Proportionality in Spatial Keyword Search

Georgios Kalamatianos
Department of Information Technology, Uppsala University
Uppsala, Sweden
georgios.kalamatianos@it.uu.se

Georgios J. Fakas
Department of Information Technology, Uppsala University
Uppsala, Sweden
georgios.fakas@it.uu.se

Nikos Mamoulis
Department of Computer Science and Engineering, University of Ioannina
Ioannina, Greece
nikos@cs.uoi.gr

ABSTRACT

More often than not, spatial objects are associated with some context, in the form of text, descriptive tags (e.g. points of interest, flickr photos), or linked entities in semantic graphs (e.g. Yago2, DBpedia). Hence, location-based retrieval should be extended to consider not only the locations but also the context of the objects, especially when the retrieved objects are too many and the query result is overwhelming. In this paper, we study the problem of selecting a subset of the query result, which is the most representative. We argue that objects with similar context and nearby locations should proportionally be represented in the selection. Proportionality dictates the pairwise comparison of all retrieved objects and hence bears a high cost. We propose novel algorithms which greatly reduce the cost of proportional object selection in practice. Extensive empirical studies on real datasets show that our algorithms are effective and efficient. A user evaluation verifies that proportional selection is more preferable than random selection and selection based on object diversification.

CCS CONCEPTS

• **Information systems** → **Data management systems**; **Spatial-temporal systems**; **Retrieval models and ranking**; • **Theory of computation** → **Design and analysis of algorithms**.

KEYWORDS

Proportionality, diversity, keyword search, Ptolemy's spatial diversity, spatial data, ranking.

ACM Reference Format:
Georgios Kalamatianos, Georgios J. Fakas, and Nikos Mamoulis. 2021. Proportionality in Spatial Keyword Search. In *Proceedings of the 2021 International Conference on Management of Data (SIGMOD '21), June 20–25, 2021, Virtual Event, China.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3448016.3457309

1 INTRODUCTION

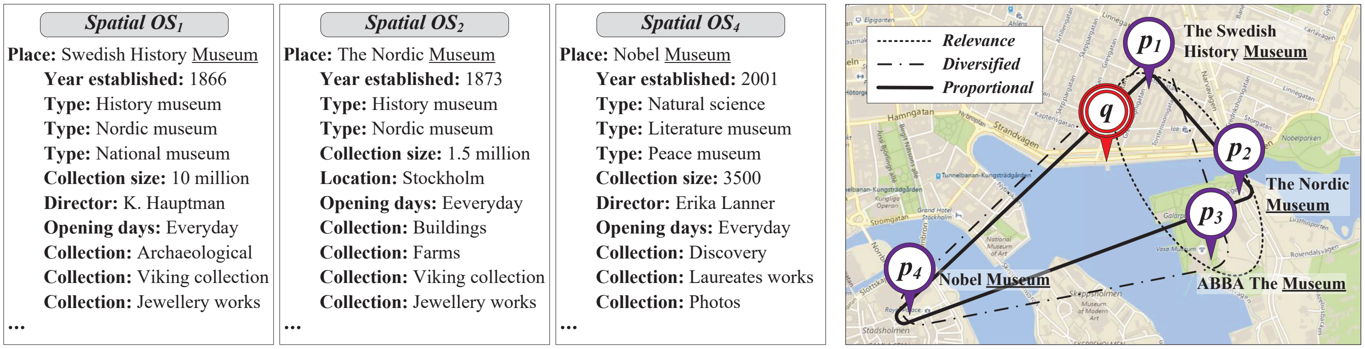
Thousands of public and private datasets which include geo-spatial information exist. For instance, on the web, there are datasets with GIS objects or POIs (e.g. spatialhadoop datasets[1]), datasets with geo-tagged photographs (e.g. flickr), online social networks (e.g. Facebook, Gowalla), semantic knowledge graphs (e.g. YAGO [27], DBpedia), etc. Acknowledging the significance of discovering datasets and making them universally accessible and useful, Google recently introduced *Google Dataset Search*[2], which facilitates the discovering of web-accessible datasets. Acknowledging also the need for retrieval, various search paradigms have been proposed by the research community. For instance, keyword search paradigms liberate users from technical details such as understanding the nature and structure of the data or a programming language [1, 14, 21, 28, 29, 33, 39, 40].

In this paper, we focus on the *location-based* retrieval of spatial entities in datasets. We assume that the spatial objects, besides having a location, are also enriched with some *context*. The context could be either *explicit*, i.e. in the form of descriptive text or tags, or *implicit*, i.e. it could be derived by *linked* neighboring objects in semantic RDF (resource description framework) graphs. Retrieval models that consider the context of spatial objects, typically combine proximity to a query location and contextual relevance to a set of query keywords [9]. If the context is explicit, popular information retrieval models, such as cosine similarity or tf-idf, can be used to model relevance [3]. Examples of datasets on which such models apply are collections of POIs or geo-located flickr photographs annotated with description tags. If the context is implicit, contextual relevance can be defined by considering the linked entities in subgraphs which include the query keywords. For instance, the search paradigms of [5, 38] consider minimal subgraphs of nodes that collectively contain the keywords, whereas the object summaries (*OS*s) paradigm [13–18] considers trees rooted at nodes containing the keywords. Examples of datasets on which such models apply are RDF knowledge graphs (e.g., YAGO, DBpedia) and social networks (e.g. Facebook, Gowalla). It is important to note that, regardless of the type of spatial objects and datasets, contextual similarity between objects can be measured using Jaccard similarity between the corresponding sets of items in their context. Namely, the items can be keywords, tags, data set nodes, RDF graph nodes.

The *OS* paradigm summarises information about entities and constitutes an example of implicit context in graphs. A spatial *OS* (*sOS*) is a tree rooted at a spatial entity in a database (i.e., a tuple with a location attribute) or an RDF graph and its context is derived by the set of neighbouring important entities (linked either directly or indirectly to the spatial root via foreign key links or RDF predicates). For example, consider a user that wishes to get information about museums in Stockholm from DBpedia (Figure 1). A spatial *OS* will comprise a node representing the "Swedish

[1] http://spatialhadoop.cs.umn.edu
[2] toolbox.google.com/datasetsearch

SIGMOD '21, June 20–25, 2021, Virtual Event, China
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8343-1/21/06…$15.00
https://doi.org/10.1145/3448016.3457309



Spatial OS_1

Place: Swedish History Museum
Year established: 1866
Type: History museum
Type: Nordic museum
Type: National museum
Collection size: 10 million
Director: K. Hauptman
Opening days: Everyday
Collection: Archaeological
Collection: Viking collection
Collection: Jewellery works
...

Spatial OS_2

Place: The Nordic Museum
Year established: 1873
Type: History museum
Type: Nordic museum
Collection size: 1.5 million
Location: Stockholm
Opening days: Eeveryday
Collection: Buildings
Collection: Farms
Collection: Viking collection
Collection: Jewellery works
...

Spatial OS_4

Place: Nobel Museum
Year established: 2001
Type: Natural science
Type: Literature museum
Type: Peace museum
Collection size: 3500
Director: Erika Lanner
Opening days: Everyday
Collection: Discovery
Collection: Laureates works
Collection: Photos
...

(a) Spatial Object Summaries

(b) Map of Places

Figure 1: Example of Proportionality (querying for *museums* in *Stockholm*)

History Museum" as a root and child nodes including contextual information, e.g. "Nordic museum", "History museum", "Viking collections", etc. (spatial OS_1).

The retrieval goal is finding spatial objects, which are near the query location and relevant to the query context (e.g., keywords, entities). A *retrieval score* for each query result can be defined by combining spatial distance with contextual relevance (e.g., to query keywords). Still, the query results could be too many and may overwhelm the user. A typical approach is to *rank* the results based on their score and return the top-k objects [9, 38]. However, the most relevant spatial objects could be in the same direction w.r.t. the query location and/or could be too similar to each other in terms of context [12, 31, 41]. Consider a user at location q in Figure 1, who is searching for nearby museums; the top-3 places p_1, p_2, and p_3 are all located in the same direction with respect to the query and have almost similar context (2 out of 3 are history museums).

Several studies reveal that users strongly prefer spatially and contextually diversified query results over un-diversified ones and propose algorithms which select a small number of results which are not only relevant, but also *spatially and contextually diverse* [43, 48]. Recently, Cai et al. [5] introduced diversification on spatial keyword search by combining relevance and diversity. Namely, the output places, in addition to being relevant to the query, should be diverse w.r.t. their context and location. For instance, a diversified query result for Figure 1 could include p_1 (a history museum), p_3 (ABBA museum) and p_4 (Nobel museum). These places are close to the query and at the same time they are diverse because they are located in different directions w.r.t. the query location and they have quite different context.

Still, simple diversity measures disregard the spatial and contextual distribution of the objects; hence, they may fail to retrieve a *representative* subset of the query results, compromising the quality of results. For instance in our example, we see that 2 out of 4 places are history museums in the same direction w.r.t query location. More precisely, these two places share many common nodes (e.g. common Type and Collection nodes) and are located in the same direction w.r.t. query. This reveals that the general area is dominated by (history) museums located on the right side of the query. Therefore, by representing *proportionally* these properties (at the same time facilitating diversity), we assist users to comprehend the area; diversification fails to reveal such insightful information. Thus, in this paper, we study selecting a subset of the query results by combining (1) relevance, (2) spatial proportionality w.r.t. the query location and (3) contextual proportionality w.r.t. the descriptive entities of the objects. In our running example, a proportional result will include p_1, p_2 and p_4; where similar and proportional p_1 and p_2 places are diverse to p_4. Our problem definition and solutions are general and can be applied to any search paradigm where the output is a (ranked) set of spatial entities with context.

The proportionality problem introduces efficiency challenges as we need to perform pairwise comparison to all retrieved objects, in order to determine the frequent common properties. Hence, we propose novel efficient algorithms addressing contextual and spatial proportionality. Our contributions can be summarized as follows:

- We introduce the problem of proportionality in location-based retrieval for objects with context and show that it is NP-hard. We also propose novel proportionality measures w.r.t. location and context.
- We propose a novel efficient algorithm for contextual proportionality (i.e. Micro set Jaccard hashing).
- We propose novel efficient algorithms for the calculation of spatial proportionality (i.e., grid based algorithms).
- We propose a generic algorithmic framework that adapts existing greedy diversification algorithms (i.e. IAdU and ABP) [5]. We prove the approximation bounds of IAdU and ABP for the proportional selection problem.
- We present a thorough evaluation on real datasets demonstrating the efficiency of our algorithms. We conduct a user evaluation verifying that proportional results are more preferable than non-proportional or diversified results.

The rest of the paper is organized as follows. Sections 2 and 3 present related and background work. Sections 4 and 5 formalize our problem and introduce the general framework. Sections 6 and 7 propose efficient contextual and spatial proportionality algorithms. Section 8 provides approximation bounds of IAdU and ABP. Section 9 contains our experimental evaluation and Section 10 conclusions.

2 RELATED WORK

Our proposed proportional selection framework considers (1) the relevance of the objects to the query (i.e., spatial distance and keywords similarity) (2) contextual proportionality and (3) spatial proportionality w.r.t. the query location. To the best of our knowledge,

Table 1: Related Work vs. Our Work ([this])

Contextual Proportionality		Spatial Proportionality		Relevance
Entities	Topics	Query Location	Regions	
[this], [19]	[7, 11, 42, 46]	**[this]**	[23]	**[this]**, [19, 46]

there is no previous work that considers all these together in proportional selection, as Table 1 shows. Hereby, we discuss and compare related work in diversification and proportionality.

