Multiple Radii DisC Diversity: Result Diversification based on Dissimilarity and Coverage

MARINA DROSOU and EVAGGELIA PITOURA, University of Ioannina

Recently, result diversification has attracted a lot of attention as a means to improve the quality of results retrieved by user queries. In this paper, we introduce a novel definition of diversity called DisC diversity. Given a tuning parameter $r$, which we call radius, we consider two items to be similar if their distance is smaller than or equal to $r$. A DisC diverse subset of a result contains items such that each item in the result is represented by a similar item in the diverse subset and the items in the diverse subset are dissimilar to each other. We show that locating a minimum DisC diverse subset is an NP-hard problem and provide algorithms for its approximation. We extend our definition to the multiple radii case, where each item is associated with a different radius based on its importance, relevance or other factors. We also propose adapting DisC diverse subsets to a different degree of diversification by adjusting $r$, i.e., increasing the radius (or zooming-out) and decreasing the radius (or zooming-in). We present efficient implementations of our algorithms based on the M-tree, a spatial index structure, and experimentally evaluate their performance.

Categories and Subject Descriptors: Information systems [Retrieval models and ranking]: Information retrieval diversity, Top-k retrieval in databases

General Terms: Information retrieval diversity, Top-k retrieval in databases

Additional Key Words and Phrases: Diversity, Search Process, Information filtering, Algorithms, Top-k

ACM Reference Format:
Marina Drosou and Evaggelia Pitoura. 2014. Multiple Radii DisC Diversity: Result Diversification based on Dissimilarity and Coverage. ACM Trans. Embedd. Comput. Syst. V, N, Article A (January YYYY), 45 pages. DOI: http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

Result diversification has attracted considerable attention as a means of enhancing the quality of the results presented to users (e.g., [Vee et al. 2008; Ziegler et al. 2005]). Consider, for example, a user who wants to buy a camera and submits a related query. A diverse result, i.e., a result containing various brands and models with different pixel counts and other technical characteristics is intuitively more informative than a homogeneous result containing only cameras with similar features.

There have been various definitions of diversity; they can be roughly categorized [Drosou and Pitoura 2010] as based on: (i) content (or similarity), i.e., items that are dissimilar to each other (e.g., [Ziegler et al. 2005]), (ii) novelty, i.e., items that contain new information when compared to what was previously presented (e.g., [Clarke et al. 2008]) and (iii) coverage, i.e., items that belong to different categories or topics (e.g., [Agrawal et al. 2009]).

This is a preliminary release of an article accepted by ACM Transactions on Database Systems. The definitive version is currently in production at ACM and, when released, will supersede this version.

Copyright 2014 by the Association for Computing Machinery, Inc. Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Publications Dept, ACM Inc., fax +1 (212) 869-0481, or permissions@acm.org.
Most previous approaches to diversification rely on assigning a diversity score to each item and then selecting as diverse either the \( k \) items with the largest score for a given \( k \) (e.g., [Angel and Koudas 2011; Catallo et al. 2013]), or the items with score larger than some predefined threshold (e.g., [Yu et al. 2009]). Diversity is often combined with other ranking criteria, such as relevance (e.g., [Szpektor et al. 2013]). In this case, the selected items must be both highly relevant individually and diverse as a set. The two criteria are often conflicting with each other, since the items most relevant to a specific user need are often similar to each other. A number of different approaches have been proposed to achieve a trade-off between high relevance and high diversity (e.g., [Carbonell and Goldstein 1998; Gollapudi and Sharma 2009]), usually based on assigning weights to the two factors, resulting again in a unified score and a corresponding top-\( k \) or threshold problem.

In this paper, we address diversity through a different perspective. In contrary to previous approaches, we aim at selecting a representative subset of the result set that contains items that are both dissimilar with each other and cover the whole result set. Let \( \mathcal{P} \) be the set of items in a query result, \( d \) a distance metric, and \( r \) a real number, that we call radius. We consider that two items \( p_i, p_j \) in \( \mathcal{P} \) are similar to each other, if and only if, \( d(p_i, p_j) \leq r \). We also say that they cover each other. Our goal is to select a subset \( S \) of \( \mathcal{P} \), such that (i) for each item \( p_i \in \mathcal{P} \), there is at least one item \( p_j \in S \), such that \( d(p_i, p_j) \leq r \), and (ii) for any pair of items, \( p_i, p_j \in S \), it holds \( d(p_i, p_j) > r \). The first condition ensures that all items in \( \mathcal{P} \) are represented, or covered, by at least one similar item in the selected subset. The second condition ensures that the selected items are dissimilar to each other. We call the set \( S \) Dissimilar and Covering subset or \( \text{DisC} \) diverse subset.

A novel aspect of our approach is that, instead of specifying a required size \( k \) of the diverse set or a threshold, our tuning parameter \( r \) explicitly expresses the degree of diversification and determines the size of the diverse set. Increasing \( r \) results in a smaller, more diverse subset, while decreasing \( r \) results in a larger, less diverse subset. We call these operations zooming-out and zooming-in respectively. At one extreme, a radius equal to the diameter of the result set gives a singleton diverse subset. At the other extreme, a radius smaller than the smallest pairwise distance in the result set gives a diverse subset equal to the original result set.

Since there may be more than one \( \text{DisC} \) diverse subset of a result set, for attaining a concise representation, we aim at selecting the one with the smallest number of items, termed \( \text{Minimum} \ r-\text{DisC} \) diverse subset. Furthermore, when the items in the result set are associated with weights, besides the size, we take weights into account and select a \( \text{Minimum Weighted} \ r-\text{DisC} \) diverse subset. When all weights are equal, a Minimum Weighted \( r \)-\( \text{DisC} \) diverse subset reduces to a Minimum \( r \)-\( \text{DisC} \) diverse subset. As an example, Figure 1(a) and Figure 1(b) depict the selected diverse subset for a set of items representing major cities in our world, without and with weights respectively. In this example, weights were set based on population.

Further, we would like to allow different areas of the result set to contribute more or less items to the diverse subset. To this end, we extend the definition of \( \text{DisC} \) diverse subsets to allow each item \( p_i \) to be associated with a different radius \( r(p_i) \). The radius of an item may depend on its relevance to the query, on the density of its surrounding area, or other factors. Figure 1(c) depicts the selected subset of the world cities example in the case of multiple radii, where a smaller radii is used for cities in Europe, resulting in more items being selected from this area.

We formalize the problem of locating minimum \( \text{DisC} \) diverse subsets as an independent dominating set problem on graphs. In the case of a single radius, the correspond-
Fig. 1: DisC diversity: (a) no weights, single radii, (b) weights, single radii, and (c) no weights, multiple radii. Selected items are shown as solid circles with size proportional to their weight. (Non solid) circles denote the radius around the selected items.

ing graph is undirected, whereas in the case of multiple radii, the corresponding graph is directed. We show that locating a DisC diverse subset is equivalent to locating an independent and dominating set for the corresponding graph. Although, for directed graphs, there are graphs for which there is no independent and dominating set, we show that for the graphs modeling the DisC problem, there is always such a set. Locating a minimum independent and dominating set is an NP-hard problem. We provide a suite of greedy algorithms for locating approximate solutions along with bounds for the size of the produced diverse subsets.

Then, we consider the problem of incrementally adjusting the radius \( r \), or zooming. We explore the relation among DisC diverse subsets of different radii and provide algorithms for adapting an already computed DisC diverse subset to a new radius along with corresponding theoretical upper bounds for the size of the diverse subsets produced. Figure 2 shows an example of zooming-in and zooming-out.

Although the examples presented so far concern two-dimensional points, DisC diversity is applicable to any type of data set, including the case of non-categorical attributes, as long as an appropriate distance \( d \) is provided. As an example, consider searching for cameras, where diversity refers to cameras with different features. Figure 3 depicts an initial most diverse result and the result of zooming-in one individual camera in this result.

Since the crux of the efficiency of all proposed algorithms is locating neighbors, we take advantage of spatial data structures. In particular, we propose efficient implementations based on the M-tree [Ciaccia et al. 1997]. We evaluate our algorithms for the different DisC diversity versions using both real and synthetic datasets and draw various conclusions regarding their effectiveness and efficiency.

In a nutshell, in this paper, we make the following contributions:

- we use a new, intuitive definition of diversity, called DisC diversity, based on using a radius \( r \) rather than a size limit \( k \) and extend it to support a different radius for each item,
- in addition to the geometrical interpretation of DisC diversity, we present an equivalent graph-based model of the problem,
- we introduce incremental diversification through zooming-in and zooming-out,
- we show that locating DisC diverse subsets is an NP-hard problem, provide efficient algorithms for their computation along with theoretical approximation bounds,
A:4

Fig. 2: Zooming. Selected items are shown as solid circles with size proportional to their weight. (Non solid) circles denote the radius around the selected items.

Fig. 3: Zooming-in a specific camera.

provide efficient M-tree tailored implementations and experimentally evaluate their performance,
– we compare DisC diversity with other popular diversity models, both analytically and qualitatively.

DisC diversity was first introduced in [Drosou and Pitoura 2012a]. Here, we generalize the DisC model to the weighted case and to the case of different radii for different items.

The rest of the paper is structured as follows. In Section 2, we introduce DisC diversity for single and multiple radii, while in Section 3, we provide a graph-based view of DisC diversity and study its complexity. In Section 4, we present algorithms for computing DisC diverse subsets. In Section 5, we focus on incremental diversification, i.e., adjusting the radius of a DisC diverse subset, while in Section 6, we compare our approach with other diversification methods, both analytically and qualitatively.
In Section 7, we employ the M-tree for the efficient implementation of our algorithms, while in Section 8, we present experimental results. Finally, Section 9 presents related work and Section 10 concludes the paper.

2. DISC DIVERSITY

In this section, we first provide a formal definition of DisC diversity. We define the Minimum $r$-DisC and Minimum Weighted $r$-DisC diverse subsets and provide various
Fig. 4: (a) A set of items: the (minimum) $r$-DisC diverse subset $\{p_2, p_4, p_6\}$ is preferred over the larger $r$-DisC diverse subset $\{p_1, p_3, p_4, p_6\}$, (b) their graph representation.

theoretical bounds for the size of an $r$-DisC diverse subset with regards to the minimum ones. Then, we extend our definition of DisC diversity to support a different radius for each item.

2.1. Definition of DisC Diversity

Let $P$ be a set of items returned as the result of a user query. We want to select a representative subset $S$ of these items such that each item of $P$ is represented by a similar item in $S$ and the items selected to be included in $S$ are dissimilar to each other.

We define similarity between two items using a distance metric $d : P \times P \to \mathbb{R}^+$. For a real number $r, r \geq 0$, we use $N_r(p_i)$ to denote the set of neighborhoods (or, the neighborhood) of an item $p_i \in P$, i.e., the items lying at distance at most $r$ from $p_i$:

$$N_r(p_i) = \{p_j | p_i \neq p_j \land d(p_i, p_j) \leq r\}$$

We use $N_{r}^+(p_i)$ to denote the set $N_r(p_i) \cup \{p_i\}$, i.e., the neighborhood of $p_i$ including $p_i$ itself. Items in the neighborhood of $p_i$ are considered similar to $p_i$, while items outside its neighborhood are considered dissimilar to $p_i$. We define an $r$-DisC diverse subset as follows:

**Definition 2.1.** ($r$-DisC Diverse Subset) Let $P$ be a set of items and $r, r \geq 0$, a real number. A subset $S$ of $P$ is an $r$-Dissimilar-and-Covering diverse subset, or $r$-DisC diverse subset, of $P$, if the following two conditions hold: (i) (coverage condition) $\forall p_i \in P, \exists p_j \in N_r^+(p_i)$, such that $p_j \in S$ and (ii) (dissimilarity condition) $\forall p_i, p_j \in S$ with $p_i \neq p_j$, it holds that $d(p_i, p_j) > r$.

The first condition ensures that all items in $P$ are represented by at least one similar item in $S$ and the second condition that the items in $S$ are dissimilar to each other. We call every item $p_i \in S$ an $r$-DisC diverse item and $r$ the radius of $S$. When the value of $r$ is clear from context, we simply refer to $r$-DisC diverse items as diverse items.

There may be more than one dissimilar and covering diverse subsets for the same set of items $P$. Since we want a concise representation of $P$, we select the smallest one (see Figure 4(a) for an example). Formally, we define the Minimum $r$-DisC diverse subset problem as follows:

**Definition 2.2.** (Minimum $r$-DisC Diverse Subset Problem) Given a set $P$ of items and a radius $r, r \geq 0$, find an $r$-DisC diverse subset $S^*$ of $P$, such that, for every $r$-DisC diverse subset $S$ of $P$, it holds that $f(S^*) \leq f(S)$, where $f(S) = |S|$.
Often, items are associated with a weight indicating their importance under some specific context, e.g., their relevance to the information need of the user issuing the query. We use $w(p_i)$ to denote the weight of $p_i$. Larger weights indicate items of higher importance. For simplicity, we consider that all weights are in $[0, 1]$. Now, given $\mathcal{P}$, we want to select items that are both diverse to each other and also highly relevant. We define the Minimum Weighted $r$-DisC diverse subset problem as follows:

Definition 2.3. (Minimum Weighted $r$-DisC Diverse Subset Problem) Given a set of items $\mathcal{P}$, a weight function $w : \mathcal{P} \rightarrow [0, 1]$ and a radius $r$, $r \geq 0$, find a DisC diverse subset $S^*$ of $\mathcal{P}$, such that, for every DisC diverse subset $S$ of $\mathcal{P}$, it holds that $f(S^*) \leq f(S)$, where

$$f(S) = \sum_{p_i \in S} \frac{1}{w(p_i)}$$

If we consider all weights equal, i.e., when we are only interested in the diversity of the selected set and not the individual weights of the selected items, then the Minimum Weighted $r$-DisC diverse subset problem is reduced to the Minimum $r$-DisC diverse subset problem, i.e., locating the minimum sized subset of dissimilar items that can
cover the available space. Between two subsets of equal size, Definition 2.3 selects the one with the largest sum of weights.

Figure 5 shows an example. The set \( \mathcal{P} \) consists of items forming clusters in the 2D Euclidean space. For the weighted case, we use two different ways to assign weights: (i) assigning a uniformly distributed weight to each item and (ii) assigning larger weights to items closer to the center of their cluster. The second approach models the common case in which we have a number of different interpretations of the query (e.g., “jaguar” as an animal and “jaguar” as a car). These different interpretations correspond to the center of the clusters and items close to each one of them are considered more important than more distant ones.

Figure 5(a) and Figure 5(d) depict the unweighted \( r \)-DisC diverse subsets, while Figure 5(b) and Figure 5(e) depict the weighted \( r \)-DisC diverse subsets. For comparison, we used the size \( k \) of the weighted \( r \)-DisC diverse subsets as the input for retrieving the top-\( k \) items with the largest weights (without enforcing diversity). Figure 5(c) and Figure 5(f) show the corresponding results. Clearly, the top-weighted items in the clustered case are very close to each other. In the uniform case they are more spread out but, still, not highly diverse. The unweighted DisC selects diverse items without considering weight. Using the weighted DisC, we manage to get items with large weights that in addition are distant to each other.

2.2. General Bounds

Next, we present a number of theoretical results concerning the size of an \( r \)-DisC diverse subset. In the following, we use the terms dominance and coverage, as well as, independence and dissimilarity interchangeably. In particular, two items \( p_i \) and \( p_j \) are independent, if \( d(p_i, p_j) > r \). We also say that an item covers all items in its neighborhood.

