not sufficient to correctly judge the quality of the approximation. In fact, it is obtained as the average error for all possible values of r_x and r_y so that some peculiar behaviors may remain hidden.

In this respect, the stability of the error must also be considered. For this purpose, we computed the variance ϵ'_σ. Note that high average errors and small variances may also provide good approximations. To illustrate this, suppose that we want to use the proximity to order (rank) a set of regions with respect to a reference region. The ranking results obtained through the actual and approximate proximity may turn out to be identical even though ϵ'_μ is quite high. In fact, when the variance of error is very small, it means that the error is almost constant, and the approximation somehow follows the behavior of the actual proximity. In this case, it is highly probable that the approximated proximity increases (or decreases) according to the trend of the actual one, thus guaranteeing the correct ordering.

On the other hand, ϵ''_μ represents the average error from a different point of view and complements ϵ'_μ. It is determined for a given pair of radii (r_x, r_y) by varying d_{xy}. This measure offers a finer grained view on the error behavior, since the average is only computed varying the distance d_{xy}.

4.6.2 Discussion on the experimental results

For all data sets, the actual proximity was compared with our techniques and the trivial proximity (defined by Equation 4.2.1). Figures 4.6, 4.7, and 4.8 presents the average error ϵ'_μ and its variance ϵ'_σ. Note that all the approximation methods outperform the trivial one, and the error of the trivial method may even be one order of magnitude higher. The same holds for the variance of the errors. For all the proposed techniques, ϵ'_σ is one order of magnitude smaller than the value obtained with the trivial technique. This implies that the trivial proximity may provide results that significantly differ from the actual proximity. In addition, the proposed methods

