Journal of Discrete Algorithms 18 (2013) 75–88

Contents lists available at SciVerse ScienceDirect

Journal of Discrete Algorithms

www.elsevier.com/locate/jda

Modelling efficient novelty-based search result diversification in metric spaces

Veronica Gil-Costa [b,c,*], Rodrygo L.T. Santos [a], Craig Macdonald [a], Iadh Ounis [a]

[a] University of Glasgow, UK
[b] Yahoo! Research, Santiago de Chile, Chile
[c] CONICET, Argentina

ARTICLE INFO

Article history:
Available online 7 August 2012

Keywords:
Diversification
Metric spaces
Clustering
Pivoting
Permutation

ABSTRACT

Novelty-based diversification provides a way to tackle ambiguous queries by re-ranking a set of retrieved documents. Current approaches are typically greedy, requiring $O(n^2)$ document–document comparisons in order to diversify a ranking of n documents. In this article, we introduce a new approach for novelty-based search result diversification to reduce the overhead incurred by document–document comparisons. To this end, we model novelty promotion as a similarity search in a metric space, exploiting the properties of this space to efficiently identify novel documents. We investigate three different approaches: pivoting-based, clustering-based, and permutation-based. In the first two, a novel document is one that lies outside the range of a pivot or outside a cluster. In the latter, a novel document is one that has a different signature (i.e., the document's relative distance to a distinguished set of fixed objects called permutants) compared to previously selected documents. Thorough experiments using two TREC test collections for diversity evaluation, as well as a large sample of the query stream of a commercial search engine show that our approaches perform at least as effectively as well-known novelty-based diversification approaches in the literature, while dramatically improving their efficiency.

1. Introduction

Search result diversification has emerged as an effective approach for tackling ambiguous queries. In particular, a diverse ranking aims to satisfy as many *aspects* of an ambiguous query as possible, and as early as possible. By satisfying multiple query aspects, a high *coverage* of these aspects is achieved. By having different aspects satisfied as early as possible, a high *novelty* is also attained [26].

Promoting coverage is typically more efficient than promoting novelty: while coverage can be estimated for different documents independently, the same is not true for novelty. In particular, the notion of novelty entails a dependence between the relevance of different documents—i.e., a novel document is one that covers aspects not covered by the other documents. As a result, novelty-based diversification becomes essentially the problem of finding a set of documents that together cover most of the aspects of a query at a given rank cutoff. In this general formulation, this is an NP-hard problem [1]. Most novelty-based approaches proposed in the literature for this problem deploy a greedy approximation algorithm: at each iteration, the algorithm selects a document that covers the most aspects not yet covered by the documents selected in the previous iterations. In a typical case, after the system retrieves n documents to be diversified, this greedy algorithm performs

* Corresponding author to: Yahoo! Research, Santiago de Chile, Chile
E-mail addresses: gvcosta@yahoo-inc.com (V. Gil-Costa), rodrygo@dcs.gla.ac.uk (R.L.T. Santos), craigm@dcs.gla.ac.uk (C. Macdonald), ounis@dcs.gla.ac.uk (I. Ounis).

1570-8667/$ – see front matter
http://dx.doi.org/10.1016/j.jda.2012.07.004

$O(n^2)$ document–document comparisons—i.e., $O(n)$ similarity searches across n iterations [21]—which can severely impact the efficiency of these approaches.

In this article, we propose to reduce the number of required similarity computations in novelty-based diversification approaches, by modelling novelty in a metric space [21]. Metric spaces have been typically used to locate user-relevant information in collections of objects. In general terms, a metric space consists of a set of objects. To estimate the closeness of objects, their similarity is quantified using a pairwise distance metric. Hence, we can measure the similarity or dissimilarity between any two objects.

Many modern database applications [6] use metric spaces to model the similarity between objects. Similarity search in metric spaces focuses on retrieving objects, which are similar to a query point. Certain properties of such metric spaces offer an opportunity for enhancing the efficiency of similarity searches. In particular, by representing the retrieved documents as m-dimensional vectors in a metric space, we can exploit the properties of such spaces to dramatically reduce the number of similarity computations required to diversify these documents.

Metric spaces applied to search result diversification were first studied in the context of image search [56]. The first attempt to leverage the properties of metric spaces for diversifying textual documents was made by Gil-Costa et al. [32]. In particular, they showed that the number of required similarity computations for novelty-based diversification can be reduced using a pivoting algorithm [9], which selects sparse pivot documents from the result set at running time. In other words, the algorithm selects dissimilar pivot documents which are far away from each other in the metric space.

In this article,[1] we show that effective and efficient diversification can be attained using metric spaces algorithms based upon different principles. In particular, besides the pivoting-based diversification approach presented in [32] we introduce permutation- and clustering-based approaches. More precisely, we employ pivoting-, permutation-, and clustering-based construction algorithms without actually storing extra information neither using additional data structures. To the best of our knowledge, this work is the first attempt to use permutation- and clustering-based approaches for diversifying textual documents.

In our pivoting-based diversification approach, novelty is promoted by selecting sparse documents as pivots. In particular, we use a pivoting technique that can adapt itself to different spaces by dynamically determining the number of pivots. Each pivot has a radius. The geometric shape of the pivot along with its radius, corresponds to a ball around the pivot. Hence each pivot covers a specific area in the metric space. Documents within the area are considered redundant, as they are too similar to the already selected pivot.

Our second approach deploys clustering to cover different areas in the space. The idea is to divide the set of documents into compact groups or clusters, meaning that the documents in each cluster are close to each other. A representative document, named centre, is chosen from each cluster. Documents representing a cluster centre are considered novel, whereas the remaining documents in a cluster are considered redundant, as they are too similar to the already selected centres. Our clustering-based approach is built on top of a data structure that has been proven to be efficient on high-dimensional metric spaces [17]. In such metric spaces, the probability distribution of distance among documents has a very concentrated histogram, with a larger mean as the dimension grows, rendering the search for novel documents more difficult to perform. In our approach clusters are built with a fixed number of documents k. By fixing k, we can adjust the desired level of diversification. The cluster radius is given by the distance from the centre to the k-nearest document. The geometric shape of the centre along with its radius corresponds to a ball around the centre. In high-dimensional spaces, the value of k can be dynamically increased as many documents may have the same k-th distance to the centre. In this case, the number of clusters can be dramatically reduced.

Our third approach is based on probabilistic and approximate approaches [3,18]. It does not use the concept of covered areas in a metric space. Instead, it uses the idea that documents are similar or dissimilar according to their distances to a set of fixed documents, called permutants. Each document is associated with a *signature* (vector of permutants sorted from closest to farthest to the document). The basic idea is to select a random set of permutants and produce all the signatures by comparing every document against the permutants. Then, these permutants are sorted (permuted) in increasing order to obtain the signature for each document. Two documents with the same *signature* are considered redundant.

The efficiency of both pivoting- and permutation-based approaches depends on the number of selected pivots or permutants. With a larger set of pivots/permutants, more document–document comparisons will be required to diversify the ranking of retrieved documents. The efficiency of the clustering-based approach relies on the number of centres. But this number of centres can be quickly limited in high-dimensional spaces due to more documents being very close to one another; therefore, they may have the same distances to the centres. Hence, each cluster may contain more than k documents.

The contributions of this article are two-fold:

(1) We model novelty-based diversification as a similarity search in a metric space. At this point we include, besides the pivoting-based approach presented in [32], two more approaches to explore and take advantage of the properties of metric spaces.

[1] An early version of this work appeared in the Proceedings of the 18th International Symposium on String Processing and Information Retrieval [32].

(2) We thoroughly investigate the effectiveness and efficiency of our three proposed metric space approaches, using two publicly available test collections for diversity evaluation, as well as a large sample of the query stream of a commercial search engine.

Our experimental results attest both the effectiveness and the efficiency of our approach compared to existing novelty-based diversification approaches. In terms of effectiveness, our results show that the proposed approaches perform as well as current well-known novelty-based diversification approaches. They also present similar effectiveness to one another. Regarding the efficiency, our approaches improve the quadratic cost of current greedy approaches presented in the literature. Our clustering-based approach reports the best running time.

The remainder of this article is organised as follows. In Section 2, we review existing approaches for search result diversification and similarity search in metric spaces. Section 3 shows how novelty-based diversification can be modelled in a metric space. Sections 4 and 5 detail our experimental setup and evaluation, respectively. Conclusions follow in Section 6.

2. Background and related work

In this section, we describe related approaches to the ones presented in this article. In particular, in Section 2.1, we review existing approaches to search result diversification, with a focus on novelty-based approaches. In Section 2.2, we provide background on metric spaces.