**Theorem 2.4.** Let \( B \) be the maximum number of independent neighbors of any item in \( \mathcal{P} \). Any \( r \)-DisC diverse subset \( S \) of \( \mathcal{P} \) is at most \( B \) times larger than any minimum \( r \)-DisC diverse subset \( S^* \).

**Proof.** Since \( S \) is an independent set, any item in \( S^* \) can cover at most \( B \) items in \( S \) and thus \( |S| \leq B|S^*| \).

Note that, since a minimum weighted \( r \)-DisC subset for \( \mathcal{P} \) cannot be smaller than a minimum \( r \)-DisC diverse subset \( \mathcal{P} \), it also holds that any \( r \)-DisC diverse subset \( S \) of \( \mathcal{P} \) is at most \( B \) times larger than any minimum weighted \( r \)-DisC diverse subset \( S^* \).

The value of \( B \) depends on the distance metric used and also on the dimensionality \( \text{dim} \) of the data space. For many distance metrics, \( B \) is a constant for a specific dimensionality. Next, we show how \( B \) is bounded for specific combinations of the distance metric and the data dimensionality.

**Lemma 2.5.** If \( d \) is the Euclidean distance and \( \text{dim} = 2 \), each item \( p_i \) in \( \mathcal{P} \) has at most \( B = 5 \) neighbors that are independent from each other.

**Proof.** Let \( p_1, p_2 \) be two independent neighbors of \( p \). Then, it must hold that \( \angle p_1 p p_2 \) is larger than \( \frac{\pi}{4} \). Otherwise, \( d(p_1, p_2) \leq \max\{d(p, p_1), d(p, p_2)\} \leq r \) which contradicts the independence of \( p_1 \) and \( p_2 \). Therefore, \( p \) can have at most \( \frac{2\pi}{\frac{\pi}{4}} - 1 = 5 \) independent neighbors.
LEMMA 2.6. If $d$ is the Manhattan distance and $\dim = 2$, each item $p_i$ in $\mathcal{P}$ has at most $B = 7$ neighbors that are independent from each other.

Proof. Let $p_1, p_2$ be two independent neighbors of $p$. Then, it must hold that $\angle p_1pp_2$ (in the Euclidean space) is larger than $\frac{\pi}{4}$. We will prove this using contradiction. $p_1, p_2$ are neighbors of $p$ so they must reside in the shaded area of Figure 6. Without loss of generality, assume that one of them, say $p_1$, is aligned to the vertical axis. Assume that $\angle p_1pp_2 \leq \frac{\pi}{4}$. Then $\cos(\angle p_1pp_2) \geq \frac{\sqrt{2}}{2}$. It holds that $b \leq r$ and $c \leq r$, thus, using the cosine law we get that $a^2 \leq r^2(2 - \sqrt{2})$ (1). The Manhattan distance of $p_1, p_2$ is equal to $x + y = \sqrt{a^2 + 2xy}$ (2). Also, the following hold: $x = \sqrt{b^2 - z^2}$, $y = c - z$ and $z = b \cos(\angle p_1pp_2) \geq \frac{br}{\sqrt{2}}$. Substituting $z$ and $c$ in the first two equations, we get $x \leq \frac{b}{\sqrt{2}}$ and $y \leq r - \frac{br}{\sqrt{2}}$. From (1), (2) we now get that $x + y \leq r$, which contradicts the independence of $p_1$ and $p_2$. Therefore, $p$ can have at most $(2\pi/\frac{\pi}{4}) - 1 = 7$ independent neighbors. □

LEMMA 2.7. If $d$ is the Euclidean distance and $\dim = 3$, each item $p_i$ in $\mathcal{P}$ has at most $B = 24$ neighbors that are independent from each other.

Proof. Assume a sphere of radius $r$ centered at $p_i$. To fit as many independent items in the sphere as possible, we place them on the surface of the sphere at distance $r$ from each other. Let $p_1, p_2$ be two such items. Since the radius of the sphere is also $r$, it holds that $\angle p_1pp_2 = \frac{\pi}{3}$. Thus, the arc on the surface of the sphere between $p_1$ and $p_2$ is equal to $\frac{\pi}{3}r$. The problem of how many such independent items can be placed on the surface of the sphere is equivalent to that of how many equilateral spherical triangles of side length $\frac{\pi}{3}r$ can be packed on the surface of the sphere, without overlap except at the edges. This number is not known exactly but it has been shown to be between 20 and 22 (e.g., [Wilhelm 2008]). The proof is based on dividing the area of the surface of the sphere by the area of such a triangle. To form these triangles, 24 items are required (3 for the first triangle plus 1 for each of the rest of the triangles), which proves the lemma. □

2.3. DisC Diversity with Multiple Radii

The radius $r$ specifies the granularity with which the selected DisC diverse subset represents the underlying result space. A large $r$ results in a small subset, whereas a small $r$ results in a large subset. So far, we considered that the radius $r$ is global, i.e., $r$ is the same for all items in $\mathcal{P}$. There may be cases, however, in which we want different parts of the data space to be represented with more or less items in the DisC diverse subset. To allow this, we consider the more general case where each item $p_i$ is
associated with a different radius \( r(p_i) \), i.e., \( r \) is not a constant but, instead, a function \( r : \mathcal{P} \rightarrow \mathbb{R}^+ \) assigning a radius \( r(p_i) \in \mathbb{R}^+ \) to each item \( p_i \in \mathcal{P} \).

The problem now loses its symmetry, since an item \( p_i \) may be in the neighborhood of an item \( p_j \), while \( p_j \) is not in the neighborhood of \( p_i \). This gives rise to two different interpretations of radius. One interpretation is that \( p_i \) can represent all items in its neighborhood (i.e., all items lying at a distance at most \( r(p_i) \) around it). The other interpretation is that \( p_i \) can be represented by all items in its neighborhood. We call the first problem **Covering DisC diverse subset problem** and the second one **CoveredBy DisC diverse subset problem**.

For example, in Figure 7(a), under the Covering DisC semantics, the radius of \( p_3 \) means that \( p_3 \) represents, or covers, \( p_2 \) and \( p_1 \), whereas, based on their radius, neither \( p_2 \) nor \( p_1 \) can represent, or cover, \( p_3 \). On the contrary, under the CoveredBy semantics, the radius of \( p_3 \) means that \( p_3 \) can be represented, or covered, by \( p_2 \) and \( p_1 \), but neither \( p_2 \) nor \( p_1 \) can be represented, or covered, by \( p_3 \).

Next, we present the corresponding formal definitions.

**Definition 2.8. (Covering DisC Diverse Subset)** Let \( \mathcal{P} \) be a set of items and \( r : \mathcal{P} \rightarrow \mathbb{R}^+ \) be a function determining the radius of each item in \( \mathcal{P} \). A subset \( S \) of \( \mathcal{P} \) is a Covering DisC diverse subset of \( \mathcal{P} \), if the following two conditions hold: (i) (coverage condition) \( \forall p_i \in \mathcal{P}, \exists p_j \in S \) such that \( d(p_i, p_j) \leq r(p_j) \) and (ii) (dissimilarity condition) \( \forall p_i, p_j \in S \) with \( p_i \neq p_j \), it holds that \( d(p_i, p_j) \geq \max\{r(p_i), r(p_j)\} \).

For example, in Figure 7(a), \( \{p_3, p_5, p_6, p_7\} \) is a Covering DisC subset of the depicted set of items.

**Definition 2.9. (CoveredBy DisC Diverse Subset)** Let \( \mathcal{P} \) be a set of items and \( r : \mathcal{P} \rightarrow \mathbb{R}^+ \) be a function determining the radius of each item in \( \mathcal{P} \). A subset \( S \) of \( \mathcal{P} \) is a CoveredBy DisC diverse subset of \( \mathcal{P} \), if the following two conditions hold: (i) (coverage condition) \( \forall p_i \in \mathcal{P}, \exists p_j \in S \) such that \( d(p_i, p_j) \leq r(p_i) \) and (ii) (dissimilarity condition) \( \forall p_i, p_j \in S \) with \( p_i \neq p_j \), it holds that \( d(p_i, p_j) \geq \max\{r(p_i), r(p_j)\} \).

For example, in Figure 7(a), \( \{p_2, p_4, p_7\} \) is a CoveredBy DisC subset of the depicted set of items.

Our motivation behind multiple radii is to allow placing different importance to different items in the dataset. Next, we present three different scenarios for assigning radii to items.

The first one corresponds to the case where some parts of the dataset are considered more important than others and we want these parts to be represented with more
Fig. 8: Using multiple radii based on areas of interest ((a)-(b)), density ((c)-(d)) and weights ((e)-(h)). Selected items are shown as solid circles. (In the CoveredBy case, each item $p_i$ not in the diverse subset is represented by an item in the diverse subset within distance $r(p_i)$ from it. Since there is a large number of such items, we do not draw their radii for clarity.)

items in the selected diverse subset. In Figure 8(a) and Figure 8(b), we see such an example, where each of the four quadrants is assumed to have different importance, with the most important one being the bottom left quadrant and importance decreasing as we move clockwise. To achieve a representation corresponding to importance, we assign to each area clockwise increasing radius values. As seen, areas associated with smaller radii (i.e., more important ones) are represented by more items in the diverse set, since items in these areas have to be closer together to be considered similar. The basic difference between the results of the Covering and the CoveredBy approach is near the boundaries of the quadrants. In the Covering approach, items in the quadrant with the larger radii cover the items in the neighboring quadrant, thus excluding them from the diverse set.

The second scenario corresponds to the case in which we want to relate representation with density, so that dense areas are not under-represented in the diverse subset. To achieve this, we assign smaller radii to items in denser areas of the dataset. Figure 8(c) and Figure 8(d) show the retrieved solutions for the Covering and CoveredBy variations of the multiple radii problem. In both cases, the dense areas of the dataset are better represented in the diverse set due to their items being associated with smaller radii. Again, in the CoveredBy case, different behavior appears at the boundaries (see, for example, the items selected from the outskirts of the top cluster).

The third scenario corresponds to the case in which we want to relate representation with weights. For the Covering problem, we assign larger radii to items with larger weights. This is to model the case where we want highly relevant items to cover a large area around them. For the CoveredBy problem, we assign smaller radii to items with larger weights. This ensures that each item can be covered only by items that have a larger weight than it. We consider again the uniform and clustered distribution of weights (Figure 8(e)-Figure 8(h)). In this case, as shown, Covering sets are much dif-
Fig. 9: (a) Minimum dominating set \( \{v_2, v_5\} \) and (b) a minimum independent dominating set \( \{v_2, v_4, v_6\} \) of the depicted graph.

Different than CoveredBy sets. The CoveredBy approach produces more relevant results in the sense that it allows more relevant items to be included in the diverse set. In the Covering approach, the large radius of a relevant item excludes other relevant items in its neighborhood from entering the diverse set. This is more evident in the clustered weights case, for which the CoveredBy approach is a better fit.

3. GRAPH REPRESENTATION AND NP-HARDNESS

In this section, we present a graph based model of DisC diversity and show that locating a DisC diverse subset is equivalent to finding an independent and dominating set of the corresponding graph.

The various DisC subsets presented so far have a corresponding graph representation. Consider first a single radius \( r \) and let \( G_{P,r} = (V, E) \) be an undirected graph such that there is a vertex \( v_i \in V \) for each item \( p_i \in P \) and an edge \( (v_i, v_j) \in E \), if and only if, \( d(p_i, p_j) \leq r \) for the corresponding items \( p_i, p_j \). An example is shown in Figure 4(b).

Let us recall a couple of graph-related definitions. A \textit{dominating set} \( D \) for a graph \( G \) is a subset of vertices of \( G \) such that every vertex of \( G \) not in \( D \) is joined to at least one vertex in \( D \) by some edge. An \textit{independent set} \( I \) for a graph \( G \) is a set of vertices of \( G \) such that for every two vertices in \( I \), there is no edge connecting them. It is easy to see that a dominating set of \( G_{P,r} \) satisfies the covering condition of Definition 2.1, whereas an independent set of \( G_{P,r} \) satisfies the dissimilarity condition of Definition 2.1. Thus:

**Lemma 3.1.** A set \( S, S \subseteq P \), is an \( r \)-DisC diverse subset for a set \( P \), if and only if, it is an independent dominating set of the corresponding graph \( G_{P,r} \).

We next present some useful properties that relate the coverage (i.e., dominance) and dissimilarity (i.e., independence) conditions. A \textit{maximal independent set} of a graph is an independent set such that adding any other vertex to the set forces the set to contain an edge, that is, an independent set that is not a subset of any other independent set. It is known that [Berge 1962]:

**Lemma 3.2.** An independent set of a graph is maximal, if and only if, it is dominating.

From Lemma 3.2, we conclude that:

**Observation 1.** A minimum maximal independent set is also a minimum independent dominating set.
Observation 2. A minimum dominating set is not necessarily independent.

For example, in Figure 9, the minimum dominating set of the depicted items is of size 2, while the minimum independent dominating set is of size 3.

The above also holds for the Minimum Weighted r-DisC diverse subset problem.

Let us consider the multiple radii case. Our graph-based view of the problem is now the following. Let $G_{P,r}(r) = (V,E)$ be a directed graph such that there is a vertex $v_i \in V$ for each item $p_i \in P$ and a (directed) edge $(v_i, v_j) \in E$, if and only if, for the corresponding items $p_i, p_j$, it holds that $d(p_i, p_j) \leq r(p_i)$ (Covering problem) or $d(p_i, p_j) \leq r(p_j)$ (CoveredBy problem). In Figure 7, we see an example. The coverage relationship is not symmetric anymore. In Figure 7(b), for example, item $p_1$ covers $p_i$ and $p_2$, but neither $p_1$ nor $p_2$ cover $p_3$. Independence between two items means that none of them covers the other. In Figure 7(b), $p_4$ and $p_7$ are independent, but items $p_3$ and $p_1$ are not.

A dominating, or covering, set $D$ for a directed graph $G$ is a subset of vertices of $G$ such that every vertex of $G$ not in $D$ is joined to at least one vertex of $D$ by some incoming edge. An independent set $I$ for a directed graph $G$ is a set of vertices of $G$ such that, for every two vertices in $I$, there is no edge connecting them.

Lemma 3.3. A set $S$, $S \subseteq P$, is a Covering or CoveredBy DisC diverse subset for a set $P$, if and only if, it is an independent dominating set of the corresponding graph $G_{P,r}(r)$.