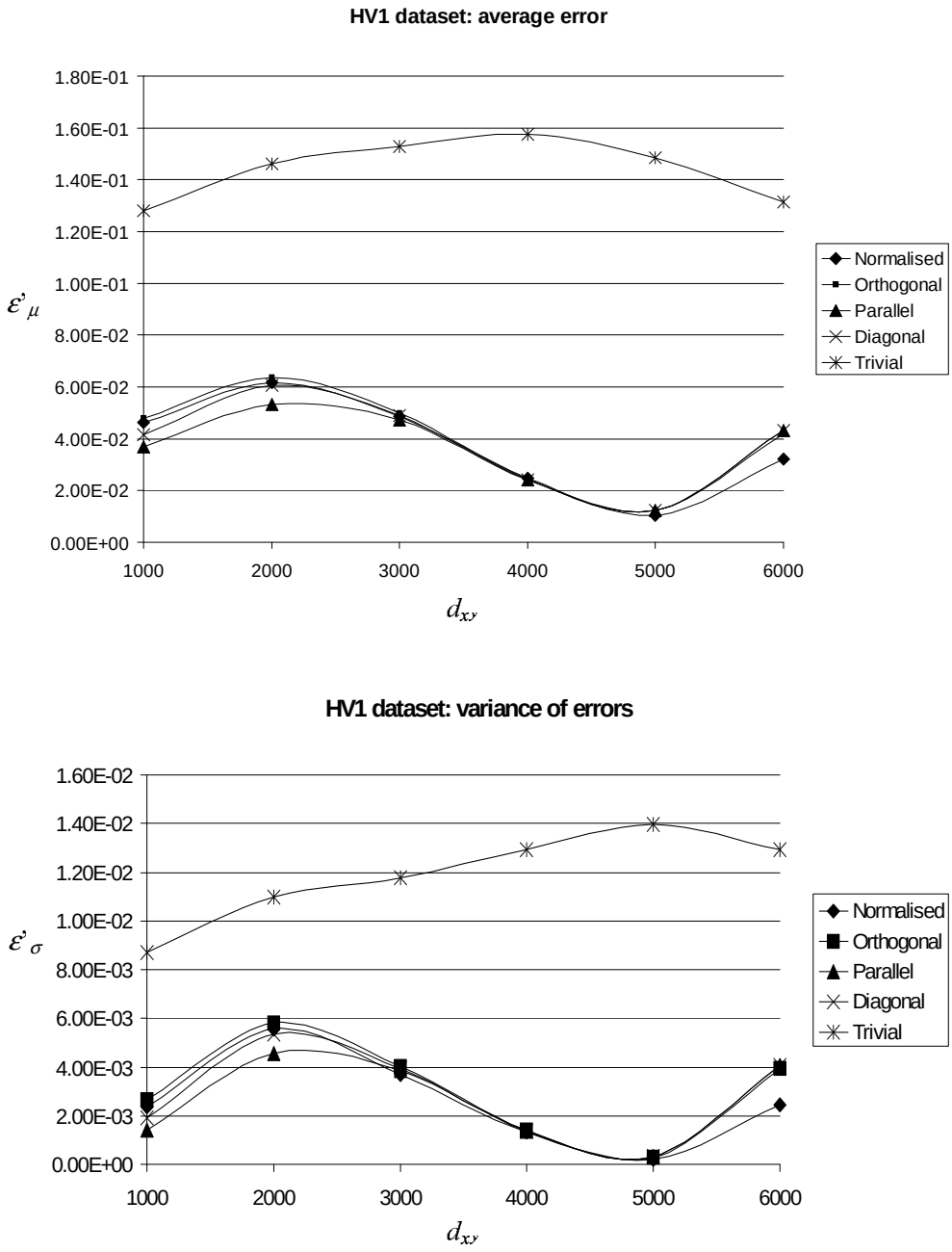


Figure 4.6: Average and variance of errors given d_{xy} in HV1

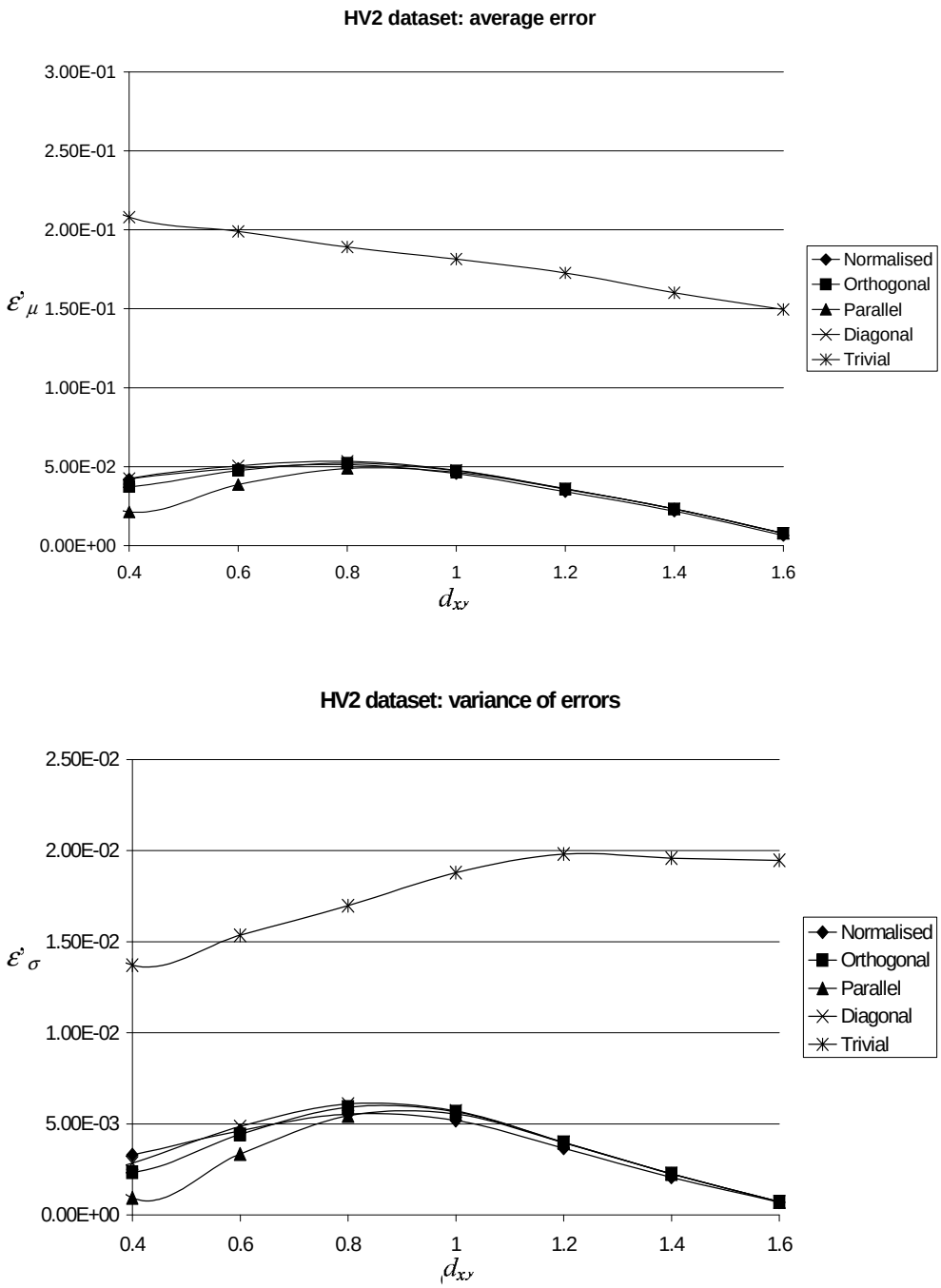