Diversification. Diversification of query results has attracted a lot of attention as a method for improving the quality of results by balancing relevance to the query and dissimilarity among results [8, 19, 20, 25, 45]. The motivation is that, in non-diversified search methods, users are overwhelmed with many similar answers with minor differences [31]. PerK [41] and DivQ [12] address the diversification problem in keyword search over relational databases; they use Jaccard distance as a measure of similarity between the keywords in the node-sets that constitute the query results.

Spatial Diversification. Several works consider spatial diversification, which selects objects that are well spread in the region of interest [6, 37]. In [24, 30], diversity is defined as a function of the distances between pairs of objects. However, considering only the distance between a pair and disregarding their orientation could be inappropriate. In view of this, van Kreveld et al. [44] incorporate the notion of angular diversity, wherein a maximum objective function controls the size of the angle made by a selected object, the query location, and an unselected object. Recently, Cai et al. [5] combine both spatial and contextual diversity and propose a new measure for spatial diversity (to be described in detail in Section 3).

Contextual Proportionality. [7, 10, 42, 46] facilitate proportional diversity by considering *topics* (categories) on items' characteristics and then by proportionally representing these topics. In contrast, our work considers proportionality directly on entities (words, nodes, etc.) which is more dynamic and avoids complications of classifying results in topics (Table 1). In [10] (an early work on this area), an election-based method is proposed to address proportionality. However, this method disregards the relevance of items to the query and thus they may result in picking irrelevant items. In [19, 46], this limitation is addressed by considering relevance in the objective function. Proportionality has also been studied in recommendation systems. For instance, [47] facilitates proportionality by considering topics on both users and items' characteristics. Previous work does not solve the proportionality problem, considering spatial relevance and diversity in space and context.

Spatial Proportionality has also been studied on Geographical data. For instance, [23] facilitates proportionality by clustering POIs in sub-regions and then by proportionally recommending POIs from these sub-regions. This approach is restrictive since proportionality is based on *static regions* rather than dynamic areas around a query location (which is what we propose); in addition, this approach uses the locations of POIs, but disregards their *context* (Table 1).

Jaccard Similarity Computation. Our approach involves Jaccard similarity computations for numerous pairs of (small) sets. Existing work on efficient Jaccard similarity calculation between sets focuses on the scalability w.r.t. both (1) the size of sets and (2) the number of sets. For instance, minhash is an approximation algorithm that detects near duplicate web pages. Many of these algorithms are top-k (or threshold based) and thus are designed to terminate fast by pre-processing sets (e.g., sorting or LSH (locality-sensitive hashing) [4, 34]). Such a processing can be an effective investment for top-k searches; on the other hand, in our case where we need to compare all pairs, it is an unnecessary overhead. Some algorithms (e.g., minhash) construct signatures in order to speed-up comparisons. Similarly, signatures require preprocessing, which is a useful investment on very large sets; however, for moderate to small sets (as in our case), signatures are not effective and this preprocessing does not pay off. In summary, existing eminent techniques that address scalability in operations that involve Jaccard similarity computations are not appropriate for our problem.

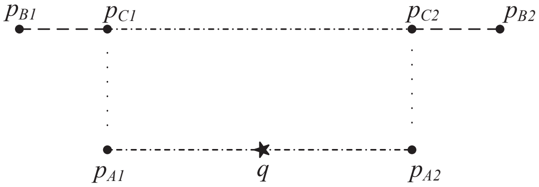


Figure 2: Ptolemy's Spatial Diversity
$dS(p_{A1}, p_{A2}) > dS(p_{B1}, p_{B2}) > dS(p_{C1}, p_{C2})$

3 BACKGROUND

In this section, we describe the type of data that we focus on and discuss applications which manage or use such data. In addition, we discuss in more details definitions of relevance and diversity that apply on the data.

Spatial objects with context. We consider a large collection of objects which have spatial locations and some form of context. The spatial locations are described by a set of coordinates and common distance metrics apply on them (e.g., Euclidean distance). The context can be in different forms [22, 32]. Specifically, the context can simply be a set of descriptive keywords or tags. Another type of context could be the set of nodes (or RDF entities), which are linked to the object in a graph. Regardless the form of the context and without loss of generality, we use Jaccard similarity to model the similarity between the contexts of two objects.

Spatial diversity. Cai et al. [5] propose Ptolemy's diversity, a new spatial diversity metric, which considers the query location and relative direction of objects from it. Ptolemy's diversity between two places p_i and p_j with respect to a query location q is defined as follows:

$$dS(p_i, p_j) = \frac{||p_i, p_j||}{||p_i, q|| + ||p_j, q||}, \tag{1}$$

where $||p_i, p_j||$ is the Euclidean distance between p_i and p_j. $dS(p_i, p_j)$ is naturally normalized to take values in $[0, 1]$, since $||q, p_i|| + ||q, p_j|| \geq ||p_i, p_j||$ (triangle inequality). Two places p_i and p_j receive a maximum diversity score $dS(p_i, p_j) = 1$, if they are diametrically opposite to each other w.r.t. to q; e.g., points p_{A1} and p_{A2} in Figure 2.

In the same figure, pair of places (p_{C1}, p_{C2}) have the same distance as pair (p_{A1}, p_{A2}), but $dS(p_{C1}, p_{C2}) < dS(p_{A1}, p_{A2})$, because p_{C1} and p_{C2} are in the same direction w.r.t q (i.e., north of q). Pair (p_{B1}, p_{B2}) are further from each other compared to the places in pair (p_{C1}, p_{C2}) and consequently have a higher diversity score (this can be shown using Pythagorean theorem).

Table 2: Notations

Notation	Definition
p_i	A place (p_i also denotes the location and context of the place)
$C(p_i)$	Set of contextual items of p_i (e.g., keywords or vertices)
$\lvert C(p_i)\rvert$ or $\lvert p_i\rvert$	Number of elements in contextual set of place p_i
$rF(p_i)$	Relevance score of p_i w.r.t. q
$sC(p_i, p_j)$	Contextual (Jaccard) similarity
$sS(p_i, p_j)$	Ptolemy's spatial similarity; i.e. $1 - dS(p_i, p_j)$ (Eq. 1)
$sF(p_i, p_j)$	Weighted similarity of p_i and p_j (Eq. 13)
$pC\mathcal{S}(p_i)$	Contextual proportionality of p_i w.r.t. $\mathcal{S}$ (Eq. 3)
$pC\mathcal{R}(p_i)$	Contextual proportionality of p_i w.r.t. $\mathcal{R}$ (Eq. 4)
$pS\mathcal{S}(p_i)$	Spatial proportionality of p_i w.r.t. $\mathcal{S}$ (Eq. 6)
$pS\mathcal{R}(p_i)$	Spatial proportionality of p_i w.r.t. $\mathcal{R}$ (Eq. 7)
$pF\mathcal{S}(p_i)$	Weighted summation of $pC\mathcal{S}(p_i)$ and $pS\mathcal{S}(p_i)$ (Eq. 11)
$pF\mathcal{R}(p_i)$	Weighted summation of $pC\mathcal{R}(p_i)$ and $pS\mathcal{R}(p_i)$ (Eq. 12)
$pC(p_i)$	Contextual proportionality score of p_i (Eq. 2)
$pS(p_i)$	Spatial proportionality score of p_i (Eq. 5)
$pF(p_i)$	Combined (contextual and spatial) proportionality of p_i (Eq. 8)
$HPF(p_i, p_j)$	Holistic proportionality between p_i and p_j (Eq. 15)
$HPF(\mathcal{R})$	Holistic proportionality score of $\mathcal{R}$ (Eq. 10)
$cHPF(p_i)$	Proportional contribution of p_i if added to $\mathcal{R}$ (used by IAdU)

4 PROPORTIONAL SELECTION PROBLEM

Consider a query q and its result $\mathcal{S}$, a set of retrieved places. Each place $p_i \in \mathcal{S}$ carries (1) a relevance score $rF(p_i)$ combining the distance to q and potentially other criteria (such as relevance to a set of query keywords [38]), (2) a location and (3) a context (i.e. a set of contextual items such as keywords, nodes, etc.). Our objective is to find a subset $\mathcal{R}$ of $\mathcal{S}$ that combines a *relevance function* to the query and a *proportionality function* that considers the location and the context of each place. If K and k denote the sizes of $\mathcal{S}$ and $\mathcal{R}$, respectively, then it should be $k < K$. Note that our problem definition is general and is independent from any paradigm used to derive the set $\mathcal{S}$ of retrieved objects. For instance, the places can be geo-textual search results [9], spatial object summaries [14], spatial keyword search results over RDF graphs [38], etc.

For each place p_i in the retrieved set of places $\mathcal{S}$, we assume that the relevance score $rF(p_i)$ of p_i to the query is known. The exact definition of the relevance function $rF(p_i)$ depends on the retrieval model used; e.g., it could be a linear combination of the Euclidean distance between p_i and the query location q and the relevance of p_i's context to the query keywords [9, 38].

In this section, we first define proportionality with respect to context and location; then, we define a holistic score that trades off relevance and proportionality; finally, we define the problem formally. For a place p_i, we overload the notation p_i to denote its location and contextual set; we also use $C(p_i)$ to denote the contextual set wherever necessary. Table 2 shows the most frequently used notation in the paper.

4.1 Proportionality Function

Contextual proportionality. We observe in the example of Figure 1 that the places in the retrieved set $\mathcal{S}$ may have common elements in their contexts. For instance, "History museum", "Nordic museum", "Viking collections", "Jewelry works" appear in both spatial OS_1 and OS_2 of $\mathcal{S}$. These contextual elements are representative for the spatial region which includes OS_1 and OS_2. Therefore, we argue that in the selection of the subset $\mathcal{R}$, we should favor proportionally places that include such frequent contextual elements. At the same time, we argue that results forming $\mathcal{R}$ should be dissimilar as to facilitate diversity. In view of these properties we define the proportional score of a place p_i w.r.t. its context as follows:

$$pC(p_i) = pC\mathcal{S}(p_i) - pC\mathcal{R}(p_i), \qquad (2)$$

where

$$pC\mathcal{S}(p_i) = \sum_{p_j \in \mathcal{S}, p_i \neq p_j} sC(p_i, p_j), \qquad (3)$$

$$pC\mathcal{R}(p_i) = \sum_{p_j \in \mathcal{R}, p_i \neq p_j} sC(p_i, p_j). \qquad (4)$$

Here, $sC(p_i, p_j)$ measures the contextual similarity of two places as the Jaccard similarity between the corresponding sets of elements $C(p_i), C(p_j)$ (e.g., keywords, graph vertices, etc.) in their contexts; i.e. $sC(p_i, p_j) = \frac{|C(p_i) \cap C(p_j)|}{|C(p_i) \cup C(p_j)|}$. $pC\mathcal{S}(p_i)$ aggregates the similarity between p_i and all other places in $\mathcal{S}$. We also define $pC\mathcal{R}(p_i)$ as the similarity of p_i to the rest of places in $\mathcal{R}$. The rationale is that, in our selection, we should penalize p_i if it has large similarity $pC\mathcal{R}(p_i)$ with the rest places in $\mathcal{R}$. Hence, to assess the value of p_i in R, we subtract $pC\mathcal{R}(p_i)$ from $pC\mathcal{S}(p_i)$. This is inspired by earlier work in proportionality [10, 19] that follows the same strategy. The proportional score $pC(p_i)$ of a place ranges in $[0, K - k]$, where K and k denote the amount of elements in $\mathcal{S}$ and $\mathcal{R}$ respectively, since each $sC(p_i, p_j)$ ranges in $[0, 1]$.