Proof. Let $S$ be a DisC diverse subset for $P$. Due to the coverage condition, for every item $p_i$ not in $S$, there must be an item $p_j$ in $S$ with $d(p_i, p_j) \leq r(p_j)$ (Covering problem) or $d(p_i, p_j) \leq r(p_j)$ (CoveredBy problem), thus $S$ is a dominating set of $G_{P,r}(r)$. Also, due to the dissimilarity condition, no item $p_i$ in $S$ can cover some other item $p_j$ in $S$. Thus, $S$ is also an independent set. Now, let $S$ be an independent dominating set $S$ of the directed graph $G_{P,r}(r)$. Then, for every item $p_i$ not in $S$ there is some item $p_j$ in $S$ such that an edge $(p_j, p_i)$ exists, i.e., $d(p_i, p_j) \leq r(p_j)$ (Covering problem) or $d(p_i, p_j) \leq r(p_j)$ (CoveredBy problem). Also, there is no edge connecting $p_i, p_j$ in $S$, i.e., $d(p_i, p_j) > \max\{r(p_i), r(p_j)\}$. Thus, both the coverage and dissimilarity conditions of Definition 2.8 (Covering problem) or Definition 2.9 (CoveredBy problem) hold and $S$ is also a DisC diverse subset for $P$. \hfill \Box

Note that, when all items are associated with equal radii, it holds that $d(p_i, p_j) \leq r(p_i)$, if and only if, $d(p_i, p_j) \leq r(p_i)$. In this case, the graph representation of a set $P$ for the Covering problem is equivalent to that for the CoveredBy problem and, in addition, all edges of the graph are bidirectional, i.e., the graph can be reduced to an undirected graph.

Finding a minimum independent dominating set of a graph has been proven to be NP-hard [Garey and Johnson 1979], even for special cases of graphs such as unit disk graphs [Clark et al. 1990], i.e., graphs in the Euclidean space whose vertices can be put in one to one correspondence with equisized circles in a plane such that two vertices are joined by an edge, if and only if, the corresponding circles intersect.

4. Computing Disc Diverse Subsets

In this section, we present a suite of algorithms for locating DisC diverse subsets. We first present a baseline algorithm for constructing $r$-DisC diverse subsets and then show a greedy variation that aims at minimizing the size of the produced subsets. We also consider the multiple radii case and present an algorithm for constructing DisC
Algorithm 1 Locating DisC diverse subsets.

Input: A set of items $P$, a radius function $r(.)$ and a selection criterion $C(.)$.
Output: A DisC diverse subset $S$ of $P$.

1: $S \leftarrow \emptyset$
2: for all $p_i \in P$ do
3:   color $p_i$ white
4:   end for
5: while there exist white items do
6:   select the white item $p_i$ with the largest value of $C(p_i)$
7:   $S = S \cup \{p_i\}$
8:   color $p_i$ black
9:   for all $p_j \in N^W_{r(p_i)}(p_i)$ (Covering) or $p_j$ s.t. $p_i \in N_r(p_j)$ (CoveredBy) do
10:      color $p_j$ grey
11:   end for
12: end while
13: return $S$

subsets for the Covering and CoveredBy problems. Finally, we elaborate on locating subsets that satisfy only the coverage condition and present a greedy algorithm for locating such sets.

The baseline algorithm (shown in Algorithm 1) selects items to be included in the diverse subset at rounds, one at a time. For presentation convenience, let us call black the items of $P$ that are in the diverse subset $S$, grey the items covered by some item in $S$ and white the items that are neither black nor grey. $N^W_r(p_i)$ denotes the set of white neighbors of $p_i$. Initially, $S$ is empty and all items are white. At each round, we select a white item that satisfies a selection condition $C$.

4.1. Basic-DisC for Single Radius

We next show that any selection criterion in Algorithm 1 results in an $r$-DisC diverse subset.

**Lemma 4.1.** In the single radius case, Algorithm 1 produces an $r$-DisC diverse subset $S$ of $P$ for any selection criterion.

**Proof.** At first all items are white. Once an item enters $S$, all its neighbors become grey and are withdrawn from consideration. Any white item is independent from all selected items in $S$ and thus can be selected to be included in $S$. To see that, assume for the purpose of contradiction, that for a white item $p_i$ and black item $p_j$, it holds $d(p_i, p_j) \leq r$, then $p_j \in N_r(p_i)$, thus it should have been colored grey, when $p_i$ was selected for inclusion in $S$. Thus, the set produced by selecting any white item is an independent set. It is also a maximal independent set, since at the end there are only grey items left, thus adding any of them to $S$ would violate the independence of $S$. From Lemma 3.2, $S$ is an $r$-DisC diverse subset. □

We call this algorithm Basic-DisC. As shown in Lemma 2.4, the size of any DisC subset, and thus the size of the DisC subset produced by Basic-DisC is at most $B$ times larger than that of a minimum $r$-DisC diverse subset or a minimum weighted $r$-DisC diverse subset.
We normalize the size of a white neighborhood in the white item that has the best combination of weight and white neighborhood size. In the case of the Minimum Weighted problem, all weights are in \((0, 1]\). That is, in the case of the Minimum Weighted DisC diverse subset problem, we set \(C(p_i) = |N^W_i(p_i)|\), i.e., we select the white item that covers the largest number of uncovered items. In the case of the Minimum Weighted DisC diverse subset problem, we select the white item that has the best combination of weight and white neighborhood size.

We now consider the following intuitive greedy variation of Basic-DisC, that we call Greedy-DisC. Instead of selecting white items arbitrarily at each step for inclusion in the diverse subset \(S\), we select the white item that minimizes the objective function \(f\). That is, in the case of the Minimum \(r\)-DisC diverse subset problem, we set \(C(p_i) = |N^W_i(p_i)|\), i.e., we select the white item that covers the largest number of uncovered items. In the case of the Minimum Weighted \(r\)-DisC diverse subset problem, we select the white item that has the best combination of weight and white neighborhood size.

We normalize the size of a white neighborhood in \([0, 1]\) (recall that we have assumed that weights are in \((0, 1]\) and use \(C(p_i) = w(p_i) (|N^W_i(p_i)|/\max_{p_j \in \mathcal{P}, v \in \mathcal{V}} |N^W_i(p_j)|)\). In case of ties, we select the white item with the largest number of white neighbors.

2.3. Multiple-DisC for Multiple Radii

While an undirected graph always has an independent dominating subset, this is not the case for directed graphs (e.g., [Milner and Woodrow 1989]). To illustrate this, consider the following simple example of Figure 10 where \(V = \{v_1, v_2, v_3\}\) and \(E = \{(v_1, v_2), (v_2, v_3), (v_3, v_1)\}\). In this graph, no single item is able to cover the whole set, while, at the same time, no two items are independent from each other. Thus, this graph has no independent dominating subset. However, such a graph cannot exist in our case. Assume, for the purposes of contradiction, that such as a graph exists. For the Covering case, since there is an edge \((v_2, v_3)\) but there is no edge \((v_2, v_1)\), we get that \(r(p_3) \geq d(p_1, p_3) > r(p_2)\). Similarly, it also holds that \(r(p_2) > r(p_3)\) and \(r(p_3) > r(p_1)\). Therefore, we get that \(r(p_3) > r(p_1)\) which cannot be true. Similarly, for the CoveredBy problem, we get that \(r(p_3) < r(p_1)\) which, also, cannot be true.

In our case, where \(d\) is a distance metric, we can always construct an independent dominating subset for our directed graphs. To achieve this, we need to select white items using some specific criterion as the following lemma shows. Note, that the proof of Lemma 4.1 does not hold, since not all white items are necessarily independent from the selected diverse items. For example, consider the Covering problem for the graph of Figure 11 and assume that \(v_1\) is selected first. Then, \(v_1\) will be colored black and \(v_2\) will be colored grey, while \(v_3\) will remain white. However, \(v_3\) cannot enter the selected subset in a following step, since it is not independent from \(v_1\). Thus, one can choose the right selection criterion \(C\) in Algorithm 1 for the case of multiple radii.

**Lemma 4.2.** Algorithm 1 produces a (multiple radii) DisC diverse subset \(S\) of \(\mathcal{P}\) when selecting white items in (i) decreasing order of their radius, i.e., \(C(p_i) = r(p_i)\), for the Covering problem and (ii) increasing order of their radius, i.e., \(C(p_i) = 1/\rho(p_i)\), for the CoveredBy problem.

**Proof.** We prove the lemma for the Covering problem. The proof for the CoveredBy problem is similar. At first all items are white. Upon selecting an item for inclusion,
all its neighbors become grey and are thus withdrawn from consideration. Let \( p_i \) be a white item considered at some round and \( p_j \) be an already selected, i.e., black, item. Since \( p_i \) is white, there can be no directed edge \((p_i, p_j)\). Also, since \( p_i \) was considered for inclusion in \( S \) prior to \( p_j \), it holds that \( r(p_i) \leq r(p_j) \). Thus, since \( r(p_i) \leq r(p_j) \) and there can be no directed edge \((p_i, p_j)\) we have that \( d(p_i, p_j) > r(p_j) \geq r(p_i) \). That is, there can be no directed edge \((p_i, p_j)\). Therefore, each white item selected to be colored black at some round is independent from all previously selected items, i.e., the produced set is an independent one. It is also a maximal independent set, since at the end there are only grey items left (line 5), thus selecting any of them would violate the independence of \( S \). From Lemma 3.2, \( S \) is a DisC diverse subset.

We call this algorithm \text{Multiple-DisC}. For the example of Figure 11, \text{Multiple-DisC} would select \( v_3 \) first, since \( v_3 \) has the largest radius (it covers both \( v_1 \) and \( v_2 \)) and the Covering DisC diverse subset \( \{p_3\} \) will be produced. Furthermore, in our example of Figure 7, by visiting white items in decreasing order of their radius (solving the Covering problem), \text{Multiple-DisC} would first select \( p_5 \), followed by \( p_6 \) and \( p_7 \), in that order, resulting in a DisC diverse subset of \( P \). Visiting white items in increasing order of their radius (solving the CoveredBy problem) would result in the selection of \( p_2, p_7 \) and \( p_4 \), in that order.

### 4.4. Greedy Algorithm for Coverage Only

While the size of the subsets produced by \text{Greedy-DisC} is expected to be smaller than that of the subsets produced by \text{Basic-DisC}, the fact that we consider for inclusion in \( S \) only white, i.e., independent, items may still not reduce the size of \( S \) as much as expected. From Observation 2, it is possible that an independent covering set is larger than a covering set that also includes dependent items. For example, consider the vertices (or equivalently the corresponding items) in Figure 9. Assume that \( v_2 \) is inserted in \( S \) first, resulting in \( v_1, v_3 \) and \( v_5 \) becoming grey. Then, we need two more vertices, namely, \( v_4 \) and \( v_6 \), for covering the whole set. However, if we consider for inclusion grey items as well, then \( v_5 \) can join \( S \), resulting in a smaller covering set.

Motivated by this observation, we also define \( r \)-C diverse subsets that satisfy only the coverage condition of Definition 2.1. Figure 12(a) and Figure 12(b) show how the unweighted DisC solution is affected when the dissimilarity condition is raised, i.e., we have a covering but not necessarily independent subset of the data. Raising the dissimilarity condition slightly decreases the size (by one item) in this example. However, the selected items are close together (see, for example, the cluster on the right of the dataset). Figure 12(c) and Figure 12(d) show how the weighted DisC solution is affected when the dissimilarity condition is raised. While, in our example, the size remains the same, the selected items are closed together, i.e., not as dissimilar. However, this allows us to get more items with larger weights, since the selected items are not required to be dissimilar to each other.

To compute \( r \)-C diverse sets, we modify \text{Greedy-DisC} accordingly. The only change required is that for selecting the next diverse item, we consider both white and grey objects. This allows us to select at each step the item that covers the largest possible number of uncovered items, even if this item is grey. We call this variation \text{Greedy-C}.

For \text{Greedy-C}, we get a covering but not necessarily dissimilar subset of \( P \), whose size is generally different than the size of the subset produced by \text{Greedy-DisC} for the same radius \( r \). In this case, we get a different bound for the size of the produced \( r \)-C diverse subset \( S \).
Fig. 12: Solutions for the unweighted ((a)-(b)) and weighted ((c)-(d)) Dissimilar-and-Covering (r-DisC) and Covering-only (r-C) problems. Selected items are shown as solid circles. (Non solid) circles around items denote the radius \( r \) of the selected items.

**Theorem 4.3.** Let \( \Delta \) be the maximum number of neighbors of any item in \( \mathcal{P} \). The weighted \( r-C \) diverse set produced by Greedy-C is at most \( \ln \Delta \) times larger than the minimum weighted r-DisC diverse set \( S^* \).

**Proof.** In this proof, we shall use the graph representation of the problem. We consider that inserting a vertex (item) \( p \) into \( S \) has cost \( 1/w(p) \). We distribute this cost equally among all covered vertices, i.e., after being labeled grey, vertices are not charged anymore. Assume an optimal minimum dominating set \( S^* \). The graph \( G \) can be decomposed into a number of star-shaped subgraphs, each of which has one vertex from \( S^* \) at its center. The cost of an optimal minimum dominating set is exactly \( 1/w(p) \) for each star-shaped subgraph centered around \( p \). We show that for a non-optimal set \( S \), the cost for each star-shaped subgraph is at most \( \ln \Delta \), where \( \Delta \) is the maximum degree of the graph. Consider a star-shaped subgraph of \( S^* \) with \( p \) at its center and let \( N_W^p \) be the number of white vertices in it. If a vertex in the star is labeled grey by Greedy-C, these vertices are charged some cost. By the greedy condition of the algorithm, this cost can be at most \( 1/w(p)|N_W^p| \) per newly covered vertex. Otherwise, the algorithm would rather have chosen \( p \) for the dominating set because \( p \) would contribute \( 1/w(p)|N_W^p| \) to the selected set. In this case, the first vertex that is labeled grey is charged at most \( 1/w(p)(\delta(p) + 1) \), the second vertex is charged at most \( 1/w(p)\delta(p) \) and so on, where \( \delta(p) \) is the degree of \( p \). Therefore, the total cost for covering the star of \( p \) is at most:

\[
\frac{1}{w(p)} \left( \frac{1}{\delta(p) + 1} + \frac{1}{\delta(p)} + \ldots + \frac{1}{2} + 1 \right) = \frac{1}{w(p)} H(\delta(p) + 1) \leq \frac{1}{w(p)} H(\Delta + 1)
\]

where \( H(i) \) is the \( i \)th harmonic number. The total cost of the set \( S \) produced by Greedy-C for covering all the stars of \( S^* \) is:

\[
f(S) \leq \sum_{p_i \in S^*} \frac{1}{w(p_i)} H(\Delta + 1) = H(\Delta + 1) f(S^*) \approx \ln \Delta f(S^*)
\]

Since the size of a minimum weighted dominating set is equal or smaller than the size of a minimum weighted independent dominating set, the theorem holds. \( \square \)

Note that, Theorem 4.3 also holds in the case of the (unweighted) Minimum \( r-\text{DisC} \) diverse subset problem, where we simply consider the cost of each star-shaped subgraph around \( p \) to be equal to 1.
As an example, consider Figure 13 and let $p$ be an item in the $r$-DisC optimal diverse subset $S^*$. The cost of covering all items in the neighborhood of $p$ for $S^*$ is equal to $1/w(p)$. Let $S$ be a subset produced by Greedy-C and assume $p_1$ is covered first by some item other than $p$. Since $p$ has not been colored black, the cost corresponding to the covering of $p_1$ can be at most equal to $1/4w(p)$, since there are still 4 uncovered items inside the neighborhood of $p$. Assume that $p, p_2, p_3$ are then covered by some other item. The cost corresponding to the covering of each of them can be at most equal to $1/3w(p)$. Thus, the cost of $S$ for covering the area inside the neighborhood of $p$ is $1/w(p)(1/4 + 1/3 + 1/3 + 1/3) = 1.25(1/w(p))$.

Greedy-C is used also in the case of multiple radii. In this case, we do not need to select items based on the size of their radii based on Lemma 4.2, since we do not require that they are independent. Thus, the greedy selection and Theorem 4.3 apply as is.

