Fine-Grained Diversification of Proximity Constrained Queries on Road Networks

Xu Teng
Iowa State University
xuteng@iastate.edu

Jingchao Yang
George Mason University
jyang43@gmu.edu

Joon-Seok Kim
George Mason University
jkim258@gmu.edu

Goce Trajcevski
Iowa State University
gocet25@iastate.edu

Andreas Züfle
George Mason University
azufle@gmu.edu

Mario A. Nascimento
University of Alberta
mario.nascimento@ualberta.ca

ABSTRACT

Proximity-oriented spatial queries, such as range queries and k-nearest neighbors (kNNs), are common in many applications, notably in Location Based Services (LBS). However, in many settings, users may also desire that the returned proximal objects exhibit (likely) maximal and fine-grained semantic diversity. For instance, nearby restaurants with different menu items are more interesting than close ones offering similar menus. Towards that goal, we propose a topic modeling approach based on the *Latent Dirichlet Allocation*, a generative statistical model, to effectively model and exploit a fine-grained notion of diversity, namely based on sets of keywords (e.g., menu items) instead of a coarser user-given category (e.g., a restaurant's cuisine). In addition, and relying on the notion of *Distance Signatures*, we propose an index structure that can be used to effectively extract the k objects that are within a range distance from a given query location, and which are also semantically diverse. Our experimental evaluations using real datasets demonstrate that the proposed methodology is able to provide highly diversified answers to cardinality-wise constrained range queries much more efficiently than a straightforward alternative solution.

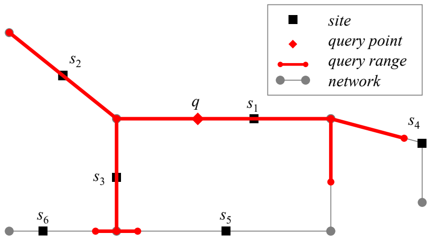
CCS CONCEPTS

• **Information systems** → **Spatial-temporal systems**; **Location based services**.

ACM Reference Format:
Xu Teng, Jingchao Yang, Joon-Seok Kim, Goce Trajcevski, Andreas Züfle, and Mario A. Nascimento. 2019. Fine-Grained Diversification of Proximity Constrained Queries on Road Networks. In *16th International Symposium on Spatial and Temporal Databases (SSTD '19), August 19–21, 2019, Vienna, Austria.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3340964.3340970

1 INTRODUCTION

Range and *k-Nearest Neighbor (kNN)* queries are among the most popular categories of queries in many applications relying on Location-Based Services (LBS) [19]. These spatial queries are particularly useful when users seek Points of Interest (PoIs) in their vicinity [12], e.g.: closest restaurants or friends. While those types of queries have been subject to extensive research for over a decade [15, 20], the main motivation for our work is that in many practical scenarios, in addition to the proximity, the users may be interested in the *semantic diversity* in terms of the various descriptors of nearby entities. For example, in a geo-social network setting, a user may want to spend time with groups of nearby friends with varying interests. Similarly, as shown in Fig. 1 (to be used as our running example), a user located in q may be interested not only in the the restaurants within a given bounded distance, but also in experiencing a wider variety of menu items.



Restaurant	Cuisine	Menu Items
s_1	Japanese	{Sushi, Sashimi, Sake}
s_2	Japanese	{Udon, Tempura, Sake}
s_3	Seafood	{Fish, Salmon, Wine}
s_4	Asian	{Dumplings, Sushi, Beer}
s_5	Italian	{Pizza, Pasta, Wine}
s_6	German	{Schnitzel, Pasta, Beer}

Figure 1: Searching restaurants by cuisines and menu items.

Motivated by this, we introduce a novel type of a query called *k-Diversified Range Query* (kDRQ), which aims at *maximizing the semantic diversity of the answer set of spatial queries within a bounded range.* Although we focus on LBS-applications in our discussion and examples, kDRQs are useful in many other settings in which coupling the notion of semantic diversity with spatio-temporal attributes (e.g., [9]) is meaningful. Existing works have tackled problems requiring simultaneous consideration of spatial and non-spatial properties of data objects. For example, queries pertaining to similarity of spatio-textually enriched trajectories (i.e., semantic/activity trajectories), e.g., [23, 24], take PoIs and textual tags into account. Similarly, [22] presents solutions that aim at diversifying the answer-set in terms of kNN on road networks. However, in broad terms, spatio-textual kNN query returns the set of k nearest locations containing a certain keyword, say "restaurant", not considering the details of the restaurants in the respective locations. Thus, keywords are merely used for an additional filtering/selection.

SSTD '19, August 19–21, 2019, Vienna, Austria

ACM ISBN 978-1-4503-6280-1/19/08…$15.00
https://doi.org/10.1145/3340964.3340970

One way to improve the query diversification is to use a fine(r) classification of the PoIs. For example, instead of referring to a PoI only as a "restaurant," one can describe its category – e.g., "Chinese", "German", "Vegetarian", etc [6]. While this helps in the sense that a diversified query would now tend to return restaurants of different categories, it may be difficult to define the classes and to manually label them, for several reasons: (1) some PoIs may not belong to only one clearly defined category; (2) restaurants of the same category may have sufficiently different menus to be considered "diverse"; (3) often, a restaurant may provide its menu, but not a particular type (as it is the case with many restaurants on Yelp).

