

# Evaluating the Diversification of Similarity Query Results

Lúcio F. D. Santos, Willian D. Oliveira, Mônica R. P. Ferreira,  
Robson L. F. Cordeiro, Agma J. M. Traina, Caetano Traina Jr.

Database and Images Group – GBdl  
Institute of Mathematics and Computer Science – ICMC  
University of São Paulo – USP  
São Carlos, SP, Brazil  
{luciodb, willian, monika, robson, agma, caetano}@icmc.usp.br

**Abstract.** The data currently generated and collected increase not only in volume, but also in complexity, requiring new query operators to be searched. Similarity queries have been acknowledged as one of the most useful resources to retrieve complex data, but the basic similarity operators are not enough to meet the requirements of the applications, largely because their result sets tend to include many elements too similar to the query center and among themselves. To tackle this problem, variations and extensions of basic operators have been studied pursuing result diversification, i.e., to search for elements sufficiently similar to the query center, but also diverse from each other. Result diversification has been studied considering either extra information related to the data or the distance among result set elements. The problem with the former approach is that “extra information” rarely exists and, even when it does, the corresponding processing cost is commonly too high. Moreover, the distance-based algorithms are often good alternatives even for data domains that can rely on other information, besides the elements and their distances. The main drawback of distance-based algorithms is the lack of evaluation methods to understand how diverse the retrieved answer is. This article reports on the development of several statistical measurements able to evaluate the diversity of the result set. The concept of the “answer space”, has also been created, aimed at highlighting the distribution of the several result sets that can be the answers to a given similarity-diversified query, which enables the evaluation of the query quality regarding several different criteria. Finally, we describe an extensive set of experiments to validate our proposals and highlight the analysis that could be performed by the system analyst, using four real datasets that span up to 72k elements and 761 dimensions.

Categories and Subject Descriptors: H.2 [Database Management]: Miscellaneous; H.3 [Information Storage and Retrieval]: Miscellaneous

Keywords: Evaluation methods, Result diversification, Similarity search, Space mapping

## 1. INTRODUCTION

Recently, several studies have been conducted aimed at getting more efficient similarity query execution [Skopal et al. 2009]. Better performance is commonly obtained by indexing structures, which is even more demanded when applied to large, complex data. Another issue of foremost importance is the improvement of the efficacy of the answers, i.e., by avoiding returning too similar elements in the result set. For example, assume that a student will join a conference in São Paulo and decides to seek information in a search engine on the Internet. It is easy to observe that a search for the term “São Paulo” has more interesting results if there are references to the city, the state, the soccer team, the aircraft, the saint, restaurant guides and cultural spots, than if they are all concentrated in one of those topics. To deal with this problem, research areas, such as information retrieval [Ziegler et al.

---

This research was supported by FAPESP, CAPES and CNPq.

Copyright©2012 Permission to copy without fee all or part of the material printed in JIDM is granted provided that the copies are not made or distributed for commercial advantage, and that notice is given that copying is by permission of the Sociedade Brasileira de Computação.

2005] and recommendation systems [Agrawal et al. 2009] have introduced the “diversity property” to result sets, whose goal is to retrieve elements similar enough to the query center to meet the query predicate, but also diverse enough to generate a heterogeneous, more interesting and useful result set.

Most of the result diversification approaches take advantage of extra information related to the data (i.e., metadata), such as taxonomies [Ziegler et al. 2005; Agrawal et al. 2009], spatial structure of the data [Vee et al. 2008], cluster attributes [Chen and Li 2007; van Leuken et al. 2009], query logs and users’ expectation [Capannini et al. 2011]. Following this idea, the quality of query answers is dependent on the existence of *a priori* extra information. Unfortunately, extra information is rare in many real scenarios, and processing it always leads to higher computational costs.

Another result diversification approach is based on the exploration of the domain of distances among elements in the target data space. The distance-based approaches do not require any other information besides the data elements and their pair-wise similarities, which are evaluated by a distance function [Carbonell and Goldstein 1998; Vieira et al. 2011; Yu et al. 2009; Gollapudi and Sharma 2009; Drosou and Pitoura 2012; Skopal et al. 2009]. However, there are few evaluation methods able to accurately measure the diversity of the result sets retrieved by such algorithms, aimed at highlighting one’s understanding of what kind of diversity is retrieved by each of the existing approaches.

In this article we focus on the effectiveness evaluation of distance-based approaches, and we propose a novel set of methods to measure the accuracy of similarity-with-diversity retrieval algorithms, based on the use of several statistical analyses obtained from different strategies for measuring result set diversity. We also propose the definition of an “answer space”, in which each element is a possible query solution. The space enables the comparison of queries and the definition of properties that allow developing guidelines to choose the best-suited similarity-with-diversity retrieval algorithm to answer queries over a given dataset. To validate our proposals, we performed an extensive experimental evaluation using four real datasets that span up to 72k elements and 761 dimensions. Our experiments show how each existing algorithm compares with each other, and confirm that those considered to the best ones indeed perform better than the others in most situations. Moreover, our results pinpoint for which types of datasets these algorithms in fact thrive and for which ones they do not, thus indicating “where” there is still room for improvement in result diversification research.

The remainder of the article follows a traditional organization: related work (Section 2), proposed techniques (Section 3), experiments (Section 4), and conclusions (Section 5). The symbols of our notation are listed in Table I.

Table I: Table of Symbols

Symbols	Definitions	Symbols	Definitions
$\mathfrak{F}$	Diversity algorithm	$R_t$	Target result set ( $R_t \in R$ )
$\mathfrak{L}$	Diversity metric extractor	$R_r$	Reference result set ( $R_r \in R$ )
$\mathbb{S}$	Similarity domain	$r_u, r_v$	Result set elements ( $r_u, r_v \in R_i$ )
$\mathbb{L}$	Answer space	$k$	Number of elements in the result set
$S$	Dataset ( $S \in \mathbb{S}$ )	$\delta_{div}$	Diversity distance
$s_q$	Query center ( $s_q \in \mathbb{S}$ )	$\delta_{sim}$	Similarity distance
$R$	Set of result sets ( $R = \bigcup R_{is}$ )	$DV$	Set of diversity feature vectors
$R_i$	$i^{th}$ result set ( $R_i \subset S, R_i \in R$ )	$dv_i$	$i^{th}$ diversity feature vectors ( $dv_i \in \mathbb{L}$ )

## 2. RELATED WORK

### 2.1 Result Diversification Methods

Result diversification has been examined in various ways of different areas [Drosou and Pitoura 2012]. Most approaches use metadata, besides similarity and diversity distances among the elements [Vieira et al. 2011; Agrawal et al. 2009; Gollapudi and Sharma 2009; Carbonell and Goldstein 1998]. Examples of such additional information are user expectation and query logs in web searches [Angel and Koudas 2011; Capannini et al. 2011], taxonomies of terms when searching in textual datasets [Agrawal et al. 2009; Ziegler et al. 2005], and cluster attributes in annotated data [Chen and Li 2007; van Leuken et al. 2009]. However, the processing of external information is often computationally expensive and it commonly provides suboptimal results, since workloads and query requirements are seldom known in advance [Vieira et al. 2011].

Other approaches for result diversification have also been pursued without extra information. The existing methods can be classified into two main groups: optimization and separation distance. The optimization approaches target  $k$ -nearest neighbor queries only. In such approaches, similarity and diversity compete with each other, taking one user-defined diversity preference ( $\lambda$ ) as input, so that the result of basic similarity algorithms can be re-ranked, inducing diversity among elements based on a *trade-off* objective function  $F$  [Carbonell and Goldstein 1998; Yu et al. 2009; Vieira et al. 2011].