5. INCREMENTAL DISC

The radius determines the desired degree of diversification. A large radius corresponds to fewer and less similar to each other representative items, whereas a small radius results in more and less dissimilar representative items. At one extreme, a radius equal to the largest distance between any two items results in a single item being selected and, at the other extreme, a radius smaller than the smallest pairwise distance in the result set results in all items of $P$ being selected.

In this section, we consider an interactive mode of operation where, after being presented with an initial set of results for some radius, a user asks to see either more or less results by correspondingly decreasing or increasing the radius. First, we present results relating the size of DisC diverse subsets for different radii and then propose algorithms for incrementally changing the radius.

5.1. Zooming

Consider first the single radius case. Given a set of items $P$ and an $r$-DisC diverse subset $S$ of $P$ for some specific radius, we want to compute an $r'$-DisC diverse subset $S'$ of $P$. There are two cases: (i) $r' < r$ and (ii) $r' > r$ which we call zooming-in and zooming-out respectively.

Since we want to support an incremental mode of operation, the set $S'$ should be as close as possible to the already seen result $S$. Ideally, $S' \supseteq S$, for $r' < r$ and $S' \subseteq S$, for $r' > r$. However, this is not always possible as the following lemma shows.

**Lemma 5.1.** Let $S$ be a covering and dissimilar subset of $P$ for $r$.

(i) $S$ is a covering but not necessarily dissimilar subset of $P$ for $r' > r$.
(ii) $S$ is a dissimilar but not necessarily covering subset of $P$ for $r' < r$.
Fig. 14: Zooming in (a) and out (b) for the single radius case. Dashed and solid circles correspond to radius $r$ and $r'$ respectively.

**Proof.** (i) Let $r' > r$. Since $S$ is a covering subset of $P$ for $r$, for each item $p_i \in P$, there is an item $p_j \in S$ such that $d(p_i, p_j) \leq r < r'$, thus $S$ is also a covering subset of $P$ for $r'$. However, for two items $p_i, p_j \in S$, it is possible that $p_j \notin N_{r'}(p_i)$. Therefore, $S$ may not be a dissimilar subset of $P$ for $r'$. (ii) Let $r' < r$. Since $S$ is a dissimilar subset of $P$ for $r$, for any two items $p_i, p_j \in P$, it holds $d(p_i, p_j) > r > r'$, thus $S$ is also a dissimilar subset of $P$ for $r'$. However, it is possible that there exists some item $p_k \in P$ for which there does not exist an item $p_j \in S$ with $p_j \in N_r(p_k)$. Therefore, $S$ may not be a covering subset of $P$ for $r'$. □

For the multiple radii case, instead of a single new radius $r'$, we assume a new function $r'(p_i)$. In the zooming-in case, $r'(p_i) < r(p_i)$ for all $p_i$, while in the zooming-out case, $r'(p_i) > r(p_i)$ for all $p_i$. It is easy to see that Lemma 5.1 holds for both the Covering and CoveredBy problems if we replace $r'$ and $r$ with the corresponding functions.

To study the relationship between $S$ and $S'$ when changing the radius, we focus on the items lying at distance between $r$ and $r'$ from the selected items of the initial DisC diverse subset (Figure 14). These are items of interest since they are possibly either left uncovered when the radius decreases (zooming-in), thus violating the covering property, or covered by other diverse items when the radius increases (zooming-out), thus violating the dissimilarity property.

For two radii $r_1, r_2, r_2 > r_1$, we define the set $N^I_{r_1, r_2}(p_i)$, as the set of items at distance at most $r_2$ from $p_i$ which are at distance larger than $r_1$ from each other, i.e., items in $N_{r_1}(p_i) \setminus N_{r_1}(p_i)$ that are independent from each other considering the radius $r_1$.

The following lemma bounds the size of $N^I_{r_1, r_2}(p_i)$ for specific distance metrics and dimensionality.

**Lemma 5.2.** Let $r_1, r_2$ be two radii with $r_2 > r_1$. Then, for $dim = 2$:

(i) if $d$ is the Euclidean distance:

$$|N^I_{r_1, r_2}(p_i)| \leq 9 \left[ \log_\beta (r_2/r_1) \right], \text{ where } \beta = \frac{1 + \sqrt{5}}{2}$$

(ii) if $d$ is the Manhattan distance:

$$|N^I_{r_1, r_2}(p_i)| \leq 4 \sum_{i=1}^{\gamma} (2i + 1), \text{ where } \gamma = \left\lceil \frac{r_2 - r_1}{r_1} \right\rceil$$

**Proof.** Euclidean distance: For the proof, we use a technique for partitioning the annulus between $r_1$ and $r_2$ similar to the one in [Thai et al. 2007a] and [Xing et al.}
neighbors of each other. Let \( r \) be the radius of an item \( p \) (Figure 15(a)) and \( \alpha \) a real number with \( 0 < \alpha < \frac{\pi}{2} \). We draw circles around the item \( p \) with radii \((2\cos\alpha)^r, (2\cos\alpha)^{r+1}, (2\cos\alpha)^{r+2}, \ldots, (2\cos\alpha)^{y_n-1}, (2\cos\alpha)^{y_n}\), such that \((2\cos\alpha)^r \leq r_1 \) and \((2\cos\alpha)^{r+1} > r_2 \) and \((2\cos\alpha)^{y_n-1} < r_2 \) and \((2\cos\alpha)^{y_n} \geq r_2 \). It holds that \( x_p = \frac{\ln r_1}{\ln(2\cos\alpha)} \) and \( y_p = \frac{\ln r_2}{\ln(2\cos\alpha)} \). In this way, the area around \( p \) is partitioned into \( y_p - x_p \) annuli plus the \( r_1 \)-disk around \( p \). Consider an annulus \( A \). Let \( p_1 \) and \( p_2 \) be two neighbors of \( p \) in \( A \) with \( d(p_1, p_2) > r_1 \). Then, it must hold that \( \angle p_1 pp_2 > \alpha \). To see this, we draw two segments from \( p \) crossing the inner and outer circles of \( A \) at \( a \), \( b \), \( c \), \( d \) such that \( p_1 \) resides in \( pb \) and \( \angle bpd = \alpha \), as shown in the figure. Due to the construction of the circles, it holds that \( \frac{|pb|}{|pd|} = \frac{|pa|}{|pe|} = 2 \cos \alpha \). From the cosine law for \( \angle abd \), we get that \(|ad| = |pa| \) and, therefore, it holds that \(|cb| = |ad| = |pa| = |pe| \). Therefore, for any item \( p_3 \) in the area \( abcd \) of \( A \), it holds that \(|pp_3| > |bp_3| \) which means that all items in that area are neighbors of \( p_1 \), i.e., at distance less or equal to \( r_1 \). For this reason, \( p_2 \) must reside outside this area which means that \( \angle p_1 pp_2 > \alpha \). Based on this, we see that there exist at most \( \frac{2\pi}{\alpha} - 1 \) independent (for \( r_1 \)) nodes in \( A \). The same holds for all annuli. Therefore, we have at most \((yn-1)\left(\frac{2\pi}{\alpha} - 1\right)\) independent nodes in the annuli. For \( 0 < \alpha < \frac{\pi}{2} \), this has a minimum when \( \alpha \) is close to \( \frac{\pi}{2} \) and that minimum value is \( # A:19 \). Let \( r \) be the radius of an item \( p \). We draw Manhattan circles around the item \( p \) with radii \( r_1, r_2, \ldots \) until the radius \( r_2 \) is reached. In this way, the area around \( p \) is partitioned into \( \gamma = \left\lfloor \frac{r_2 - r_1}{r_1} \right\rfloor \) Manhattan annuluses plus the \( r_1 \)-Manhattan-disk around \( p \). Consider an annulus \( A \). The items shown in Figure 15(b) cover the whole annulus and their Manhattan pairwise distances are all greater or equal to \( r_1 \). Assume that the annulus spans among distance \( ir_1 \) and \( (i+1)r_1 \) from \( p \), where \( i \) is an integer with \( i > 1 \). Then, \(|ab| = \sqrt{2(ir_1 + r_1/2)^2} \). Also, for two items \( p_1, p_2 \) it holds that \(|p_1p_2| = \sqrt{2(r_1/2)^2} \). Therefore, at one quadrant of the annulus there are \( \frac{|ab|}{|p_1p_2|} = 2i + 1 \) independent neighbors which means that there are \( 4(2i + 1) \) independent neighbors in \( A \). Therefore, there are in total \( \sum_{i=1}^{\gamma} 4(2i + 1) \) independent (for \( r_1 \)) neighbors of \( p \). □

5.2. Incremental Zooming Algorithms for the Single Radius Case

Next, we describe our algorithms for incrementally adapting an \( r \)-DisC diverse subset \( S \) to an \( r' \)-DisC diverse subset \( S' \) and provide bounds concerning the size relationship of \( S \) and \( S' \).

Fig. 15: Independent neighbors.
Algorithm 2 Greedy-Zoom-In.

Input: A set of items $P$, a solution $S$ and initial and new radii $r(p_i), r'(p_i), r'(p_i) < r(p_i)$, for each item $p_i$ in $P$.

Output: An adapted DisC diverse subset of $P$.

1: $S' \leftarrow S$
2: for all $p_i \in S$ do
3: color items in $\{N_r(p_i) \setminus N_{r'}(p_i)\}$ white
4: end for
5: while there exist white items do
6: choose the white item $p_i$ with the largest $|N_{r'}(p_i)|$
7: color $p_i$ black
8: $S' \leftarrow S' \cup \{p_i\}$
9: for all $p_j \in N_{r'}(p_i)$ do
10: color $p_j$ grey
11: end for
12: end while
13: return $S'$

Zooming-in. Let us first consider the case of zooming-in to a smaller radius, i.e., $r' < r$. Here, we aim at producing a small independent covering solution $S'$, such that, $S' \supseteq S$. Since $S$ is an independent but not necessarily dominating set for $r'$, we construct $r'$-DisC diverse subsets by adding items to $S$ to make it a maximal independent, and thus dominating, set using Lemma 3.2.

Consider an item of $S$, for example $p_1$ in Figure 14(a). Items at distance at most $r'$ from $p_1$ are still covered by $p_1$ and cannot enter $S'$. Items at distance greater than $r'$ and at most $r$ may be uncovered and join $S'$. Each of these items can enter $S'$ as long as it is not covered by some other item of $S$ that lays outside the former neighborhood of $p_1$. For example, in Figure 14(a), $p_4$ and $p_5$ may enter $S'$ while $p_3$ cannot, since, even with the smaller radius $r'$, $p_3$ is covered by $p_2$.

To adapt a DisC diverse subset, we consider such items in turn. This turn can be either arbitrary (Basic-Zoom-In algorithm) or proceed in a greedy way, where at each turn the item that covers the largest number of uncovered items is selected (Greedy-Zoom-In, Algorithm 2).

Concerning the size relationship between $S$ and $S'$, the following lemma holds.

Lemma 5.3. Let $S$ be the initial DisC diverse set and $S'$ be the adapted one generated by the Basic-Zoom-In or Greedy-Zoom-In algorithm. It holds that:

(i) $S \subseteq S'$ and
(ii) $|S'| \leq |S| + \sum_{p_i \in S} |N_{r',r}(p_i)|$

Proof. Condition (i) trivially holds from step 1 of the algorithm. Condition (ii) holds since for each item in $S$ there are at most $|N_{r',r}(p_i)|$ independent items at distance greater than $r'(p_i)$ from each other that can enter $S'$. 

In practice, items selected to enter $S'$, such as $p_4$ and $p_5$ in Figure 14(a), are likely to cover other items left uncovered by the same or similar items in $S$. Therefore, the size difference between $S$ and $S'$ is expected to be smaller than this theoretical upper bound.

To produce the new adapted DisC diverse subset, we proceed in two passes. In the first pass, we examine all items of $S$ in some order and remove their diverse neighbors that are now covered by them. At the second pass, items from any uncovered
Algorithm 3 Greedy-Zoom-Out(a).

**Input:** A set of items $P$, a solution $S$ and initial and new radii $r(p_i), r'(p_i), r'(p_i) > r(p_i)$, for each item $p_i$ in $P$.

**Output:** An adapted DisC diverse subset of $P$.

1: $S' \leftarrow \emptyset$
2: color all black items red
3: color all grey items white
4: while there exist red items do
5: select the red item $p_i$ with the largest $|N_r(p_i)|$
6: color $p_i$ black
7: $S' = S' \cup \{p_i\}$
8: for all $p_j \in N_r(p_i)$ do
9: color $p_j$ grey
10: end for
11: end while
12: while there exist white items do
13: select the white item $p_i$ with the larger $|N_w(p_i)|$
14: color $p_i$ black
15: $S' = S' \cup \{p_i\}$
16: for all $p_j \in N_w(p_i)$ do
17: color $p_j$ grey
18: end for
19: end while
20: return $S'$

areas are added to $S'$. Again, we have an arbitrary and a greedy variation, denoted Basic-Zoom-Out and Greedy-Zoom-Out respectively. Algorithm 3 shows the greedy variation; the first pass (lines 4-11) considers $S \setminus S'$, while the second pass (lines 12-19) considers $S' \setminus S$. Initially, we color all previously black items red. All other items are colored white. We consider three variations for the first pass of the greedy algorithm: selecting the red items with (a) the largest number of red neighbors, (b) the smallest number of red neighbors and (c) the largest number of white neighbors. Variations (a) and (c) aim at minimizing the items to be added in the second pass, that is, $S' \setminus S$, while variation (b) aims at maximizing $S \cap S'$. Algorithm 3 depicts variation (a), where $N_r(p_i)$ denotes the set of red neighbors of item $p_i$.

Concerning the size relationship between $S$ and $S'$, the following lemma holds.

**Lemma 5.4.** For the solution $S'$ generated by the Basic-Zoom-Out or Greedy-Zoom-Out algorithm, it holds that:

(i) There are at most $\sum_{p_i \in S} |N_{r'}(p_i)|$ items in $S \setminus S'$.

(ii) For each item of $S$ not included in $S'$, at most $B - 1$ items are added to $S'$, where $B$ is the maximum number of independent neighbors of any item in $P$.

**Proof.** Condition (i) is a direct consequence of the definition of $N_{r'}(p_i)$. Concerning condition (ii), recall that each removed item $p_i$ has at most $B$ independent neighbors for $r'(p_i)$. Since $p_i$ is covered by some neighbor, there are at most $B - 1$ other independent items that can potentially enter $S'$. □

5.3. Incremental Zooming for the Multiple Radii Case

We next elaborate on extending zooming operators to the multiple radii case. Our main goal remains adapting the previous DisC subset $S$ to a new subset $S'$, i.e., maintaining
Fig. 16: Zooming-in example for the Covering (a) and Covered By (b) problems for the multiple radii case. Dashed and solid circles correspond to radius before and after zooming respectively.

as many as possible (independent) items from the previous solution and selecting new items to cover any uncovered areas.

As previously detailed, for both zooming-in and for the second pass of zooming-out, we need to select items to cover any areas that are left uncovered. For the single radius case, as shown in Lemma 5.3 and Lemma 5.4, selecting any uncovered (i.e., white) item works. However, this is not the case for multiple radii. In this case, items must be selected in some specific order, i.e., in decreasing or increasing order of their radius for the Covering and CoveredBy problems respectively (see Section 4). Otherwise, it is possible that an uncovered item is selected that is not independent from other already selected items.