Spatial proportionality. Similarly, we define the proportionality score of a place w.r.t the query location. For instance in our running example, we observe that the area containing places p_1, p_2, p_3 is a representative neighborhood for the given query (i.e. for both keywords and location), as opposed to the area containing the spatial outlier p_4. Therefore, we argue that we should favor proportionally places located in such representative neighborhoods w.r.t. the query location. At the same time, we argue that places should be located in diverse directions w.r.t the query location. In view of these properties, we define the proportionality score of a place as follows:

$$pS(p_i) = pS\mathcal{S}(p_i) - pS\mathcal{R}(p_i), \qquad (5)$$

where

$$pS\mathcal{S}(p_i) = \sum_{p_j \in \mathcal{S}, p_i \neq p_j} sS(p_i, p_j), \qquad (6)$$

$$pS\mathcal{R}(p_i) = \sum_{p_j \in \mathcal{R}, p_i \neq p_j} sS(p_i, p_j). \qquad (7)$$

Here, $sS(p_i, p_j)$ measures the pairwise spatial similarity of two points w.r.t. q by using the complementary of their Ptolemy's spatial diversity (i.e. $sS(p_i, p_j) = 1 - dS(p_i, p_j)$, Eq. 1). The rationale of the $pS\mathcal{S}(p_i)$ definition is to favor a place with many neighbors in $\mathcal{S}$ w.r.t q. Similarly, $pS\mathcal{R}(p_i)$ favors places spatially diverse to the rest of the places in $\mathcal{R}$. Thus, both $pS\mathcal{S}(p_i)$ and $pS\mathcal{R}(p_i)$ consider the query location q. $pS(p_i)$ score also ranges in $[0, K - k]$. Like $pC\mathcal{S}(p_i)$, $pS\mathcal{S}(p_i)$ also requires computing $sS(p_i, p_j)$ for all pairs. In Section 6, we propose algorithms that accelerate these computations.

Combined scores. We can combine contextual and spatial proportionality to a *proportionality* score as follows:

$$pF(p_i) = (1 - \gamma) \cdot pC(p_i) + \gamma \cdot pS(p_i), \qquad (8)$$

where $\gamma \in [0, 1]$ controls the relative importance of the two factors. We can combine proportionality and relevance to a holistic score:

$$HPF(p_i) = (1 - \lambda) \cdot (K - k) \cdot rF(p_i) + \lambda \cdot pF(p_i), \qquad (9)$$

where $\lambda \in [0,1]$ controls the relative importance of relevance and proportionality. We multiply the relevance score $rF(p_i)$ by $K-k$ in order to normalize against $pF(p_i)$ that ranges in $[0, K-k]$. Finally, we can combine these scores for all places in $\mathcal{R}$:

$$HPF(\mathcal{R}) = \sum_{p_i \in \mathcal{R}} HPF(p_i). \tag{10}$$

Additional useful definitions. Before we proceed with the problem definition, we also introduce additional definitions that are used through the paper. First, we introduce weighted (γ) scores:

$$pF\mathcal{S}(p_i) = (1-\gamma) \cdot pC\mathcal{S}(p_i) + \gamma \cdot pS\mathcal{S}(p_i), \tag{11}$$

$$pF\mathcal{R}(p_i) = (1-\gamma) \cdot pC\mathcal{R}(p_i) + \gamma \cdot pS\mathcal{R}(p_i) \tag{12}$$

$$sF(p_i, p_j) = (1-\gamma) \cdot sC(p_i, p_j) + \gamma \cdot sS(p_i, p_j) \tag{13}$$

$pF\mathcal{S}(p_i)$ (resp. $pF\mathcal{R}(p_i)$) is the combined similarity (contextual and spatial) of p_i and all other places in $\mathcal{S}$ (resp. $\mathcal{R}$), whereas $sF(p_i, p_j)$ is the combined similarity between p_i and p_j. Based on the above equations, we can also define the proportionality score $pF(p_i)$ as:

$$pF(p_i) = pF\mathcal{S}(p_i) - pF\mathcal{R}(p_i) \tag{14}$$

We also introduce the following pairwise holistic score that can facilitate the heuristics of our greedy algorithms (Section 5):

$$\begin{aligned} HPF(p_i, p_j) = (1-\lambda) \cdot (K-k) \cdot \frac{rF(p_i) + rF(p_j)}{k-1} + \\ \lambda \cdot \left(\frac{pF\mathcal{S}(p_i) + pF\mathcal{S}(p_j)}{k-1} - 2 \cdot sF(p_i, p_j)\right). \end{aligned} \tag{15}$$

This score is defined in such a way that the summation of $HPF(p_i, p_j)$ scores of all pairs of places in $\mathcal{R}$ will give us the same score as the summation of $HPF(p_i)$ scores of all places in $\mathcal{R}$ (i.e. $HPF(\mathcal{R}) = \sum_{p_i \in \mathcal{R}} HPF(p_i) = \sum_{p_i, p_j \in \mathcal{R}, p_i \neq p_j} HPF(p_i, p_j)$). (Note that Equation 9 can also be defined as $HPF(p_i) = (1-\lambda) \cdot (K-k) \cdot rF(p_i) + \lambda \cdot (pF\mathcal{S}(p_i) - pF\mathcal{R}(p_i))$.) Then, the summations of $rF(p_i)$, $pF\mathcal{S}(p_i)$ and $pF\mathcal{R}(p_i)$ for all places in $\mathcal{R}$ are equal to the summations of $\frac{rF(p_i)+rF(p_j)}{k-1}$, $\frac{pF\mathcal{S}(p_i)+pF\mathcal{S}(p_j)}{k-1}$ and $2 \cdot sF(p_i, p_j)$ for all pairs in $\mathcal{R}$ respectively, i.e. $\sum_{p_i \in \mathcal{R}} rF(p_i)$, $\sum_{p_i \in \mathcal{R}} pF\mathcal{S}(p_i)$ and $\sum_{p_i \in \mathcal{R}} pF\mathcal{R}(p_i)$ respectively. Thus, we have:

$$\begin{aligned} HPF(\mathcal{R}) = (1-\lambda) \cdot (K-k) \cdot \sum_{p_i \in \mathcal{R}} rF(p_i) + \\ +\lambda \cdot \left(\sum_{p_i \in \mathcal{R}} pF\mathcal{S}(p_i) - \sum_{p_i \in \mathcal{R}} pF\mathcal{R}(p_i)\right). \end{aligned} \tag{16}$$

4.2 Problem Definition

Hereby, we define the proportional selection problem. As proven below this problem is NP-hard; thus, we resort to greedy algorithms for solving it.

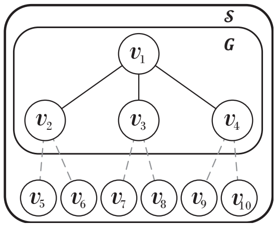
Problem Definition 1. *Given a set of K places $\mathcal{S}$ (where each place carries a relevance score, location and set of contextual items), a query location q, and an integer $k < K$, find a set $\mathcal{R}$ of k places that have the highest $HPF(\mathcal{R})$ among all k-subsets of $\mathcal{S}$.*

Theorem 4.1. *Problem 1 is NP-hard.*

Proof. In order to prove the hardness of our proportionality problem, we construct a reduction from the independent set problem. Given an undirected graph $G(V, E)$ and a positive integer k, ($k \leq |V|$), the independent set problem is to decide if G contains an independent set $\mathcal{R}$ of size k (i.e. there is not any edge connecting any pair of nodes in $\mathcal{R}$).

We generate an instance of our problem as follows. Each vertex v_i in V corresponds to a place p_i with a contextual set $C(p_i)$. For every edge (v_i, v_j) in E, we add an element $v_{i,j}$ to the contextual sets of both p_i and p_j. We now construct the complete set of places $\mathcal{S}$ as follows. First, we add to $\mathcal{S}$ all places that correspond to vertices of V. Let d be the maximum degree of any vertex in V. For each vertex $v_i \in V$, for which the degree $deg(v_i)$ is less than d, we add $d - deg(v_i)$ new places in $\mathcal{S}$ and "connect" them to v_i. Namely, for each such new place p_j, we add an element $v_{i,j}$ to the contextual sets of both p_i and p_j. Finally, we add to the contextual set $C(p_j)$ of each new place p_j $d-1$ elements which are unique to p_j (i.e., no other place has any of these elements in its contextual set). As a result, each p_i corresponding to a vertex in V with a degree less than d will have *exactly one common element* with each of the new places linked to it. In general, all places p_i, which correspond to vertices in V will have identical $pC\mathcal{S}(p_i)$ scores because (1) they all have exactly one common element with exactly d places in $\mathcal{S}$ and (2) all places in $\mathcal{S}$ have exactly d elements in their contextual sets. In addition, all places p_j which do not correspond to vertices in V (i.e., all places added later), will have exactly one common element with exactly one place in $\mathcal{S}$. This means that the $pC\mathcal{S}(p_i)$ scores of all p_is corresponding to vertices in V are equal and strictly larger than the $pC\mathcal{S}(p_j)$ scores of all other places p_j.

We can now prove that the k-subset $\mathcal{R}$ of $\mathcal{S}$, which maximizes $HPF(\mathcal{R})$ is a k independent set in the original graph G. We consider a special case of our problem, where $\lambda = 1$ (i.e., we disregard relevance) and $\gamma = 0$ (i.e., we disregard Ptolemy's diversity). First, all k-subsets of $\mathcal{S}$, which include only vertices in V have a common $\sum_{p_i \in \mathcal{R}} pF\mathcal{S}(p_i)$ score (equal to $\sum_{p_i \in \mathcal{R}} pC\mathcal{S}(p_i)$, since $\gamma = 0$), which is higher than the corresponding score of all k-subsets which include some vertex outside V. This is because all vertices in such a subset have the maximum possible $pC\mathcal{S}(p_i)$ score (as discussed above). Second, all k independent sets from V correspond to k-subsets for which the quantity $\sum_{p_i \in \mathcal{R}} pF\mathcal{R}(p_i)$ is zero. This is because, all pairs of places in such a set have no common elements. The reduction takes polynomial time, since the maximum degree of any vertex in $|V|$ is $|V|-1$, which means that we should add at most $|V| \cdot (|V|-1)$ edges and vertices. This completes the proof. □



(a) Independent set

$$\begin{aligned} C(p_1) &= \{v_{1,2}, v_{1,3}, v_{1,4}\} \\ C(p_2) &= \{v_{1,2}, v_{2,5}, v_{2,6}\} \\ C(p_3) &= \{v_{1,3}, v_{3,7}, v_{3,8}\} \\ C(p_4) &= \{v_{1,4}, v_{4,9}, v_{4,10}\} \\ C(p_5) &= \{v_{2,5}, v_{5a}, v_{5b}\} \\ C(p_6) &= \{v_{2,6}, v_{6a}, v_{6b}\} \\ C(p_7) &= \{v_{3,7}, v_{7a}, v_{7b}\} \\ C(p_8) &= \{v_{3,8}, v_{8a}, v_{8b}\} \\ C(p_9) &= \{v_{4,9}, v_{9a}, v_{9b}\} \\ C(p_{10}) &= \{v_{4,10}, v_{10a}, v_{10b}\} \end{aligned}$$

(b) Corresponding contextual sets

Figure 3: Example of Reduction

Figure 3 shows an example of the reduction. Consider the graph shown in Figure 3(a), which includes four vertices, such that v_1 is connected to all vertices and there are no other edges. A 3-independent set in this graph is $\{v_2, v_3, v_4\}$. For the reduction, we initially define $C(p_1) = \{v_{1,2}, v_{1,3}, v_{1,4}\}$, $C(p_2) = \{v_{1,2}\}$, $C(p_3) =$

$\{v_{1,3}\}$, and $C(p_4) = \{v_{1,4}\}$. Then, for each one of the vertices $\{v_2, v_3, v_4\}$, we connect it to two new vertices, add the corresponding new places to $\mathcal{S}$, and update the corresponding contexts. This results in all four original vertices in V to have the same (maximum) degree 3; hence, all corresponding places have 3 elements in their contexts and any subset with $k = 3$ such vertices have the same (maximum) sum of $pF\mathcal{S}(p_i)$ scores. At the same time, each vertex in the independent set $\mathcal{R} = \{v_2, v_3, v_4\}$ has a zero $pF\mathcal{R}(p_i)$ score. Overall, any k independent set problem can be converted to a special case of our problem for $\lambda = 1$ and $\gamma = 0$.