Figure 4.7: Average and variance of errors given d_{xy} in HV2

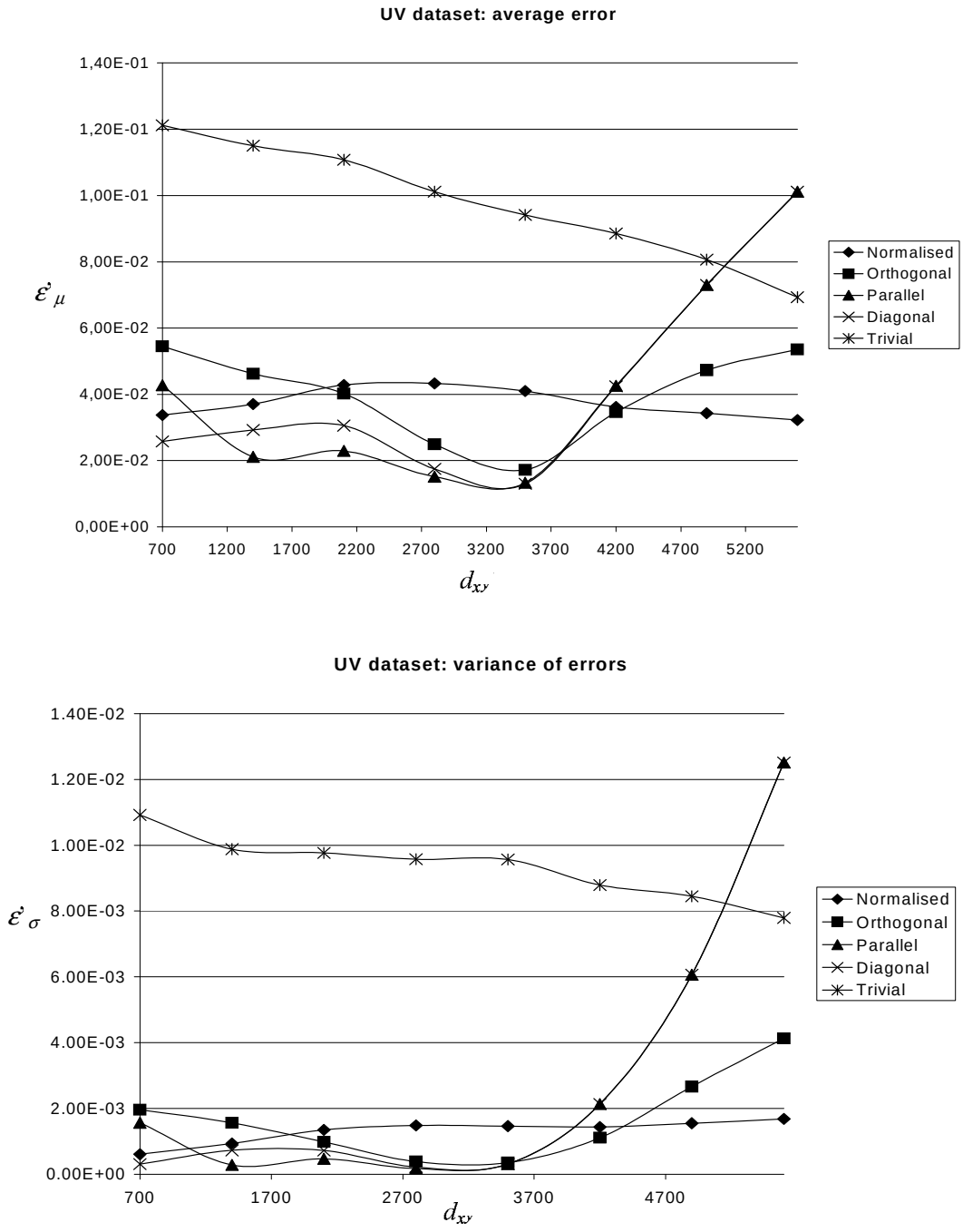


Figure 4.8: Average and variance of errors given d_{xy} in UV