To illustrate, consider the Covering example of Figure 16(a), where all radii have been reduced to half and assume that $p_1$ and $p_3$ were in $S$. $p_1$ and $p_3$ remain independent for the new radii, while $p_2$ is now left uncovered. $p_2$ cannot enter $S$, since it covers $p_3$ and is thus not independent from it. A similar example for the CoveredBy case is depicted in Figure 16(b).

To overcome such issues, we proceed as follows. Whenever we encounter an uncovered item $p_i$ that cannot enter $S'$, we exclude from $S'$ items that are not independent from $p_i$. In our example of Figure 14(a), $p_3$ is excluded from $S'$. Now, $p_2$ can enter $S'$. In this example, $S'$ is now a DisC diverse subset. Note that, the exclusion of $p_3$ could have led to new uncovered areas of the dataset, which would be covered in a similar fashion.

6. COMPARISON WITH OTHER MODELS

In this section, we show how our DisC model relates to other diversification methods.

Theoretical Results. Two widely used diversification models are MAXMIN and MAXSUM that aim at selecting a subset $S$ of $\mathcal{P}$ so as the minimum or the average pairwise distance of the selected items is maximized (e.g., [Gollapudi and Sharma 2009; Vieira et al. 2011; Borodin et al. 2012]). More formally, an optimal MAXMIN (resp., MAXSUM) subset of $\mathcal{P}$ is a subset $S$ with the maximum $f_{\text{MIN}}(S) = \min_{p_i, p_j \in S} \text{dist}(p_i, p_j)$ (resp., $f_{\text{SUM}}(S) = \sum_{p_i, p_j \in S} \text{dist}(p_i, p_j)$) over all subsets of the same size. Input in both approaches is the size $k$ of the diverse subset.
The following lemma provides a bound for the \( f_{\text{MIN}} \) distance of the items in any \( r\text{-DisC} \) set with regards to the optimal distance \( f_{\text{MIN}} \) of a subset of the same size.

**Lemma 6.1.** Let \( \mathcal{P} \) be a set of items, \( S \) be an \( r\text{-DisC} \) diverse subset of \( \mathcal{P} \) and \( \lambda \) be the \( f_{\text{MIN}} \) distance between the items of \( S \). Let \( S^* \) be an optimal MAXIN subset of \( \mathcal{P} \) for \( k = |S| \) and \( \lambda^* \) be the \( f_{\text{MIN}} \) distance of \( S^* \). Then, \( \lambda^* \leq 3 \lambda \).

**Proof.** Each item in \( S^* \) is covered by (at least) one item in \( S \). There are two cases, either (i) all items \( p_1^*, p_2^* \in S^* \), \( p_1^* \neq p_2^* \), are covered by different items is \( S \), or (ii) there are at least two items in \( S^* \), \( p_1^*, p_2^* \in S^* \), that are both covered by the same item \( p \) in \( S \). Case (i): Let \( p_1 \) and \( p_2 \) be two items in \( S \) such that \( d(p_1, p_2) = \lambda \) and \( p_1^* \) and \( p_2^* \) respectively be the items in \( S^* \) that each covers. Then, by applying the triangle inequality twice, we get: \( d(p_1, p_2) \leq d(p_1, p_1^*) + d(p_2, p_2^*) \). By coverage, we get: \( d(p_1, p_2) \leq r + \lambda + r \leq 3 \lambda \), thus \( \lambda^* \leq 3 \lambda \). Case (b): Let \( p_1^* \) and \( p_2^* \) be two items in \( S^* \) that are covered by the same item \( p \) in \( S \). Then, by coverage and the triangle inequality, we get \( d(p_1^*, p_2^*) \leq d(p_1, p) + d(p, p_2) \leq 2 \), thus \( \lambda^* \leq 2 \lambda \).

Lemma 6.1 shows how much smaller the \( f_{\text{MIN}} \) distance of an \( r\text{-DisC} \) subset is with regards to the optimal \( f_{\text{MIN}} \) of a subset of the same size. The following lemma looks into the size of an \( r\text{-DisC} \) subset that attains the same \( f_{\text{MIN}} \) distance as an optimal MAXIN subset of size \( k \).

**Lemma 6.2.** Let \( \mathcal{P} \) be a set of items, \( S^* \) be an optimal subset of \( \mathcal{P} \) of size \( k \) and \( \lambda^* \) be the \( f_{\text{MIN}} \) distance of \( S^* \). Let \( S \) be an \( r\text{-DisC} \) diverse subset with \( r = \lambda^* \). It holds that \( |S| < k' \), where \( k' \) is the first integer larger than \( k \) for which the corresponding optimal MAXIN subset of \( \mathcal{P} \) \( S^{*'} \) has \( f_{\text{MIN}} \) distance equal to \( \lambda^* \), with \( \lambda^{*'} < \lambda^* \).

**Proof.** Since the optimal (i.e., maximum) minimum distance for \( k' \) is smaller than \( \lambda^* \), then there can be no DisC set for \( r = \lambda^* \) with size equal or larger than \( k' \). Therefore, a DisC diverse subset for \( r = \lambda^* \) can be of size up to \( k' \).

A bound similar to that of Lemma 6.1 does not exist for the MAXSUM case\(^1\). To illustrate this, consider the example of Figure 17, where \( k - 1 \) of the items are located in a line, at distance \( r \) from each other, while the rest of the items are very close to each other and located at a large distance from the other items, i.e., \( a \gg 1 \). Let us call the two groups of items “group X” and “group Y” respectively. For simplicity, let \( r = 1 \).

---

\(^1\)We would like to thank Dr. Anirban Dasgupta from Yahoo! Research for suggesting this example.
Fig. 18: Solutions by the various diversification methods for a clustered dataset. All items are associated with equal weights and radii. Selected items are shown as solid circles. (Non solid) circles around items of the DisC solution denote the radius $r$ of the selected items.

An optimal DisC diverse set for $r = 1$ is of size $k$ and consists of the $k - 1$ items of group X plus one item from group Y. Let $G$ be the sum of the pairwise distances of all items of group X. The sum of pairwise distances $f_{\text{DisC}}$ of the optimal DisC set is approximately equal to $a(k - 1) + G$. A MAXSUM solution for $k$ would instead select $k/2$ items from group X and $k/2$ items from group Y. The corresponding sum of pairwise distances $f_{\text{MAXSUM}}$ in this case would be $a(k/2)^2 + G'$, where $G' < G$ is the sum of the pairwise distances of all items from group X. Since $a$ can be arbitrarily large, for a sufficiently large value of $a$, we can assume that $f_{\text{DisC}} \approx a(k - 1)$ and $f_{\text{MAXSUM}} \approx a(k/2)^2$. Therefore, for a sufficiently large value of $a$, it holds that $f_{\text{MAXSUM}} / f_{\text{DisC}} = \frac{k^2}{4(k-1)}$. Thus, $f_{\text{MAXSUM}} / f_{\text{DisC}}$ can grow arbitrarily large as $k$ increases.

**Qualitative Results.** Next, we present some qualitative results of applying different approaches for selecting diverse items, namely DisC, MAXMIN and MAXSUM. We also show results for $k$-medoids, a widespread clustering algorithm that seeks to minimize $\frac{1}{|P|} \sum_{p_i \in P} d(p_i, c(p_i))$, where $c(p_i)$ is the closest item of $p_i$ in the selected subset, since the located medoids can be viewed as a representative subset of the dataset. We used a 2-dimensional “Clustered” dataset. To implement MAXMIN and MAXSUM, we used greedy heuristics which have been shown to achieve good solutions [Drosou and Pitoura 2010; Gollapudi and Sharma 2009]. To allow for a comparison, we first run Greedy-DisC for a given $r$ and then use the size of the produced diverse subset as the input $k$ of the other approaches. In this example, $k = 12$ for $r = 0.15$ (Figure 18).

MAXSUM diversification and $k$-medoids fail to cover all areas of the dataset; MAXSUM tends to focus on the outskirts of the dataset, whereas $k$-medoids clustering reports only central items, ignoring items that are further away. MAXMIN performs better in this aspect. However, since MAXMIN seeks to retrieve items that are as far apart as possible, it fails to retrieve items from dense areas; see, for example, the central areas of the clusters in Figure 18. Note also that MAXSUM and $k$-medoids may select near duplicates, as opposed to DisC and MAXMIN. We also experimented with variations of MAXSUM proposed in [Vieira et al. 2011] but the results did not differ substantially from the ones in Figure 18(b).

Note that, while specifying a size $k$ may often be intuitive, fixed size methods do not provide any guarantees concerning the quality of the result in terms of dissimilarity and coverage. DisC aims at providing an alternative approach for the case where the user is interested in retrieving diverse results that cover the whole result space. While some knowledge on the distances among the items may be useful when setting $r$, it is
not necessary; a user may start with some random radius \( r \) and later on use zooming to tune the size of the diverse subset. We revisit this issue in our experiments (Section 8).

7. IMPLEMENTATION

A central operation in computing DisC diverse subsets is locating neighbors. For this reason, we introduce implementations of our algorithms that exploit a spatial index structure, namely, the M-tree [Ciaccia et al. 1997]. An M-tree is a balanced tree index that can handle large volumes of dynamic data of any dimensionality in general metric spaces. In particular, an M-tree partitions space around some of the indexed items, called pivots, by forming a bounding ball region of some bounding radius around them. Let \( c \) be the maximum node capacity of the tree. Internal nodes have at most \( c \) entries, each containing a pivot item \( p_v \), the bounding radius \( br_v \) around \( p_v \), the distance of \( p_v \) from its parent pivot and a pointer to the subtree \( t_v \) rooted at \( p_v \). All items in the subtree \( t_v \) are within distance at most equal to the bounding radius \( br_v \) from \( p_v \). Leaf nodes have entries containing the indexed items and their distance from their parent pivot.

New items are inserted in the M-tree in such a way so as to minimize any possible increase of the bounding radii of the various pivots. Assume, for example, the M-tree of Figure 19(a) and let \( p_7 \) be a new item. \( p_7 \) will be inserted in the leftmost leaf node, since the increase required for the bounding radius of \( p_1 \) to accommodate \( p_7 \) is smaller than the respective increase for the bounding radius of \( p_4 \). The construction of an M-tree is also influenced by the splitting policy that determines how nodes are split when they exceed their maximum capacity \( c \). For instance, in our example, \( c = 3 \). The insertion of \( p_7 \) in the M-tree of Figure 19(a) would require splitting the leftmost leaf node and promoting one of its items to its parent node as a new pivot (see Figure 19(b)). Splitting policies indicate (i) which two of the \( c+1 \) available pivots will be promoted to the parent node to index the two new nodes (promote policy) and (ii) how the rest of the pivots will be assigned to the two new nodes (partition policy). These policies affect the overlap among the nodes of the tree. For computing diverse subsets, we use an M-tree to index the items to be diversified. In addition:

(i) We link together all leaf nodes. This allows us to visit all items in a single left-to-right traversal of the leaf nodes and exploit some degree of locality in covering the items.

(ii) To compute the neighbors \( N_r(p_i) \) of an item \( p_i \) at radius \( r \), we perform a range query centered around \( p_i \) with distance \( r \), denoted \( Q(p_i, r) \).

(iii) We build trees using splitting policies that minimize overlap. In most cases, the policy that resulted in the lowest overlap was (a) promoting as new pivots the pivot \( p_i \) of the overflowed node and the item \( p_j \) with the maximum distance from \( p_i \) and (b) partitioning the items by assigning each item to the node whose pivot has
the closest distance with the item. We call this policy “MinOverlap”. Figure 19(b) shows an example of applying “MinOverlap” after the arrival of \( p_7 \).

7.1. Computing Diverse Subsets

We next present implementations of the various algorithms presented for computing diverse subsets that use the M-tree to improve efficiency.

**Basic-DisC.** The Basic-DisC algorithm selects white items in random order. In the M-tree implementation of Basic-DisC, we consider items in the order they appear in the leaves of the M-tree, thus taking advantage of locality. Upon encountering a white item \( p_i \) in a leaf, the algorithm colors it black and executes a range query \( Q(p_i, r) \) to retrieve and color grey its neighbors. Since the neighbors of an indexed item are expected to reside in nearby leaf nodes, such range queries are in general efficient. We can visualize the progress of Basic-DisC as gradually coloring all items in the leaf nodes from left-to-right until all items become either grey or black.

**Greedy-DisC.** The Greedy-DisC algorithm selects at each round the best white item according to the selection criterion \( C \) (line 6 of Algorithm 1). To efficiently implement this selection, we maintain a sorted list, \( L \), of all white items in:
- decreasing order of the size of their white neighborhood for the Minimum \( r \)-DisC diverse subset problem,
- decreasing order of the product of their weight and the (normalized) size of their white neighborhood for the Minimum Weighted \( r \)-DisC diverse subset problem.

Instead of performing one range query per item after building the tree to compute the size of the white neighborhoods to initialize \( L \), we compute such values incrementally as we build the M-tree. At first, for each item \( p_i \), it holds that \( N^W_r(p_i) = N_r(p_i) \). To compute the neighborhood size of each item incrementally, when an item \( p_i \) is inserted into the M-tree, a range query \( Q(p_i, r) \) is executed, the white neighborhood of \( p_i \) is initialized to \( |Q(p_i, r)| \) and the size of the white neighborhoods of all items retrieved by the range query are incremented by one. We found that this incremental approach reduces node accesses up to 45%.

For maintaining \( L \), each time an item \( p_i \) is selected and colored black, first, a range query \( Q(p, r) \) is executed to retrieve its neighbors and color them grey. In addition to this, we need to update the ordering of a number of affected items, i.e., neighbors of these newly colored grey items whose white neighborhoods have been affected. To do this, we exploit a one-pass traversal of the M-tree, where we descend the M-tree looking for pivots whose subtrees may contain neighbors of newly colored grey items. We found this one-pass approach reduces node accesses up to 70% as compared to executing one range query separately per newly colored grey item to update the white neighborhoods of all affected items. To further reduce the cost of maintaining the exact size of the white neighborhoods, we also consider a lazy variation (Lazy-Greedy-DisC) that only considers affected items at some distance smaller than \( r \) for any newly colored grey item \( p_j \) during the one-pass traversal of the M-tree.

**Multiple-DisC.** The Multiple-DisC algorithm selects at each round the white item with the largest (resp. smallest) radius for the Covering (resp. CoveredBy) problem. Ties are resolved by selecting the item with the largest white neighborhood. For updating the white neighborhoods of the various items, we utilize a one-pass traversal of the M-tree as for Greedy-DisC. For the Covering problem, any items at some distance smaller than \( r_j \) from any newly colored grey item \( p_j \) are retrieved and updated. For the CoveredBy problem, any items at some distance smaller than \( r_{\text{max}} \) from any newly colored grey item \( p_j \) are retrieved and updated, where \( r_{\text{max}} \) is the largest radius in the dataset. Again, we also consider lazy variations.
Pruning. We make the following observation that allows us to prune subtrees while executing range queries for all algorithms above. Items that are already grey do not need to be colored grey again when some other of their neighbors is colored black.

PRUNING RULE: A leaf node that contains no white items is colored grey. When all its children become grey, an internal node is colored grey. While executing range queries, any top-down search of the tree does not need to follow subtrees rooted at grey nodes.