When  $\lambda > 0$ , the exhaustive solution for the  $k$ -nearest diversification set can be found by a brute force algorithm. It tests every possible subset  $R \subseteq S$  of size  $k$  to find the highest  $F$  value. As the worst case solution is NP-hard, greedy algorithms are employed to build the result set incrementally. Such algorithms often use the initial result set produced by a similarity-based algorithm in which the result elements are as similar as possible to the query center, but ask for more elements than the  $k$  required. Thereafter, the desired  $k$  elements are selected considering the objective function  $F$ . In this article we focus on the max-sum diversification approach, i.e., maximize both, the sum of similarity and diversity distances between result set elements, since it seems the most widely accepted among previous approaches [Carbonell and Goldstein 1998; Gollapudi and Sharma 2009; Yu et al. 2009; Drosou and Pitoura 2010; Vieira et al. 2011].

The performance of the greedy algorithms also differs in function of the construction strategy adopted for the result set  $R$ . Thus, we can classify the existing methods based on this characteristic:

*Incremental*: The result set  $R$  starts empty and is iteratively increased by selecting the element in  $S$  that maximizes the objective function.

*Exchanging*: An initial result set  $R$  is chosen. Thereafter, the remaining elements in  $S$  are evaluated as candidates to replace an element from the current solution  $R$ .

*Meta-heuristic*: An initial result set  $R$  is chosen by a heuristic-based ranking function. Then, a local search improves the current solution  $R$  by iteratively swapping an element in the result set for another.

Several optimization methods have been proposed to efficiently compute  $R$ , based on one of those construction strategies. Table II classifies some representative methods according to the construction strategy used.

The Maximal Marginal Relevance (MMR) [Carbonell and Goldstein 1998] iteratively constructs the result set  $R$  by selecting a new element in  $S$  that maximizes the following function:

$$MMR(s_i) = (1 - \lambda)\delta_{sim}(s_i, s_q) + \frac{\lambda}{|R|} \sum_{s_j \in R} \delta_{div}(s_i s_j) .$$

Table II: Description of some representative optimization approaches in diversification quality separated by the construction strategies.

abbreviation	method name	construction strategy
MMR	Maximal Marginal Relevance	incremental
Swap	Swap	exchanging
GNE	GRASP with Neighbor Expansion	meta-heuristic

The MMR method has two critical properties that influence the elements of the result set  $R$ . First,  $R$  always starts with the element of the highest  $\delta_{sim}$  in  $S$ , regardless of the value  $\lambda$ . Second, since the result is incrementally constructed by inserting a new element into previous results, the first element chosen has a larger influence on the quality of the final result set  $R$ , which may display lower or higher quality in terms of  $F$ , according to the first element chosen.

The *Swap* method [Yu et al. 2009] is twofold. In the first step, the top- $k$  relevant elements in  $S$  define the initial result  $R$ . In the second phase, each remaining element in  $S$ , ordered by decreasing  $\delta_{sim}$  values, is tested to replace an element of the current solution  $R$ . If the tested element improves  $F$ , then a replace operation is permanently applied to  $R$ . This process continues until every element in the candidate set  $S$  has been checked. The final result set may not be optimal, since the candidate set  $S$  is analyzed with respect to their  $\delta_{sim}$  order and does not consider the order of  $\delta_{div}$  values in  $S$ , which can result in solutions that do not maximize  $F$ .

The GNE method was proposed by Vieira et al. [2011]. It uses a Greedy Randomized Adaptive Search Procedure (GRASP) [Feo and Resende 1995] for diversifying query results. Each iteration chooses a random element among the top ranked ones and builds the result set  $R$  by selecting the element of highest maximum marginal contribution (*mmc*) to the current solution, using the function:

$$mmc(s_i) = (1 - \lambda)\delta_{sim}(s_i, s_q) + \frac{\lambda}{k-1} \sum_{s_j \in R_{p-1}} \delta_{div}(s_i, s_j) + \frac{\lambda}{k-1} \sum_{l=1}^{l \leq k-p} \sum_{s_j \in S - s_i} \delta_{div}^l(s_i, s_j) .$$

In this equation,  $R_{p-1}$  is the partial result of size  $p-1$ ,  $1 \leq p \leq k$ , and  $\delta_{div}^l(s_i, s_j)$  gives the  $l^{th}$  highest  $\delta_{div}$  value in  $\{\delta_{div}^l(s_i, s_j) : s_j \in S - R_{p-1-s_i}\}$ .

GNE has two phases. In the Construction Phase, at each iteration, the choice of the next element to be added in  $R$  is determined by a greedy randomized ranking function, which ranks the elements in  $S$  according to *mmc*. Only elements of highest *mmc*, are considered to be stored in a list named the Restricted Candidate List (RCL). An initial result set  $R$  is then randomly chosen from the RCL. Note that it may not be the element of highest contribution in the RCL. In the Local Search Phase, the initial result set is progressively improved by applying a series of local modifications to the neighborhood of the current solution. The local search algorithm swaps elements in the result set  $R$  with the most diverse elements regarding a reference element in  $R$ , whenever it improves the current solution.

The separation distance approach considers that there must exist a minimum distance  $\xi_p$  among pairs of elements in the answer. Pairs of elements closer than  $\xi_p$  are considered too similar to each other and only one element of each pair is included in the answer [Skopal et al. 2009; Haritsa 2009; Gil-Costa et al. 2011; Drosou and Pitoura 2012]. An example of the usage of this approach appears in the  $k$ -Distinct Nearest Neighbors ( $k$ DNN) query, proposed by Skopal et al. [2009]. The First-Match (FM) algorithm uses a fixed user-defined separation distance that specifies the required diversity between result set elements. The  $k$ DNN query builds on the classic  $k$ -NN query, but excluding all elements that are too similar to any of the reported elements.

The First-Match (FM) algorithm is a representative  $k$ DNN variant that can be considered as an extension of the classic  $k$ -NN one. Given a query element  $s_q$ , the algorithm retrieves elements in ascending order with respect to their distances to  $s_q$ . Whenever a distinct element is retrieved from the ordering, it is added to the query result, therefore the elements already reported to the query result have to be far enough from each other.

## 2.2 Diversification Evaluation Methods

In order to evaluate the accuracy of diversification, various measurements have been proposed in the literature [Drosou and Pitoura 2010; Agrawal et al. 2009]. The field of Information Retrieval (IR) systems has been adapting the traditional evaluation measures to take into account the diversity level of query results. The traditional measures, namely NDCG (*normalized discounted cumulative gain*), MRR (*Mean Reciprocal Rank*) and MAP (*Mean Average Precision*) evaluate the result sets based on the position that the elements appears in an ordered list of relevant elements to the query.

An example of this adaptation is provided by Clarke et al. [2008]. The  $\alpha$ -NDCG extends the traditional NDCG to measure the gain of an item being at a specific position of the list taking into account the items that precede it. This measurement is based on the concept of *information nuggets*, which represents a small piece of similar information, as it is commonly referred to in the summarization and question answering communities [Clarke et al. 2008]. The main drawback of this measurement is to require *a priori* knowledge of the nuggets and also considerable amount of human effort to judge the relevance of elements in the list [Drosou and Pitoura 2010]. The *Intent-Aware Normalized Discounted Cumulative Gain Measure* (NDCG-IA) was proposed by Agrawal et al. [2009] and considered the importance of the elements in different categories for the same query, forcing a trade-off between adding elements with higher relevance scores and those that cover additional categories. The same intuition is applied to adapt the MRR and MAP. The main drawback of these measurements is being dependent of on extra information, such as taxonomies. Thus, the adapted version provided by the IR field cannot be applied on distance-based approach, where the only information available are the elements and the distance among them.

