On the Expected Exclusion Power of Binary Partitions for Metric Search

Lucia Vadicamo*[1][0000−0001−7182−7038], Alan Dearle[2][0000−0002−1157−2421], and Richard Connor[2][0000−0003−4734−8103]

[1] Institute of Information Science and Technologies (ISTI), CNR, Pisa, Italy
[2] University of St Andrews, St Andrews, Scotland, UK
lucia.vadicamo@isti.cnr.it, al@st-andrews.ac.uk, rchc@st-andrews.ac.uk

Abstract. The entire history and, we dare say, future of similarity search is governed by the underlying notion of partition. A partition is an equivalence relation defined over the space, therefore each element of the space is contained within precisely one of the equivalence classes of the partition. All attempts to search a finite space efficiently, whether exactly or approximately, rely on some set of principles which imply that if the query is within one equivalence class, then one or more other classes either cannot, or probably do not, contain any of its solutions.

In most early research, partitions relied only on the metric postulates, and logarithmic search time could be obtained on low dimensional spaces. In these cases, it was straightforward to identify multiple partitions, each of which gave a relatively high probability of identifying subsets of the space which could not contain solutions. Over time the datasets being searched have become more complex, leading to higher dimensional spaces. It is now understood that even an approximate search in a very high-dimensional space is destined to require $\mathcal{O}(n)$ time and space.

Almost entirely missing from the research literature however is any analysis of exactly when this effect takes over. In this paper, we make a start on tackling this important issue. Using a quantitative approach, we aim to shed some light on the notion of the exclusion power of partitions, in an attempt to better understand their nature with respect to increasing dimensionality.

Keywords: Metric Search · Binary Partitioning · Exclusion power · Curse of Dimensionality

1 Introduction

We are interested in similarity search spaces of the form (U, d) where U is some universe of objects and d is a distance function $d : U \times U \to \mathbb{R}^+$ satisfying the metric postulates [**?**]. The function d is typically the only meaningful defined operation over U. The task is normally to search a finite (but typically very large) set $S \subset U$ for a small set of objects which are similar to a query object

[*] Corresponding author.

$q \in U$, i.e. to find some small subset $\mathcal{Q}(q,t) = \{s \in S \,|\, d(q,s) \leq t\}$ for some appropriate t. We refer to t as the query threshold. In this paper this definition suffices to encompass both range and nearest neighbour queries and we do not distinguish between them[3].

We use the term *partition* to refer to an equivalence relation defined over U, such that each element is contained in precisely one of the equivalence classes defined by the relation. In the domain of metric search, since d is the only operation available over elements of U, such partitions must be defined in terms of distances to objects identified within the set. For example, for a distinguished value $p \in U$, a simple ball partition may be defined as $\mathcal{F} = \{F_0, F_1\}$

$$\begin{aligned} F_0 &= \{u \in U \,|\, d(p,u) > \tau\} \\ F_1 &= \{u \in U \,|\, d(p,u) \leq \tau\ \} \end{aligned} \tag{1}$$

for some constant value τ.

The processing of similarity queries normally takes place in two distinct phases. In a first *pre-processing* phase, a set of partitions is defined over U. Each element of S is analysed with respect to a number of these, and information about the inclusion of each element within the defined equivalence classes is noted.

In the second *query* phase, the query is analysed with respect to the same set of partitions, at which point deductions may be made about whether solutions to the query are likely to be present in the defined equivalence classes. With reference to the previous example, if $q \in F_1$, it may be possible to reason that any solution to q is more likely to be in F_1 than F_0. The more similar q is to p, the higher the likelihood that this is true. If the space in question is a metric space, and $d(q,p) \leq \tau - t$, then it is impossible for F_0 to contain any values within distance t of the query.

In general, the set of partitions identified at pre-processing time contains the only information which can be used in order to avoid a full scan of the database. In all cases, the choice of partitions is thus critical to the efficacy of the mechanism.