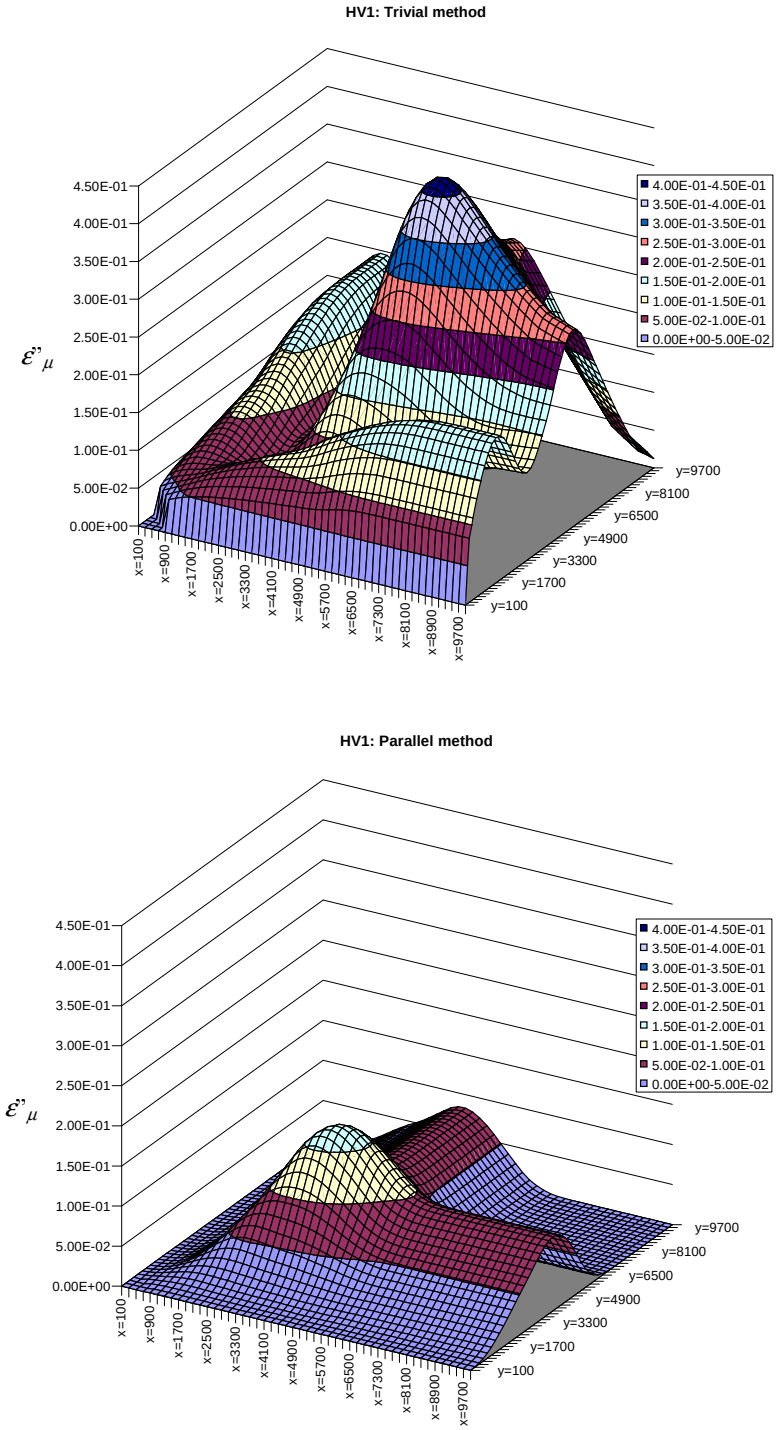


Figure 4.9: Comparison between the errors of the trivial method and the parallel method given r_x and r_y in HV1