To evaluate the accuracy of diversification in distance-based approaches, two measurements are commonly used: the Gap and the objective function measurements [Vieira et al. 2011; Yu et al. 2009; Drosou and Pitoura 2010]. The main difference of these measurements to those from the IR field is that the results are viewed as a set instead of an ordered list. The objective function measure evaluates the maximization in the result sets based on its defined diversity function. For example, considering that two algorithms ( $A$  and  $B$ ) were defined using the same objective function  $F$ . The algorithm  $A$  is considered better than  $B$  if  $F_A$  value is higher than  $F_B$ . The Gap measure is a version of the objective function measure that normalizes the results using the optimal value provided by an exhaustive algorithm (optimal value) [Vieira et al. 2011]. For example, considering that  $F_A$  and  $F_O$  are the values reached by the algorithm  $A$  and the exhaustive algorithm  $O$ , respectively. The Gap measure is the difference between  $F_O$  and  $F_A$ , divided by  $F_O$ .

## 3. DIVERSITY EVALUATION

This section presents our novel concept of the *answer space* based on a set of statistical measurements over query answer sets, aims at comparing the different answers from the algorithms with the plain query answer. We conduct our discussion assuming that  $S = \{s_1, \dots, s_n\}$  is a dataset of  $n$  elements taken from a domain  $\mathbb{S}$  and  $s_q \in \mathbb{S}$  is a query center. Let  $R_i \subset S$  be a result set for the query centered at  $s_q$ , which selects in  $S$  elements similar to  $s_q$ , and also diverse among themselves, following a diversity algorithm  $\mathfrak{F}$ . The set of result sets  $R$  is the union of all result sets  $R_i$  centered at the

same  $s_q$  ( $R = \bigcup R_i$ ). Our goal is to evaluate the quality of the distance-based result diversification algorithms following two strategies to measure diversity: 1) result-based statistics and 2) result set comparisons. The first strategy extracts features from each  $R_i \in R$  and compares them using our proposed Dissimilarity Feature method ( $DiF_M$ ), which is detailed in the upcoming Section 3.2.1. The features are obtained by the diversity metric extractor  $\mathfrak{L}$ , which is based on the distances between elements in  $R_i$ , since it is the only information available. The second strategy of evaluation directly compares the elements of the result sets applying the new Dissimilarity ( $D_M$ ) and the Dissimilarity Error ( $DE_M$ ) evaluation methods, proposed in the upcoming Sections 3.2.2 and 3.2.3, respectively.

Figure 1 illustrates the main components of our proposal. The toy dataset  $S = \{s_1, s_2, \dots, s_{10}\}$  shown in Figure 1(a) is the search space composed of elements from domain  $\mathbb{S}$ . The diversity algorithm  $\mathfrak{F}$  is executed using  $S$  and query center  $s_q$ . Figure 1(b) shows its execution with three different input parameter configurations, generating the set of result sets  $R = \{R_1, R_2, R_3\}$ . For each result set  $R_i$ , the diversity metric extractor  $\mathfrak{L}$  extracts appropriate features and maps them into the answer space  $\mathbb{L}$  (Definition 3.1), as shown in Figure 1(c). Aimed at evaluating the quality of the result, the result sets  $R_1, R_2$  and  $R_3$  can be analyzed by our proposed ‘Diversification Evaluation Method’, using our Dissimilarity and/or Dissimilarity Error evaluation methods (Arrow ① of Figure 1), and/or  $\mathbb{L}$  by our Dissimilarity Feature Method, also presented in the ‘Diversification Evaluation Method’ (Arrow ② of Figure 1). This choice depends on the information pursued by the user during the analysis.

**Definition 3.1. Answer Space ( $\mathbb{L}$ ):** Given  $R$  a set of result sets and a diversity metric extractor  $\mathfrak{L}$ , an answer space  $\mathbb{L}$  is an  $m$ -dimensional space in which  $m$  is the number of features extracted by  $\mathfrak{L}$  and each element in  $\mathbb{L}$  is a distance distribution of a possible diversity solution for a query center  $s_q$ .

The following sections detail the features extracted by our Diversity Metric Extractor  $\mathfrak{L}$  from each result set and the evaluation methods that compose the ‘Diversification Evaluation Method’.

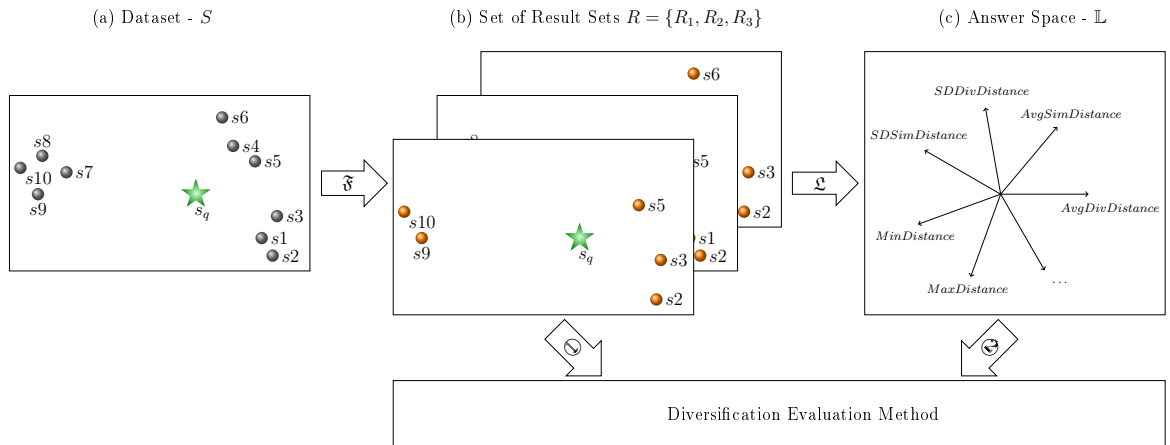


Fig. 1: Answer space mapping.

### 3.1 The Diversity Feature Vector

The Diversity Feature Vector  $dv_i$  represents the features extracted from each result set  $R_i \in R$  in the Answer Space  $\mathbb{L}$ . In this process, the Diversity Metric Extractor  $\mathfrak{L}$  uses each  $R_i$  to extract features that will be used for comparisons among the algorithms. We propose quantitative features that can be directly calculated by  $\mathfrak{L}$  using only the distances between elements of the result set, which is the

only information available. These features capture different statistics that are used by the result-based statistics evaluation method to measure the quality of a result set in the answer space  $\mathbb{L}$ .

Let  $DV = \{dv_1, \dots, dv_n\}$  be the set of diversity feature vectors extracted by  $\mathfrak{L}$  from the set of result sets  $R = \{R_1, \dots, R_n\}$ , that is,  $\mathfrak{L}(R_i) = dv_i$ . Therefore, each  $dv_i$  has features  $(f_1, \dots, f_m)$ , in which  $m$  is the number of features extracted by  $\mathfrak{L}$ , i.e., the number of dimensions of the answer space  $\mathbb{L}$ . We have assumed that  $\mathbb{L}$  is 6-dimensional and  $dv_i$  is composed of the six features described as follows:

- i. Average Diversity Distance (*AvgDivDistance*), which represents how diverse the result set elements are among each other. It is estimated by the average diversity distance from the result set elements, as shown in Equation 1.

$$AvgDivDistance(R_i) = \frac{\sum_{u=1}^{|k-1|} \sum_{v=u+1}^{|k|} \delta_{div}(r_u, r_v)}{k \cdot (k-1)} . \quad (1)$$

- ii. Average Similarity Distance (*AvgSimDistance*), which represents how similar the result set elements are to the query center. It refers to the average similarity distance between  $s_q$  and every result set element, as shown in Equation 2.