1.1 Binary Partitions

To simplify the domain, we restrict our analysis to binary partitions used in a simplified *exact* search mechanism. To avoid committing the discussion to a particular search mechanism, we consider a notional metric search framework with the following properties:

- A finite set of n binary partitions $\{\mathcal{F}^j\}_{j=1}^n$, where $\mathcal{F}_j = \{F_0^j, F_1^j\}$ is made of two classes, is established at pre-processing time, with respect to a fixed set of m reference objects $p_1, \ldots, p_m \in U$

[3] A nearest neighbour query can be formulated as a range query where the the query threshold is not known in advance but it is set iteratively as the distance to the current k-th nearest neighbour [**?**].

- At query time, the distances from the query q to all reference objects are calculated
- A set of classes which cannot contain any solution to the query is thus established
- All objects which cannot be thus excluded comprise a *candidate result set* whose objects must be tested individually against the query.

Note that many different indexing and filtering mechanisms fall within this general description. In the most general sense, the success of search for solutions to an individual query is related to the following properties of the set of partitions used during the process:

1. the number of available partitions;
2. for each partition, the probability of the distances between the query and the reference objects allowing exclusion of one of the classes of the partition;
3. for any such partition and query, the size of the class which can be thus excluded, and
4. the independence of the set of classes which can be excluded for a given query. For example if all the excluded classes have a common intersection, the value of each one is diminished.

In this article, we address only properties (2) and (3). They are clearly in tension with each other: for example, a partition class which defines only a very small volume of the infinite space is likely to have a high probability of exclusion for an arbitrary query, but is likely to contain only a small number of objects from the finite set. Similarly, a class defining a relatively large volume of the space, thus likely to contain many objects, is less likely to be excluded.

The main contribution of this article is a quantified study of this effect in various metric spaces of different dimensionality.

2 Related Work

Chávez et al. [?] proposed a unifying model to analyse existing indexing algorithms for proximity search by observing that all indexing algorithms for proximity searching consist of building a set of equivalence classes. They remark that every partition of a space induces an equivalence relation, and conversely, every equivalence relation induces a data partitioning. At query time some classes are discarded and the others form a candidate results set that should be exhaustively searched for query solutions. Therefore, the most important tradeoff when designing the data partitioning is to balance the cost of finding the candidate results set (*internal complexity*) and the cost of refining it (*external complexity*). The internal complexity is evaluated as the number of distance calculations d needed to compute the candidate result set C and the external complexity is $|C|$ distance computations. They defined the *discriminative power* of a search algorithm as the ratio of internal complexity to external complexity, which serves as an indicator of the performance fitness of the equivalence relation. Moreover, they observed that two classes of techniques exist based on equivalence relations, namely, pivoting and compact partitions.

Pivoting based techniques rely on building a relation based on the distances between an element and a number of preselected pivots (also called reference points, vantage points, keys). The distances between elements and pivots and between the query q and the pivots are used together with the triangle inequality to filter out elements of the database without actually measuring their distance to q. For example, using ball pivoting the equivalence classes correspond to a family of "rings" or "sphere shells" centered on a pivot. Points within the same sphere shell (i.e., at the same distance from a pivot) are in the same equivalence class. In [**?**] Chávez points out that in this class of algorithms generally improve as more pivots are added.

Compact partitions are based on the class of the points that have some preselected object as their closest center. Thus the partitions induced using this technique correspond to a Delaunay tessellation over the space. Thus using this approach, the universe is divided into a set of spatial zones and complete zones may be discarded by performing a few distance evaluations. Chávez demonstrates that compact partitioning algorithms deal better with high dimensional metric spaces.

In [**?**,**?**] Hetland describes the problem of metric indexing as storing the points from a dataset in some data structure which is later traversed to efficiently extract those points relevant to some query. This data structure is described as a bipartite digraph of points and regions which he defines as a *sprawl*. Each region is defined with respect to a set of source points, called foci or pivots $p_1, .., p_m$. Region membership is defined in terms of distances $x = [d(u, p_1), \ldots, d(u, p_m)]$. Hetland also defines an *ambit* to be a function $f(x)$ (remoteness map) and a threshold or radius r, that describe a partition region (i.e., a partition class). Such ambits are equivalent to the partition functions described in this paper, which also correspond to the *certification functions* introduced by Pestov and Stojmirović [**?**]. In [**?**] Hetland describes a number of different bifocal linear ambits which include ball and hyperboloid remoteness. Using Hetland's classification the 4-point hyperplane partitioning (defined below) is a nonlinear ambit based on a non-metric-preserving power transform. In [**?**] he gives other examples of nonlinear ambits including those based on a Hamacher product and a Cantor function.