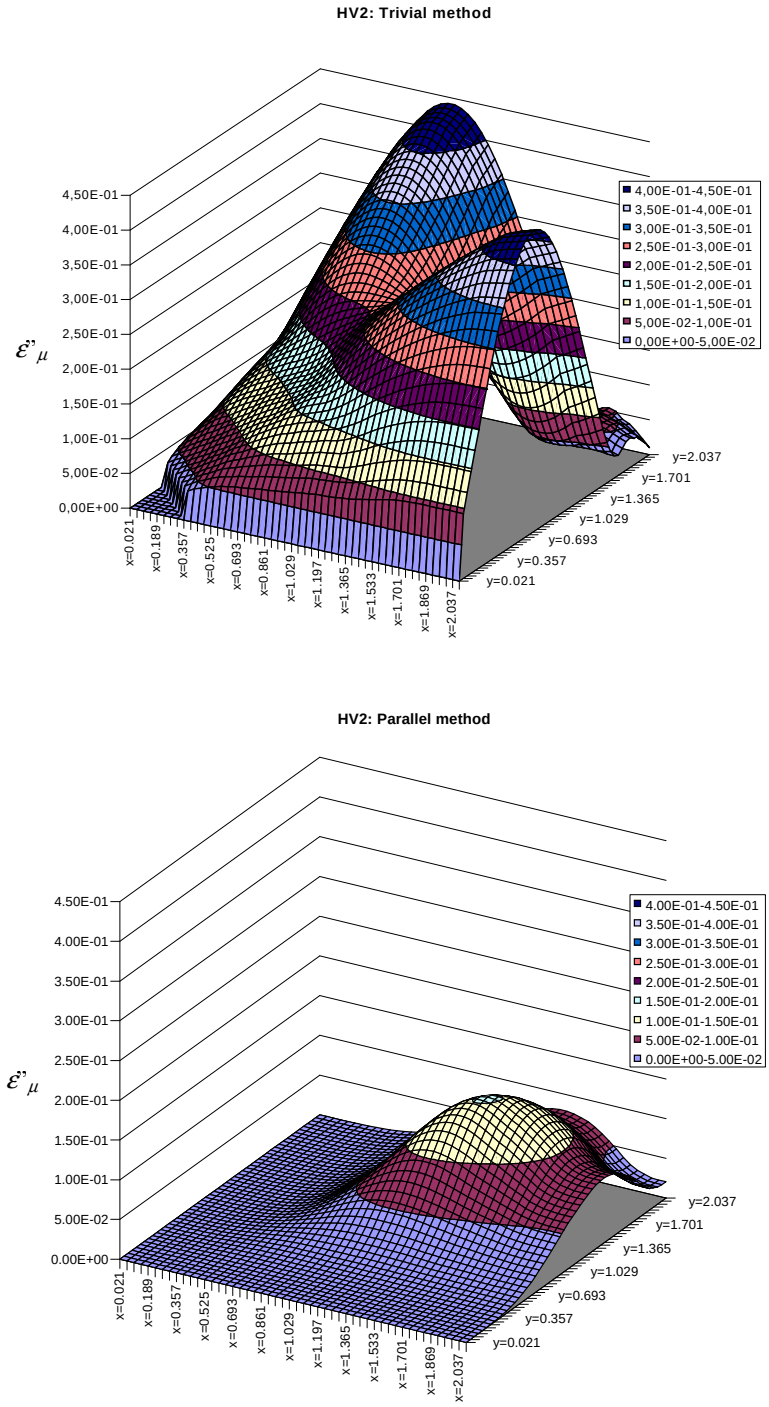


Figure 4.10: Comparison between the errors of the trivial method and the parallel method given r_x and r_y in HV2