As the algorithms progress, more and more nodes become grey, and thus, the cost of range queries reduces over time. For example, we can visualize the progress of the Basic-DisC (Pruned) algorithm as gradually coloring all tree nodes grey in a post-order manner.

Greedy-C. The Greedy-C algorithm considers at each round both grey and white items. A sorted structure \( L \) has to be maintained as well, which now includes both white and grey items and is substantially larger. Furthermore, the pruning rule is no longer useful, because grey items and nodes need to be accessed again, since they are potential candidates.

7.2. Adapting the Radius

For zooming-in, given an \( r \)-DisC diverse subset \( S \) of \( \mathcal{P} \), we would like to compute an \( r' \)-DisC diverse subset \( S' \) of \( \mathcal{P} \), \( r' < r \), such that, \( S' \supseteq S \). A naive implementation would require as a first step locating the items in \( N_{r'}(p_i) \) (line 3 of Algorithm 2) by invoking two range queries for each \( p_i \) (with radius \( r \) and \( r' \) respectively). Then, a diverse subset of the items in \( N_{r'}(p_i) \) is computed either in a basic or in a greedy manner. However, during the construction of \( S \), items in the corresponding M-tree have already been colored black or grey. We use this information in the following rule.

ZOOMING RULE: Black items of \( S \) maintain their color in \( S' \). Grey items maintain their color as long as there exists a black item at distance at most \( r' \) from them.

Therefore, only grey nodes with no black neighbors at distance \( r' \) may turn black and enter \( S' \). To apply this rule, we augment the leaf nodes of the M-tree with the distance of each indexed item \( p_i \) to its closest black neighbor \( p_j \), since \( p_i \) will continue to be covered by \( p_j \) for all \( r' \leq d(p_i, p_j) \).

The Basic-Zoom-In algorithm requires one pass of the leaf nodes. Each time a grey item \( p_i \) is encountered, we check whether it is still covered, i.e., whether its distance from its closest black neighbor is smaller or equal to \( r' \). If not, \( p_i \) is colored black and a range query \( Q(p_i, r') \) is executed to locate and color grey the items for which \( p_i \) is now their closest black neighbor. At the end of the pass, the black items of the leaves form \( S' \). The Greedy-Zoom-In algorithm involves the maintenance of a sorted structure \( L \) of all white items. To build this structure, the leaf nodes are traversed, grey items that are now found to be uncovered are colored white and inserted into \( L \). After this, \( L \) is sorted accordingly.

Zooming-out algorithms are implemented similarly to the zooming-in case.

8. EXPERIMENTAL EVALUATION

In this section, we evaluate the efficiency and effectiveness of our algorithms using both synthetic and real datasets. We first describe our datasets and algorithms and then present experimental results.
Table I: Input parameters.

| Parameter                      | Default value | Range       |
|--------------------------------|---------------|-------------|
| M-tree node capacity           | 50            | 25 - 100    |
| M-tree splitting policy        | MinOverlap    | various     |
| Dataset cardinality            | 100000        | 579 - 150000|
| Dataset dimensionality         | 2             | 2 - 10      |
| Dataset spatial distribution   | clustered     | uniform, clustered |
| Dataset weight distribution    | clustered     | uniform, clustered |
| Distance metric                | Euclidean     | Euclidean, Hamming |
| Radius assignment              | uniform       | uniform, weight-based |

Table II: Algorithms.

| Algorithm          | Abbreviation | Description |
|--------------------|--------------|-------------|
| Basic-DisC         | B-DisC       | Selects items in order of appearance in the leaf level of the M-tree. |
| Greedy-DisC        | G-DisC       | Selects at each round the white item \(p_i\) with the largest value of \(C(p_i)\). |
| Multiple-DisC      | M-DisC       | Selects white items in order of their radius. |
| Greedy-C           | G-C          | Selects at each round the non-black item \(p_i\) with the largest value of \(C(p_i)\). |
| Basic-LinearScan   | B-LinearScan | Selects items in random order. |
| Greedy-LinearScan  | G-LinearScan | Selects at each round the white item \(p_i\) with the largest value of \(C(p_i)\). |

8.1. Datasets and Algorithms

Our synthetic datasets consist of multidimensional items, which are either uniformly distributed in space or form (hyper) spherical clusters of different sizes. We assign weights to items either uniformly or in a “clustered” manner around specific target items, i.e., items that are considered very important, so that items that are closer to the target items get larger weights than items further away. Thus, we have four combinations for our synthetic data based on the spatial and weight distributions, namely “Uniform-Uniform”, “Uniform-Clustered”, “Clustered-Uniform” and “Clustered-Clustered”. We also employ four real datasets. Three of them contain geographical information about (i) 5922 cities and villages in Greece (“Greek Cities”)\(^2\), (ii) the 590 highest populated cities in the world (“World Cities”)\(^3\) and (iii) 1000 apartments for sale in London (“Nestoria”)\(^4\). Weights are assigned uniformly for “Greek Cities”, based on higher population for “World Cities” and based on lower price for “Nestoria”. The fourth real dataset (“Cameras”) consists of 7 characteristics for 579 digital cameras, such as brand and storage type\(^5\). We assign weights based on a combination of the megapixels and the optical zoom of the cameras.

We normalize the values of all datasets in \([0, 1]\). We use the Euclidean distance for the synthetic and geographical datasets, while for “Cameras”, whose attributes are categorical, we use \(d(p_i, p_j) = \sum_i \delta(p_i, p_j)\), where \(\delta(p_i, p_j)\) is equal to 1, if \(p_i\) and \(p_j\) differ in the \(i\)th dimension and 0 otherwise, i.e., the Hamming distance. Note that the choice of an appropriate distance metric is an important but orthogonal to our problem issue.

We next briefly summarize the algorithms used throughout this section.

— **Unweighted DisC**: We employ the Basic-DisC and Greedy-DisC algorithms. Basic-DisC simply selects a DisC diverse subset, while Greedy-DisC also attempts to minimize size by selecting at each round the white item \(p_i\) with the largest \(|N^W(p_i)|\). We also use Lazy-Greedy-DisC for \(r/2\), as described in Section 7.

\(^2\)http://www.rtreeportal.org
\(^3\)http://nordpil.com/go/resources/world-database-of-large-cities
\(^4\)http://www.nestoria.co.uk
\(^5\)http://acme.com/digicams
Fig. 20: Block accesses for Basic-DisC, Greedy-DisC, Lazy-Greedy-DisC and Greedy-C with and without pruning for the unweighted case.

— **Weighted DisC:** Again, we employ Basic-DisC, Greedy-DisC and Lazy-Greedy-DisC. The difference now is that the greedy algorithms select at each step the white item $p_i$ with the largest $w(p_i)\left(|N^W(p_i)|/\max_{p_j \in P} |N^W(p_j)|\right)$. In case of ties, the item $p_i$ with the largest $|N^W(p_i)|$ is selected.

— **Multiple Radii DisC:** We employ Multiple-DisC. In case of ties, the item $p_i$ with the largest $|N^W(p_i)|$ is selected.

In all cases, we also consider the Greedy-C algorithm that produces covering but not necessarily independent sets. Greedy-C works as Greedy-DisC with the difference that both white and grey items are considered as candidates at each step.

We also compare our M-tree implementation with a linear scan implementation that does not use an index. Specifically, at each round, the linear scan implementation of Basic-DisC (Basic-LinearScan) reads all white items, colors some of them grey and writes back all items that still remain white. The linear scan implementation of Greedy-DisC (Greedy-LinearScan) operates in a similar way. However, at each round, white items need to be read twice: once for coloring some of them grey and once for updating the size of the white neighborhoods. Note that both linear scan algorithms are optimized in that, at each round, they do not operate on the whole dataset but only on the remaining white objects.

Table I summarizes the values of the input parameters used in our experiments and Table II summarizes the algorithms employed.

We have implemented our algorithms in Java and run our experiments on a 3.10GHz Intel Core i3-2100 PC with 3GB of RAM and a SATA hard disk achieving speeds around 35MB/sec for random accesses and 110MB/sec for sequential accesses.
8.2. Comparison of the Different Algorithms

In the first set of the experiments, we evaluate the different algorithms in terms of performance and quality of results. Performance is measured in terms of I/O operations, while quality in terms of the size of the computed diverse subset. For these experiments, we use the unweighted DisC; we evaluate the weighted DisC in following sections. We also evaluate the effect of the data set and M-tree characteristics.

Performance. We first measure the cost of our algorithms in terms of block accesses in the employed M-trees. Figure 20 reports this cost, as well as, the cost savings when the pruning rule of Section 7 is employed for Basic-DisC, Greedy-DisC and Lazy-Greedy-DisC (as previously detailed, this pruning cannot be applied to Greedy-C). Greedy-DisC has higher cost than Basic-DisC. Both algorithms benefit from pruning (up to 30% for small radii). Lazy-Greedy-DisC does not reduce the computational cost significantly, while in some cases its cost is higher than that of Greedy-DisC. This happens because Lazy-Greedy-DisC produces larger DisC subsets than those of Greedy-DisC, which means that more passes of the M-tree are performed.

We also evaluate the benefits of using the M-tree over using the linear scan implementations of our algorithms. Figure 21 shows the number of I/O accesses for Basic-DisC and Greedy-DisC using our synthetic datasets of 100,000 items. The block...
size of the linear scan implementations is equal to the node size of the M-tree. We see that the M-tree implementations require less I/O accesses, thus reducing cost. Note also that the linear scan implementation of Greedy-DisC performs better than the linear scan implementation of Basic-DisC for the “Clustered” dataset, since more items are colored grey at the first rounds of the algorithm as compared to the “Uniform” case, and thus, less white items have to be read and written in subsequent rounds.

Next, we report real CPU time when all data resides in memory using our synthetic datasets of 10,000 items. (Figure 22). We see that the M-Tree implementations are faster than their linear scan counterparts. For Basic-DisC, the reason is that the linear scan implementation retrieves all remaining white items at each round in order to decide which ones to color grey, while the M-Tree implementation takes advantage of the spatial index structure and retrieves a much lower number of items. For Greedy-DisC, this disadvantage of the linear scan implementation is not as obvious, since Greedy-DisC colors grey more items than Basic-DisC at early rounds, thus reducing the number of white items read at later iterations. However, the M-Tree implementation of Greedy-DisC is much more efficient during the initialization phase of the algorithm, where the neighborhood sizes of the various items need to be computed. The M-tree speeds up these computations (see Section 7), thus achieving better execution time in total.

Both the I/O cost and the CPU time of Basic-DisC decrease for larger radii, since larger radii result in more items being colored grey at each round. For Greedy-DisC, the I/O cost decreases as well, for the same reason. However, the CPU time increases, since larger radii result in more computations for updating the sizes of white neighborhoods at each round. This increase is more evident for the “Clustered” dataset, where items selected at early rounds have more neighbors than in the “Uniform” case.

Finally, we use our 100,000 items datasets to draw some conclusions concerning the execution time trade-off between I/O and CPU operations (Figure 23). For smaller radii, the execution time required by Basic-DisC is dominated by I/O operations. The reason is that, at each round, less items are disqualified and, thus, more items are transferred between the disk and the memory. While the I/O cost gets larger for Greedy-DisC as well (see also Figure 21), the degree of domination is not as high,
Table III: Solution size for the unweighted (“no”) and weighted (“yes”) cases.

(a) Uniform-Uniform.

| r        | 0.01 | 0.02 | 0.03 | 0.04 | 0.05 | 0.06 | 0.07 |
|----------|------|------|------|------|------|------|------|
| no       |      |      |      |      |      |      |      |
| yes      |      |      |      |      |      |      |      |
| B-DisC   | 5971 | 5971 | 1649 | 1649 | 753  | 753  | 446  |
| G-DisC   | 5090 | 5614 | 1472 | 1572 | 711  | 743  | 282  |
| L-G-DisC | 5377 | 5598 | 1480 | 1501 | 682  | 705  | 466  |
| G-C      | 4941 | 5516 | 1427 | 1533 | 696  | 717  | 397  |

(b) Clustered-Uniform.

| r        | 0.01 | 0.02 | 0.03 | 0.04 | 0.05 | 0.06 | 0.07 |
|----------|------|------|------|------|------|------|------|
| no       |      |      |      |      |      |      |      |
| yes      |      |      |      |      |      |      |      |
| B-DisC   | 1874 | 1874 | 591  | 591  | 278  | 278  | 176  |
| G-DisC   | 1683 | 1736 | 531  | 547  | 262  | 271  | 154  |
| L-G-DisC | 1660 | 1724 | 501  | 533  | 255  | 254  | 149  |
| G-C      | 1637 | 1720 | 520  | 544  | 250  | 271  | 151  |

(c) Uniform-Clustered (weighted only).

| r        | 0.01 | 0.02 | 0.03 | 0.04 | 0.05 | 0.06 | 0.07 |
|----------|------|------|------|------|------|------|------|
| no       |      |      |      |      |      |      |      |
| yes      |      |      |      |      |      |      |      |
| G-DisC   | 5352 | 1554 | 740  | 440  | 234  | 206  | 146  |
| L-G-DisC | 5511 | 1514 | 697  | 397  | 266  | 189  | 139  |
| G-C      | 5218 | 1506 | 708  | 422  | 278  | 197  | 136  |

since Greedy-DisC also requires more CPU time to compute the neighborhood sizes of the various items during the initialization phase. As noted before, the M-Tree variation reduces this extra cost (see also Figure 22). For larger radii, the execution time required by Greedy-DisC is dominated by CPU operations, since larger radii result in items having more neighbors, whose pairwise distances must be computed, while at the same time, more items are disqualified at early rounds, resulting in reduced I/O cost.

Solution Size. We next compare our various algorithms in terms of the size of the computed diverse subset (Table III). We present results for our synthetic and one of our real datasets. Results are similar for the omitted datasets. We consider Basic-DisC, Greedy-DisC, Lazy-Greedy-DisC and Greedy-C. Greedy-DisC locates a smaller DisC diverse subset than Basic-DisC in all cases. Lazy-Greedy-DisC also performs better than Basic-DisC and comparable with Greedy-DisC. Generally, all algorithms locate smaller DisC sets for clustered datasets. Greedy-C produces subsets with size similar with those produced by Greedy-DisC. This means that raising the independence assumption does not lead to substantially smaller diverse subsets in our datasets.

Impact of Dataset Cardinality and Dimensionality. In the rest of this section, unless otherwise noted, we use the Greedy-DisC algorithm.

For this experiment, we employ the “Clustered” dataset. We vary its cardinality from 50000 to 150000 items and its dimensionality from 2 to 10 dimensions. Figure 24 shows the corresponding cost and solution size as computed by Greedy-DisC. We observe that the solution size is more sensitive to changes in cardinality when the radius is small. The reason for this is that for large radii, a selected item covers a large area in space. Therefore, even when the cardinality increases and there are many available items to choose from, these items are quickly covered by the selected ones. In Figure 24(a), the increase in the cost is due to the increase of range queries required to maintain correct information about the size of the white neighborhoods.
Increasing the dimensionality of the dataset causes more items to be selected as diverse as shown in Figure 24(d). This is due to the “curse of dimensionality” effect, since space becomes sparser at higher dimensions. The computational cost may however vary as dimensionality increases, since it is influenced by the cost of computing the neighborhood size of the items that are colored grey. For smaller radii, in which case more items are selected, the cost decreases due to the effective pruning during the one-pass traversal of the M-tree.