3 Quantifying the Value of a Partition Set

3.1 Unifying Partition Functions

To unify the quantitative treatment of different kinds of binary partition with their associated distance constraints, we recently introduced [**?**] the concept of a binary partition $\mathcal{F} = \{F_0, F_1\}$ characterised by a *partition function* $f : U \to \mathbb{R}$ and a *balancing factor* $\tau \in \mathbb{R}$ with the following properties:

1. $F_0 = \{s \in U \mid f(s) > \tau\}$ and $F_1 = \{s \in U \mid f(s) \leq \tau\}$

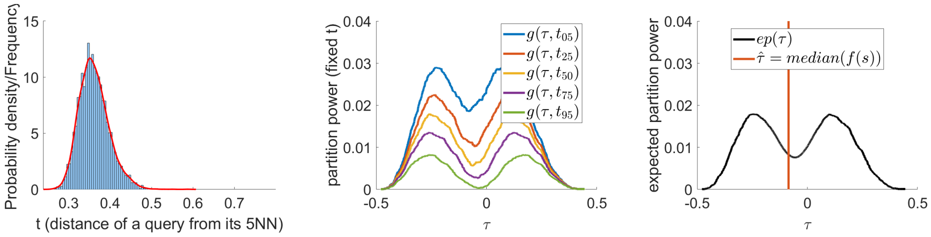
2. $d(s_1, s_2) \geq |f(s_1) - f(s_2)|$ for all $s_1, s_2 \in U$ (*distance lower-bound* property)

Note that if $f(s) = \tau$, then s is on the partition boundary and by convention we include the partition boundary in F_1. Moreover f should be defined in a way so that it both determines the classes F_0, F_1 and provides a rule to estimate a lower-bound of the actual distance between two data points. The lower-bound is used to derive the exclusion rules used at query time. Specifically, given a query q and a query threshold t, then we have that

- if $f(q) \leq \tau - t$ then F_0 can be excluded
- if $f(q) > \tau + t$ then F_1 can be excluded

This characterisation provides us with a unified framework to describe the most common metric binary partitioning principles, namely *ball partitioning* [**?**,**?**], *generalised hyperplane partitioning* [**?**,**?**], and *4-point hyperplane partitioning* [**?**,**?**], together with their exclusion rules. Specifically, as proved in [**?**], we have that

- a *ball partitioning* given a pivot p and a radius r is characterised by the function
$$f_{\text{Ball}}(s) = d(s, p), \qquad \forall\, s \in U$$
and the balancing factor $\tau = r$;
- a *generalised hyperplane partitioning* of the form
$$\begin{aligned} F_0 &= \{s \in U | d(s, p_1) - d(s, p_0) > \alpha\} \\ F_1 &= \{s \in U | d(s, p_1) - d(s, p_0) \leq \alpha\} \end{aligned} \tag{2}$$
for two given pivots p_0 and p_1 and offset α, is characterised by the function
$$f_{\text{Hyp}}(s) = \frac{d(s, p_1) - d(s, p_0)}{2}, \qquad \forall\, s \in U \tag{3}$$
and balancing factor $\tau = \alpha/2$;
- a *4-point hyperplane partitioning*
$$\begin{aligned} F_0 &= \{s \in U | d(s, p_1)^2 - d(s, p_0)^2 > \alpha\} \\ F_1 &= \{s \in U | d(s, p_1)^2 - d(s, p_0)^2 \leq \alpha\} \end{aligned} \tag{4}$$
that can be characterised by the function
$$f_{\text{4pHyp}}(s) = \frac{d(s, p_1)^2 - d(s, p_0)^2}{2d(p_0, p_1)}, \qquad \forall\, s \in U \tag{5}$$
and balancing factor $\tau = \alpha/2d(p_0, p_1)$. This kind of partition is valid only on the large class of Supermetric Spaces meeting the 4-point property [**?**]. The partition boundary can be visualised as a hyperplane in a 2D Euclidean space obtained using the nSimplex projection [**?**] to transform the data; with the hyperplane being orthogonal to the line containing the two pivots in the 2D Euclidean space. Moreover, if $\tau = \alpha = 0$ then the classes F_0 and F_1 are exactly the same as the generalised hyperplane partitioning above, but the 4-point property [**?**,**?**], rather than the triangle inequality, is used for estimating the distance lower-bound.