$$AvgSimDistance(R_i, s_q) = \frac{\sum_{u=1}^{|k|} \delta_{sim}(r_u, s_q)}{k} . \quad (2)$$

- iii. Standard Deviation Diversity Distance (*SDDivDistance*), which measures the dispersion of individual diversity distances on  $R_i$  in comparison to *AvgDivDistance*, i.e., *SDDivDistance* checks whether or not the distances between the elements in the result set are equally distributed. If *SDDivDistance* has a value close to zero, the elements are equally spaced, while higher values indicate the presence of clusters in  $R_i$ . *SDDivDistance* is calculated by Equation 3.

$$SDDivDistance(R_i, s_q) = \sqrt{\frac{\sum_{u=1}^{|k-1|} \sum_{v=u+1}^{|k|} (\delta_{div}(r_u, r_v) - AvgDivDistance(R_i))^2}{k \cdot (k-1)}} . \quad (3)$$

- iv. Standard Deviation Similarity Distance (*SDSimDistance*), which measures the dispersion of individual similarity distances in comparison to *AvgSimDistance*, i.e., *SDSimDistance* checks if the distances between every result set element and  $s_q$  are equally distributed. *SDSimDistance* is calculated by Equation 4.

$$SDSimDistance(R_i, s_q) = \sqrt{\frac{\sum_{u=1}^{|k|} (\delta_{sim}(r_u, s_q) - AvgSimDistance(R_i))^2}{k}} . \quad (4)$$

- v. Minimum Distance (*MinDistance*), which represents the smallest diversity distance between any pair of elements in the result set. It is defined in Equation 5.

$$MinDistance(R_i) = \min_{r_u, r_v \in R_i} (\delta_{div}(r_u, r_v)) . \quad (5)$$

- vi. Maximum Distance (*MaxDistance*), which represents the largest similarity distance between the query center and any result set element. It is obtained by Equation 6.

$$MaxDistance(R_i, s_q) = \max_{r_u \in R_i} (\delta_{sim}(r_u, s_q)) . \quad (6)$$

Once the features of the result set  $R_i$  have been obtained, the resulting Diversity Feature Vector  $dv_i$  is mapped into the answer space  $\mathbb{L}$ . Thereafter, the accuracy of each result set  $R_i$  is evaluated by analyzing  $\mathbb{L}$ , as described in the following sections.

### 3.2 Diversification Evaluation Method

The diversification evaluation method is composed of result-based statistics and result set comparisons. The former compares pairs of diversity feature vectors  $dv_1$  and  $dv_2$ , mapped into the answer space  $\mathbb{L}$  ( $dv_1, dv_2 \in \mathbb{L}$ ), while the latter measures how dissimilar two result sets  $R_r$  and  $R_t$  ( $R_r, R_t \in R$ ) are. The reference result set  $R_r$  refers to the best possible solution, which must be provided by an exhaustive algorithm. The target result set  $R_t$  is the one to be evaluated. Generally speaking, all methods that we have proposed for result set comparisons receive as input a result set, generated by the diversity algorithm to be evaluated and return a value that expresses the diversity between the result set elements regarding diversity criteria.

#### 3.2.1 Dissimilarity Feature Method ( $DiF_M$ ).

This section proposes the Dissimilarity Feature Method  $DiF_M$  to represent the dissimilarity between a target result set  $R_t$  and the reference result set  $R_r$ , based on the evaluation of their feature vectors  $dv_t$  and  $dv_r$ . The main benefit of  $DiF_M$  is that it allows evaluating result sets without comparing the elements, while still allowing the user to define and use personalized features with  $DiF_M$  besides those proposed in Section 3.1. Low values for  $DiF_M$  imply that  $dv_t$  is very similar to  $dv_r$ , while higher values imply larger dissimilarity. Intuitively, our Diversity Feature Extractor  $\mathfrak{L}$  assumes that it is possible to interpret the distance distribution between result set elements as a probability distribution, aimed at describing the similarity and diversity distances by a Gaussian distribution. To establish the dissimilarity between  $dv_t$  and  $dv_r$ , we propose the use of the distance function defined in Equation 7, i.e. the weighted sum of the differences between features of the diversity vectors. For the sake of simplicity, we used  $W_i = 1$  for all features in our experiments, but different weights  $W_i$  can be used for each feature when it is previously known that some features are more relevant than others to perform a specific diversity task.

$$DiF_M(dv_r, dv_t) = \sum_{i=1}^n |dv_r[i] - dv_t[i]| * W_i . \quad (7)$$

#### 3.2.2 Dissimilarity Evaluation Method ( $D_M$ ).

This section presents the Dissimilarity Evaluation Method  $D_M$ . It represents the discrepancy between a target result set  $R_t$  and the reference result set  $R_r$ . This relationship is defined as the relative similarity between both result sets. Aimed at establishing the dissimilarity between  $R_t$  and  $R_r$ , we propose to apply the Jaccard distance to compare their result set elements, as defined in Equation 8. Low values of  $D_M$  imply that  $R_t$  is similar to  $R_r$ , while higher values imply larger dissimilarity. Thus, the  $D_M$  method targets on result sets that are as similar as possible.

$$D_M(R_r, R_t) = 1 - \frac{|R_r \cap R_t|}{|R_r \cup R_t|} . \quad (8)$$

Figure 2 illustrates our  $D_M$  method for two distinct cases, (a) and (b). The black circles represent the data elements of each result set, while the black diamonds represent those elements in the intersection of  $R_r$  and  $R_t$  (areas in gray). In Figure 2(a), the result sets share only one element, thereafter,  $D_M = 0.8$ . On the other hand, in Figure 2(b),  $R_t$  shares all elements with  $R_r$ , leading to  $D_M = 0.0$ , and our  $D_M$  measurement correctly spots that the result sets in Figure 2(a) are more dissimilar than those in Figure 2(b).



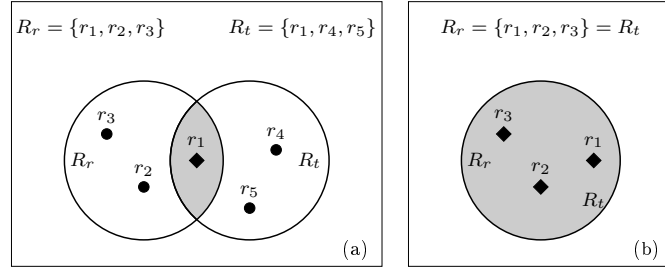


Fig. 2:  $D_M$  method. (a) Dissimilar result sets ( $D_M = 0.8$ ). (b) Similar(same) result sets ( $D_M = 0.0$ )

### 3.2.3 Dissimilarity Error Evaluation Method ( $DE_M$ ).

The  $D_M$  method presented in the previous subsection compares the result set elements in a binary way, i.e. it spots whether or not the elements in  $R_t$  are the same ones of  $R_r$ . Therefore,  $D_M$  is insensitive to whether each target result set element is a close neighbor of a counterpart in the reference result set or if they are far apart, which restricts  $D_M$  to evaluate data elements only by exact match (equality). To tackle this issue, here we extend  $D_M$  to present the Dissimilarity Error Evaluation Method  $DE_M$ . If  $R_t$  has the same elements as  $R_r$ , their dissimilarity remains zero as in  $D_M$ . However, if the elements in  $R_t$  are similar but not quite the same of  $R_r$ , a degree of similarity will be considered. To achieve that goal, we have assumed that the elements in  $R_r$  are cluster centers to which the elements in  $R_t$  must be assigned to. Each element in  $R_t$  is associated with a unique element in  $R_r$ : its nearest neighbor. Thereafter,  $DE_M$  represents the error estimated by the distance from each element  $R_{ti} \in R_t$  to its “cluster representative”  $R_{ri} \in R_r$ , as shown in Equation 9. Low values indicate that  $R_t$  is similar to  $R_r$ , while higher values imply larger dissimilarity.