**Impact of M-tree Characteristics.** Next, we evaluate how the characteristics of the employed M-trees affect the computational cost of computed DisC diverse subsets. Note that, different tree characteristics do not have an impact on which items are selected as diverse.

Different degree of overlap among the nodes of an M-tree may affect its efficiency for executing range queries. To quantify such overlap, we employ the fat-factor [Jr. et al. 2002] of a tree $T$ defined as:

$$f(T) = \frac{Z - nh}{n} \cdot \frac{1}{m - h}$$

where $Z$ denotes the total number of block accesses required to answer point queries for all items stored in the tree, $n$ the number of these items, $h$ the height of the tree and $m$ the number of nodes in the tree. Ideally, the tree would require accessing one node per level for each point query which yields a fat-factor of zero. The worst tree would visit all nodes for every point query and its fat-factor would be equal to one.

We created various M-trees using different splitting policies which result in different fat-factors. We present results for four different policies. The lowest fat-factor was acquired by employing the “MinOverlap” policy. Selecting as new pivots the two items with the greatest distance from each other resulted in increased fat-factor. Even higher fat-factors were observed when assigning an equal number of items to each new node (instead of assigning each item to the node with the closest pivot) and, finally, selecting the new pivots randomly produced trees with the highest fat-factor among all policies.

Figure 25 reports our results for our uniform and clustered 2-dimensional datasets with cardinality equal to 100000. For the uniform dataset, we see that a high fat-factor leads to more block accesses being performed for the same solution. This is not the case for the clustered dataset, where items are gathered in dense areas and thus increasing the fat-factor does not have the same impact as in the uniform case, due to pruning and locality. As the radius of the computed subset becomes very large, the solution size becomes very small, since a single item covers almost the entire dataset, this is why all lines of Figure 25 begin to converge for $r > 0.5$. 

ACM Transactions on Embedded Computing Systems, Vol. V, No. N, Article A, Publication date: January YYYY.
8.3. Weighted DisC

Next, we consider the weighted DisC problem and evaluate the quality (i.e., size, weight and diversity) of the results produced by various algorithms when the weight of the items is taken into account. We report results for Basic-DisC, Greedy-DisC, Lazy-Greedy-DisC and Greedy-C.

Solution Size. The produced subsets are slightly larger than those of the unweighted version (Table III) in all cases (except from Basic-DisC which is the same, since Basic-DisC produces an independent and covering subset without considering the size or the weight of the resulting subset). This happens because our algorithms are now selecting items based on both the weight and the neighborhood size of the items and, thus, items with large weights enter the diverse set even if they do not cover as many other items. Finally, note that, when weights are considered, the subsets produced for “Uniform-Clustered” are generally smaller than those of “Uniform-Uniform”, since in this case items with larger weights are closer to each other.

Average Weight. Figure 26 shows the average weight of the subsets produced by our algorithms. We also report the average weight of the top-k items with the largest weights, for k equal to the size of the subset generated by Greedy-DisC. Greedy-C achieves a larger average weight. This happens because Greedy-C is not restricted to selecting dissimilar items and, thus, nearby items with large weights can all be selected.

Minimum Distance. Next, we report the minimum pairwise distance among the selected items for the weighted version of the problem as compared to selecting the top-k items with the largest weights, for k equal to the size of the subset generated by Greedy-DisC. Figure 27 shows the results. We see that the results produced by just selecting the items with the top-k weights are very close to each other and thus exhibit very poor diversity. We also see that the increased average weight of Greedy-C (Figure 26) has the trade-off of selecting items that are much closer to each other, especially for smaller radii.

8.4. Multiple Radii DisC

Next, we evaluate some interesting issues concerning using multiple radii.
First, we see how we can use multiple radii so as to tune the number of diverse items selected from a specific area of the dataset. For this experiment, we consider our “Uniform” dataset. We partition the dataset in four areas of equal size and set the radius of all items in the first three areas equal to 0.05, while varying the radius of the items in the fourth area from 0.01 to 0.10. Figure 28(a) and Figure 28(d) report the percentage of selected items from the fourth, “tunable” area, as well as, each of the other three “non-tunable” areas for the Covering and CoveredBy problems respectively. We see that by varying the radius of the “tunable” area, we can over- or under-represent it in the diverse subset, according to our liking.

Figure 28 also shows the size and average weight for the Covering and CoveredBy problems for a clustered dataset, when weights are assigned both uniformly (“Clustered-Uniform”) or in a clustered manner (“Clustered-Clustered”). For the Covering problem, we assign larger radii to items with larger weight, while for the CoveredBy problem, we assign smaller radii to items with larger weight. In all cases, radii values are in \( (0, 2r] \), where \( r \) is the one shown in the x-axis of the figures. We see that, in both cases, the CoveredBy variation achieves a larger average weight at the tradeoff of larger diverse subset size. The difference in average weight is more evident in the case of clustered weights, since, using CoveredBy, it is more difficult for items with large weight to block other items with large weight that are close to them from entering the diverse subset.

8.5. Zooming.

In the following, we evaluate our zooming algorithms. We begin with the zooming-in algorithms. To do this, we first generate solutions with Greedy-DisC for a specific radius \( r \) and then adapt these solutions for radius \( r' \). We use Greedy-DisC because it
A:36

Fig. 27: Minimum distance among the selected items for Greedy-DisC, Greedy-C and the top-weighted items for the weighted case.

The Jaccard distance is 0 when $S_1 = S_2$ and 1 when the $S_1$ and $S_2$ are disjoint. Figures 29(a), 29(d) report the corresponding results for different radii. Due to space limitations, we report results for the “Clustered” and “Cities” datasets. Similar results are obtained for the other datasets as well. Each solution reported for the zooming-in algorithms is adapted from the Greedy-DisC solution for the immediately larger radius and, thus, the x-axis is reversed for clarity; e.g., the zooming solutions for $r = 0.02$ in Figure 29(b) and Figure 29(c) are adapted from the Greedy-DisC solution for $r = 0.03$.

We observe that the zooming-in algorithms provide similar solution sizes with Greedy-DisC in most cases, while their computational cost is smaller, even for Greedy-Zoom-In. More importantly, the Jaccard distance of the adapted solutions for $r'$ to the Greedy-DisC solution for $r$ is much smaller than the corresponding distance of the Greedy-DisC solution for $r'$. This means that computing a new solution for $r'$ from scratch changes most of the items returned to the user, while a solution computed by a zooming-in algorithm maintains many common items in the new solution. Therefore, the new diverse subset is intuitively closer to what the user expects to receive.
Similar results hold for the zooming-out algorithms (we omit related figures). Greedy-Zoom-Out(c) achieves the smallest adapted DisC diverse subsets. However, its computational cost is very high and generally exceeds the cost of computing a new solution from scratch. Greedy-Zoom-Out(a), which selects in the first pass red items with large number of red neighbors, also achieves similar solution sizes with Greedy-Zoom-Out(c), which selects in the first pass red items with large number of white neighbors, while its computational cost is much lower. The non-greedy algorithm has the lowest computational cost. Again, all the Jaccard distances of the zooming-out algorithms to the previously computed solution are smaller than that of Greedy-DisC, which indicates that a solution computed from scratch has only a few items in common from the initial DisC diverse set.

8.6. Comparison with Other Methods.

Next, we compare the diverse subsets produced by DisC with those produced by other diversification methods, in particular by MAXMIN and MAXSUM. To implement MAXMIN and MAXSUM, we used the widely used greedy algorithms for these problems [Drosou and Pitoura 2010; Gollapudi and Sharma 2009]. MAXMIN selects items so as to maximize their minimum pairwise distance, while MAXSUM so as to maximize their average pairwise distance. Both algorithms take as input the desired size $k$ of the result. To compare the results, we compute DisC for various values of $r$. For each value of $r$, we compute MAXMIN and MAXSUM setting $k$ equal to the size of the result produced by DisC. In Figure 30, we report the minimum and average pairwise distances of the results. For comparison, we also report the minimum and average distance among $k$ random items. Note that, MAXMIN and MAXSUM attempt to optimize only the minimum and average distance respectively and do not consider coverage or
other criteria. We see that the DisC performance resembles that of MAXMIN more than that of MAXSUM. This is because MAXMIN disencourages nearby items from entering the diverse subset, having an effect similar to that of the dissimilarity condition of DisC. MAXSUM, on the other hand, often results in a minimum distance close to zero, since no such restriction applies. For this reason, since many items from the outskirts of the dataset can be selected by MAXSUM (see also Figure 18), the average distance of the diverse set is higher than that in the case of both DisC and MAXMIN.

**Fixing the size of the diverse subset to** $k$. Most diversification methods in the literature aim at selecting a diverse subset of fixed size $k$. The DisC model takes as input a radius $r$ and produces a diverse subset that satisfies the dissimilarity and coverage conditions, without placing a limit on its size. In Figure 31, we use a limit $k$ on the number of selected items retrieved by Greedy-DisC and measure the coverage of the produced diverse sets. Clearly, coverage increases more rapidly for larger radii, since the same number of items are able to cover a larger area.

For comparison, we also measure the coverage achieved by the greedy MAXMIN and MAXSUM algorithms. To do this, we first execute the algorithms for various $k$ values and then measure the minimum distance among the selected $k$ items. We assume this distance to be the “radius” of the respective diverse subsets, since for any “radius” larger than this, the corresponding diverse subsets contain items that are not independent. We see that the coverage achieved by both MAXMIN and MAXSUM is smaller than that achieved by Greedy-DisC for the same $k$. This difference is even more evident for the “Clustered” dataset, since in that case items are closer to each other.
A table similar to Table IV can be exploited to estimate a good value for $r$ in case the user wants to place a budget $k$ on the number of retrieved DisC diverse items. Aiming at maximizing coverage, we execute Greedy-DisC for different values of $r$ and locate the $r$ that produces a diverse subset whose size is the closest to the requested $k$. For efficiency, we can employ a binary-search approach for selecting values of $r$ to use.

9. RELATED WORK
In this section, we first overview other diversity definitions and discuss how our DisC definition of diversity is related to them. Then, we review related work on the use of indices for the efficient implementation of diverse item selection and present related work from the field of graph theory.

Other Diversity Definitions. Diversity has recently attracted a lot of attention as a means of counter-acting the over-specialization problem and enhancing user satisfaction [Vieira et al. 2011; Angel and Koudas 2011; Gollapudi and Sharma 2009; Borodin et al. 2012]. Diverse results have been defined in various ways [Drosou and Pitoura 2010], namely in terms of content (or similarity), novelty and semantic coverage.

Most content-based definitions (e.g., [Zhang and Hurley 2008]) interpret diversity as an instance of the $p$-dispersion problem, which is generally defined as selecting $p$ out of $n$ items, so that some objective function based on the chosen items is optimized. A number of variations have been extensively studied (e.g., [Erkut et al. 1994; S. S. Ravi and Tayi 1994; Chandra and Halldórsson 1996]). The objective most usually employed is
Table IV: Radius values and corresponding \( k \) values.

| \( r \) | 0.005 | 0.01 | 0.02 | 0.03 | 0.04 | 0.05 | 0.06 | 0.07 | 0.08 | 0.09 | 0.10 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| \( k \) (Uniform) | 6832 | 3213 | 1125 | 565 | 352 | 233 | 173 | 134 | 106 | 84 | 69 |
| \( k \) (Clustered) | 2585 | 1060 | 378 | 207 | 130 | 89 | 65 | 48 | 40 | 32 | 30 |

Fig. 31: Coverage when limiting the number of diverse items to \( k \).

that of maximizing the minimum distance among any pair of selected items. (MAXMIN diversification problem, e.g., [Drosou and Pitoura 2012b]). Other works consider the MAXSUM diversification problem instead (e.g., [Vieira et al. 2011; Borodin et al. 2012]), whose objective is to select \( p \) out of \( n \) items, so that the average distance between the chosen items is maximized. Our approach differs from those two traditional diversification problems, in that the size of the diverse subset is not an input parameter. Instead, users can explicitly define the desired degree of diversification via the radius \( r \) and later adapt the retrieved solutions by tuning \( r \) to see more or less diverse results.

Diversity is often combined with other criteria, most often that of relevance, to select items that are both highly relevant to a user query, as well as, diverse to each other. Assuming that each item \( p_i \) is associated with some weight \( w(p_i) \), a common way to combine the two criteria is to use a diversification factor \( \lambda \), \( \lambda > 0 \), and select the subset with the largest value of \( \min_{p_i \in S} w(p_i) + \lambda \min_{p_i, p_j \in S} d(p_i, p_j) \) and

\[
(k - 1) \sum_{p_i \in S} w(p_i) + 2\lambda \sum_{p_i, p_j \in S} d(p_i, p_j)
\]

for the MAXMIN and MAXSUM problems respectively (e.g., [Gollapudi and Sharma 2009]). Another common approach is Maximal Marginal Relevance (MMR) [Carbonell and Goldstein 1998], in which weights and diversity are linearly combined when items are selected. Our approach here is different, since we are not restricted by the number \( k \) of selected items but, instead, seek a subset of dissimilar items that can cover the available space. Therefore, we favor smaller DisC subsets containing highly relevant items. Also, we can employ multiple radii to tune the importance of different areas in the dataset, as opposed to treating all items in the same way as in those traditional approach.

Another line of research aims at selecting diverse results similarly to top-\( k \) results by employing some sort of threshold algorithm, often attempting to incorporate weights (e.g., [Qin et al. 2012; Capannini et al. 2011]). This approach is more common in novelty-based definitions of diversity (e.g., [Clarke et al. 2008; Zhang et al. 2002]). There is a crucial difference between the two problems, however, in that the diversity of a single item cannot be computed independently from the other items as in the
top-$k$ case, since all diversity measures require comparing the item with any previously selected ones. In this spirit, [Catallo et al. 2013] presents an approach for low-dimensional vector spaces in which the computation of the solution requires the availability of both relevance-based and distance-based sorted access methods. A number of variations of sorted and random accesses are also employed in [Angel and Koudas 2011] However, the focus of that work is on scheduling the order of the various accesses for cost efficiency rather than maximizing the quality of the retrieved solutions. Generally, such diversity threshold scores are hard to interpret, since they do not depend solely on the item. Instead, the score of each item is relative to which items precede it in the rank. Our approach is fundamentally different in that we treat the result as a whole and select DisC diverse subsets of it that fully cover it.

Another related problem is that of extending nearest neighbor search to selecting $k$ neighbors that are not only spatially close to the query item but are also diverse to each other [Santos et al. 2013; Jain et al. 2004; Abbar et al. 2013]. Such works usually focus on exploiting thresholds and space pruning techniques to enforce diversity during the retrieval of the nearest neighbors of a query item. Our work is different since our goal is not to locate the nearest and most diverse neighbors of a single item but rather to locate an independent and covering subset of the whole dataset. On a related issue, selecting $k$ representative skyline items is considered in [Tao et al. 2009], where the distance between a non-selected skyline item from its nearest selected item is minimized and [Valkanas et al. 2013], where the dominance relationships among items are exploited.