(a) Query threshold distances (b) Exclusion power (c) Expected exclusion power

Fig. 1: *8-dimensional Euclidean dataset*: Example of typical query threshold distances (a), power graphs for a 8-dimensional Euclidean dataset (b), and expected exclusion power (c). The left-hand figure shows the distribution of the fifth nearest neighbour distances for a set of 5000 queries. The middle figure show the exclusion power graphs over τ for five representative t values (0.05, 0.25, 0.5, 0.75, 0.95-th percentiles of the query threshold distribution) in the case of a generalised hyperplane partitioning. The right-hand figure shows the Expected exclusion power over τ

We define the *balance ratio* of a binary partition $\{F_0, F_1\}$ of the finite search set S as the ratio of the smaller of $|F_0|$ or $|F_1|$ to $|S|$, giving a value in the range $[0, 0.5]$ where a higher value means a more even balance ratio. Note that when changing the balancing factor τ, the partition boundary moves and thus its balance ratio changes as well.

This unification (f, τ) allows the characterisation of the *balance ratio* and *power* of a partition as the value of τ is altered, as shown in the next Section.

3.2 Partition Exclusion Power

We introduce the notion of partition *exclusion power* to represent the amount of exclusion possible for a partition characterised by some given value of τ and a function f. In essence, the power of a partition is an estimate of the probability of being able to deduce that $d(q, s) > t$, for some distance t, for arbitrarily selected $q \in U$ and $s \in S$.

For the remainder of this article, we use the assumption that the distribution of query and data within the sampled spaces are equivalent. This is probably a reasonable assumption in most metric query scenarios, although there are likely to be specialist examples where it is not the case. The same analysis may be performed whenever the distribution of both query and data can be characterised, whether they are equivalent or not.

In [**?**], for a range query $\mathcal{Q}(q, t)$, we defined the exclusion power of the partition $\mathcal{F} = \{F_0, F_1\}$ as the probability of excluding one element s on the basis of the data partition to which it belongs:

$$P(s \in F_0) \cdot P(\mathcal{Q}(q, t) \subset F_1) + P(s \in F_1) \cdot P(\mathcal{Q}(q, t) \subset F_0) \qquad (6)$$

which can be rewritten in terms of f and τ as

$$P(f(s) > \tau) \cdot P(f(q) \leq \tau - t) + P(f(s) \leq \tau) \cdot P(f(q) > \tau + t) \tag{7}$$

If $\texttt{CDF}(x)$ is the cumulative distribution function of $f(s)$ for $s \in S$ (assuming that the distribution is the same for data and query points, as noted above) then the exclusion power can be expressed as

$$g(\tau, t) = (1 - \texttt{CDF}(\tau)) \cdot \texttt{CDF}(\tau - t) + \texttt{CDF}(\tau) \cdot (1 - \texttt{CDF}(\tau + t)) \tag{8}$$

This provides a mechanism for estimating exclusion power of a partition for a fixed τ and query threshold t. Therefore, to understand the effect of different values of τ an *exclusion power graph* may be constructed which is plotted across the range of τ for a fixed value of t. This allows the optimum value of τ to be deduced for a range query with threshold t. The exclusion power graph is dependent on the query threshold. Thus queries with different thresholds will result in different power graphs. Figure **??** shows the resultant power graphs for various thresholds over eight dimensional euclidean data as described in the caption.

To define a *general exclusion power measure* independent from the specific query threshold, in this paper we propose to use the *expected partition power*:

$$ep(\tau) = \int g(\tau, t)h(t)dt \tag{9}$$

where $h(t)$ is the probability density function associated with the query threshold distribution (e.g., the red curve in Figure **??**). In Figure **??**, we show the expected partition power graph for the same 8-dimensional Euclidean data used above.