2.1. Search result diversification

Diversification approaches can be classified as *implicit* or *explicit* [53]. Implicit approaches assume that different documents will cover different query aspects. As a result, these approaches promote novel documents as a means to indirectly cover multiple aspects. The definition of a 'novel' document is precisely what distinguishes the approaches in this family. For instance, Carbonell and Goldstein [13] proposed to compare documents based on their cosine similarity. Zhai et al. [62] proposed an extension of this idea, by comparing documents with respect to the divergence of their language models. Recently, Wang and Zhu [58] proposed to use the correlation of documents' relevance scores.

Instead of assuming that different documents cover different aspects, explicit diversification approaches directly model these aspects as part of their strategy. For instance, Agrawal et al. [1] proposed a diversification approach based on an explicit representation of query aspects as taxonomy classes, in order to promote documents that cover classes also covered by the query. A similar approach was proposed by Carterette and Chandar [14], but with query aspects represented as topic models built from the top retrieved results for the query. Finally, Santos et al. [51] proposed to represent the aspects underlying a query as 'sub-queries'. In their approach, documents are promoted according to their estimated relevance to multiple sub-queries, as well as to the estimated importance of each sub-query.

Although having the same goal, these two families of approaches deploy rather distinct strategies. While implicit diversification approaches are driven by novelty, explicit ones usually target coverage. In general, coverage focuses on tackling query ambiguity, by actively promoting documents that are potentially relevant to multiple query aspects. Novelty, on the other hand, combats redundancy, by demoting documents that cover already well covered aspects. While promoting coverage is generally a more effective diversification strategy than promoting novelty, their combination as a *hybrid* strategy can outperform both strategies individually. In particular, novelty can effectively break the tie between documents with similar coverage estimates [52].

Despite having the potential to improve upon the application of coverage alone in terms of effectiveness, promoting novelty is generally more computationally expensive. Indeed, while coverage can be estimated independently for different documents, promoting novelty incurs additional costs. For explicit diversification approaches, the additional cost comes from the need to iteratively update the marginal utility of the identified query aspects, given how much each aspect is covered by the documents selected in the previous iterations of these approaches' greedy process [1,51]. To overcome this problem, Capannini et al. [12] proposed to adapt hybrid diversification approaches to avoid having to update the aspect marginal utility estimates at every iteration. Instead, they approximate the effect of novelty by pre-scoring each retrieved document with respect to each query aspect and the document's dissimilarity to all other documents retrieved for this aspect. Although efficient, this approximation of novelty can be unsafe in terms of effectiveness, as it does not consider the position of each document in the final diversified ranking. As a result, it may promote a document that is novel with respect to documents ranked after it in the final ranking, as opposed to the documents placed at higher ranks.

For implicit diversification approaches, the additional costs incurred by promoting novelty equate to directly comparing the retrieved documents to one another, so as to identify documents that carry novel content [13,58,62]. To overcome the inefficiency of existing novelty-based diversification approaches, we propose to tackle novelty seeking as a search in a metric space. In particular, the properties of a metric space offer an opportunity for reducing the number of document comparisons required to identify novel documents.

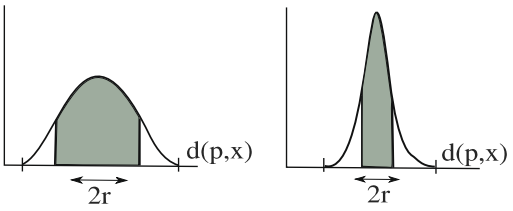


Fig. 1. Disperse (left) and concentrated (right) distance histograms. The shaded region in both histograms denotes the fraction of objects within the range of the query object. The latter histogram denotes a more challenging search space.

2.2. Metric spaces

Metric spaces have been widely studied to perform sequentially searches for objects which are similar to a given query object (e.g. [39,42,43]). Additionally, some works have used metric spaces in other settings like distributed environments [11, 31,48] and peer-to-peer (P2P) systems [7,15,37].

Formally, a *metric space* $(\mathcal{U}, \delta)$ comprises a universe of objects $\mathcal{U}$ and a *distance function* $\delta : \mathcal{U} \times \mathcal{U} \to \mathcal{R}^+$, which determines the similarity between any pair of objects [21]. The function δ can be regarded as a measure of object dissimilarity. Therefore, the smaller the distance between two objects, the more "similar" they are. The definition of the distance function depends on the type of the objects being compared. In an m-dimensional vector space—a particular case of metric spaces in which every object is represented by a vector of m real coordinates—δ could be a distance function of the family $L_s(x, y) = (\sum_{1 \leqslant i \leqslant m} |x_i - y_i|^s)^{\frac{1}{s}}$. For example, $s = 2$ yields the Euclidean distance. For any $x, y, z \in \mathcal{U}$, the function δ holds several properties: non-negativity ($\delta(x, y) \geqslant 0$), reflexivity ($\delta(x, y) = 0$ iff $x = y$), symmetry ($\delta(x, y) = \delta(y, x)$), and the triangle inequality ($\delta(x, z) \leqslant \delta(x, y) + \delta(y, z)$). Some good surveys about metric spaces can be found in [21,61] and [50].

Although general metric spaces do not have an explicit dimensionality (as in vector spaces), we can discuss their *intrinsic dimensionality* [21]. Metric spaces with high intrinsic dimensionality typically present highly concentrated histograms of distances among objects. In practice, this has a negative impact on the efficiency of any similarity search algorithm [60]. This observation is illustrated in Fig. 1. Given an object p and a query q with radius search r, the triangle inequality implies that all objects x such that $|\delta(q, p) - \delta(p, x)| > r$ are outside the query range, and hence can be safely discarded. However, in a concentrated histogram, the probability of discarding an object is lower, because the objects tend to be close to one another. The intrinsic dimensionality provides a means to quantify the inherent difficulty of performing a similarity search in this space.

Metric space search algorithms preprocess the working set of objects $\mathcal{X}$ to build an index $\mathcal{I}$. Different indexing algorithms have been proposed in the literature to speed up similarity searches [50], most of them fall into one of two categories: clustering and pivoting.

Clustering techniques divide the working set of objects into groups (called clusters), such that similar objects fall into the same group [17,21,22,41]. Thus, the space is divided into zones as compact as possible, usually in a recursive fashion. This technique stores a representative point ("centre") for each zone plus extra information that permits quickly discarding the zone at query time. Two criteria can be used to delimit a zone. The first one is the *Voronoi region* [4], where we select a set of centres and put every other point inside the zone of its closest centre. The regions are bounded by hyperplanes and the zones are analogous to Voronoi regions in vector spaces. The second criterion is the *covering radius* $cr(c_i)$ (e.g. [22]), which is the maximum distance between c_i and any object in its zone. Moreover, these techniques can be combined. Some tree-like data structures (e.g., [8,10,55,59,60]) use the clustering technique in a more indirect way: they select a pivot as the root of the tree and divide the space according to the distances to the root.

Pivoting techniques select some objects as pivots and calculate the distance between every other object and each pivot. The resulting index is a data structure that can be seen as a table T with the pivots in the columns and the object identifiers in the rows, where each cell $T_{i,j}$ stores the distance between the object i and the pivot j. Several algorithms (e.g., [5,19,20, 40,44,57]) are almost direct implementations of this idea. Essentially, only their extra data structure (e.g. additional memory space is assigned to store pre-computed distances and other relevant information) is used to reduce the cost of finding the candidate points. A key challenge for pivoting techniques is to determine the number of pivots needed to cover all objects in the working set. Moreover, the number of pivots tends to increase with the size of the working set. Some hybrid approaches (e.g., [30,38]) combine clustering techniques with pivoting techniques.

Recently, a new class of approaches was proposed for approximate similarity search [3,18]. An approximated answer is formed by those objects, which are close to the current query, but that are not necessarily the k closest ones. In particular, Amato and Savino [3] proposed a method using inverted files based on the idea that two objects are similar if they have the same distance to each object of a fixed set. Independently, Chavez et al. [18] presented a similar idea. They proposed an algorithm that predicts the proximity between objects by taking into account the order of the distances between these objects and a set of reference objects, called permutants. The idea is to compute the distances between objects and the permutants. To this end, each object is associated with a list of permutants (called *permutation* or *signature*), sorted by their distance to the object. Formally, given a set of permutants $\mathcal{P}$, with $|\mathcal{P}| = k$, an object x_i is associated with a signature

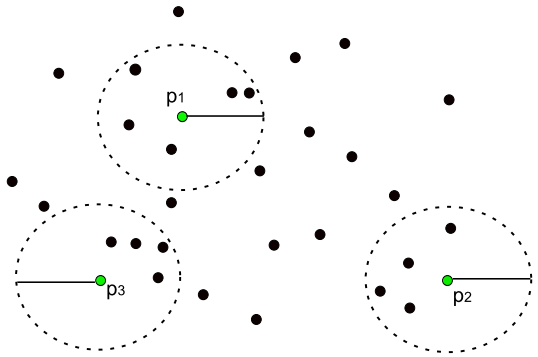


Fig. 2. Objects in the range of pivots p_1, p_2, and p_3 are considered redundant.