5 GENERIC PROPORTIONALITY ALGORITHMIC FRAMEWORK

Our problem (Definition 1) necessitates the pairwise comparison of all places in $\mathcal{S}$ in order to compute all the proportionality scores $HPF(p_i, p_j)$, essential to determine the k-sized subset $\mathcal{R}$ with the highest $HPF(\mathcal{R})$. We propose a two-step algorithmic framework, which first computes and stores the pairwise scores, which are then used for finding the solution. As we explain below, our main contribution is in the first step, since we use previously known greedy algorithms for the second step.

Step 1: Compute proportionality scores of $\mathcal{S}$. The greedy algorithms utilise Equation 15 which requires $rF(p_i)$, $pC\mathcal{S}(p_i)$, $p\mathcal{S}\mathcal{S}(p_i)$ and $sF(p_i, p_j)$. In contrast to the $rF(p_i)$ score which is given to us, the calculation of $pC\mathcal{S}(p_i)$ and $p\mathcal{S}\mathcal{S}(p_i)$ is very challenging as it dictates the comparison of all pairs (p_i, p_j) of places in $\mathcal{S}$ (i.e., a quadratic number of pairs), in order to calculate their $sF(p_i, p_j)$. As we discuss in the following sections, baseline approaches for calculating sub functions $sC(p_i, p_j)$ and $sS(p_i, p_j)$ require up to $|C(p_i)|$ (size of the contextual set) and 20 operations, respectively. Hence, we need a total of $O(K^2 \cdot (|C(p_i)| + 20))$ operations for all pairs of places in $\mathcal{S}$. We introduce tailored algorithms that greatly reduce this complexity in practice (Sections 6 and 7). We also compare them with such baseline approaches [5]. In order to avoid performing redundant computations, after an $sF(p_i, p_j)$ score is computed, it is cached and reused whenever necessary during the execution of our greedy algorithms.

Step 2: Compute $\mathcal{R}$. The problem is NP-hard, as we have already shown. We use two alternative greedy algorithms from previous work [5], i.e., IAdU and APB. The two algorithms use thresholds in order to facilitate a faster termination, which we adapt accordingly. Hereby, we will focus our description on the heuristics, the respective adaptations and their complexity (efficiency aspects can be found in [5]). In Section 8, we study their approximation bounds.

IAdU. This algorithm iteratively constructs the result set $\mathcal{R}$ by selecting each time the place from $\mathcal{S}$ that maximizes the contribution it can make towards the overall score $HPF(\mathcal{R})$. The contribution $cHPF(p_i)$ of p_i to be added to the current result set $\mathcal{R}$ is defined as follows:

$$cHPF(p_i) = \begin{cases} rF(p_i), & \text{if } \mathcal{R} = \emptyset, \\ \sum_{p_j \in \mathcal{R}} HPF(p_i, p_j), & \text{otherwise.} \end{cases} \tag{17}$$

$cHPF(p_i)$ considers the relevance score and the proportionality of p_i against existing elements in $\mathcal{R}$. In the first iteration, $\mathcal{R}$ is empty, thus the available contribution of a place can only be the corresponding $rF(p_i)$ score. The contributions of all other places are then updated to consider the new entry in $\mathcal{R}$. Then, the algorithm iteratively selects the place p_i that maximizes $cHPF(p_i)$ w.r.t. the current $\mathcal{R}$, adds p_i to $\mathcal{R}$, and updates the contribution of the places not in $\mathcal{R}$. The complexity of the algorithm is $O(K \cdot k \cdot logK + K^2)$.

ABP. This algorithm greedily constructs the result set $\mathcal{R}$ by iteratively selecting the pair of places (p_i, p_j) with the largest $HPF(p_i, p_j)$ score (Equation 15). ABP selects the next pair (p_i, p_j) based on only its $HPF(p_i, p_j)$ value, independently of the relationships of p_i or p_j to places already in $\mathcal{R}$ (in contrast to IAdU). Once a pair is selected, both its constituent elements and any pairs they make are removed from further consideration by the algorithm (in a *lazy fashion*). Since a single pair is selected in each iteration, $\lfloor k/2 \rfloor$ iterations apply when the value of k is even. When k is odd, an arbitrary place is chosen to be inserted in the result set $\mathcal{R}$ as its last entity. The worst case complexity of the algorithm is $O(K^2 \cdot \log(K^2))$ which is higher than that of IAdU.

6 CONTEXTUAL PROPORTIONALITY CALCULATION

$pC\mathcal{S}(p_i)$ scores require the calculation of Jaccard similarity of all pairs of contextual sets of places in $\mathcal{S}$, which can be an expensive process. We propose a novel *micro set Jaccard hashing (msJh)* algorithm, which is tailored to the characteristics of our sets (i.e., numerous sets of moderate size). Jaccard similarity is a generic measure, appropriate for any type of contextual items (e.g. for sets of keywords, tags, RDF entities, nodes, etc.).

Baseline approach. We first discuss a baseline approach for computing the contextual similarities of all pairs of places in $\mathcal{S}$. This approach, for each pair, first creates a hash table with the elements of the first set and then uses it to check for each element in the second set if it appears in the first set. For comparing all pairs in $\mathcal{S}$, we still need to hash all K sets in $\mathcal{S}$. Assume, for simplicity, that all sets have the same size $|p_i|$. The hashing phase costs $O(K \cdot |p_i|)$, as we have to scan all elements from all sets. The comparison phase costs $O(K^2 \cdot |p_i|)$, because for each of the $O(K^2)$ pairs, we need $|p_i|$ checks in the worst case. The baseline approach is expensive if $\mathcal{S}$ contains many places; for instance, for $K = 100$ and $|p_i| = 5$, we need approximately 25,000 operations.

Minhash is an eminent technique for the fast calculation of Jaccard similarity on vast amounts of sets of big size. This approach works in two steps. During the first step, we apply t hash functions (i.e. $K \cdot t$ operations) on each set (where we get t minimum values). During the second step each pair is compared against the respective t minimum values (i.e. in total $K^2 \cdot t/2$ operations). Thus, in order to compare all pairs, we need in total of $K^2 \cdot t/2 + K \cdot t$ operations. Apparently, this approach can be very efficient when the number of elements $|p_i|$ in the contextual set of each place p_i is large, as $|p_i|$ does not affect the cost. We implemented this algorithm, in order to compare it with our proposed msJh algorithm, but it failed to perform well on our data, where the sets are relatively small.

6.1 Micro Set Jaccard hashing (msJh) Algorithm

In view of the limitations of the previous algorithms, we propose the micro set Jaccard hashing (*msJh*) algorithm. The algorithm generates an inverted list for each element with the sets wherein the element appears. The rationale of the *msJh* index is that we can

Algorithm 1 Micro set Jaccard hashing (msJh)

```
1: for each $p_i$ in $\mathcal{S}$ do
2:     for each element $v$ in $p_i$ do
3:         if $msHT[v]$ does not exist then
4:             Generate new $msHT[v]$ list
5:         Add $p_i$ in the front of $msHT[v]$ list
6: for each $p_i$ in $\mathcal{S}$ do
7:     for each element $v$ in $p_i$ do
8:         for each $p_j$ in $msHT[v]$ with $j > i$ do
9:             Update Jaccard Score $(p_i, p_j)$
```

	p_1	p_2	p_3	p_4	p_5
p_1:$\{a, b, c, d\}$		$\{a, d\} \Rightarrow 2/4$	$\{\} \Rightarrow 0$	$\{a, b\} \Rightarrow 2/5$	$\{b, c\} \Rightarrow 2/5$
p_2:$\{a, d\}$			$\{\} \Rightarrow 0$	$\{a\} \Rightarrow 1/4$	$\{\} \Rightarrow 0$
p_3:$\{e, f, g\}$				$\{\} \Rightarrow 0$	$\{\} \Rightarrow 0$
p_4:$\{a, b, h\}$					$\{b\} \Rightarrow 1/5$
p_5:$\{b, c, i\}$					

Figure 4: Example of the *msJh* Algorithm

facilitate a targeted Jaccard comparison. Namely, we facilitate the comparisons of sets only if we know they have common elements (by using *msJh*). Our technique is very efficient for small sets and at the same time computes it is exact (in contrast to minhash which is efficient on large sets with a minor approximation loss). The algorithm consists of two steps (i.e. Algorithm 1). Figure 4 illustrates an example.

Step 1: Generate *msJh*. We parse all sets and add on a hash table all elements and the sets wherein they appear (i.e., micro set hash table, denoted as *msht*; lines 1-5). More precisely, for each element we maintain a reverse list of the sets wherein the element appears (the reverse order of the places in the inverted list facilitates avoidance of redundant checks and we explain this in the following step). Figure 5 illustrates the *msht* for the example of Figure 4.

Step 2: Compare sets. We compare pairs in an economical fashion by utilising *msht*. More precisely, we calculate the intersection of any pair p_i and p_j, for pairs with $i < j$ and for each element v_l in p_i (lines 6-10). For instance, in our example of Figure 4, we will process first p_1. For each element in p_1 (i.e., $\{a, b, c, d\}$), we consult the *msht* as to see in which sets these elements appear. Then, we update the Jaccard (partial) scores accordingly. E.g. a of p_1 appears in p_4 and p_2. Then, we process b of p_1, which appears in p_5 and p_4. Recall that we add elements on *msht* in a reverse order. Thus, we can stop processing an element against sets that have been previously processed or against the set itself. For instance, while processing p_1, we will not compare a against p_1; also, while comparing p_2, we will not compare a against p_2 and p_1. An illustrative example of the savings of this algorithm (against the baseline algorithm) can be shown in the comparison of p_1 and p_3. Where, according to *msht*, the two sets have no common elements and this will result in zero operations. On the other hand, the baseline approach will still have to compare these two sets. Finally, given the intersection of $|p_i|$ and $|p_j|$, we can infer the union by subtracting the size of the intersection from $|p_i| + |p_j|$.

The algorithm has the following time costs. During the first step, we need to create the micro hash table, which requires $O(K|p_i|)$ time, where $|p_i|$ is the average number of elements in a set in $\mathcal{S}$.

Micro Hash	Sets (Reverse order)
a	p_4, p_2, p_1
b	p_5, p_4, p_1
c	p_5, p_1
d	p_2, p_1
e	p_3
f	p_3
g	p_3
h	p_4
i	p_5

Figure 5: Micro set hash table (*msht*) for example of Fig. 4

During the second step, we build the intersections of all pairs of p_is. Thus, assuming again for simplicity that all sets have common size $|p_i|$, we need $O(K^2 * |p_i|)$ time (i.e., the worst case is when all sets are equal), i.e., the same cost as the baseline approach in the worst case. However, in practice, the pairs of sets will not have high overlap; hence, the algorithm is much faster than the baseline approach as we verify experimentally.