The exclusion power defined here is closely related to the concept of *discriminative power* (i.e., the ratio of internal complexity to external complexity) defined by Chávez in [**?**]. Adjusting the τ values thus changes the discriminative power. In this paper we show how exclusion power may be used to optimise τ so that for the same internal complexity we minimise the external complexity i.e. we find the τ that optimises the discriminative power.

4 Power Analysis in High(er) Dimensional Data

It is clear that if a partition has a balance ratio of 0 (i.e., all the data objects are in the same partition class) then it is of no value in terms of exclusion, whereas a value of 0.5 is unlikely to be optimal in a high dimensional space. In fact, it has long been known, if only as a rule of thumb, that balanced tree-structured indexes lose their performance as dimensionality increases, and unbalanced structures perform better. For example, the List of Clusters [**?**,**?**] is known to perform better than a Balanced Vantage Point Tree [**?**] in *higher* dimensions, although we lack a formal definition of the meaning of *higher* in this context. Here, we investigate this phenomenon from a new point of view by using the expected exclusion power estimation.

For a partition $\{F_0, F_1\}$, defined by a pair (f, τ), a set of *witness* data values may be used to calculate approximations of the different values of balance ratio and expected partition power (Eq. **??**) varying τ. Note that if τ is selected as the median of $\{f(s), s \in S\}$ then the partition classes are balanced (i.e., the balance ratio is 0.5). Therefore, if an exclusion occurs, half of the dataset will be excluded. Moving τ from the median value will produce partitions with a different balance ratio. To understand the effect of different values of τ the expected exclusion power graph may be constructed and optimum value(s) of τ can be deduced.

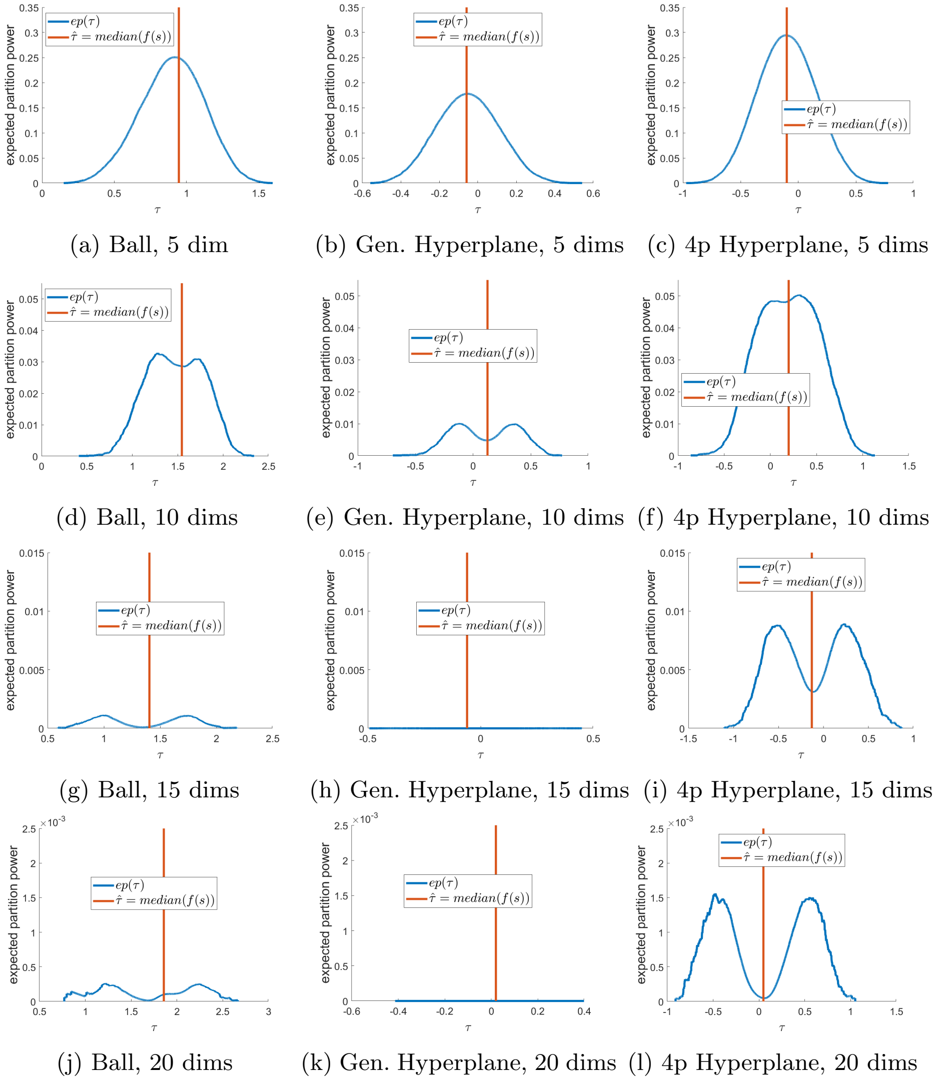
Figure **??** shows the change in the power graphs as dimensionality increases for a ball partitioning, a generalised hyperplane partitioning and a 4p hyperplane partitioning. The plots are for Euclidean data of dimensions 5, 10, 15, 20[4]. In the low dimensional settings we can observe that the maximum power is achieved when the partition is balanced, i.e. τ is equal to the median of the $f(s)$ values. By contrast, as the dimension of data increases, choosing the median value will work very badly. As can be seen in Figure **??** such a value is unlikely to result in any successful exclusions. Therefore, for high(er) dimensions a better strategy is to pick two values for τ corresponding to the two peaks in the power distributions. It also interesting to note that as dimension increases, we expect that no exclusion is possible using ball and generalised hyperplane partitioning whatever τ value is chosen, confirming the well know curse of dimensionality phenomenon [**?**,**?**]. This effect is also visible in the case of 4p hyperplane partitioning with Euclidean dimensions bigger than 20.