$\langle p_1, p_2, \ldots, p_k \rangle$ where $p_i \in \mathcal{P}$ for $1 \leqslant i \leqslant k$ and $\delta(p_1, x_i) \leqslant \delta(p_2, x_i) \leqslant \cdots \leqslant \delta(p_k, x_i)$. In order to determine whether two objects of the working set are similar, their corresponding orderings of permutants are compared.

Additional work on permutation-based algorithms was described by Skala [54], who evaluated the number of possible permutations that can occur in a given space. Esuli [28] presented a permutation prefix index, called PP-Index, which produces effective approximate answers to queries. In particular, dataset objects are kept on secondary memory. For each object permutation, the PP-Index selects a prefix of size ℓ. These prefixes are indexed in a main memory tree-based data structure. The leaves keep the information required to retrieve the disk blocks relative to the objects represented by the prefix permutation obtained in the path of the tree. In a previous work, Esuli [27] also used the PP-Index in the MiPai image-retrieval system. Gennaro et al. [29] presented a permutation-based approach building upon the full-text retrieval library Lucene.[2] A similar approach integrating the Lucene library was presented by Lux et al. [35]. Recently, Novak et al. [45,46] presented a distributed metric space index for P2P systems called M-Index.

In the next section, we propose to adapt three metric space index construction algorithms for novelty-based search result diversification. In particular, we exploit pivoting-, clustering-, and permutation-based approaches for identifying novel documents in the ranking.

3. Metric space diversification

Let $\mathcal{D}$ contain the documents initially retrieved for a query q. Novelty-based diversification approaches typically build a re-ranking $\mathcal{S}$ by iteratively selecting a document $d \in \mathcal{D} \setminus \mathcal{S}$ that is both *relevant* (with respect to the initial query) and *novel* (with respect to the documents already selected in $\mathcal{S}$). To estimate relevance, any standard retrieval model (e.g., BM25 [49]) can be used. To estimate novelty, every document $d_i \in \mathcal{D} \setminus \mathcal{S}$ is compared to every document $d_j \in \mathcal{S}$, where $\mathcal{S}$ comprises the documents selected in the previous iterations. This way, the document d_i that differs most from the already selected documents in $\mathcal{S}$ is itself included in $\mathcal{S}$. Such document–document comparisons are usually performed as distance computations in an m-dimensional term-frequency space, where m is the number of unique terms in the underlying document collection. As discussed in Section 2.1, these approaches differ mainly in their choice of a distance function (e.g., cosine [13], divergence [62], or correlation [58]). Regardless of the chosen distance function, these approaches require $O(n^2)$ distance computations to diversify a list of n documents. An $O(n)$ similarity search is performed across n iterations. To reduce the quadratic number of distance computations incurred by the existing greedy novelty-based diversification approaches, we propose to exploit metric spaces properties.

In the following sections, we present three approaches to model novelty-based diversification over metric spaces. As discussed in Section 2.2, the first one is a pivoting algorithm, which selects sparse pivots. In other words, pivots which are far away from each other. The second approach is a clustering-based algorithm, which uses a covering radius criterion to cluster a set of objects in the space. The last one is a permutation-based algorithm, which represents documents as sorted list of distances to a set of fixed permutant documents.

3.1. Pivoting-based diversification

Our first approach is based on an efficient pivoting algorithm named Sparse Spatial Selection (SSS) algorithm [9]. As illustrated in Fig. 2, the SSS algorithm identifies a set of k "pivots" among the n objects in the metric space.

We propose a novelty-based diversification approach inspired by the SSS pivoting algorithm. Our novel Sparse Spatial Selection Diversification (SSSD) approach incorporates the notion of pivots to reduce the number of distance computations required to diversify a set of documents. SSSD builds upon the SSS algorithm in order to skip redundant documents in the ranking. As described in Algorithm 1, SSSD takes as input a query q, an initial set of documents $\mathcal{D}$ retrieved for this

[2] http://lucene.apache.org.

query, a distance function δ with upper-bound M, and the covering radius ϕ, with $0 \leqslant \phi \leqslant 1$, such that $r = \phi M$ determines the covering range of each pivot. The parameter M is the maximum distance between any two objects of the space. The parameter ϕ controls the density of pivots with which the space is covered. The higher the value of ϕ, the less pivots are selected. The value of this parameter is empirically selected to reduce the number of distance evaluations, without compromising the resulting diversification effectiveness of the algorithm.

```
SSSD[q, D = {d_1, ..., d_n}, δ, M, φ]
1: P ⇐ {d_1}
2: for all d_i ∈ D \ {d_1} do
3:     if δ(d_i, p_j) ⩾ φM ∀p_j ∈ P then
4:         P ⇐ P ∪ {d_i}
5:     end if
6: end for
7: P ⇐ P ∪ (D \ P)
```

Algorithm 1. Sparse Spatial Selection Diversification (SSSD).

The core of SSSD is the selection of pivots (lines 1–4 in Algorithm 1). To this end, let $(\mathcal{U}, \delta)$ be a metric space, with $\mathcal{D} \subseteq \mathcal{U}$ comprising the documents retrieved for the query q. The pivot set $\mathcal{P}$ is initialised with the first retrieved document, i.e., $d_1 \in \mathcal{D}$. For each remaining document $d_i \in \mathcal{D} \setminus \{d_1\}$, d_i is chosen as a new pivot if its distance to every pivot in $\mathcal{P}$ is greater than or equal to the range ϕM. Hence, a retrieved document becomes a new pivot if and only if it is located outside the range of all current pivots. Moreover, documents within the range of an already selected pivot are considered redundant and skipped, and are added later to the end of the ranking (line 5), in the same order as they were originally retrieved in the initial ranking $\mathcal{D}$.

Importantly, during the selection of pivots, it is not necessary that all documents in $\mathcal{D}$ be compared against all pivots. When a document d_i is compared against a pivot p_j and does not satisfy the condition $\delta(d_i, p_j) \geqslant \phi M$, this document is discarded and no additional comparisons are required. In the best case scenario, documents are compared only with the first pivot when they are within the range of this pivot. Assuming that an unseen document d_i has a constant probability $\nu = f(\mathcal{U}, \delta, M, \phi)$ of lying outside the range of all pivots $p_j \in \mathcal{P}$ given the metric space $(\mathcal{U}, \delta)$ and the range ϕM, it can easily be shown that Algorithm 1 requires $\sum_{i=1}^{n-1} \nu^{i-1}(n-i)$ document–pivot comparisons to diversify n documents. In the worst scenario, when $\nu = 1$ (i.e., all documents are outside the range of all pivots and become themselves pivots), this algorithm exhibits the same quadratic complexity as the greedy novelty seeking approach. However, in practical deployments, $\nu \ll 1$, which results in a drastic reduction in the number of required document–pivot comparisons.

In Fig. 3, we have an initial set $\mathcal{D}$ with 15 retrieved documents. The SSSD algorithm processes the documents in the same order as they were originally retrieved in the initial ranking. Thus, it selects d_1 as the first pivot (novel document) and computes the range ϕM for d_1. Documents d_2, d_3 and d_4 are considered redundant because they are within the range of d_1. Then d_5 is selected as the second novel document because it is outside the range of d_1. Documents d_6 and d_7 are considered redundant as they are within the range of d_1. The same happens with d_8 and d_9 because they are within the range of d_5. The last pivot is d_{10} because it is located outside the range of all current pivots (d_1, d_2). All remaining documents are redundant as they are within the range of the already selected pivots. The final set of documents is rearranged so that the pivot–documents are located at the beginning.

3.2. Clustering-based diversification

In this section, we propose a clustering-based diversification approach. In particular, we use the List of Clusters (LC) data structure. LC has been shown to be efficient in high-dimensional metric space searches [17]. The basic idea is to build clusters (c, r). Each cluster has a centre c with a covering radius r, so that documents in the cluster are within the covering radius of the centre.