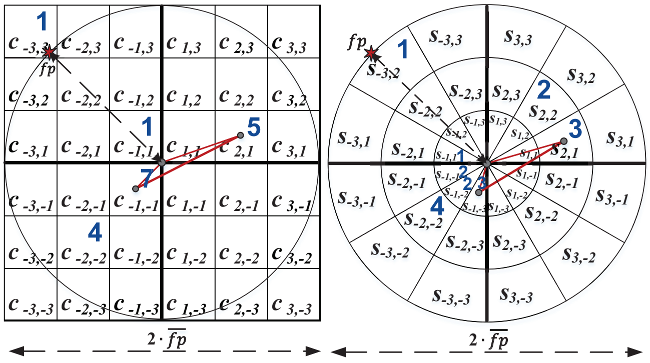
7 SPATIAL PROPORTIONALITY CALCULATION

The computation of $p\mathcal{SS}(.)$ is demanding as we need to compare all $O(K^2)$ pairs in $\mathcal{S}$. Furthermore, computing Ptolemy's $sS(p_i, p_j)$ is expensive. Specifically, for each distance $||p_i, p_j||$ between two places we need 6 operations, i.e. $\sqrt{(p_i.x - p_j.x)^2 + (p_i.y - p_j.y)^2}$. We need three distance computations per pair (i.e. for $||p_i, p_j||$, $||p_i, q||$ and $||p_j, q||$). Finally, we also need 2 more operations, i.e.: (1) the addition of $||p_i, q||$ and $||p_j, q||$ at the denominator and finally (2) the division of the nominator and denominator. Thus in total we need 20 operations for each $dS(p_i, p_j)$. We refer to this brute-force computation approach as the ***baseline* algorithm**. Considering its high cost, we propose Grid based $p\mathcal{SS}(.)$ approaches which reduce the cost by one order of magnitude (at some approximation loss).

7.1 Grid based $p\mathcal{SS}(.)$ calculation

We propose an efficient grid based algorithm for $p\mathcal{SS}(.)$, which accelerates the computation of Ptolemy's similarity $sS(p_i, p_j)$. We investigate its application on two grid structures, i.e. a squared and a radial grid structure. More precisely, we create a regular grid centered on q, which covers the locations of all places in $\mathcal{S}$ and assign each place p_i in $\mathcal{S}$ to the corresponding cell. We approximate $sS(p_i, p_j)$ of any pair of places by replacing their real coordinates with the coordinates of the centres of the respective cells. This approach can decrease drastically the computational cost of $p\mathcal{SS}(p_i)$ at a small compromise on approximation. The grid-based approach also has an important and useful property (which we prove). Namely, the $sS(,)$ of the centres of any two cells is independent from the size of the cells. Thus, we can pre-compute the $sS(,)$ scores for the centers of any pair of cells and use these scores for any query. Recall that $sS(,)$ calculation requires up to 20 operations. Hence, if we use the pre-computed scores, we reduce this cost to 1 operation only. Algorithm 2 illustrates the algorithm with a pseudo code, and Figure 6 illustrates a running example.

7.1.1 Squared grid and algorithm. Hereby, we describe the steps of the algorithm when using a squared grid.



(a) $p\mathcal{SS}$ Squared Grid (b) $p\mathcal{SS}$ Radial Grid

Figure 6: $p\mathcal{SS}$ Grid Examples (annotated with $|c_i|$)

Step 1: Generate the $p\mathcal{SS}(.)$ grid. We define the grid G by a triplet $G(G_c, G_z, |G|)$. The grid is divided into square cells and hence itself is a square. G_c is the center of the grid and it is aligned to the query location q. G_z is the length of each of the grid's sides, which is set to $2\overline{fp}$, where $\overline{fp}$ is the distance between q and the furthest point from q in $\mathcal{S}$ (see the example of Figure 6(a)). $|G|$ is the number of cells in the grid. A larger $|G|$ decreases the approximation error but also increases the cost of $p\mathcal{SS}(p_i)$ computation.

Each grid row or column has $|g|$ cells, where $|g| = \sqrt{|G|}$. Value $|g|$ should be an even number, because the number of cells on the left (bottom) of the grid's centre G_C is equal to the number of cells on the right (top) of G_C, as determined by $\overline{fp}$. Each cell c_i contains a number of places, denoted by $|c_i|$. For each query, a good choice of $|G|$ should be such that $|G| \approx K$, according to our experiments.

Step 2. Allocate places to cells. We allocate each place p to the cell that contains p and maintain a counter $|c_i|$ for the number of places in each cell. For each cell c_i, its centre, denoted as cc_i, represents (i.e., approximates) the locations of all places in c_i.

Step 3. Calculate $p\mathcal{SS}(,)$. Let us assume that $sS(cc_i, cc_j)$ between the centres (cc_i, cc_j) of every pair of cells (c_i, c_j) has been pre-computed and is accesible from a matrix sSM. We calculate the $p\mathcal{SS}(c_i)$ of a cell, by considering the cardinality $|c_i|$ and the cardinality $|c_j|$ of all other cells together with the precomputed $sS(cc_i, cc_j)$ scores, by adapting Equation 6 as follows:

$$p\mathcal{SS}(c_i) = \sum_{c_j \in G} |c_j| \cdot (sS(cc_i, cc_j)) - 1. \tag{18}$$

$p\mathcal{SS}(c_i)$ represents the score for any place p residing in c_i and will be the same for all places in c_i, i.e. $p\mathcal{SS}(p) = p\mathcal{SS}(c_i)$ for each p in c_i. In the computation of $p\mathcal{SS}(c_i)$, we also consider all places in c_i; c_i includes $|c_i|$ collocated places with $sS(p, p_j) = 1$ for all p, p_j in c_i. We subtract 1 in order to disregard the comparison of a place against itself. We consider all cells with $|c_i| > 0$.

Precomputation. The algorithm requires that the $sS(cc_i, cc_j)$ scores between all cell centres are pre-computed for any resolution and position of G. This is possible because of the nature of Ptolemy's similarity, which makes it independent from the scale of distances between points; only their relative orientation to q matters. We prove this property in Theorem 7.1, at the end of this section. Specifically, the $sS(cc_i, cc_j)$ score depends on the relative position of cells c_i and c_j w.r.t. the center of the grid, where this position is measured in terms of number of cells. For example in Figure 6, $sS(cc_{-1,1}, cc_{-1,-1})$ equals to $1 - 1/\sqrt{2}$ and depends only on the relative positions of the cells w.r.t. the grid centre, but not on their sizes. Hence, by pre-computing all scores for a large grid G_{MAX} which can be superimposed on top of any query, we can use the pre-computed values. If the query requires a smaller grid (recall that $|G| \approx K$), where $|G| \le |G_{MAX}|$, then we use only the pre-computed scores of the respective subset of G_{MAX}.

Algorithm 2 Grid based $p\mathcal{SS}()$ calculation

```
Generate empty grid $G(q, 2 \cdot \overline{fp}, |G|)$                {Step 1}
for each $p$ in $\mathcal{S}$ do
    Assign $p$ to the cell $c_i$ which contains $p$                {Step 2}
    $|c_i| = |c_i| + 1$
for each cell $c_i$ with $|c_i| > 0$ do
    for each cell $c_j$ with $|c_j| > 0$ and $j \ge i$ do
        $p\mathcal{SS}(c_i) = p\mathcal{SS}(c_i) + |c_j| \cdot sS(cc_i, cc_j)$       {Step 3}
    $p\mathcal{SS}(c_i) = p\mathcal{SS}(c_i) - 1$
```

Complexity. For Step 1, in order to generate the grid, we need $O(|G|)$ time. During Step 2, we need $O(K)$ operations to assign K places to cells. For Step 3, in order to calculate the $p\mathcal{SS}()$ for a pair of cells, we need two operations (i.e. multiplying $|c_j|$ by $sS(cc_i, cc_j)$). In the worst case, the K places will be in different cells. Thus, for calculating $p\mathcal{SS}(c_i)$, we will need $2 \cdot K$ operations. Hence, for the whole grid with K cells, we will need $O(K^2)$ operations in the worst case. The space complexity is $O(K)$, since $|G| \approx K$, while the storage requirements for pre-computation are $O(|G_{MAX}|)$.

7.1.2 *Radial grid.*

An alternative to the square grid approximation is a radial grid R, which is defined by sectors formed by (1) circles and (2) lines as follows. We use a set of r_c homocentric circles, all centered at the grid center R_c (i.e., the query location q). These circles have as radii multiples of a constant c_z, where the outmost circle has diameter $2 \cdot \overline{fp}$. We also use a set of R_d lines that divides the space into equal slices (any two consecutive lines have a common angle). These lines' lengths are set to the diameter of the outmost circle (Figure 6(b)). The algorithm (i.e., Algorithm 2) remains the same; but here we have a radial grid and sectors (instead of cells). The rationale of using a radial grid is that it has smaller cell sizes near the query location and could give a better approximation when many places are located very close to q. We set $R_d = 2 \cdot r_c$ which results in $|R| = 2 \cdot R_d \cdot r_c$ sectors. Hence, the radial grid can be denoted by $R(R_c, R_z, |R|)$, where (1) R_c is the centre of the grid (q), (2) R_z is the length of the diameter and is set to $2 \cdot \overline{fp}$, and (3) $|R|$ is the number of sectors (cells) in the grid (i.e. R_d^2). Note that $R_z = 2 \cdot r_c \cdot c_z$. Each s_i may contain a number of places, denoted as $|s_i|$. We use the center sc_i of a sector s_i as the representative point, defined by the intersection between a circle having as radius the average radii of the two cirles that define it and the diameter having as angle the average angle of the two diameters that define the sector. Finally, we can easily see that the time and space analysis and Theorem 7.1 (i.e. we can pre-compute and reuse the $sS(,)$ of sectors) apply here as well.

7.1.3 *Scale-free property of Ptolemy's similarity.*

Given a pair of points (p_i, p_j) and a query location q, we now prove that their $sS(p_i, p_j)$ score remains the same if we multiply their difference to q in all dimensions by the same factor f. Formally:

THEOREM 7.1. *Let p_i and p_j be two points with coordinates (x_i, y_i) and (x_j, y_j), respectively. Let q be a query location with coordinates (x_q, y_q). Let p'_i and p'_j be two points with coordinates (x'_i, y'_i) and (x'_j, y'_j), respectively, such that $(x'_i - x_q) = f \cdot (x_i - x_q)$, $(y'_i - y_q) = f \cdot (y_i - y_q)$, $(x'_j - x_q) = f \cdot (x_j - x_q)$, and $(y'_j - y_q) = f \cdot (y_j - y_q)$. It holds that $sS(p_i, p_j) = sS(p'_i, p'_j)$.*

PROOF. We have $sS(p'_i, p'_j) = 1 - \frac{||p'_i,p'_j||}{||p'_i,q||+||p'_j,q||} =$
$1 - \frac{\sqrt{(x'_i-x'_j)^2+(y'_i-y'_j)^2}}{\sqrt{(x'_i-x_q)^2+(y'_i-y_q)^2}+\sqrt{(x'_j-x_q)^2+(y'_j-y_q)^2}}$
We also have $x'_i - x'_j = f \cdot (x_i - x_q) - f \cdot (x_j - x_q) = f \cdot (x_i - x_j)$ and similarly $y'_i - y'_j = f \cdot (y_i - y_j)$, $x'_i - x_q = f \cdot (x_i - x_q)$, $y'_i - y_q = f \cdot (y_i - y_q)$, $x'_j - x_q = f \cdot (x_j - x_q)$, $y'_j - y_q = f \cdot (y_j - y_q)$. Hence, $sS(p'_i, p'_j) =$
$1 - \frac{\sqrt{f \cdot (x_i-x_j)^2+f \cdot (y_i-y_j)^2}}{\sqrt{f \cdot (x_i-x_q)^2+f \cdot (y_i-y_q)^2}+\sqrt{f \cdot (x_j-x_q)^2+f \cdot (y_j-y_q)^2}} =$
$1 - \frac{\sqrt{(x_i-x_j)^2+(y_i-y_j)^2}}{\sqrt{(x_i-x_q)^2+(y_i-y_q)^2}+\sqrt{(x_j-x_q)^2+(y_j-y_q)^2}} = 1 - \frac{||p_i,p_j||}{||p_i,q||+||p_j,q||} =$
$sS(p_i, p_j)$. □

Now, consider a grid G that is centered at q. For every pair of cells c_i, c_j in the grid G, let (cc_i, cc_j) be the corresponding pair of cell centres. Based on Theorem 7.1, score $sS(cc_i, cc_j)$ is independent from the cell size c_z and only depends on the relative positions of c_i, c_j w.r.t. the grid's centre, measured in terms of number of cells. For example, in Figure 6, the grid cells are given identifiers, based on their relative position (in number of cells) to the grid centre. Based on Theorem 7.1, the $sS(cc_{a,b}, cc_{c,d})$ score between any two cell centres $cc_{a,b}$ and $cc_{c,d}$ depends only on the grid-based coordinates (a, b) and (c, d) of cells $c_{a,b}$ and $c_{c,d}$ and not on the sizes of the cells. This is because in two grids G and G', the ratio of the differences between cell centres $cc_{a,b} \in G$ and $cc'_{a,b} \in G'$ and the corresponding grid centres in each dimension is the same for any (a, b). In addition, $sS(cc_{a,b}, cc_{c,d})$ is the same for any position of the grid centre. Summing up, the same pre-computed $sS(cc_{a,b}, cc_{c,d})$ values are used for any query location q and any grid size G_z and number of cells $|G|$.