These diagrams explain behaviour observed by many researchers into metric search, that choosing unbalanced indexing structures often works better for higher dimensional data. Moreover, it also confirms the observations made in [**?**,**?**] regarding the better distance bounds that can be obtained using the 4-point property instead of the triangle inequality.

5 Experimental Validation

In this section we confirm the observations deduced from the analysis of the expected power graph experimentally. To illustrate we report the results for 15 dimensional Euclidean data using 10K data points and 100 random pivots. For each pivot pair (p_i, p_j) we build a 4point hyperplane partition $\mathcal{F}_{ij}$ characterised by the function $f_{ij} = \frac{d(s,p_j)^2 - d(s,p_i)^2}{2d(p_i,p_j)}$ and balancing factor τ_{ij}. At query time, a candidate result set is build by considering the intersection of all the partition classes that cannot be excluded from the search on the basis of the distance lower-bound property only (see Section **??**). Lastly the candidate set is refined using the actual distance function d to establish the final (exact) result set. Therefore the size of the candidate result set is equivalent to the percentage of the data that must be accessed to answer a query.

[4] All results in this article are derived using randomly generated uniformly distributed Euclidean data in different dimensions as stated. All code is available on request from the authors.



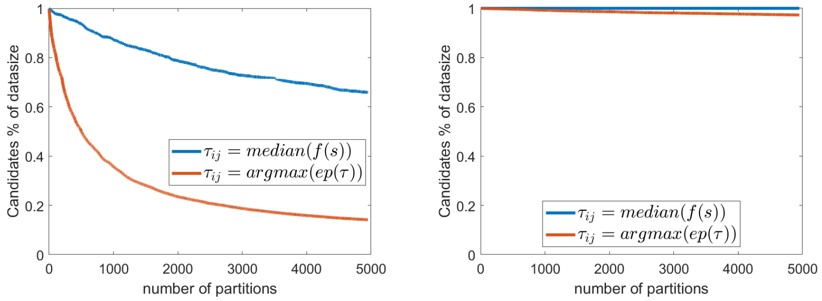
(a) Ball, 5 dim (b) Gen. Hyperplane, 5 dims (c) 4p Hyperplane, 5 dims

(d) Ball, 10 dims (e) Gen. Hyperplane, 10 dims (f) 4p Hyperplane, 10 dims

(g) Ball, 15 dims (h) Gen. Hyperplane, 15 dims (i) 4p Hyperplane, 15 dims

(j) Ball, 20 dims (k) Gen. Hyperplane, 20 dims (l) 4p Hyperplane, 20 dims

Fig. 2: Expected powers for Euclidean data at dimensions 5, 10, 15, 20, for ball partitioning (left), generalised hyperplane partitioning (middle), 4p hyperplane partitioning (right). The expected power was evaluated using 100 queries over 10K witness data points. Two pivots p_1, p_2 were randomly selected for each dataset; p_1 is used to build the ball partition, both p_1 and p_2 are used for the hyperplane partitions.