In order to diversify a ranking $\mathcal{D}$ of documents initially retrieved for a query q, we first choose a centre $c \in \mathcal{D}$ and a radius r. The cluster (c, r) comprises the subset of documents of $\mathcal{D}$ which are at distance of at most r from c. We define:

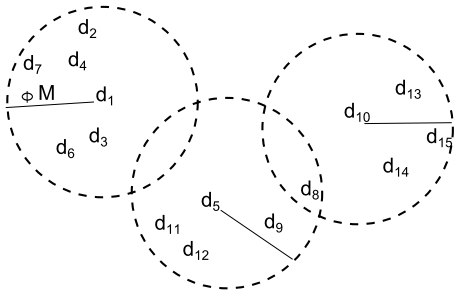
$$\mathcal{I}_{\mathcal{D},c,r} = \left\{d \in \mathcal{D} \setminus \{c\}\colon\ \delta(c,d) \leqslant r\right\}$$

as the set of *internal* documents, i.e., which lie inside the cluster (c, r), and

$$\mathcal{E}_{\mathcal{D},c,r} = \left\{d \in \mathcal{D}\colon\ \delta(c,d) > r\right\}$$

as the set of *external* documents. Clustering is recursively applied in $\mathcal{E}_{\mathcal{D},c,r}$.

Algorithm 2 describes our List of Clusters Diversification (LCD) approach. The centre set $\mathcal{C}$ is initialised with the first retrieved document $d_1 \in \mathcal{D}$ (line 1). Then, d_1 is removed from the set of external documents in line 2. In lines 5–6, we compute the distance between all documents $d_i \in \mathcal{E}$ and the current centre c. These distances are stored in a vector $\mathcal{V}$ and sorted in line 7. In line 8, we obtain the cluster radius r as the distance from the centre c to its k-nearest document. All documents within the cluster radius are added to the internal cluster, as they are considered too similar to the already



Ranking of documents $\mathcal{D}$ = { d1 d2 d3 d4 d5 d6 d7 d8 d9 d10 d11 d12 d13 d14 d15 }

SSSD = { d1 d5 d10 d2 d3 d4 d6 d7 d8 d9 d11 d12 d13 d14 d15 }

Fig. 3. Diversification example using the SSSD algorithm.

selected centre. These documents are removed from $\mathcal{E}$ in line 10 and are later added to the bottom of the ranking (line 14), in the same order as they were originally retrieved in the initial ranking $\mathcal{D}$.

Although clusters are of size k, when working with high-dimensional metric spaces, the value of k can be dynamically increased as many documents may have the same k-th distance to the centre. In this case the number of clusters can be reduced and therefore the number of computations required to select the centres of the clusters are reduced. On the other hand, fewer clusters reduces the number of diversified documents which may impact on the effectiveness of the proposed algorithm. As explained in Fig. 5, if we use $k = 5$ the first cluster with centre d_1 contains seven documents. This is because documents d_2, d_4, d_7 and d_8 have the same distance to the centre.

Chavez and Navarro [17] investigated different heuristics to select cluster centres, and experimentally showed that the best strategy is to choose the next centre as the object that maximises the sum of distances to the previous centres. In this work, we use this heuristic to select the centres, as shown in line 11. In line 12, the new centre is added to the set $\mathcal{C}$ and removed from $\mathcal{E}$ in line 13. The algorithm ends when all documents in $\mathcal{D}$ have been processed.

LCD$[q, \mathcal{D} = \{d_1, \ldots, d_n\}, k]$
1: $\mathcal{C} \Leftarrow \{d_1\}$
2: $\mathcal{E} \Leftarrow \mathcal{D} \setminus \{d_1\}$
3: $c \Leftarrow d_1$ // *current centre*
4: **while** $|\mathcal{E}| > 0$ **do**
5: **for all** $d_i \in \mathcal{E}$ **do**
6: $\mathcal{V} \Leftarrow \mathcal{V} \cup \{\delta(d_i, c)\}$
7: **end for**
8: Sort $\mathcal{V}$
9: $r \Leftarrow \mathcal{V}[k]$
10: $\mathcal{I} \Leftarrow \{d_j \in \mathcal{E}:\ \delta(c, d_j) \leqslant r\}$
11: $\mathcal{E} \Leftarrow \mathcal{E} \setminus \mathcal{I}$
12: $c \Leftarrow d_i \in \mathcal{E}:\ \delta(d_i, c) > \delta(d_j, c)\ \forall d_j \in \mathcal{E}$
13: $\mathcal{C} \Leftarrow \mathcal{C} \cup \{c\}$
14: $\mathcal{E} \Leftarrow \mathcal{E} \setminus \{c\}$
15: **end while**
16: $\mathcal{C} \Leftarrow \mathcal{C} \cup (\mathcal{D} \setminus \mathcal{C})$

Algorithm 2. List of Clusters Diversification (LCD).

The LCD algorithm produces a list of triples $\langle c_i, r_{c_i}, I_i \rangle$. The centres are promoted to the top of the ranking. A centre chosen first has preference over the subsequent ones. All documents that lie inside the covering radius of the first centre belong to its cluster, despite that they may also lie inside the clusters of subsequent centres. As a result, these documents are considered redundant and are added to the bottom of the ranking. Fig. 4 shows three clusters with centres (in order of construction) c_1, c_2, and c_3. Objects found at the intersection of clusters c_2 and c_3 are evaluated only by the centre c_2.

Fig. 5 shows how the LCD algorithm proceeds using $k = 5$. Document d_1 is selected as the first centre (novel document). The remaining documents are sorted by distance to d_1. The covering radius is computed as the distance from d_1 to its k-nearest document. In this case, we have four documents, denoted d_2, d_4, d_8, d_7 with the same distance to d_1. All these documents are considered redundant, inserted into $\mathcal{I}_{\mathcal{D},d_1,r_c}$ and removed from $\mathcal{E}$. The second centre, d_{16}, is selected as a novel document because it has the maximum distance to d_1. The documents in $\mathcal{E}$ are sorted by distance to d_{16}. The first 5 nearest documents to d_{16} are considered redundant, inserted into $\mathcal{I}_{\mathcal{D},d_2,r_c}$ and removed from $\mathcal{E}$. At this point, there are only two remaining documents to be processed. d_{11} is selected as a novel document because it maximises the sum of distances to all current centres. Hence, d_{14} is considered redundant. The final ranking of documents is rearranged so that the centres are placed at the top.

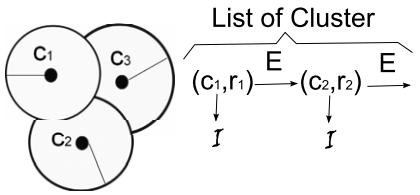


Fig. 4. The influence zone of centres c_1, c_2 and c_3. Objects within the intersection of two clusters are evaluated by the cluster built first. Clusters are composed by a centre c_i, a covering radius r_{c_i} and a set of internal objects $\mathcal{I}$.

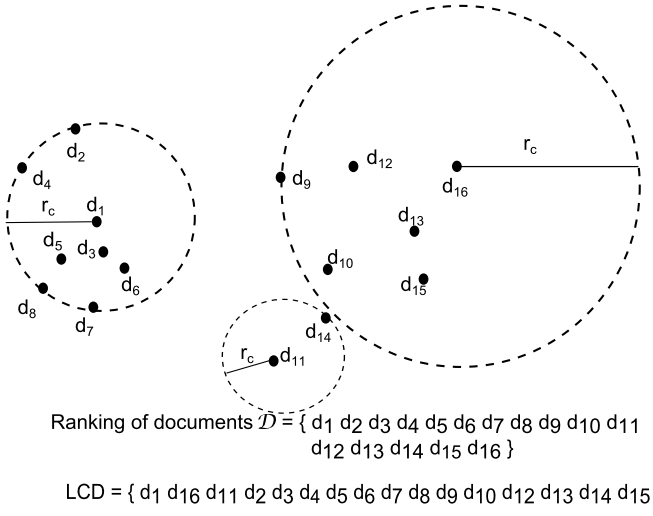


Fig. 5. Diversification example using the LCD algorithm with $k = 5$.

As explained above, the LCD algorithm promotes novelty by selecting documents as centres that are far away from each other. After selecting the first document as a centre, the second centre is selected as the document that maximises the distance to the first one, namely, a centre dissimilar to the already selected ones. The asymptotic cost is $O(n^2)$ [36], but in practice this cost depends on the value of k. For a given k value and a set of n retrieved documents for a query q, the cost of diversifying this set is $\sum_{i=1}^{\frac{n}{k+1}} n - i - (i-1)k$. We can avoid more distance computations with a larger k.