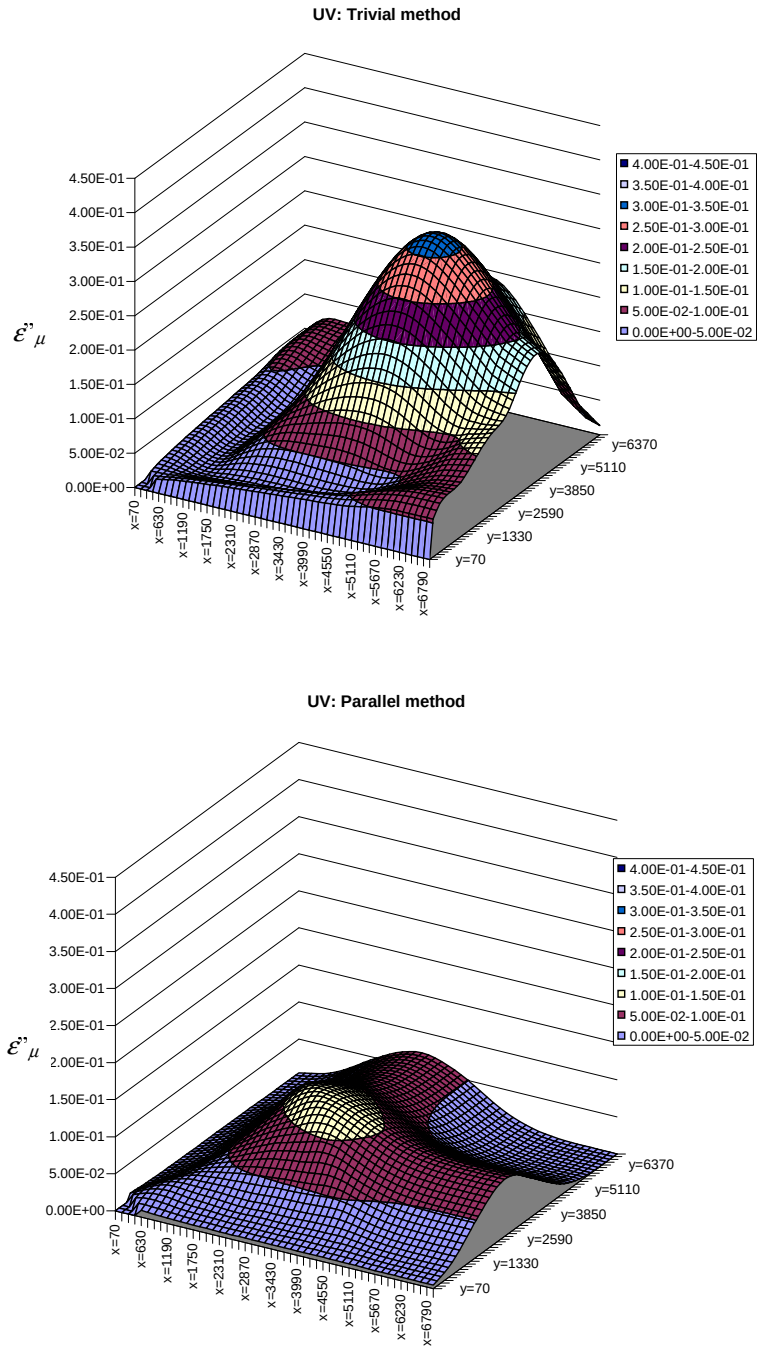


Figure 4.11: Comparison between the errors of the trivial method and the parallel method given r_x and r_y in UV

provide very good and stable results. They have a small variance as well as small errors, so that they can be reliably used in practice.

Although there is not a clear winner among the proposed methods, the Parallel methods gives the best results in the most frequently used range of distances. If we compare the proposed methods for the UV data set, we can see that the Parallel method provides good and stable results. The quality of this method deteriorates, both in terms of ϵ'_μ and ϵ'_σ, for high values of d_{xy}, which are not likely to occur in practice. Here the best results are obtained through the Normalized method. In the HV1 and HV2 data sets we can see again that the Parallel method provides the best performance, though the differences with respect to the other techniques described are even less significant.

Consider now the average error for a given pair of radii ϵ''_μ. For the sake of simplicity, we only compare ϵ''_μ for the Parallel and the trivial method. The results are shown in Figures 4.9, 4.10, and 4.11. As an additional confirmation of the observation that we have made for ϵ'_μ and ϵ'_σ, the error ϵ''_μ for our approximations is again significantly smaller than the one measured for the trivial method. In particular, the error of the trivial method is always quite high, while for a substantial range of r_x and r_y values, the error of the Parallel method is close to 0.