Clustering is a research field related to that of diversification, since cluster medoids can be viewed as representative items. Medoids were extended in [Boim et al. 2011] to include some sense of relevance (priority medoids). However, there are fundamental differences between the two problems, since clustering aims at selecting representatives that minimize some intra-cluster distance, which leads medoids to be drawn to dense areas of items. Thus, items selected by clustering algorithms may not be as distant from each other as items selected by diversification algorithms. Perhaps the clustering works mostly related to our are those on detecting distance-based outliers (e.g., [Knorr and Ng 1998]), i.e., items for which there are less than $m$ other items lying at distance at most $r$ from it. A major difference to our work is that the decision whether an item is an outlier depends only on the size of its neighborhood. This is not the case for $r$-DisC diverse items, since the decision to select an item as $r$-DisC diverse affects other items as well. Our zooming operations bear some resemblance to hierarchical clustering [Reddy and Vinzamuri 2013], which aims at building a hierarchy of clusters, so that smaller clusters at the bottom of the hierarchy are gradually merged into larger ones as we move up the hierarchy. The diversification approach in [Liu and Jagadish 2009] is built upon the same basic idea, where a tree similar to a dendrogram is first constructed and is later exploited by moving - or “zooming” - between levels, each time retrieving the items indexed at the specific level.

Finally, the problem of diversifying continuous data has been recently considered in [Drosou and Pitoura 2012b; Panigrahi et al. 2012; Minack et al. 2011] using a number of variations of the MAXMIN and MAXSUM diversification models.

Index-Based Implementations of Diversification Algorithms. Due to the NP-hardness of the diversification problem (e.g., [Deng and Fan 2013]), many heuristics have been proposed for locating approximate solutions. Most of these can be classified as either greedy or interchange (swap) heuristics [Drosou and Pitoura 2010]. In greedy heuristics, items are iteratively selected in rounds. Their complexity in terms of computed distances ranges from $O(k^2n)$ to $O(n^2)$ depending on the initialization step, while
\(1/2\)-approximations of the optimal solutions can be achieved for both the MaxMin and MaxSum problems (e.g., [Tamir 1991]). Interchange heuristics initialize \(S\) with a random solution and then iteratively attempt to improve it by interchanging an item in \(S\) with another item that is not in \(S\). Their worst case complexity is \(O(n^k)\).

Here, we used the M-tree to implement our approach and exploited its properties, as well as our pruning rule, to reduce computational cost. Indices have been used in the past for diversification, most recently in [Drosou and Pitoura 2012b], where a number of algorithms based on Cover Trees are proposed for the MaxMin diversification problem. In [Vee et al. 2008], a Dewey encoding of database tuples enables them to be organized in a tree structure which is later exploited to select the \(k\) most diverse of them. A similar approach is followed in [Li and Chan 2013]. However, the proposed methods are limited to a specific diversity measure. A spatial index is also exploited in [Haritsa 2009] to locate those relevant nearest neighbors of an item that are the most distant to each other.

Results from Graph Theory. The properties of independent and dominating (or covering) subsets have been extensively studied in graph theory. A number of different variations exist (e.g., [Halldórsson 1993; Gibson and Pirwani 2010]). Among these, the Minimum Independent Dominating Set problem is equivalent to the \(r\)-DisC diversity problem. This problem has been shown to have some of the strongest negative approximation results: in the general case, it cannot be approximated in polynomial time within a factor of \(n^{1-\epsilon}\) for any \(\epsilon > 0\) unless \(P = NP\) [Halldórsson 1993]. However, some approximation results and faster algorithms have been found for special graph cases (e.g., [Chlebík and Chlebíková 2008; Gamarnik et al. 2010; da Fonseca et al. 2012; Bourgeois et al. 2013]). Weighted variations of the problem also exist (e.g., [Zou et al. 2011; Kako et al. 2009]). In our work, rather than providing polynomial approximation bounds for DisC diversity, we focus on the efficient computation of non-minimum but small DisC diverse subsets and their adaptation to new radii.

Related work on directed graphs is considerably more limited. An extra challenge in this case is that not all directed graphs have an independent dominating set. In [Milner and Woodrow 1989] a number of conditions for the existence of an independent dominating set for a number of different kinds of graphs are provided. Some related work also exists on locating minimum dominating (but not independent) sets (e.g., [Pang et al. 2010; Chartrand et al. 1999]).

Finally, there is a substantial amount of related work concerning Minimum Connected Dominating Sets (e.g., [Thai et al. 2007b]). However, allowing the dominating set to be connected has an impact on the complexity of the problem and allows different algorithms to be designed. Here, we require diverse items to be dissimilar to each other, thus, such approaches cannot be exploited.

10. SUMMARY

In this paper, we proposed a novel definition of diversity as the problem of selecting a minimum representative subset \(S\) of a result \(P\), such that each item in \(P\) is represented by a similar item in \(S\) and that the items included in \(S\) are not similar to each other. Similarity is modeled by a radius \(r\) around each item. We call such subsets \(r\)-DisC diverse subsets of \(P\). We introduced weighted and multiple radii variations of DisC subsets and, also, adaptive diversification through decreasing \(r\), termed zooming-in, and increasing \(r\), called zooming-out. Since locating minimum \(r\)-DisC diverse subsets is an NP-hard problem, we introduced algorithms for computing approximate solu-
tions, including incremental ones for zooming, and provided corresponding theoretical bounds. We also presented efficient implementations based on spatial indexing.

REFERENCES

Sofiane Abbar, Sihem Amer-Yahia, Piotr Indyk, Sepideh Mahabadi, and Kasturi R. Varadarajan. 2013. Diverse near neighbor problem. In SoCG. 207–214.

Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. 2009. Diversifying search results. In WSDM. 5–14.

Albert Angel and Nick Koudas. 2011. Efficient diversity-aware search. In SIGMOD. 781–792.

Claude Berge. 1962. The Theory of Graphs and Its Applications. Methuen.

Rubi Boim, Tova Milo, and Slava Novgorodov. 2011. Diversification and refinement in collaborative filtering recommender. In CIKM. 739–744.

Allan Borodin, Hyun Chul Lee, and Yuli Ye. 2012. Max-Sum diversification, monotone submodular functions and dynamic updates. In PODS. 155–166.

Nicolas Bourgeois, Federico Della Croce, Bruno Escoffier, and Vangelis Th. Paschos. 2013. Fast algorithms for min independent dominating set. Discrete Applied Mathematics 161, 4-5 (2013), 558–572.

Gabriele Capannini, Franco Maria Nardini, Raffaele Perego, and Fabrizio Silvestri. 2011. Efficient Diversification of Web Search Results. PVLDB 4, 7 (2011), 451–459.

Jaime G. Carbonell and Jade Goldstein. 1998. The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. In SIGIR. 335–336.

Ilios Bourgeois, Federico Della Croce, Bruno Escoffier, and Vangelis Th. Paschos. 2013. Top-k diversity queries over bounded regions. ACM Trans. Database Syst. 38, 2 (2013), 10.

Barun Chandra and Magnus M. Halldörsdóttir. 1996. Facility Dispersion and Remote Subgraphs. In SWAT. 53–65.

Gary Chartrand, Frank Harary, and Bill Quan Yue. 1999. On the out-domination and in-domination numbers of a digraph. Discrete Mathematics 197-198 (1999), 179–183.

Miroslav Chlebík and Janka Chlebíková. 2008. Approximation hardness of dominating set problems in bounded degree graphs. Inf. Comput. 206, 11 (2008), 1264–1275.

Paolo Ciaccia, Marco Patella, and Pavel Zezula. 1997. M-tree: An Efficient Access Method for Similarity Search in Metric Spaces. In VLDB. 426–435.

Brent N. Clark, Charles J. Colbourn, and David S. Johnson. 1990. Unit disk graphs. Discrete Mathematics 86, 1-3 (1990), 165–177.

Charles L. A. Clarke, Maheedhar Kolla, Gordon V. Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. 2008. Novelty and diversity in information retrieval evaluation. In SIGIR. 659–666.

Guilherme Dias da Fonseca, Celina M. Herrera de Figueiredo, Vinicius G. P. de Sa, and Raphael Machado. 2012. Linear Time Approximation for Dominating Sets and Independent Dominating Sets in Unit Disk Graphs. CoRR abs/1204.3488 (2012).

Ting Deng and Wenfei Fan. 2013. On the Complexity of Query Result Diversification. PVLDB 6, 8 (2013), 577–588.

Marina Drosou and Evaggelia Pitoura. 2010. Search result diversification. SIGMOD Record 39, 1 (2010), 41–47.

Marina Drosou and Evaggelia Pitoura. 2012a. DisC diversity: result diversification based on dissimilarity and coverage. PVLDB 6, 1 (2012), 13–24.

Marina Drosou and Evaggelia Pitoura. 2012b. Dynamic diversification of continuous data. In EDBT. 216–227.

Erhan Erkut, Yilmaz Uluksal, and Oktay Yeniçerioğlu. 1994. A comparison of p-dispersion heuristics. Computers & OR 21, 10 (1994), 1103–1113.

David Gamarnik, David Goldberg, and Theophane Weber. 2010. PTAS for Maximum Weight Independent Set Problem with Random Weights in Bounded Degree Graphs. In SODA. 268–278.

M. R. Garey and David S. Johnson. 1979. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman.

Matt Gibson and Imran A. Pirwani. 2010. Approximation Algorithms for Dominating Set in Disk Graphs. CoRR abs/1004.3320 (2010).

Sreenivas Gollapudi and Anesh Sharma. 2009. An axiomatic approach for result diversification. In WWW. 381–390.
A:44

Magnús M. Halldórsson. 1993. Approximating the Minimum Maximal Independence Number. *Inf. Process. Lett.* 46, 4 (1993), 169–172.

Jayant R. Haritsa. 2009. The KNDN Problem: A Quest for Unity in Diversity. *IEEE Data Eng. Bull.* 32, 4 (2009), 15–22.

Anoop Jain, Parag Sarda, and Jayant R. Haritsa. 2004. Providing Diversity in K-Nearest Neighbor Query Results. In *PAKDD*. 404–413.

Caetano Traina Jr., Agma J. M. Traina, Christos Faloutsos, and Bernhard Seeger. 2002. Fast Indexing and Visualization of Metric Data Sets using Slim-Trees. *IEEE Trans. Knowl. Data Eng.* 14, 2 (2002), 244–260.

Akihisa Kako, Takao Ono, Tomio Hirata, and Magnús M. Halldórsson. 2009. Approximation algorithms for the weighted independent set problem in sparse graphs. *Discrete Applied Mathematics* 157, 4 (2009), 617–626.

Edwin M. Knorr and Raymond T. Ng. 1998. Algorithms for Mining Distance-Based Outliers in Large Datasets. In *VLDB*. 392–403.

Lu Li and Chee-Yong Chan. 2013. Efficient Indexing for Diverse Query Results. *PVLDB* 6, 9 (2013), 745–756.

Bin Liu and H. V. Jagadish. 2009. Using Trees to Depict a Forest. *PVLDB* 2, 1 (2009), 133–144.

Eric C. Milner and Robert E. Woodrow. 1989. On directed graphs with an independent covering set. *Graphs and Combinatorics* 5, 1 (1989), 363–369.

Enrico Minack, Wolf Siberski, and Wolfgang Nejdl. 2011. Incremental diversification for very large sets: a streaming-based approach. In *SIGIR*. 585–594.

Chaoyi Pang, Rui Zhang, Qing Zhang, and Junhu Wang. 2010. Dominating sets in directed graphs. *Inf. Sci.* 180, 19 (2010), 3647–3652.

Debmalya Panigrahi, Atish Das Sarma, Gagan Aggarwal, and Andrew Tomkins. 2012. Online selection of diverse results. In *WSDM*. 263–272.

Lu Qin, Jeffrey Xu Yu, and Lijun Chang. 2012. Diversifying Top-K Results. *PVLDB* 5, 11 (2012), 1124–1135.

Chandan K. Reddy and Bhanukiran Vinzamuri. 2013. A Survey of Partitional and Hierarchical Clustering Algorithms. In *Data Clustering: Algorithms and Applications*. 87–110.

Daniel J. Rosenkrantz S. S. Ravi and Giri Kumar Tayi. 1994. Heuristic and Special Case Algorithms for Dispersion Problems. *Operations Research* 42, 2 (1994), 299–310.

Lucio F. D. Santos, William D. Oliveira, Mónica Ribeiro Porto Ferreira, Agma Juci Machado Traina, and Caetano Traina Jr. 2013. Parameter-free and domain-independent similarity search with diversity. In *SSDBM*. 5.

Idan Szpektor, Yoelle Maarek, and Dan Pelleg. 2013. When relevance is not enough: promoting diversity and freshness in personalized question recommendation. In *WWW*. 1249–1260.

Arie Tamir. 1991. Obnoxious Facility Location on Graphs. *SIAM J. Discrete Math.* 4, 4 (1991), 550–567.

Yufei Tao, Ling Ding, Xuemin Lin, and Jian Pei. 2009. Distance-Based Representative Skyline. In *ICDE*. 892–903.

My T. Thai, Feng Wang, Dan Liu, Shiwei Zhu, and Ding-Zhu Du. 2007a. Connected Dominating Sets in Wireless Networks with Different Transmission Ranges. *IEEE Trans. Mob. Comput.* 6, 7 (2007), 721–730.

My T. Thai, Ning Zhang, Ravi Tiwari, and Xichun Xu. 2007b. On approximation algorithms of k-connected m-dominating sets in disk graphs. *Theor. Comput. Sci.* 385, 1-3 (2007), 49–59.

George Valkanas, Apostolos N. Papadopoulos, and Dimitrios Gunopulos. 2013. SkyDiver: a framework for skyline diversification. In *EDBT*. 406–417.

Erik Vee, Utkarsh Srivastava, Jayavel Shanmugasundaram, Prashant Bhat, and Sihem Amer-Yahia. 2008. Efficient Computation of Diverse Query Results. In *ICDE*. 228–236.

Marcos R. Vieira, Humberto Luiz Razente, Maria Camila Nardini Barioni, Marios Hadjileftheriou, Divesh Srivastava, Caetano Traina Jr., and Vassilis J. Tsotras. 2011. On query result diversification. In *ICDE*. 1165–1174.

Isaac Ottoni Wilhelm. 2008. Packing Triangles on a Sphere. In http://www.math.uchicago.edu/~may/VIGRE/VIGRE2008/REUPapers/Willhelm.pdf.

Kai Xing, Wei Cheng, E. K. Park, and Shmuel Rotenstein. 2008. Distributed Connected Dominating Set Construction in Geometric k-Disk Graphs. In *ICDCS*. 673–680.

Cong Yu, Laks V. S. Lakshmanan, and Sihem Amer-Yahia. 2009. It takes variety to make a world: diversification in recommender systems. In *EDBT*. 368–378.

Mi Zhang and Neil Hurley. 2008. Avoiding monotony: improving the diversity of recommendation lists. In *RecSys*. 129–130.

ACM Transactions on Embedded Computing Systems, Vol. V, No. N, Article A, Publication date: January YYYY.
Yi Zhang, James P. Callan, and Thomas P. Minka. 2002. Novelty and redundancy detection in adaptive filtering. In SIGIR. 81–88.
Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. 2005. Improving recommendation lists through topic diversification. In WWW. 22–32.
Feng Zou, Yuexuan Wang, Xiaohua Xu, Xianyue Li, Hongwei Du, Peng-Jun Wan, and Weili Wu. 2011. New approximations for minimum-weighted dominating sets and minimum-weighted connected dominating sets on unit disk graphs. Theor. Comput. Sci. 412, 3 (2011), 198–208.