3.3. Permutation-based diversification

In the literature, permutation algorithms have been applied to solve approximate searches over metric spaces [3,18]. The basic idea is to select a set $\mathcal{P}$ of special documents, called permutants, and to compute the distance between these permutants and the remaining documents in the set of retrieved documents $\mathcal{D}$. As a result, each document is represented by a list of the considered permutants, sorted by their distance to the document. To determine whether two documents are similar, their vectors of permutants are compared.

PD$[q, \mathcal{D} = \{d_1, \ldots, d_n\}, k]$
1: $\mathcal{P} \Leftarrow Rand(\mathcal{D}, k)$
2: **for all** $d_i \in \mathcal{D}$ **do**
3: **for all** $p_j \in \mathcal{P}$ **do**
4: $\mathcal{V} \Leftarrow \mathcal{V} \cup \langle p_j, \delta(d_i, p_j)\rangle$
5: **end for**
6: Sort $\mathcal{V}$ by distance
7: $d_i.\text{Sig} \Leftarrow \langle \mathcal{V}[1].p_m, \ldots, \mathcal{V}[k].p_r\rangle$
8: **end for**
9: $\mathcal{N} \Leftarrow \{d_1\}$
10: **for all** $d_i \in \mathcal{D} \setminus \{d_1\}$ **do**
11: **if** $DistSignature(d_i, d_j, k) \neq 0\ \forall d_j \in \mathcal{N}$ **then**
12: $\mathcal{N} \Leftarrow \mathcal{N} \cup \{d_i\}$
13: **end if**
14: **end for**
15: $\mathcal{N} \Leftarrow \mathcal{N} \cup (\mathcal{D} \setminus \mathcal{N})$

Algorithm 3. Permutation Diversification (PD).

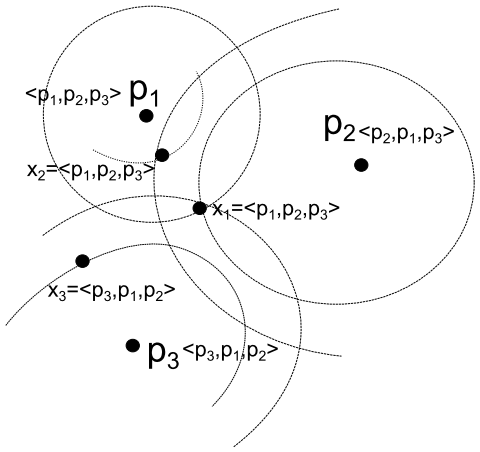


Fig. 6. Using permutant signatures to promote novel documents.

Our Permutation Diversification (PD) approach is described in Algorithm 3. In line 1, we select the set $\mathcal{P}$ of k permutants at random from the ranking $\mathcal{D}$. In particular, Chavez et al. [18] showed that different selection heuristics of linear time complexity (including the random selection) present similar performances in terms of the effectiveness of the permutation algorithm. For each document $d_i \in \mathcal{D}$, we compute its distance to all permutants $p_j \in \mathcal{P}$. In line 4, we store the pairs $\langle p_j, \delta(d_i, p_j)\rangle$ into $\mathcal{V}$, where the first component is the permutant identifier and the second component is the distance between the permutant and the document. In line 5, we sort the identifiers of the k permutants according to their distance to d_i. In line 6, we build a signature denoted $d_i.\text{Sig} = \langle \mathcal{V}[1].p_m, \ldots, \mathcal{V}[k].p_r\rangle$ where $\delta(d_i, \mathcal{V}[1].p_m) \leqslant \delta(d_i, \mathcal{V}[2].p_n) \leqslant \cdots \leqslant \delta(d_i, \mathcal{V}[k].p_r)$, and $\mathcal{V}[i].p_m$ indicates the permutant identifier allocated in the i-th position of the sorted vector of permutants.

To diversify the ranking $\mathcal{D}$, we select d_1 as part of the set $\mathcal{N}$ of non-redundant documents in line 7. To compute the distance between $d_i \in \mathcal{D}$ and $d_j \in \mathcal{N}$, we employ a permutation similarity measure described by Chavez et al. [18]. If both signatures are different, d_i is included as a novel document into $\mathcal{N}$. Otherwise, the document is considered redundant and added to the bottom of the ranking. As with classical pivoting techniques, the key challenge for the permutation algorithms is to determine the number of permutants needed to efficiently diversify a document ranking without losing effectiveness. The number of document comparisons grows linearly with the number of permutants.

In Fig. 6, we illustrate how the PD algorithm proceeds. Assuming $k = 3$ permutants, denoted p_1, p_2 and p_3, each document has a signature according to its distance to these permutants. The signature of document x_1 is $\langle p_1, p_2, p_3\rangle$. If we first include document x_1 into $\mathcal{N}$, documents x_2 and p_1 are considered redundant, because they have the same signature as x_1. The same happens with documents p_2 and x_3. One of them is included into $\mathcal{N}$ and the other is discarded. But p_2 is different from any previous document in $\mathcal{N}$ (according to the vector of permutants), therefore it is promoted as a novel document to the top of the ranking. The final set of documents is rearranged so that the non-redundant documents in $\mathcal{N}$ are located at the beginning.

4. Experimental setup

Our investigation aims to answer two major research questions:

(1) How do SSSD, LCD, and PD compare to existing novelty-based diversification approaches in terms of effectiveness?
(2) How do SSSD, LCD, and PD compare to existing novelty-based diversification approaches in terms of efficiency?

To evaluate our approach in different metric spaces, we experiment with two standard test collections for diversity evaluation, comprising both Web and newswire documents. The first collection, denoted WT, comprises 150 queries from the diversity task of the TREC 2009, 2010, and 2011 Web tracks [23–25]. The second collection, denoted IT, includes 20 queries from the Interactive track of TREC-6, TREC-7, and TREC-8 [33]. For the WT test collection, we index the TREC ClueWeb09 (cat. B) corpus, which consists of 50 million Web documents. For the IT collection, we index the Financial Times portion of TREC Disks 4&5, with 210,000 newswire documents. Both corpora are indexed using the Terrier Information Retrieval Platform [47],[3] with Porter's stemmer and standard stopword removal. In addition to using these standard test collections for evaluating both the effectiveness and the efficiency of our proposed approaches, we further evaluate their

[3] http://terrier.org.

Table 1
Diversification performance across the WT and IT topics. The best performances on top of BM25 and DPH are highlighted in bold.

	WT				IT			
	ERR-IA		α-nDCG		ERR-IA		α-nDCG	
	@20	@100	@20	@100	@20	@100	@20	@100
BM25	0.2057	0.2127	0.3043	0.3575	0.1541	0.1600	0.4703	0.5324
+MMR(c)	0.2089	0.2157	0.3091	0.3608	0.1565	0.1622	**0.4814**	0.5351
+MVA(ρ)	0.2053	0.2123	0.3032	0.3570	0.1542	0.1602	0.4706	0.5330
+SSSD(c)	0.1972$^{\nabla}$	0.2043$^{\nabla}$	0.2939$^{\nabla}$	0.3499$^{\nabla}$	0.1541	0.1605	0.4601	0.5291
+SSSD(ρ)	0.2058	0.2127	0.3047	0.3577	0.1541	0.1600	0.4703	0.5324
+LCD(c)	**0.2124**	**0.2193**	0.3095	**0.3627**	**0.1633**	**0.1693**	0.4804	**0.5430**
+LCD(ρ)	**0.2124**	**0.2193**	0.3095	**0.3627**	**0.1633**	**0.1693**	0.4804	**0.5430**
+PD(c)	0.2063	0.2129	0.3068	0.3581	0.1536	0.1600	0.4655	0.5323
+PD(ρ)	0.2110	0.2175	**0.3114**	0.3623	0.1493	0.1552	0.4574	0.5239
DPH	0.2349	0.2410	0.3387	0.3861	0.1658	0.1716	0.4833	0.5468
+MMR(c)	0.2353	0.2412	0.3394	0.3862	0.1647	0.1705	0.4798	0.5436
+MVA(ρ)	0.2360	0.2417	0.3427	0.3871	0.1655	0.1712	0.4787	0.5443
+SSSD(c)	0.2308	0.2367	0.3344	0.3814	0.1324$^{\nabla}$	0.1406$^{\nabla}$	0.3900$^{\nabla}$	0.4921$^{\nabla}$
+SSSD(ρ)	0.2349	0.2410	0.3387	0.3861	**0.1691**	**0.1749**	**0.4864**	**0.5501**
+LCD(c)	0.2316	0.2376	0.3357	0.3831	0.1643	0.1701	0.4827	0.5463
+LCD(ρ)	0.2340	0.2400	0.3381	0.3855	0.1660	0.1718	0.4859	0.5491
+PD(c)	0.2351	0.2413	0.3373	0.3858	0.1631	0.1695	0.4698	0.5419
+PD(ρ)	**0.2377**	**0.2432**	**0.3436**	**0.3877**	0.1661	0.1720	0.4779	0.5464

efficiency on a sample of the query stream from the MSN 2006 query log.[4] In particular, we selected the first 1000 queries from the log, after removing empty queries and queries with no results in the ClueWeb09 corpus.