(a) 4p Hyperplane Partitioning (b) Generalised Hyperplane Partitioning

Fig. 3: Size of candidate set as proportion of whole for τ_{ij} set to the median of $f_{ij}(s)$ values and the τ_{ij} values that maximises the expected partition powers (15D Euclidean data).

Figure **??** plots the size of the candidate set as a proportion of the entire dataset using different approaches to select the τ_{ij} values. The top (blue) curve shows the performance when all the partitions are balanced, i.e., for each partition defined by the pair (f_{ij}, τ_{ij}) the τ_{ij} is set to be the median of $f_{ij}(s)$ values. The bottom (orange) curve shows τ_{ij} set to maximise the expected power (see Equation **??**) estimated on a small set of 2,000 witness points using 100 random queries (different from those used at test time). The x-axis shows all the $\binom{100}{2}$ partitions $\mathcal{F}_{ij}$ even although a small subset of these take part in the exclusions. In both cases 500 test queries were considered and the average percentage of data accessed to answer a single query was computed and plotted in the y-axis. From this plot a clear difference can be seen in the exclusion power with τ_{ij} set to have balanced partitions and that with the τ_{ij} values set to maximise the partition powers. The balanced version manages to exclude very little data whereas the powered version excludes more than the 85% of the data. For completeness, in Figure **??** we shows the results also for generalised hyperplane partitioning, i.e. using $f_{ij} = (d(s, p_j) - d(s, p_i))/2$. Note that it does not result in any exclusions - i.e. the candidate set size is about 100% of the dataset being queried, as predicted by the expected power graph in Figure **??**.

Figure **??** only shows the exclusion for a single maximum power for each pivot pair. However, as shown in Figure **??**, the expected power graph for 15 dimensional data results in two power peaks (and consequently two different optimal τ_{ij} can be selected). In practice this results in two partitions being created for each pivot pair in the case of hyperplane partitioning. The plot shows the exclusions possible when a single optimal value and both optimal values are used. With 100 pivots and using one or two optimal τ_{ij} values for each pivot pair results in 4,950 and 9,900 partitions respectively. Figure **??** shows the size of the candidate set as proportion of whole when partitions derived from a single and both the optimal τ_{ij} values are used. In this plot the exclusions derived from the common partitions are in plotted corresponding to the leftmost part of the x-axis

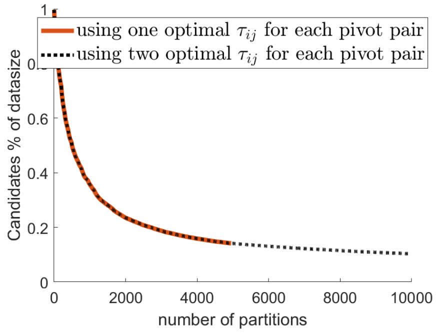


Fig. 4: Size of candidate set as proportion of whole when one optimal τ or two optimal τ values for each pivot pair are selected to maximise the expected partition powers (15D Euclidean data).

Dims	**Balanced Partitions**		**Maxpower Partitions**	
	Part. Activated	*Candidates*	*Part. Activated*	*Candidates*
5	44.62%	1.43%	44.63%	1.25%
10	3.11%	6.53%	13.76%	2.37%
15	0.05%	65.80%	6.57%	14.15%
20	0.0002%	99.60%	3.00%	49.63%

Table 1: Average percentage of partitions activated and candidate set size for 5, 10, 15 and 20 dimensional Euclidean data, 100 pivots and 500 queries over 10,000 data points

resulting in a common exclusion curve. As can be seen, the 4,950 extra partitions available when two peaks are used result in (some) more exclusion. The size of the candidate set as a fraction of the total data when the partitions derived from both power peaks are used is 10.37%, in other words 89.63% exclusion is achieved.