$$DE_M(R_r, R_t) = \sum_{i=1}^{|k|} \delta_{sim}(R_{ti}, R_{ri}) \quad (9)$$

In Equation 9,  $\delta_{sim}(R_{ti}, R_{ri})$  is the distance between the  $i^{th}$  element in  $R_t$  and its counterpart in  $R_r$ . Figure 3 illustrates the intuition of our  $DE_M$  method for two distinct cases, (a) and (b), considering  $k = 3$ . The black circles represent result set elements only in  $R_r$  or only in  $R_t$ , while the black diamonds represent elements that appear in both result sets, i.e. the distance from an element  $r_{ti} \in R_t$  to its cluster representative  $r_{ri} \in R_r$  is zero. The size of the arrow represents the error (distance) from the element  $R_{ti}$  to its counterpart  $R_{ri}$ . In Figure 3(a), the two result sets  $R_r$  and  $R_t$  have a single element in common ( $r_{r1} = r_{t1}$ ), thus their distance is zero ( $d = 0$ ). For elements  $r_{t2}$  and  $r_{t3}$ , the distances to the cluster representatives  $r_{r2}$  and  $r_{r3}$  are two ( $d = 2$ ) and three ( $d = 3$ ), respectively, therefore,  $DE_M = 5$ . On the other hand, in Figure 3(b), the two result sets have no elements in common, but the elements in  $R_t$  are similar to those in  $R_r$ . The distances from  $r_{t1}$  to  $r_{r1}$ , from  $r_{t2}$  to  $r_{r2}$  and from  $r_{t3}$  to  $r_{r3}$  are all equal to one ( $d = 1$ ), thus  $DE_M = 3$ , therefore the error score in the result in Figure 3(b) is smaller than that in Figure 3(a).

## 4. EXPERIMENTS

This article reports on the accuracy evaluation of the result diversification approaches based only on distances between result set elements. We follow two strategies to measure diversity: extraction of statistics of the result sets to compare the algorithms and a direct comparison of the elements of each result set. We conducted our experimental studies in two parts: (1) In Section 4.1, we perform analyses to *validate* our proposed evaluation methods, presented in Section 3.2, comparing them with

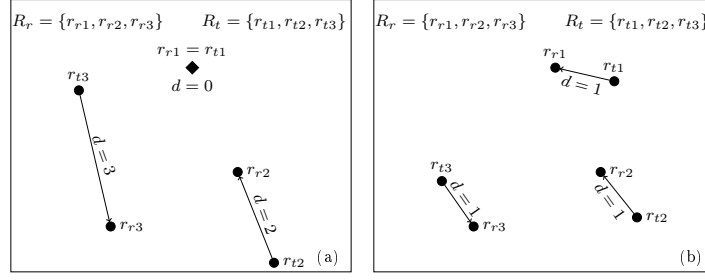
Fig. 3:  $DE_M$  evaluation method. (a)  $DE_M = 5$  (b)  $DE_M = 3$ .

Table III: Datasets used in our experimental evaluation

Dataset	# of elements	# of dimensions	$\delta_{sim}, \delta_{div}$	Description
<i>Aloi</i>	72,000	144	$L_2$	Set of color image objects rotated in 5 degree steps, obtained from the <i>Amsterdam Library of Object Images</i> website <sup>1</sup> .
<i>USCities</i>	25,375	2	$L_2$	Geographical coordinates and economic characteristics of the American cities, from the U.S. Census Bureau website <sup>2</sup> .
<i>Nasa</i>	40,150	20	$L_2$	Set of 20-dimensional vectors, extracted from NASA images. Source: <i>Metric Space Library</i> <sup>3</sup> .
<i>Faces</i>	1,016	761	$L_1$	Set of features extracted from human face images. Source: <i>Metric Space Library</i> <sup>3</sup> .

two measurements commonly used to evaluate optimization algorithms, i.e the Gap and the objective function; and (2) In Section 4.2, we explore the diversity features extracted from each result set provided by the tested algorithms, comparing them with the features of the traditional  $k$ -NN result set to identify the features transformed in the diversification process.

We implemented the well-known diversification algorithms shown in Table II (see Section 2) in C++, using the same programming framework to allow fair comparisons. The experiments were performed on a computer with an Intel Core *i7* processor and 8 GB of main memory, under Ubuntu Linux 11.10. We describe the results of the four real datasets presented in Table III. For each dataset, the table shows the dataset name, its number of elements (# of elements), its dimensionality (# of dimensions) and the distance functions used ( $\delta_{sim}$  and  $\delta_{div}$ ), along with a brief description and its source. Notice that our techniques operate on result sets returned by the algorithms tested, so they are insensitive to the dataset cardinality — instead they are sensitive to the density of the dataset, which is only indirectly related to its cardinality.

#### 4.1 Comparing Evaluation Methods

We compared the evaluation methods proposed in Section 3.2 with the Gap and the objective function measurements commonly used to evaluate optimization algorithms, considering the reference result

<sup>1</sup>Amsterdam Library of Object Images Homepage. Accessed: Jan 14, 2013. Available from: <http://staff.science.uva.nl/~aloi/>

<sup>2</sup>U.S. Census Bureau Homepage – American Census 2000. Accessed: Jan 14, 2013. Available from: <http://www.census.gov/>

<sup>3</sup>International Workshop on Similarity Search and Applications (SISAP). Accessed: Jan 14, 2013. Available from: <http://www.sisap.org/library/dbs/>

set ( $R_r$ ) provided by the exhaustive algorithm as the ground truth. Due to the high computational cost (NP-Hard) required to obtain an exhaustive result set, we restricted the size of the search space to the 200 elements most similar to the query center and the number of similar-diversified elements was always defined as  $k = 5$  for all datasets.  $F$  is the objective function used by all optimization algorithms to ensure fair comparisons,  $F(s_q, S) = (k - 1)(1 - \lambda) \cdot \delta_{sim}(s_q, S) + 2\lambda \cdot \delta_{div}(S)$ . For each dataset, the parameter  $\lambda$  varied between 0.1 to 0.9. To generate the query set, we randomly chose 100 different elements to be used as the query centers. Each point shown in the quality graphs represents the average quality measured for 100 queries with constant values of  $k$  and  $\lambda$ , but using distinct query centers. In the Graphs (a), (f), (k) and (p) of Figure 4 higher values indicate better algorithms. The remaining graphs of Figure 4, low values indicate better answers. However, the interpretation may be different for each evaluation method. For example, low values for the  $D_M$  method represent that the result set has many elements in common with the exhaustive result set, whereas low values for  $DE_M$  indicate that the result has the smallest error in the selection of elements regarding the elements of the exhaustive result set, which does not mean that the elements are the same, except when the values for  $D_M$  and  $DE_M$  are both zero.

The parameter of the FM algorithm (separation distance) was manually tuned to each dataset, varying the separation distance in steps of 0.1 until the result set has the exact number of diverse elements required in a search space of 200 elements most similar to the query center. Thus, we chose two possible values (average and highest) for the separation distance that preserves the input and output conditions of the GNE, MMR and Swap algorithms, using the same search space and returning the exact number of diverse elements, to allow fair comparison.

### Experiments using the *USCites* dataset

In the first set of experiments, we aimed at evaluating our methods over low-dimensional data. The first experiment evaluates the quality of the result provided by each algorithm in comparison to the exhaustive result set for diversity queries posed over the *USCites* dataset, with  $\lambda$  varying from 0.1 to 0.9. We set the separation distance of the FM algorithm to  $FM_{conf1} = 0.5$  and  $FM_{conf2} = 1.0$ . The experiment compared elements based on their geographical coordinates (latitude and longitude).