To retrieve an initial pool of documents to be reranked by the several considered diversification approaches, we apply either the Okapi BM25 [49] or the Divergence from Randomness DPH model [2]. On top of these ad hoc retrieval baselines, we deploy two well-known novelty-based diversification approaches as diversification baselines: Maximal Marginal Relevance (MMR [13]) and Mean-Variance Analysis (MVA [58]). As these approaches compute novelty based on cosine or correlation estimations, respectively, we deploy our proposed approaches using both cosine and Pearson's correlation as instantiations of the distance function δ, as described in Section 3. Since the standard cosine function is not a proper metric distance, we use the angular dissimilarity (i.e., one minus the normalised arccos of the inner product) between the vectors representing a pair of documents as the function δ. To cope with the quadratic complexity of MMR and MVA while keeping a uniform setting across all approaches, both of these baselines as well as our approaches are applied to diversify the top 100 documents retrieved by BM25 or DPH.

Effectiveness is assessed using the primary metrics in the diversity task of the TREC 2011 Web track [24], namely, ERR-IA [16] and α-nDCG [26]. To train the parameters of our approaches, namely, ϕ for SSSD (Algorithm 1) and k for both LCD (Algorithm 2) and PD (Algorithm 3), as well as the parameters of our baselines (λ for MMR [13]; σ and b for MVA [58]), we perform a simulated annealing [34] through a 5-fold cross validation. In particular, we train the parameters of all approaches to maximise ERR-IA@100 on the training folds, and report the results as an average across the corresponding test folds. As for efficiency, we report the number of document–document comparisons performed, as well as the time spent in performing such comparisons.

5. Experimental results

In this section, we investigate whether novelty-based diversification approaches can be made efficient without compromising their effectiveness. Before investigating the efficiency of our proposed approaches, we evaluate their effectiveness compared to MMR [13] and MVA [58] as baselines. As discussed in Section 2.1, novelty-based approaches have been recently shown to play a tie-breaking role in search result diversification, with the potential to further improve on top of a purely coverage-based ranking [52]. Table 1 shows the diversification performance of our three proposed metric space approaches as well as the two used baselines across the WT and IT test collections. As mentioned in Section 4, for the distance function δ, we consider both cosine (denoted c) and Pearson's correlation (denoted ρ). All diversification approaches are applied on top of both BM25 and DPH, which do not perform any diversification, serving as pure ad hoc retrieval baselines. The best diversification performance on top of each of the two ad hoc baselines is highlighted in bold. Statistically significant differences between each of SSSD, LCD, and PD and the best between MMR and MVA are verified by a paired t-test. The symbols $\triangle$ and ∇ denote a significant increase or decrease with $p < 0.05$.

From Table 1, we note that our approaches provide small improvements across several settings. In particular, the results over BM25 (top half of Table 1) show that LCD(c) and LCD(ρ) provide the best diversification performance for both WT and IT collections, followed by PD(ρ) on the WT collection. Over DPH (bottom half of the table), the best performing

[4] http://research.microsoft.com/en-us/um/people/nickcr/wscd09/.

diversification approaches are PD(ρ) for the WT collection, and SSSD(ρ) for the IT collection, followed by LCD(ρ) on the IT collection. These results are consistent for both ERR-IA and α-nDCG. As for the choice of distance function, Pearson's correlation (ρ) generally outperforms cosine (c). Compared to our diversification baselines, namely, MMR and MVA, the only significant decrease is observed for SSSD(c), which underperforms over BM25 for the WT collection, and over DPH for the IT collection. In all other settings, there is no significant difference between any of our proposed approaches and the two diversification baselines. This answers our first research question, by showing that our metric space approaches perform at least as effectively as existing novelty-based approaches. While our proposed approaches show a similar effectiveness to one another and to the considered baselines, their efficiency varies considerably, as we will show next.

To answer our second research question, we investigate how the organisation of a metric space around search centres impacts the efficiency of our proposed approaches. As described in Section 3, SSSD, LCD, and PD represent search centres as pivots,[5] clusters, and permutants, respectively. Accordingly, we analyse the efficiency of these approaches as a function of the number k of such centres. In particular, for the WT and IT test collections, Fig. 7 shows how the number of document–document comparisons (Figs. 7(a)–(b)) and the running time in seconds (Figs. 7(c)–(d)) of our approaches are affected by the parameter k. Running times are based on a Linux Quad-Core Intel Xeon 2.4 GHz 8 GB, and denote the time spent to compare documents, as the cost to initially retrieve and represent these documents in a vector space is the same for all approaches. Additionally, to enable the analysis of the efficiency of our approaches in context, Figs. 7(e)–(f) show how k impacts the effectiveness of our approaches, measured by ERR-IA@20, the primary diversity evaluation metric at the TREC Web track [24]. As baseline diversification approaches, we once again use MMR [13] and MVA [58], which are presented as horizontal lines, since their performance does not depend on k. As before, all approaches are applied to diversify the top 100 documents retrieved by BM25.[6]

From Figs. 7(a)–(b), we first note that, while MMR and MVA show a quadratic complexity (i.e., $\frac{n(n-1)}{2} = 4950$ comparisons for $n = 100$ documents; note that both MMR and MVA have the same complexity, hence their lines are superposed in Figs. 7(a)–(b)), all our proposed approaches scale linearly with the number of centres. In particular, SSSD shows the best performance in terms of the number of performed comparisons, followed closely by LCD. Interestingly, the latter never selects more than about $k = 35$ centres (i.e., clusters) for the considered metric spaces, even when allowed to. The PD algorithm shows a much steeper ascending curve, with a similar performance to MMR and MVA with only about $k = 15$ selected centres (i.e., permutants). These observations are consistent for both the WT and IT collections. While the number of comparisons in Figs. 7(a)–(b) is a function of the number of selected centres only, the running time demanded by the various approaches also depends on their choice of a distance function (i.e., cosine or Pearson's correlation). Indeed, as shown in Figs. 7(c)–(d), the running times of all approaches based upon the cosine function (i.e., SSSD(c), LCD(c), and PD(c)) are lower than those of their counterparts that use Pearson's correlation (i.e., SSSD(ρ), LCD(ρ), and PD(ρ)). Moreover, the cosine-based approaches are faster than the most efficient of our considered baselines (MMR(c)) for almost the entire range of k values. Notably, the most efficient among our proposed approaches, LCD incurs an almost negligible overhead in order to diversify the top retrieved results for a query, making it suitable for search environments where query latency is paramount, such as the Web. Answering our second research question, these results attest the efficiency of our proposed metric space approaches for search result diversification. To further analyse the efficiency of these approaches over a representative query stream, Figs. 8(a)–(b) show the performance of all approaches for a sample of 1000 consecutive queries from the MSN 2006 query log, as described in Section 4. These results closely match those shown in Figs. 7(a)–(d), further attesting the efficiency of our approaches.

Lastly, Figs. 7(e)–(f) bridge our two research questions, by showing the impact of increasing the number of centres k on the effectiveness of our metric space diversification approaches, in terms of ERR-IA@20. Overall, we observe that all approaches can at least match the effectiveness of MMR and MVA for some settings. In particular, both the SSSD and LCD algorithms benefit more from small k values ($k \leqslant 3$ for SSSD; $k \leqslant 4$ for LCD). Interestingly, small k values also result in the most efficient deployment of these approaches, as shown in Figs. 7(a)–(d). In turn, PD can closely match the effectiveness of both MMR and MVA for almost the entire range of k values. Once again, these results are consistent across the WT and IT test collections, and are largely independent of the choice of a distance function. On the other hand, they also show that carefully choosing an appropriate number of centres for the metric space underlying the test collection at hand is key for attaining a suitable trade-off between an effective and efficient diversification.

6. Conclusions

We have exploited the properties of metric space construction algorithms to reduce the number of document–document comparisons incurred by current novelty-based search result diversification approaches. To explore and take advantage of the properties of metric spaces, we have presented algorithms based upon pivoting, clustering, and permutations.