8 THEORETICAL ANALYSIS

In this section, we analyze the approximation bounds of the greedy algorithms (IAdU and ABP) for our proportional selection problem. Our proofs are based on the assumption that $HPF(u, v)$ satisfies the triangle inequality. For this purpose, we first investigate when does $HPF(u, v)$ satisfy the triangle inequality. Then, by using this key observation, we can trivially prove the approximation loss.

LEMMA 8.1. *Given a set of distance functions $dF_1(u, v), ..., dF_n(u, v)$ that satisfy triangle inequality, then their weighted summation (denoted as $dF(u, v) = \sum w_i \cdot dF_i(u, v)$) also satisfies triangle inequality, as given by*

$$dF(u, v) + dF(v, w) \geq dF(u, w).$$

PROOF. By definition of $dF(u, v)$, the inequality can be rewritten as: $\sum w_i \cdot dF_i(u, v) + \sum w_i \cdot dF_i(u, w) \geq \sum w_i \cdot dF_i(v, w)$. Thus,

$w_1 \cdot dF_1(u, v) + w_1 \cdot dF_1(v, w) \geq w_1 \cdot dF_1(u, w)$,
$\vdots$
$w_n \cdot dF_n(u, v) + w_n \cdot dF_n(v, w) \geq w_n \cdot dF_n(u, w)$.
The addition of these equations completes the proof. □

In general, any diversity function $dF(u, v)$ maintains its triangle inequality properties as long as the constituent components follow triangle inequality. Since from [5], we know that $dS(u, v)$ (i.e. $1 - sS(u, v)$) satisfies the inequality and from [35] we see that $dC(v, w)$ (i.e. $1 - sC(u, v)$) which is a Jaccard distance is a metric and hence satisfies the triangle inequality; then, we can infer that $dF(u, v)$ (i.e. $1 - sF(u, v)$) also satisfies triangle inequality.

THEOREM 8.2. *$HPF(u, v)$ (Eq. 15) satisfies the Triangle Inequality when $rF(v) \geq \frac{\lambda \cdot (k-1)}{(1-\lambda) \cdot (K-k)}$*

PROOF. By expanding $HPF(u, v)$ we get:
$(1 - \lambda) \cdot \frac{K-k}{k-1} \cdot (rF(u) + rF(v)) + \lambda \cdot (\frac{1}{k-1} \cdot (pF\mathcal{S}(u) + pF\mathcal{S}(v)) - 2 \cdot sF(u, v)) + (1 - \lambda) \cdot \frac{K-k}{k-1} \cdot (rF(v) + rF(w)) + \lambda \cdot (\frac{1}{k-1} \cdot (pF\mathcal{S}(v) + pF\mathcal{S}(w)) - 2 \cdot sF(v, w)) \geq (1 - \lambda) \cdot \frac{K-k}{k-1} \cdot (rF(u) + rF(w)) + \lambda \cdot (\frac{1}{k-1} \cdot (pF\mathcal{S}(u) + pF\mathcal{S}(w)) - 2 \cdot sF(u, w)) \implies$
$(1 - \lambda) \cdot \frac{K-k}{k-1} \cdot rF(v) + \lambda \cdot \frac{1}{k-1} \cdot pF\mathcal{S}(v) - \lambda \cdot sF(u, v) - \lambda \cdot sF(v, w) \geq -\lambda \cdot sF(u, w) \implies$
$(1 - \lambda) \cdot \frac{K-k}{k-1} \cdot rF(v) + \lambda \cdot \frac{1}{k-1} \cdot pF\mathcal{S}(v) - \lambda \cdot (sF(u, v) + sF(v, w) - sF(u, w)) \geq 0 \implies$
$(1 - \lambda) \cdot \frac{K-k}{k-1} \cdot rF(v) + \lambda \frac{1}{k-1} \cdot pF\mathcal{S}(v) - \lambda \cdot (1 - dF(u, v) - dF(v, w) + dF(u, w)) \geq 0$.

Considering that $dF(u, v)$ ranges in $[1, 0]$ and satisfies triangle inequality (according to Lemma 8.1), then the minimum value for $dF(u, v) + dF(v, w) - dF(u, w)$ is 0. Then we have:
$(1 - \lambda) \cdot \frac{K-k}{k-1} \cdot rF(v) + \lambda \frac{1}{k-1} \cdot pF\mathcal{S}(v) - \lambda \cdot 1 \geq 0 \implies$
$(1 - \lambda) \cdot (K - k) \cdot rF(v) + \lambda \cdot pF\mathcal{S}(v) \geq \lambda \cdot (k - 1) \implies$
$rF(v) \geq \frac{\lambda \cdot (k-1)}{(1-\lambda) \cdot (K-k)}$. □

For further simplification, we drop $pF\mathcal{S}(v)$ (which is the summation of $K - k$ elements (including $sF(u, v)$ and $sF(v, w)$) and thus should be a significant value.

If we see more carefully this inequality, it holds in most pragmatic cases and our default settings. For $\lambda = 0.5$ and $K = 10 \cdot k = 10k$, then we get: $rF(v) \geq \frac{k-1}{10k-k} \implies rF(v) \geq \frac{k}{9k} \implies rF(v) \geq 1/9$. In summary, we have triangle inequality when $rF(v) \geq 0.1$. This is a pragmatic case as results with smaller $rF(v)$ are not really relevant and they never make it in the $\mathcal{S}$.

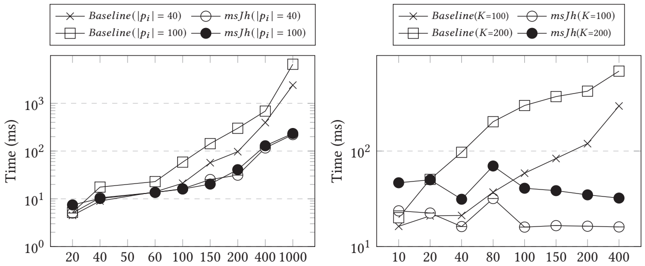
Approximation Bounds. Given $HPF(u, v)$ satisfies triangle inequality, IAdU and ABP algorithms can achieve approximation ratios of 4 and 2 respectively. For such conditions, these bounds are proved by [5] and are based on earlier work in [36] and [26].

9 EXPERIMENTS

In this section, we evaluate the efficiency and approximation quality of the proposed proportionality framework. Finally, we present a user evaluation and testing of our approach.

9.1 Setup

Datasets. We used DBpedia and Yago2 (version 2.5) datasets that have been used in [5, 38]. The DBpedia RDF graph has 8M vertices



(a) Varying K, **DBpedia**

(b) Varying $|p_i|$, **DBpedia**

Figure 7: Efficiency of msJh (Jaccard)

(0.8M of them are places) and 72M edges. Yago2 has 8M vertices (with 4.7M places) and 50M edges. In general, our techniques had similar behaviour on both datasets; due to space limitations, we present all results on DBpedia and skip some results on Yago2 if they are similar. For the performance experiments of our grids, we also used synthetically generated data, which are discussed later.

Queries. We selected locations and keywords, to form a total of 100 queries, such that the number of retrieved places per query is at least 1000. For each place p_i in the query result, we compute its relevance score $rF(p_i)$ to q by combining the Jaccard similarity to the keywords and the normalized distance of p_i to the query location (by considering the largest distance of the city) [2, 5].

Experimental settings. Our algorithms are evaluated by varying parameter values. First, we experimented with different sizes K of $\mathcal{S}$. For a given query, for each value K, we selected from the query results, the K most relevant places to form $\mathcal{S}$ according to $rF(p_i)$. K varies in $\{20, 40, 50, 60, \mathbf{100}, 150, 200, 400, 1000\}$, with 100 being the default value. Second, we experimented with different values of $|p_i|$, i.e. the number of elements in the contextual sets of p_is in $\mathcal{S}$. For a given $\mathcal{S}$, we formed the contextual sets of the places included in it, by using keywords from neighboring vertices to p_i in the corresponding RDF graph, until the desired $|p_i|$ is reached for each $|p_i|$. That is, we enriched (or constrained) the contextual sets of the places on demand by adding (or removing) keywords, in order to satisfy the requirement of the required $|p_i|$ by the experiment. The tested $|p_i|$ values range in $\{20, 40, 50, 60, \mathbf{100}, 150, 200, 400\}$, with 100 being the default value. Third, we experimented with different values of the grid size $|G|$; i.e. values in $\{36, 64, \mathbf{100}, 144, 196\}$ with a default of $|G| = 100$. Fourth, we experimented with values of k in $\{5, \mathbf{10}, 15, 20\}$ with a default value of 10. We experimented with different values of the weights λ and γ, with default $\lambda = \gamma = 0.5$.

Platform. All methods were implemented in Java and evaluated on a 2.7 GHz dual-core, quad-thread machine, with 16 GBytes of memory, running Windows 10.

9.2 Efficiency

In this section, we measured the average run-time costs of the tested algorithms on our queries for the various parameter values.

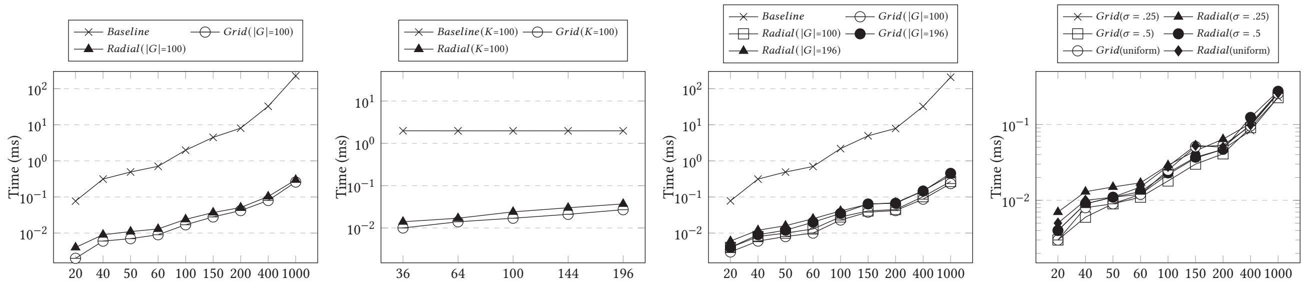
9.2.1 Contextual and Spatial Proportionality Algorithms. We study the efficiency of our solutions for contextual and spatial proportionality computation, presented in Sections 6 and 7.