Figure 4 (from (a) to (e)) shows the results for the *USCites* dataset. Figures 4(a) and (b) provide the values for the objective function ( $F$ ), besides the “gap” between the value found for a specific algorithm and that obtained for the exhaustive one, respectively. As can be seen in Figure 4(a), all algorithms, including the traditional  $k$ -NN algorithm, have similar values for the objective function and, according to Figure 4(b), the largest difference to the exhaustive value is 20%. These results suggest that the existing evaluation methods return similar values for the quality of the algorithms evaluated, even for the  $k$ -NN, which does not consider diversity to select elements.

On the other hand, using our proposed evaluation methods, it is possible to clearly understand the differences between the results of the algorithms tested. For example, Figure 4(c) shows the results for our evaluation method  $D_M$  and, as can be seen, there are discrepancies between the methods. As expected, the Swap and MMR methods practically recover the same elements returned by the exhaustive algorithm for  $\lambda < 0.3$ , but for higher values of  $\lambda$ , MMR retrieves only two elements of the exhaustive result set, while Swap recovers only one. This figure also shows that  $k$ -NN does not recover the same elements as the exhaustive algorithm. Thus,  $D_M$  can distinguish between an algorithm that uses diversity to select elements of the result set and another that does not (i.e.,  $k$ -NN). The same figure shows that  $D_M$  considered the results of the FM algorithm similar to those of the  $k$ -NN algorithm (i.e., undesired results). These results are expected, since FM has the definition of diversity for the result set keeping the elements away from each other to at least the separation distance, which is different from the optimization approaches. On the other hand, the GNE method proved superior to all the other methods tested and, selected at least 4 of the 5 elements in the exhaustive result set.

Thus,  $D_M$  was indeed able to spot the superiority of GNE in comparison to the other methods.

Figure 4(d) reports the results for our  $DE_M$  method, aimed at highlighting the distance between the elements retrieved from a specific algorithm with respect to the result set elements provided by the exhaustive algorithm. As it can be seen, the GNE algorithm has the lowest error, indicating that GNE is indeed able to choose elements very close to the elements of the exhaustive result set. For  $\lambda \leq 3$ , the  $k$ -NN algorithm has the same error as the one obtained by the diversity algorithms, indicating that for  $\lambda$  values favoring the similarity, the diversity algorithms practically do not change the answer of  $k$ -NN, including the exhaustive algorithm. The FM algorithm has a lower error in comparison to the exhaustive algorithm, when  $\lambda$  is greater than 0.5, thus it performed better than  $k$ -NN, which shows that  $DE_M$  distinguishes the quality between these algorithms. Therefore, considering only the evaluation of the result set elements in a binary way (in or out) in comparison to the exhaustive result set is not enough to ensure the quality of an algorithm.

Figure 4(e) shows the results for the  $DiF_M$  method, aimed at comparing the results of the tested algorithms based on the diversity feature vectors. The features extracted also indicate that the GNE algorithm is the one that best replicates the ideal results (those of the exhaustive algorithm), followed by MMR. However, in Figure 4(e) both GNE and MMR are considered equivalent for  $\lambda \leq 0.5$ , as opposed to what occurs to  $DE_M$  and  $D_M$ , which are equivalent only for  $\lambda \leq 0.3$ . This result shows that, despite the distinct elements selected, the distance distributions of the results sets are equivalent for GNE and MMR. Although the results for the Swap method were attenuated, it is still the third best algorithm. Regarding FM, the features were able to differentiate it from the  $k$ -NN algorithm, showing that the distance distribution varies for those algorithms.

### Experiments using the Aloi dataset

The second experiment aimed at evaluating our methods over high-dimensional data. It uses the *Aloi* dataset, which has 144 features extracted by the color moment extractor [Stricker and Orengo 1995]. We set the separation distance of the FM algorithm to  $FM_{conf1} = 0.5$  and  $FM_{conf2} = 1.0$ .

Figures 4(f) and (g) show the values for the objective function (F) and the “gap” between the algorithms with respect to the value of the exhaustive result set. As it can be seen, all algorithms reached similar values, but their differences were more evident for  $\lambda \geq 0.7$ .

Figure 4(h) shows the results for our  $D_M$  measurement. Although Swap reached a better value for the objective function in comparison to that of the traditional  $k$ -NN in Figure 4(f), when analyzed by  $D_M$ , Swap follows the  $k$ -NN, having the same value for  $\lambda = 0.5$ . GNE was again the best algorithm, but its distance to the exhaustive result increased. For the separation distance 1.0, FM had the same quality as GNE, which was better than that of MMR. In summary, compared to the exhaustive result set, all algorithms shared fewer elements for high-dimensional data. However, the proposed  $D_M$  method allows a better analysis compared to the existing evaluation methods.

Figure 4(i) shows the results regarding our  $DE_M$  measurement. For  $\lambda \geq 0.5$ , Swap, FM (for all configurations) and the  $k$ -NN algorithms had similar error scores in comparison to the exhaustive solution. By analyzing this result together with the one in Figure 4(h), we can infer that, besides choosing different elements from the exhaustive result set, these algorithms remained with the same error score, indicating a possible existence of small clusters around the exhaustive result set elements.

Figure 4(j) shows the results for our  $DiF_M$  measurement. Considering only the diversity features, the FM algorithm for separation distance 1.0 was again very dissimilar with respect to the exhaustive result set in comparison to the  $k$ -NN algorithm. Such a dissimilarity can be explained by the fact

that FM selects elements that are equally distant from each other, which is not ensured for the other algorithms.

### Experiments using the Nasa dataset

The third experiment aimed at evaluating our methods with medium-dimensionality data. It uses the *Nasa* dataset, which has 20 dimensions. We set the separation distance of the FM algorithm to  $FM_{conf1} = 0.5$  and  $FM_{conf2} = 1.0$ . Figure 4 (from (k) to (o)) shows the results. As can be seen, all the evaluation methods separated the accuracy of algorithms GNE, MMR and Swap quite well. Figure 4(m) shows that FM and  $k$ -NN display similar quality considering the  $D_M$  method. We believe that their separation distances were too small to increase diversity. This assumption is confirmed in Figure 4(n), since FM and  $k$ -NN have almost the same distance error as the exhaustive solution, showing that FM selected elements closer to the answer of  $k$ -NN. Regarding the results of  $DiF_M$  (Figure 4(o)), FM continues following the  $k$ -NN behavior, which is expected due to the results in Figures 4(k) and (l). This fact has confirmed our assumption that the separation distances were too small to change the answer of  $k$ -NN.

### Experiments using the Faces dataset

The last experiment aimed at evaluating our methods with high-dimensional data. It uses the *Faces* dataset, which has 761 dimensions. We set the separation distance of the FM algorithm to  $FM_{conf1} = 0.3$  and  $FM_{conf2} = 0.6$ . Figure 4(q) reports the results. As can be seen, FM displayed the same quality of  $k$ -NN, showing that a separation distance was again too small to diversify the answer of  $k$ -NN. Figures 4(r), (s) and (t) show that all the proposed evaluation methods separate the construction strategies for the optimization algorithms, proving that the exchange strategy is more sensitive to dimensionality variations due to its behavior and the only one that follows the  $k$ -NN among the optimization algorithms. Regarding Figure 4(t), the  $DiF_M$  method considered that FM and  $k$ -NN have the same distance distribution. Moreover, MMR and GNE display the same quality for  $\lambda \leq 0.3$ , showing that although these algorithms choose different elements (Figures 4(r) and (s)), the distance distributions among the result set elements was similar.