Our Sparse Spatial Selection Diversification (SSSD) approach selects a set of pivots from the space of documents retrieved for a query. Our List of Clusters Diversification (LCD) approach selects documents as centres to build clusters and uses the

[5] As shown in Algorithm 1, for SSSD, the number of pivots is indirectly determined by the covering radius ϕ, as opposed to being an input parameter itself.

[6] Results on top of DPH show identical trends and are hence omitted for brevity.

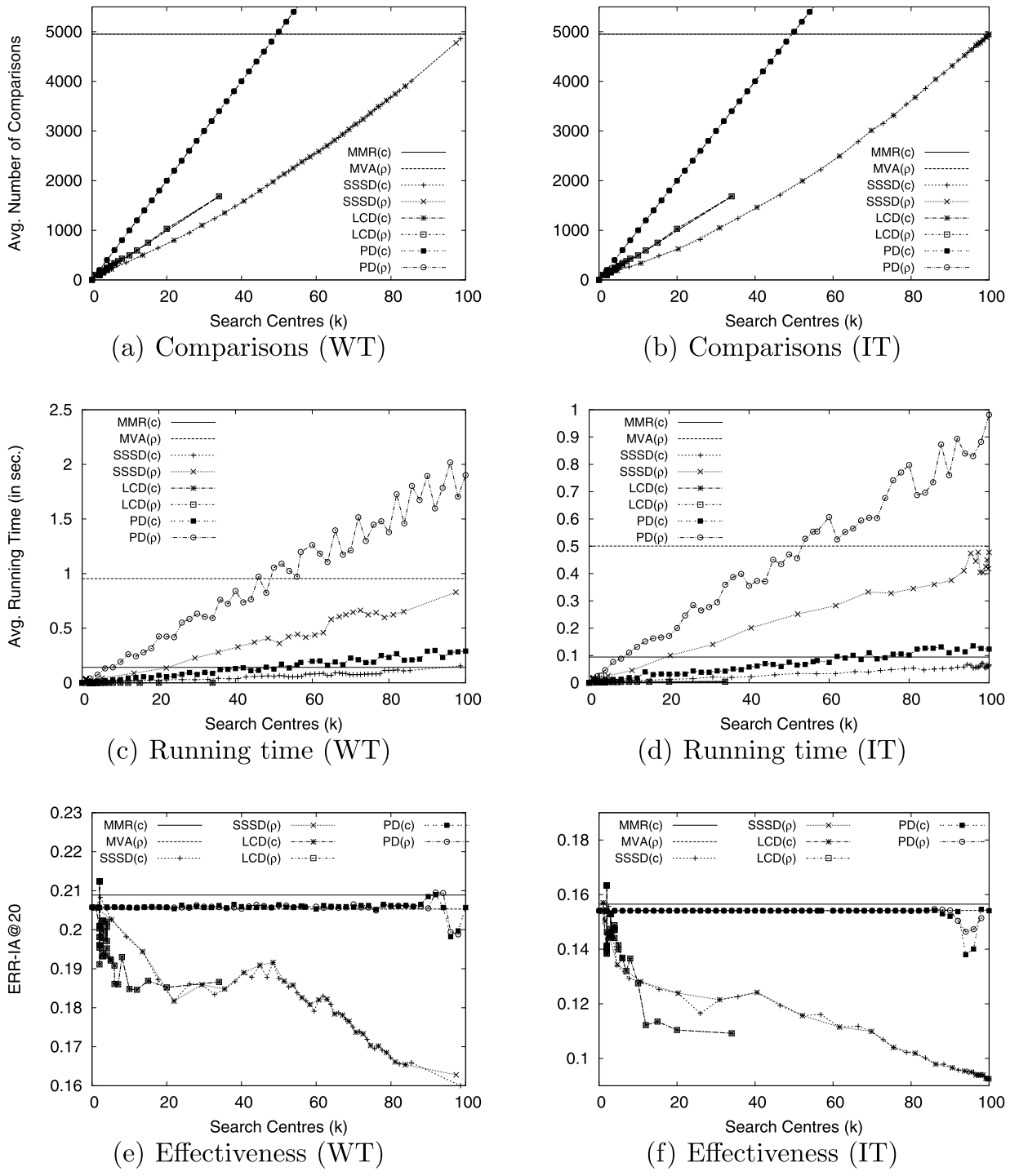


Fig. 7. Number of regions, number of document–document comparisons, running time, and diversification performance for the WT (left) and IT (right) test collections, across a range of k values. Figures are averages across the topics of the corresponding collection (150 and 20, respectively). All approaches are applied on top of BM25.

covering radius scheme to divide the space. Both approaches use the covering radius to regard documents covered by a pivot or a centre as redundant. Our third algorithm, called Permutation Diversification (PD), represents the retrieved documents for a query as vectors of permutants. Those vectors are compared to detect redundant documents.

In a thorough investigation across standard TREC test collections for diversity evaluation, we have shown that our proposed metric space approaches perform at least as effectively as well-known novelty-based diversification approaches in

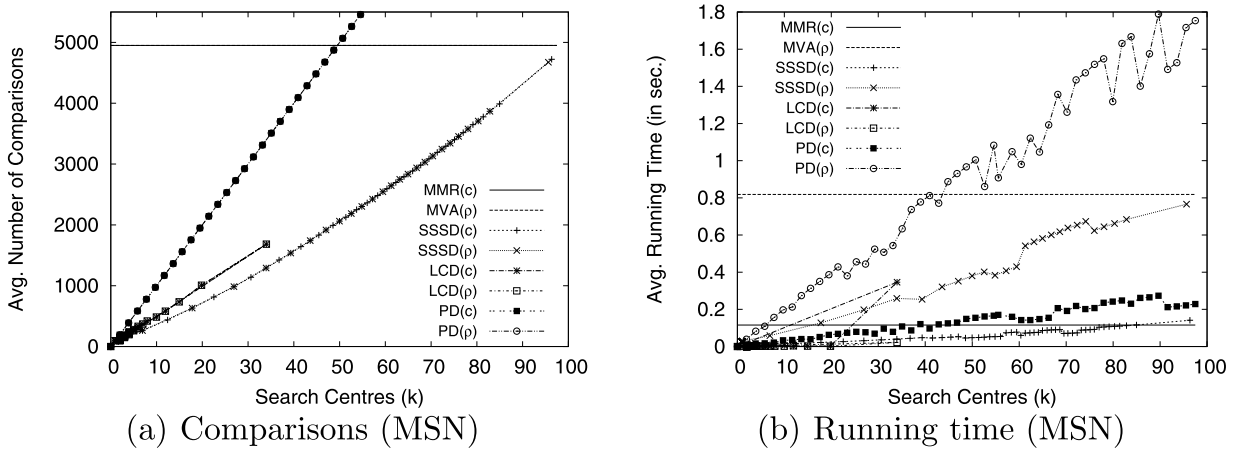


Fig. 8. Number of document–document comparisons and running time across a range of k values. All figures are averages over 1000 queries from the MSN 2006 query log.

the literature, while dramatically improving their efficiency. Moreover, by evaluating our approaches across metric spaces induced by different document collections and distance functions, we have shown that a careful division of the underlying metric space is paramount for appropriately trading off effectiveness and efficiency in novelty-based search result diversification.

Acknowledgements

The work has been performed under the HPC-EUROPA2 project (project number: 228398) with the support of the European Commission–Capacities Area–Research Infrastructures.