Contextual Proportionality. Figure 7 compares the performance of our msJh algorithm against the baseline algorithm for calculating $p\mathcal{CS}(p_i)$ for all $p_i \in \mathcal{S}$ (i.e. for all pairs in $\mathcal{S}$). Figure 7(a) reveals that the two algorithms have similar behaviour for $K \leq 40$, but for $K > 40$ our msJh becomes significantly faster. For instance for $K = 1000$ and $|p_i| = 100$, msJh and baseline require 233ms and 6567ms respectively. Similarly, we observe in Figure 7(b) that the two algorithms have similar performance for $|p_i| \leq 20$, but for $|p_i| > 40$ our msJh becomes significantly faster. msJh does not pay off for small values of $|p_i|$ due to the overhead of bookkeeping operations. We also implemented minhash and compared it with msJh, but minhash performed poorly for our settings (minhash outperforms msJh only when K and $|p_i|$ become larger than 1000 and 200, respectively); thus, we do not present further details.

Spatial Proportionality. In Figure 8, we present the performance of our squared and radial grids techniques against the baseline algorithm for calculating the $p\mathcal{SS}(p_i)$ for all $p_i \in \mathcal{S}$ (i.e. for all pairs in $\mathcal{S}$). We see that our algorithms outperform the baseline algorithm by at least one order of magnitude for all settings and datasets. We also observe that the squared grid approach is almost always slightly faster than the radial one. Figure 8(a) shows that the performance gap between the baseline and the grid-based algorithms increases with K. Figure 8(b) shows that the size of the grid $|G|$ marginally affects the time of the grid-based algorithms. We also conducted the same experiments on Yago2 for K and $|G|$ that gave similar results, thus we combine them more synoptically in Figure 8(c). Finally, in Figure 8(d), we tested the efficiency of grid-based proportionality computation on synthetically generated locations of places. For this purpose, we generated $20, ..., 200$ (K) random locations around the query location q to model the retrieved set $\mathcal{S}$, using different spatial distributions: uniform and Gaussian. In the Gaussian distributions each place coordinate was generated having as mean the corresponding coordinate of q and a standard deviation of either 0.25 or 0.5. Note that the baseline approach had much larger cost and was omitted from this sub-figure in order for the difference between the other methods to be easier to see.

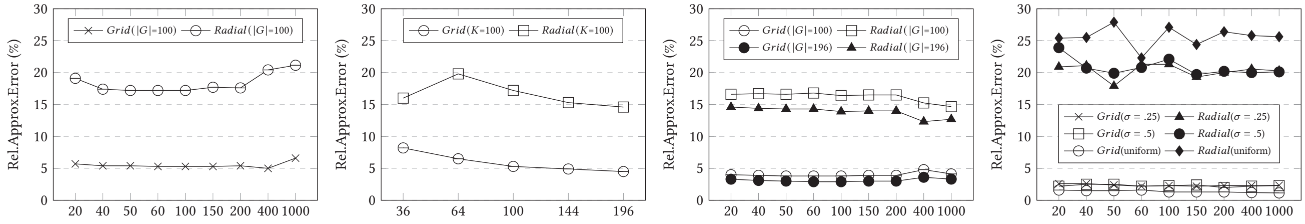
9.2.2 Greedy algorithms. Next, we measure the (average of) the combined costs of the greedy (IAdU and ABP) with the contextual and spatial proportionality algorithms. For the proportionality calculation, we compare our optimised algorithms (i.e. msJh and grid based algorithms, which are the most efficient options) against the respective baselines. Figure 11 shows the results on DBpedia for different values of K and k (the results on Yago2 are similar and they are omitted for brevity). Each bar adds up the total cost of the corresponding combination; the bottom part is the cost of the greedy algorithm, the middle part is the cost of computing spatial proportionality scores and the top part is the cost of computing spatial contextual proportionality scores.

Both optimised and baseline versions of IAdU and ABP compute proportionality scores for all pairs just once in Step 1 and then reuse these scores multiple times in Step 2. Optimised versions are about one order of magnitude faster than baseline versions; the cost difference is insensitive to K and becomes larger for smaller values of k. IAdU and ABP require similar times, e.g. for the default setting ($k = 10$ and $K = 100$), they require 0.24ms. As already discussed, msJh and grid based algorithms are faster than baselines counterparts. Their default setting's times are 6.9ms and 0.1ms, whereas the corresponding baselines take 140ms and 1.08ms respectively. We observe that the greedy algorithms costs are insignificant in



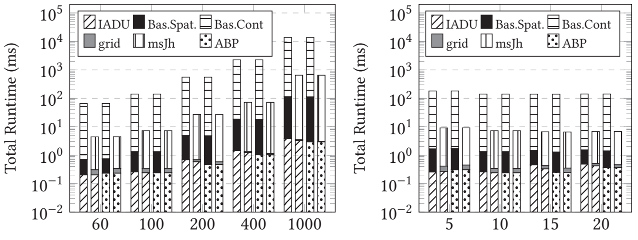
(a) Varying K, **DBpedia** (b) Varying $|G|$, **DBpedia** (c) Varying K, **Yago2** (d) Varying K ($|G| = 100$), **Synthetic**

Figure 8: Efficiency of Squared and Radial Grid algorithms



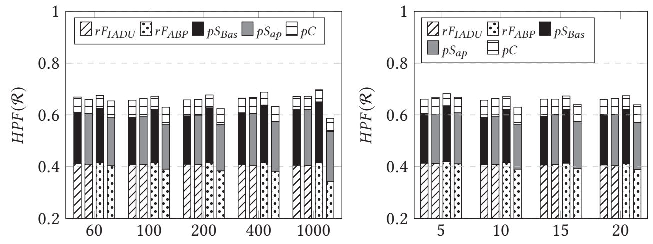
(a) Varying K, **DBpedia** (b) Varying $|G|$, **DBpedia** (c) Varying K , **Yago2** (d) Varying K, **Synthetic**

Figure 9: Effectiveness of Squared and Radial Grid algorithms (Relative Approximation Error)



(a) Varying K (b) Varying k

Figure 10: Efficiency on DBpedia



(a) Varying K (b) Varying k

Figure 11: Approximation Quality on DBpedia

comparison to the proportionality scores. As expected, the weights λ and γ have impact only on the greedy algorithms and thus their impact remains insignificant against the total time (thus we omit further discussion due to lack of space). In summary, either greedy algorithm in combination with msJh and grid based algorithms constitute the fastest approach. The experimental results justify our focus on processing efficiently the contextual and spatial proportionality scores and use them as many times as necessary in the greedy algorithms.

9.3 Approximation Quality

Grid based Algorithms. We compare the approximate $pS\mathcal{S}(p_i)$ scores for the whole $\mathcal{S}$ (i.e. $\sum_{p_i \in \mathcal{S}} pS\mathcal{S}(p_i)$) produced by the two grid approaches against the optimal one (produced by baseline). Figure 9 presents the relative approximation error of the $\sum_{p_i \in \mathcal{S}} pS\mathcal{S}(p_i)$ of the competitive approaches. We observe that the squared grid is always better than the radial grid and that K does not affect this error. We also observe that increasing $|G|$ (i.e., making the grid finer) leads to a reduction of the relative approximation error and that in general a $|G| \approx K$ is a good choice (see Figure 9(b)). We also tried various distributions (Figure 9(d)) that also present similar results. We conclude that the squared grid with $|G| \approx K$ is an appropriate choice with a negligible error of around 5% or lower in practice.

Greedy Algorithms. We assess the approximation quality of the combination of the two greedy algorithms with the approximated grid based and optimal (spatial and contextual) algorithms. Figure 11 shows the $HPF(\mathcal{R})$ scores on DBpedia for these combinations, different values of K and k and default settings (Yago2 results were similar and thus omitted). Each bar adds up the (normalised average) total score of the corresponding weighted combination; the top part represents $\sum_{p_i \in \mathcal{R}} pC(p_i)$ (denoted as pC), the middle part represents $\sum_{p_i \in \mathcal{R}} pS(p_i)$ (i.e. pS_{bas}, pS_{ap}) and the bottom part the relevance $(K - k) \cdot \sum_{p_i \in \mathcal{R}} rF(p_i)$ (i.e. rF_{IAdU}, rF_{ABP}). Recall that we cannot obtain the optimal $HPF(\mathcal{R})$ scores due to the high computational cost required. ABP always achieves (marginally) better $HPF(\mathcal{R})$ score than IAdU which reflects their (comparative) approximation quality. For instance, for the default settings, ABP performs in average 2.36% better $HPF(\mathcal{R})$ score than IAdU (i.e. $67.2 - 65.7\%$). The approximation compromise of the grid based algorithm is minor; for the default settings, the difference on $HPF(\mathcal{R})$ scores with and without using the approximated spatial scores on IAdU and ABP is 0.8% and 6.6% respectively. The λ and γ weights have marginal impact on the relative approximation quality (details are ommitted for the interest of space).

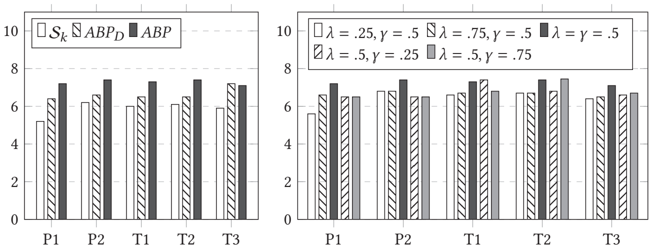
9.4 User evaluation

We also conducted a user evaluation (i.e. user preference and usability testing), which confirms the preference of users to proportional results. We asked help from ten evaluators, who are employees of our institutes (none of them was involved in this paper). First, we familiarized them with the query concepts and relevance metrics. We also explained to them the concepts of proportionality and diversity; to avoid any bias, we avoided to discuss their advantages or disadvantages. Then, we presented to them ten random queries from both data sets and their results according to the three alternative frameworks. Namely, $\mathcal{S}_k$ (i.e. the top-k places in $\mathcal{S}$ with the largest $rF(p_i)$), ABP$_D$ (i.e. diversification results produced by ABP [5], since ABP was shown superior to IAdU) and our proportional ABP. For each task, we asked them to give a score in a scale of one to ten. In order to assist evaluators with their tasks, we also presented a map with the places, their contextual sets and useful statistics (for each query). We presented the output of each method in a random order (to avoid any bias).

9.4.1 User Preference Study. In this study, we asked evaluators to evaluate and express their preference w.r.t. (**P1**) the general content of results (by considering how representative and informative they are) and (**P2**) their ranking. The P1 and P2 bars in Figure 12(a) average the evaluators' preference scores of the three methodologies, for the two criteria (i.e., general content and ranking), for $k = 10$ (using the default settings). For the first criterion (general content), we observe that the users prefer proportional, then diversified and lastly non diversified results. For the second criterion (ranking), users prefer proportional and diversified results. The study revealed that the top places are typically proportional at the same time facilitating both diversity and representation of $\mathcal{S}$; whereas, only some bottom results had some similarity to previous ones. E.g. the top 5 places are proportional and repetitions appear in the bottom 5 places (e.g. additional museums). This type of bird's eye view is preferable by users.

9.4.2 Usability Test. We conducted a comparative study of the usability of the three paradigms. Usability is the ease of use and learnability of a human-made object; namely, how efficient it is to use (for instance, whether it takes less time to accomplish a particular task), how easy it is to learn and whether it is more satisfying to use[3]. We gave them three tasks to complete (for each query and paradigm) and asked them to give a score and also to justify their answers (where possible). Namely, to score them considering (1) the ease of accomplishing each task, (2) how easy and (3) satisfying are to learn and use.