## 4.2 Analysis of Diversity Features

We also evaluated the diversity features extracted by our proposed diversity metric extractor  $\mathfrak{L}$  for each result set provided by the tested algorithms, regarding the features obtained from the traditional  $k$ -NN algorithm. We show which features are transformed in the diversification process for each tested algorithm. To capture this information, we considered the ratio of each feature extracted from each algorithm tested to the corresponding value for the  $k$ -NN algorithm.

To generate the query set, again we randomly chose 100 different elements to be used as query centers. We used the same objective function (F) used in the previous section for all optimization algorithms to ensure fair comparisons. For each dataset, the parameter  $\lambda$  varied between 0.1 and 0.9. The search space was again restricted to the 200 elements most similar to the query center, due to the high computational cost involved and the number of diverse elements, we defined  $k = 5$ . Here, we omitted the results for the FM algorithm, since its definition of diversity (separation distance) is different from the optimization approaches, considering that the feature comparisons are dependent on the extracted features.

Figure 5 shows the ratio of each diversity feature vector extracted compared to the  $k$ -NN one regarding the *Nasa* and *Faces* datasets, using 3 values of diversity preference ( $\lambda$ ) [0.1, 0.5, 0.9], respectively. Figure 5(a) shows the results for the *Nasa* dataset, which has 20 dimensions. We set  $\lambda = 0.1$ . As can be seen, all algorithms changed the same features. The increase in features mD, SDS, SDD and DD

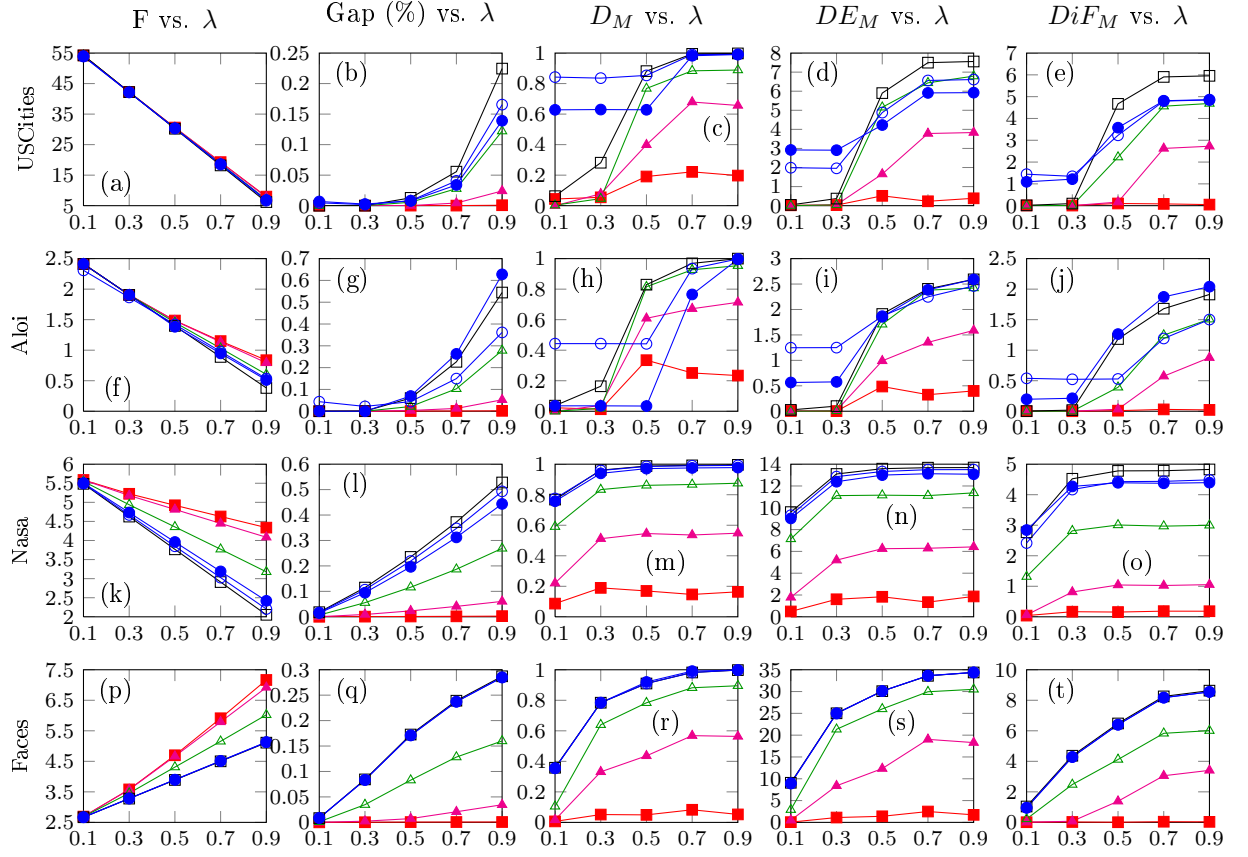


Fig. 4: Graphs comparing the 5 methods considering the measure value on the y axis and  $\lambda$  in the x axis. In the Graphs (a), (f), (k) and (p) higher values means better algorithms, and the remains graphs lower values means better algorithms. GNE —■—, MMR —▲—, SWAP —▲—, kNN —■—,  $FM_{conf1}$  —○—,  $FM_{conf2}$  —●—.

suggests that, for low values of  $\lambda$ , the algorithms change only the closest pair of elements. However, the elements are still very similar to the query center, since the SD feature has the same value of  $k$ -NN. Figure 5(b) shows the results for the *Nasa* dataset, using  $\lambda = 0.5$ . As expected, all algorithms continued to increase mD, DD and SDD. It is important to highlight that GNE was the only algorithm that changed all features similarly to the exhaustive algorithm. The MMR algorithm modifies the features very similarly to the behavior of GNE, but MMR increases the SDS, which is expected, since MMR always selects the element more similar to the query center and thereafter selects the most distant elements (incremental strategy). The results for Swap show that the commitment to the similarity measure is maintained, even for diversity preference values that benefit both similarity and diversity measures. The algorithm slightly increased the mD feature, maintaining the similarity features almost unchanged. Figure 5(c) reports the results for the *Nasa* dataset, using  $\lambda = 0.9$ , which favors the diversity measure. As expected, both GNE and exhaustive algorithm reduced the importance of similarity features in favor of diversity. It is interesting to note that the SDS, MD and SD features were greatly reduced indicating that those algorithms choose only elements distant from the query center due to the high probability of maximization of the diversity measures. On the other hand, both MMR and Swap increased the SDS feature, indicating that they sacrifice diversity to maintain the similarity to the query center, in spite of the diversity preference value.

Figure 5(d) reports the results for the *Faces* dataset, which has 761 dimensions. We set  $\lambda = 0.1$ . For high-dimensional data, all algorithms displayed feature values similar to those of  $k$ -NN, showing that for low values of  $\lambda$  the algorithm slightly increases the mD feature, different from the results for lower dimensional data (Figure 5(a)). Figures 5(e) and (f) show the results for the *Faces* dataset, using  $\lambda = 0.5$  and  $\lambda = 0.9$ , respectively. As can be seen, the results are very similar to those reported for the *Nasa* dataset, showing that, for these values of diversity preference, the algorithms changed the same features, in spite of the dataset dimensionality. Similar results were achieved for the *USCites* and *Aloi* datasets. They are omitted due to space limitations.