5.1 The Relationship Between Activated Partitions and Exclusions

We say that a partition is "activated" for a query if using the distance lower-bound property is possible to exclude one of the classes of the partition.

Table **??** shows the percentage of partitions that are activated for queries over 5, 10, 15 and 20 dimensional data. In each experiment 500 queries are executed with 100 pivots (4,950 partitions) over 10,000 data points. As before, the two columns correspond to the cases when all τ_{ij} have been set to have balanced partitions (left) and that with the τ_{ij} values set to maximimse the expected power (right). The data shown is the average over the queries. Two numbers are

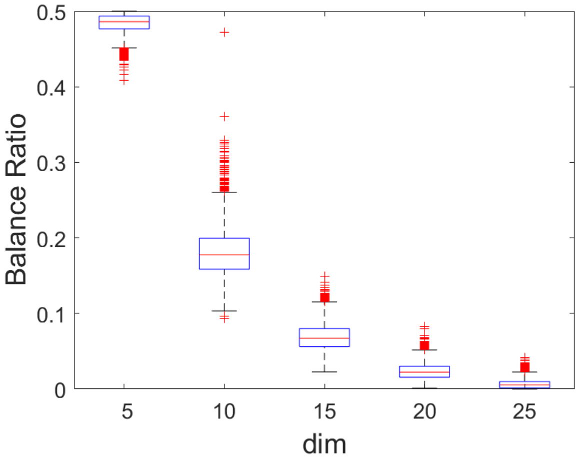


Fig. 5: Balance ratio for increasing dimensions using τ_{ij} values that maximise the expected power

presented for each experiment: the percentage of partitions that are activated and the size of the candidate set as a proportion of the dataset being queried.

As can be seen in the table, the number of partitions that are activated at query time are considerably different both in terms of the dimension of the data and the techniques used to select all the τ values.

In general, balanced partitions perform noticeably worse than the powered partitions and the number of partitions that are activated drops dramatically as the Euclidean dimension increases. Whilst the number of partitions activated also drops when the power is maximised, enough partitions are activated to permit approximately 50% of the data to be excluded in the case of 20 dimension and the partitions set to maximise the expected power even when a single power peak is employed.

We also observed, as shown in Figure **??**, that choosing the best τ values results in increasingly un-balanced partitions. Moreover, adding more partitions often does not serve to substantially increase the number of exclusions. We believe that this effect is caused by a lack of *independence* of the objects in the activated partition classes.

6 Conclusions

In this paper we have presented a generalised treatment of exclusion power for binary partitions. The model abstracts over the partition type and we have shown its application to ball partitions, generalised hyperplane partitions and 4-point partitions.

Exclusion power explains the well known differences in the number of exclusions that are possible with respect to both the dimensionality of the data and partition balance ratio.

In addition understanding how to maximise the possibility of exclusion, power diagrams also serve to indicate the probability of exclusions occurring. The understanding the probability of exclusion power determines if a dataset can be usefully queried at all using an exact metric search, i.e. if the size of candidate set is a small fraction of the size of the total dataset. This is useful in its own right since it may be applied independently of any particular algorithm to establish the amount of exclusion that is potentially possible.

In the cases where a reasonable exclusion rate can be achieved it can be used to give an indication of the number of exclusions zones that are necessary to achieve exclusion.

Additionally we have shown that by adjusting the f function and the τ values, the exclusion power may be dramatically increased in some cases. When combined with 4-point exclusion we have observed that sometimes exclusion rates rise from zero to a respectable percentage of the overall dataset.

7 Future Directions

In this paper we have attempted to shed some light on the notion of partitions in general in order to better understand their nature with respect to increasing dimensionality and their ability to exclude. Although this paper has established some general mechanisms to permit reasoning about the nature of partitions and how their construction contributes to exclusion in metric search there is clearly much more work to be done. In particular, we have only touched on the nature of the independence of partitions. Clearly the amount of exclusion that is possible, the independence of the partitions and their power are linked. We are currently investigating this issue but the work is at an early stage.

Acknowledgements This work was partially funded by AI4Media - A European Excellence Centre for Media, Society, and Democracy (EC, H2020 n. 951911) and by Economic & Social Research Council, ADR UK Programme ES/W010321/1.