The three tasks were about the understanding and the extraction of information about the queries' results and the entire $\mathcal{S}$. Task 1 (**T1**) "How easily can you infer the area with many collocated places of interest?". For instance in Stockholm, how easily can you infer that Gamla Stan is an area with many collocated museums; so someone can visit this area and can visit more than one museums. Task 2 (**T2**) "How easily can you infer the most representative type of places in the area?"; e.g. an arts or history museum in Stockholm. Task 3 (**T3**) "How easily can you infer at least three different types of places of interest in the area?"; e.g. so someone can choose from all types of museums in Stockholm.



(a) Evaluation Tasks, $k = 10$

(b) Evaluation Tasks on ABP, $k = 10$, varying λ, γ

Figure 12: User Evaluation and Usability Test

The T1–T3 bars in Figure 12(a) average the evaluators' usability scores of the three methods per query and per task. The results show that evaluators preferred firstly proportional, then diversified and lastly non-diversified results for both datasets. The average scores of ABP$_D$, ABP on tasks T1, T2, and T3 are 6.1, 6.7, and 7.5 respectively. The evaluators also provided justifications for their scores. They explained that in general they prefer the concept of proportionality as it also considers frequent properties; which is a property other types do not consider. They found diversification very useful in covering the most diverse places (addressing T2); however, they pointed out that rare but important elements may appear which again can be to some extend misleading. They found the non-diversified results more misleading as very important and relevant places are too dominant in them.

Figure 12(b) depicts the preference of users for the various values of λ and γ for $k = 10$ using the ABP algorithm. Other settings also gave interestingly good results; however, in most cases results from the default setting were more preferable.

10 CONCLUSIONS

In this work, we extend spatial keyword search to support proportional selection of the retrieved places. Our framework combines relevance and proportionality, w.r.t. both context and location. After proving the hardness of the problem, we identify the bottlenecks of proportional selection and propose techniques that greatly reduce its computational cost in practice. We use our methods as modules of two greedy algorithms (IAdU and ABP). Our experiments on real data verify the approximation quality and efficiency of our algorithms and confirm that our framework is preferred by human evaluators. More precisely, either greedy algorithm (IAdU or ABP) in combination with the msJh and squared grid algorithms appears to be the best choice for our paradigm as it is the fastest of all options. In our future work, we will study alternative scoring functions for the contextual and spatial search components (e.g., road network distance in place of Euclidean distance).

ACKNOWLEDGMENTS

Nikos Mamoulis was partially funded by Greek national funds, under the call Research-Create-Innovate (project code: T1EDK-04810).

[3] www.wikipedia.org/wiki/Usability

REFERENCES

[1] Pritom Ahmed, Mahbub Hasan, Abhijith Kashyap, Vagelis Hristidis, and Vassilis J. Tsotras. 2017. Efficient Computation of Top-k Frequent Terms over Spatio-temporal Ranges. In *SIGMOD*. 1227–1241.

[2] Ritesh Ahuja, Nikos Armenatzoglou, Dimitris Papadias, and George J. Fakas. 2015. Geo-Social Keyword Search. In *SSTD*. 431–450.

[3] Avi Arampatzis and Georgios Kalamatianos. 2018. Suggesting Points-of-Interest via Content-Based, Collaborative, and Hybrid Fusion Methods in Mobile Devices. *TOIS* 36, 3 (2018), 23:1–23:28.

[4] Roberto J. Bayardo, Yiming Ma, and Ramakrishnan Srikant. 2007. Scaling up all pairs similarity search. In *WWW*. 131–140.

[5] Zhi Cai, Georgios Kalamatianos, Georgios J. Fakas, Nikos Mamoulis, and Dimitris Papadias. 2020. Diversified spatial keyword search on RDF data. *VLDB J.* 29, 5 (2020), 1171–1189.

[6] Lisi Chen, Shuo Shang, Chengcheng Yang, and Jing Li. 2020. Spatial keyword search: a survey. *GeoInformatica* 24, 1 (2020), 85–106.

[7] Shiwen Cheng, Anastasios Arvanitis, Marek Chrobak, and Vagelis Hristidis. 2014. Multi-Query Diversification in Microblogging Posts. In *EDBT*. 133–144.

[8] Shiwen Cheng, Marek Chrobak, and Vagelis Hristidis. 2016. Slowing the Firehose: Multi-Dimensional Diversity on Social Post Streams. In *EDBT*. 17–28.

[9] Gao Cong, Christian S. Jensen, and Dingming Wu. 2009. Efficient Retrieval of the Top-k Most Relevant Spatial Web Objects. *PVLDB* 2, 1 (2009), 337–348.

[10] Van Dang and W. Bruce Croft. 2012. Diversity by proportionality: an election-based approach to search result diversification. In *SIGIR*. 65–74.

[11] Van Dang and W. Bruce Croft. 2012. Diversity by Proportionality: An Election-based Approach to Search Result Diversification. In *SIGIR*. 65–74.

[12] Elena Demidova, Peter Fankhauser, Xuan Zhou, and Wolfgang Nejdl. 2010. DivQ: diversification for keyword search over structured databases. In *SIGIR*. 331–338.

[13] Georgios J. Fakas. 2008. Automated generation of object summaries from relational databases: A novel keyword searching paradigm. In *ICDE Workshops*. IEEE Computer Society, 564–567.

[14] Georgios J. Fakas. 2011. A Novel Keyword Search Paradigm in Relational Databases: Object Summaries. *DKE* 70, 2 (2011), 208 – 229.

[15] Georgios J. Fakas, Yilun Cai, Zhi Cai, and Nikos Mamoulis. 2018. Thematic ranking of object summaries for keyword search. *DKE* 113 (2018), 1–17.

[16] Georgios J. Fakas and Zhi Cai. 2009. Ranking of Object Summaries. In *ICDE*. 1580–1583.

[17] Georgios J. Fakas, Zhi Cai, and Nikos Mamoulis. 2011. Size-l Object Summaries for Relational Keyword Search. *PVLDB* 5, 3 (2011), 229–240.

[18] Georgios J. Fakas, Zhi Cai, and Nikos Mamoulis. 2014. Versatile Size-l Object Summaries for Relational Keyword Search. *TKDE* 26, 4 (2014), 1026 – 1038.

[19] Georgios J. Fakas, Zhi Cai, and Nikos Mamoulis. 2015. Diverse and Proportional Size-l Object Summaries for Keyword Search.. In *SIGMOD*. 363–375.

[20] Georgios J. Fakas, Zhi Cai, and Nikos Mamoulis. 2016. Diverse and proportional size-l object summaries using pairwise relevance. *VLDB J.* 25, 6 (2016), 791 – 816.

[21] Georgios J. Fakas, Ben Cawley, and Zhi Cai. 2011. Automated Generation of Personal Data Reports from Relational Databases. *JIKM* 10, 2 (2011), 193–208.

[22] Marios Hadjieleftheriou, George Kollios, Vassilis J. Tsotras, and Dimitrios Gunopulos. 2006. Indexing spatiotemporal archives. *VLDB J.* 15, 2 (2006), 143–164.

[23] Jungkyu Han and Hayato Yamana. 2017. Geographical Diversification in POI Recommendation: Toward Improved Coverage on Interested Areas. In *RecSys*. ACM, 224–228.

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

[25] Mahbub Hasan, Abhijith Kashyap, Vagelis Hristidis, and Vassilis J. Tsotras. 2014. User effort minimization through adaptive diversification. In *KDD*. 203–212.

[26] Refael Hassin, Shlomi Rubinstein, and Arie Tamir. 1997. Approximation algorithms for maximum dispersion. *Operations Research Letters* 21, 3 (1997).

[27] Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. 2013. YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. *Artif. Intell.* 194 (2013), 28–61.

[28] Vagelis Hristidis, Luis Gravano, and Yannis Papakonstantinou. 2003. Efficient IR-Style Keyword Search over Relational Databases. In *VLDB*. 850–861.

[29] Vagelis Hristidis and Yannis Papakonstantinou. 2002. Discover: Keyword Search in Relational Databases. In *VLDB*. 670–681.

[30] Anoop Jain, Parag Sarda, and Jayant R Haritsa. 2004. Providing diversity in k-nearest neighbor query results. In *PAKDD*. 404–413.

[31] Mehdi Kargar, Aijun An, and Xiaohui Yu. 2014. Efficient Duplication Free and Minimal Keyword Search in Graphs. *TKDE* 26, 7 (2014), 1657 – 1669.

[32] George Kollios, Vassilis J. Tsotras, and Dimitrios Gunopulos. 2017. Mobile Object Indexing. In *Encyclopedia of GIS*. 1256–1266.

[33] Wangchao Le, Feifei Li, Anastasios Kementsietsidis, and Songyun Duan. 2014. Scalable Keyword Search on Large RDF Data. *TKDE* 26, 11 (2014), 2774–2788.

[34] Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. 2014. *Mining of Massive Datasets* (2 ed.). Cambridge University Press.

[35] Michael Levandowsky and David Winter. 1971. Distance between sets. *Nature* 234, 5323 (1971), 34–35.

[36] S. S. Ravi, D. J. Rosenkrantz, and G. K. Tayi. 1991. *Facility dispersion problems: Heuristics and special cases.* 431–450.

[37] Dimitris Sacharidis, Paras Mehta, Dimitrios Skoutas, Kostas Patroumpas, and Agnès Voisard. 2018. Selecting representative and diverse spatio-textual posts over sliding windows. In *SSDBM*. 17:1–17:12.

[38] Jieming Shi, Dingming Wu, and Nikos Mamoulis. 2016. Top-k Relevant Semantic Place Retrieval on Spatial RDF Data. In *SIGMOD*. 1977–1990.

[39] A. B. Siddique, Ahmed Eldawy, and Vagelis Hristidis. 2019. Euler++: Improved Selectivity Estimation for Rectangular Spatial Records. In *BigData*. 4129–4133.

[40] Souvik Brata Sinha, Xinge Lu, and Dimitri Theodoratos. 2018. Personalized Keyword Search on Large RDF Graphs based on Pattern Graph Similarity. In *IDEAS*. 12–21.

[41] Kostas Stefanidis, Marina Drosou, and Evaggelia Pitoura. 2010. PerK: personalized keyword search in relational databases through preferences. In *EDBT*. 585–596.

[42] Julia Stoyanovich, Ke Yang, and H. V. Jagadish. 2018. Online Set Selection with Fairness and Diversity Constraints. In *EDBT*. 241–252.

[43] Jiayu Tang and Mark Sanderson. 2010. Evaluation and User Preference Study on Spatial Diversity. In *ECIR*, Vol. 5993. 179–190.

[44] Marc Van Kreveld, Iris Reinbacher, Avi Arampatzis, and Roelof Van Zwol. 2005. Multi-dimensional scattered ranking methods for geographic information retrieval. *GeoInformatica* 9, 1 (2005), 61–84.

[45] Marcos R. Vieira, Humberto L. Razente, Maria C. N. Barioni, Marios Hadjieleftheriou, Divesh Srivastava, Caetano Traina, and Vassilis J. Tsotras. 2011. On Query Result Diversification. In *ICDE*. 1163–1174.

[46] Lin Wu, Yang Wang, John Shepherd, and Xiang Zhao. 2013. An Optimization Method for Proportionally Diversifying Search Results. In *PAKDD (1)*, Vol. 7818. 390–401.

[47] Xiaolu Xing, Chaofeng Sha, and Junyu Niu. 2017. Improving Topic Diversity in Recommendation Lists: Marginally or Proportionally?. In *APWeb/WAIM (2)*, Vol. 10367. 142–150.

[48] Chengyuan Zhang, Ying Zhang, Wenjie Zhang, Xuemin Lin, Muhammad Aamir Cheema, and Xiaoyang Wang. 2014. Diversified Spatial Keyword Search On Road Networks. In *EDBT*. 367–378.