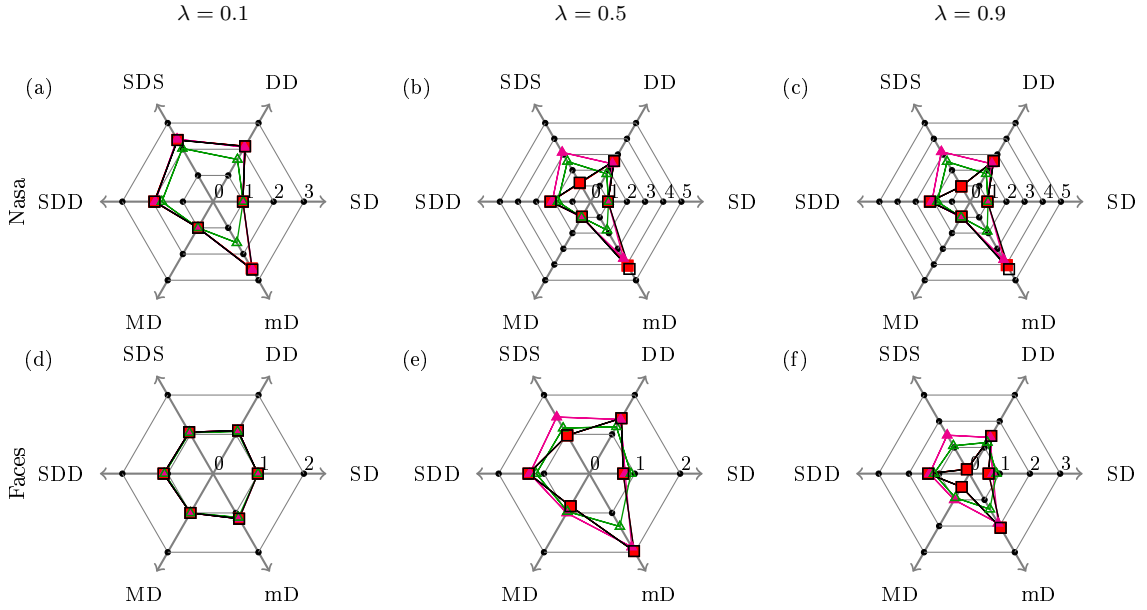


Fig. 5: Radar charts for *Nasa* and *Faces* datasets, respectively. SD stands for the AvgSimDistance feature, DD stands for the AvgDivDistance feature, SDS stands for the SDSimDistance feature, SDD stands for the SDDivDistance feature, MD stands for the MaxDistance feature and mD stands for the MinDistance feature. GNE —■—, MMR —▲—, SWAP —●—, Exhaustive —□—.

## 5. CONCLUSION

Similarity queries are one of the most pursued resources for the analysis of complex data, but the basic similarity operators do not meet the requirements of many modern applications, mainly because their result sets tend to have many elements too much similar both to the query center and among themselves. To tackle this problem, variations and extensions of the basic operators have been proposed in the literature aimed at achieving result diversification. Among the existing proposals, distance-based algorithms are the only alternative for data domains that do not have any additional information (i.e., metadata), besides the data elements and their distances. However, evaluation techniques are still largely required to help understand what type of diversity is retrieved by such methods, in which the evaluation must also rely only on distances among elements.

In this article, we tackle such a problem by creating an answer space that highlights information on the distribution of result set elements (i.e., minimal distance and maximal distance), based on a novel technique proposed to obtain several statistics from the diversified result set. To validate our proposal, we performed experiments using four real datasets that span up to 72k data elements and 761 dimensions. The experiments show how each existing algorithm compares with each other and

have confirmed that those algorithms considered the best from the literature (e.g., the GNE method) indeed perform better than the others in most situations. More importantly, our proposed techniques also pinpoint for which types of datasets the best algorithms in fact thrive and for which ones they do not, thus indicating “where” there is still room for improvement in result diversification research.

## REFERENCES

- AGRAWAL, R., GOLLAPUDI, S., HALVERSON, A., AND IEONG, S. Diversifying search results. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*. WSDM '09. ACM, New York, NY, USA, pp. 5–14, 2009.
- ANGEL, A. AND KOUDAS, N. Efficient diversity-aware search. In *ACM SIGMOD International Conference on Management of Data*. ACM, Athens, Greece, pp. 781–792, 2011.
- CAPANNINI, G., NARDINI, F. M., PEREGO, R., AND SILVESTRI, F. Efficient diversification of web search results. *PVLDB* 4 (7): 451–459, 2011.
- CARBONELL, J. AND GOLDSTEIN, J. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, USA, pp. 335–336, 1998.
- CHEN, Z. AND LI, T. Addressing diverse user preferences in SQL-query-result navigation. In *2007 ACM SIGMOD International Conference on Management of Data*. ACM, Beijing, China, pp. 641–652, 2007.
- CLARKE, C. L., KOLLA, M., CORMACK, G. V., VECHTOMOVA, O., ASHKAN, A., BÜTTCHER, S., AND MACKINNON, I. Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. SIGIR '08. ACM, New York, NY, USA, pp. 659–666, 2008.
- DROSOU, M. AND PITOURA, E. Search result diversification. *SIGMOD Rec.* 39 (1): 41–47, Sept., 2010.
- DROSOU, M. AND PITOURA, E. DisC diversity: Result diversification based on dissimilarity and coverage. *PVLDB* 6 (1): 13–24, 2012.
- FEO, T. A. AND RESENDE, M. G. C. Greedy randomized adaptive search procedures. *Journal of Global Optimization* 6 (2): 109–133, 1995.
- GIL-COSTA, V., SANTOS, R. L. T., MACDONALD, C., AND OUNIS, I. Sparse spatial selection for novelty-based search result diversification. In *18th International Conference on String Processing and Information Retrieval*. Springer-Verlag, Pisa, Italy, pp. 344–355, 2011.
- GOLLAPUDI, S. AND SHARMA, A. An axiomatic approach for result diversification. In *18th International Conference on World Wide Web*. ACM, New York, NY, USA, pp. 381–390, 2009.
- HARITSA, J. R. The kNDN problem: A quest for unity in diversity. *IEEE Data Eng. Bull.* 32 (4): 15–22, 2009.
- SKOPAL, T., DOHNAL, V., BATKO, M., AND ZEŽULA, P. Distinct nearest neighbors queries for similarity search in very large multimedia databases. In *Proceedings of the eleventh international workshop on Web information and data management*. WIDM '09. ACM, New York, NY, USA, pp. 11–14, 2009.
- STRICKER, M. A. AND ORENGO, M. Similarity of color images. In *Storage and Retrieval for Image and Video Databases*. SPIE Proceedings, vol. 2420. SPIE, San Jose, CA, USA, pp. 381–392, 1995.
- VAN LEUKEN, R. H., GARCIA, L., OLIVARES, X., AND VAN ZWOL, R. Visual diversification of image search results. In *18th International Conference on World Wide Web*. ACM, Madrid, Spain, pp. 341–350, 2009.
- VEE, E., SRIVASTAVA, U., SHANMUGASUNDARAM, J., BHAT, P., AND YAHIA, S. Efficient computation of diverse query results. In *IEEE 24th International Conference on Data Engineering*. IEEE, Cancun, Mexico, pp. 228–236, 2008.
- VIEIRA, M. R., RAZENTE, H. L., BARIONI, M. C. N., HADJIELEFTHERIOU, M., SRIVASTAVA, D., TRAINA JR, C., AND TSOTRAS, V. J. On query result diversification. In *IEEE 27th International Conference on Data Engineering*. IEEE, Hannover, Germany, pp. 1163–1174, 2011.
- YU, C., LAKSHMANAN, L., AND AMER-YAHIA, S. It takes variety to make a world: diversification in recommender systems. In *12th International Conference on Extending Database Technology: Advances in Database Technology*. ACM, New York, NY, USA, pp. 368–378, 2009.
- ZIEGLER, C.-N., MCNEE, S. M., KONSTAN, J. A., AND LAUSEN, G. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*. WWW '05. ACM, New York, NY, USA, pp. 22–32, 2005.