References

[1] R. Agrawal, S. Gollapudi, A. Halverson, S. Ieong, Diversifying search results, in: WSDM, 2009, pp. 5–14.
[2] G. Amati, E. Ambrosi, M. Bianchi, C. Gaibisso, G. Gambosi, FUB, IASI-CNR and University of Tor Vergata at TREC 2007 Blog track, in: TREC, 2007.
[3] G. Amato, P. Savino, Approximate similarity search in metric spaces using inverted files, in: Infoscale, 2008, p. 28.
[4] F. Aurenhammer, Voronoi diagrams—a survey of a fundamental geometric data structure, ACM Comput. Surv. 23 (3) (1991) 345–405.
[5] R. Baeza-Yates, W. Cunto, U. Manber, S. Wu, Proximity matching using fixed-queries trees, in: CPM, in: LNCS, vol. 807, 1994, pp. 198–212.
[6] J.M. Barrios, D. Diaz-Espinoza, B. Bustos, Text-based and content-based image retrieval on Flickr: DEMO, in: SISAP, 2009, pp. 156–157.
[7] M. Batko, D. Novak, F. Falchi, P. Zezula, Scalability comparison of peer-to-peer similarity search structures, Future Generat. Comp. Syst. 24 (8) (2008) 834–848.
[8] T. Bozkaya, M. Ozsoyoglu, Distance-based indexing for high-dimensional metric spaces, in: SIGMOD, Sigmod Record 26 (2) (1997) 357–368.
[9] N.R. Brisaboa, A. Farina, O. Pedreira, N. Reyes, Similarity search using sparse pivots for efficient multimedia information retrieval, in: ISM, 2006, pp. 881–888.
[10] W. Burkhard, R. Keller, Some approaches to best-match file searching, Comm. ACM 16 (4) (1973) 230–236.
[11] P. Burstein, A.J. Smith, Efficient search in file-sharing networks, in: ICPADS, 2007, pp. 1–9.
[12] G. Capannini, F.M. Nardini, R. Perego, F. Silvestri, Efficient diversification of web search results, Proc. VLDB Endow. 4 (7) (Apr. 2011) 451–459.
[13] J. Carbonell, J. Goldstein, The use of MMR, diversity-based reranking for reordering documents and producing summaries, in: SIGIR, 1998, pp. 335–336.
[14] B. Carterette, P. Chandar, Probabilistic models of ranking novel documents for faceted topic retrieval, in: CIKM, 2009, pp. 1287–1296.
[15] U.V. Catalyurek, E.G. Boman, K.D. Devine, D. Bozdağ, R.T. Heaphy, L.A. Riesen, A repartitioning hypergraph model for dynamic load balancing, J. Parallel Distr. Comput. 69 (2009) 711–724.
[16] O. Chapelle, D. Metlzer, Y. Zhang, P. Grinspan, Expected reciprocal rank for graded relevance, in: CIKM, 2009, pp. 621–630.
[17] E. Chavez, G. Navarro, A compact space decomposition for effective metric indexing, PRL 26 (9) (2005) 1363–1376.
[18] E. Chavez, K. Figueroa, G. Navarro, Effective proximity retrieval by ordering permutations, PAMI 30 (2008) 1647–1658.
[19] E. Chavez, J. Marroquín, R. Baeza-Yates, Spaghettis: an array based algorithm for similarity queries in metric spaces, in: SPIRE, IEEE CS Press, 1999, pp. 38–46.
[20] E. Chavez, J. Marroquín, G. Navarro, Fixed queries array: A fast and economical data structure for proximity searching, Multimed. Tool. Appl. 14 (2) (2001) 113–135.
[21] E. Chávez, G. Navarro, R. Baeza-Yates, J.L. Marroquín, Searching in metric spaces, ACM Comput. Surv. 33 (3) (2001) 273–321.
[22] P. Ciaccia, M. Patella, P. Zezula, M-tree: An efficient access method for similarity search in metric spaces, in: VLDB, 1997, pp. 426–435.
[23] C.L.A. Clarke, N. Craswell, I. Soboroff, Overview of the TREC 2009 Web track, in: TREC, 2009.
[24] C.L.A. Clarke, N. Craswell, I. Soboroff, Overview of the TREC 2011 Web track, in: TREC, 2011.
[25] C.L.A. Clarke, N. Craswell, I. Soboroff, G.V. Cormack, Overview of the TREC 2010 Web track, in: TREC, 2010.
[26] C.L.A. Clarke, M. Kolla, G.V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, I. MacKinnon, Novelty and diversity in information retrieval evaluation, in: SIGIR, 2008, pp. 659–666.
[27] A. Esuli, MIDI: Using the pp-index to build an efficient and scalable similarity search system, in: SISAP, 2009, pp. 146–148.
[28] A. Esuli, Pp-index: Using permutation prefixes for efficient and scalable similarity search, in: SEBD, 2010, pp. 318–325.
[29] C. Gennaro, G. Amato, P. Bolettieri, P. Savino, An approach to content-based image retrieval based on the Lucene search engine library, in: ECDL, 2010, pp. 55–66.

[30] C. Gennaro, M. Mordacchini, S. Orlando, F. Rabitti, A scalable distributed data structure for multi-feature similarity search, in: SEBD, 2008, pp. 302–309.
[31] V. Gil-Costa, M. Marin, N. Reyes, Parallel query processing on distributed clustering indexes, J. Discrete Algorithms 7 (1) (2009) 3–17.
[32] V. Gil-Costa, R.L.T. Santos, C. Macdonald, I. Ounis, Sparse spatial selection for novelty-based search result diversification, in: SPIRE, 2011, pp. 344–355.
[33] W. Hersh, P. Over, TREC-8 interactive track report, in: TREC, 2000.
[34] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680.
[35] M. Lux, S.A. Chatzichristofis, Lire: Lucene image retrieval: an extensible java CBIR library, in: ACM Multimedia, 2008, pp. 1085–1088.
[36] M. Mamede, F. Barbosa, Range queries in natural language dictionaries with recursive lists of clusters, in: ISCIS, 2007.
[37] M. Marin, V. Gil-Costa, C. Hernández, Dynamic P2P indexing and search based on compact clustering, in: SISAP, 2009, pp. 124–131.
[38] M. Marin, V. Gil-Costa, R. Uribe, Hybrid index for metric space databases, in: ICCS, 2008, pp. 327–336.
[39] L. Mico, J. Oncina, R.C. Carrasco, A fast branch & bound nearest neighbour classifier in metric spaces, Pattern Recogn. Lett. 17 (7) (1996) 731–739.
[40] L. Mico, J. Oncina, E. Vidal, A new version of the nearest-neighbor approximating and eliminating search (AESA) with linear preprocessing-time and memory requirements, Pattern Recogn. Lett. 15 (1994) 9–17.
[41] G. Navarro, Searching in metric spaces by spatial approximation, in: VLDB, 2002, pp. 28–46.
[42] G. Navarro, N. Reyes, Fully dynamic spatial approximation trees, in: SPIRE, 2002, pp. 254–270.
[43] G. Navarro, N. Reyes, Dynamic spatial approximation trees for massive data, in: SISAP, 2009, pp. 81–88.
[44] S. Nene, S. Nayar, A simple algorithm for nearest neighbor search in high dimensions, IEEE Trans. Pattern Anal. Mach. Intell. 19 (9) (1997) 989–1003.
[45] D. Novak, M. Batko, Metric index: An efficient and scalable solution for similarity search, in: SISAP, 2009, pp. 65–73.
[46] D. Novak, M. Batko, P. Zezula, Large-scale similarity data management with distributed metric index, IPM (2011).
[47] I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, C. Lioma, Terrier: a high performance and scalable information retrieval platform, in: OSIR, 2006, pp. 18–25.
[48] A. Papadopoulos, Y. Manolopoulos, Distributed processing of similarity queries, Distrib. Parallel Databases 9 (1) (2001) 67–92.
[49] S.E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, M. Gatford, Okapi at TREC-3, in: TREC, 1994.
[50] H. Samet, Foundations of Multidimensional and Metric Data Structures, The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling, Morgan Kaufmann Publishers Inc., 2005.
[51] R.L.T. Santos, C. Macdonald, I. Ounis, Exploiting query reformulations for Web search result diversification, in: WWW, 2010, pp. 881–890.
[52] R.L.T. Santos, C. Macdonald, I. Ounis, On the role of novelty for search result diversification, in: Information Retrieval, 2012.
[53] R.L.T. Santos, J. Peng, C. Macdonald, I. Ounis, Explicit search result diversification through sub-queries, in: ECIR, 2010, pp. 87–99.
[54] M. Skala, Counting distance permutations, J. Discrete Algorithms 7 (1) (2009) 49–61.
[55] J. Uhlmann, Satisfying general proximity/similarity queries with metric trees, Inform. Process. Lett. 40 (1991) 175–179.
[56] R.H. van Leuken, L. Garcia, X. Olivares, R. van Zwol, Visual diversification of image search results, in: WWW, 2009, pp. 341–350.
[57] E. Vidal, An algorithm for finding nearest neighbors in (approximately) constant average time, Pattern Recogn. Lett. 4 (1986) 145–157.
[58] J. Wang, J. Zhu, Portfolio theory of information retrieval, in: SIGIR, 2009, pp. 115–122.
[59] P. Yianilos, Data structures and algorithms for nearest neighbor search in general metric spaces, in: SODA, 1993, pp. 311–321.
[60] P. Yianilos, Locally lifting the curse of dimensionality for nearest neighbor search, in: SODA, 2000, pp. 361–370.
[61] P. Zezula, G. Amato, V. Dohnal, M. Batko, Similarity Search: The Metric Space Approach, Advances in Database Systems, vol. 32, Springer, 2006.
[62] C. Zhai, W.W. Cohen, J. Lafferty, Beyond independent relevance: Methods and evaluation metrics for subtopic retrieval, in: SIGIR, 2003, pp. 10–17.