We postulate that in order to better capture the diversity of PoIs, these should be described by *sets* of attributes, e.g., "keywords" such as menu items in the case of restaurants, and the *latent* topics defined by those sets. Consider the information in Fig. 1 and assume that a user requests $k = 2$ diversified restaurants within a given range which includes s_1, s_2 and s_3. If only the cuisine type is used to qualify the restaurants, it is clear that adding s_2 to an answer set that already has s_1 (or vice-versa) will not improve the overall diversity of the answer set. Thus those two restaurants will likely never appear together in any answer set, whereas the pairs $\{s_1, s_3\}$ and $\{s_2, s_3\}$ are equally diverse. Now, if one considers the restaurants' menu items instead of only their cuisine, then the diversity in $\{s_1, s_3\}$ is higher than in $\{s_1, s_2\}$ as there are no overlapping keywords in the former set. In this context, we propose the use of Latent Dirichlet Allocation (LDA) [3] to extract latent topics for each PoI and to also annotate the sites network accordingly. As we shall see, LDA yields meaningful and intuitive topics for result diversification.

Unfortunately it turns out that maximizing set-based diversity is not an easy problem. In fact, as we shall see later in this paper, considering pair-wise diversity is an NP-hard problem ([10, 22]). In order to mitigate that, we rely on the notion of LDA in order to obtain a more informed (i.e., LDA-annotated) network, and we propose a new indexing structure inspired by the concept of distance signatures [11]. Each node of the annotated network stores approximate distance information of other nodes in a bounded spatial neighborhood and also diversity information.

In summary, the main contributions of this work are:

- In Section 3 we formalize a novel type of query, named *k*DRQ, which returns the k most semantically diverse locations that are within a given range distance from a query point on road networks.
- We discuss in Section 4 the use of LDA to extract latent topics for PoIs. We then present our LDA-based topic diversity and provide a case study to show that it yields meaningful and intuitive topics for result diversification.
- In Section 5 we present the details of our proposed solution – index structures and processing algorithms – for calculating the answer-set to *k*DRQ.
- In Section 6, we report our detailed experimental evaluations using real datasets, demonstrating the benefits of the proposed approaches.

We complement our study with an an overview of the related literature in Section 2, and we conclude this work in Section 7.

2 RELATED WORK

The concept of incorporating diversity into similarity search has its origins in information retrieval. The Maximal Marginal Relevance (*MMR*) model [5] is one of the earliest proposals to consider diversity to re-rank documents in the answer set, where at each step, the element with higher marginal relevance is selected. A document has high marginal relevance if it is both relevant to the query and has minimal similarity to previously selected documents.

Several approaches have been proposed for coupling spatial and diversity settings. Finding the *k*NNs to a given query point q such that the distance between any two points is greater than a predefined minimum diversity is addressed in [13], and selecting the most diverse set within a predefined radius in Hamming space is addressed in [1]. A k-similar diversification set which optimizes a linear functions combining the similarity (i.e., closeness) and diversity for a given trade-off between them is studied in [21]. Monitoring the most diverse k-sized set over distributed sets is addressed in [2]. The main difference to these previous works is in the definition of diversity. These existing works aim at maximizing the pair-wise diversity of categories of points. In our approach, we do not assume that we have categorization of sites, nor do we assume that know the pair-wise similarities between these categories. Instead, our approach learns and models the topics of the data using textual descriptions, to maximize the topic diversity of whole result sets, rather than considering only pairs of points.

Our goal is to provide the user with a solution that offers a different kind of trade-off between spatial proximity and diversity – namely, topic-based instead of category-based diversity.

Angular diversity has been explored in [17] via Nearest Surrounder Query, which finds the nearest objects from a query point from different angles, and the angular similarity has been used for diversified *k*NN problem in [16].

Relying on the Skyline paradigm [4], finding the set of all optimal solutions for a given linear combination of two diversity notions, spatial and categorical, is presented in [6]. The categorical diversity is modeled by the difference between categories of data points – e.g., two restaurants are diverse if they are from different ethnicities. The idea of using keywords, i.e., a finer granularity in order to distinguish categories, to find diverse *k*NNs has been explored in [22]. In that work the keywords are used for filtering data points, i.e., only points that contain *all* query keywords are considered. We, on the other hand, use the concept of Latent Dirichlet Allocation in order to consider a more sophisticated notion of diversity based on the set of keywords that describe each object. Moreover, differently from the works above, we propose an indexing structure to speedup the processing of *k*DRQs.

3 PRELIMINARIES

Definition 3.1 (Site Database). Let $\mathcal{I} = \{i_1, ..., i_{|\mathcal{I}|}\}$ be a set of $|\mathcal{I}|$ items (such as terms or keywords). A *site*, s, is a pair (L, I), where L is a spatial location, and $I \subseteq \mathcal{I}$. A site database, $\mathcal{DB} = \{s_1, ..., s_{|\mathcal{DB}|}\}$, is a collection of sites.

For instance, depending on the application, sites may correspond to restaurants or individuals, in which cases items could correspond to menu entries in restaurants or personal skills, respectively.

Definition 3.2 (Site Network). Let $\mathcal{DB}$ be a collection of sites. A *site network* is a directed graph $\mathcal{G} = (V, E, W, S)$, where V is a set of sites, each consisting of a pair $(v.L, v.I)$; $E \subseteq v.L \times v.L$ is a set of edges between location-attributes ($v.L$) of the vertices; $W : E \mapsto \mathbb{R}^+$ is a function that maps each edge to a positive value representing the cost of traversing the edge, and $S : \mathcal{DB} \mapsto V$ is a function that maps a site $s \in \mathcal{DB}$, to a vertex in $\mathcal{G}$.

For the sake of simplicity and ease of exposition we assume that the site network $\mathcal{G}$, is properly embedded in a (potentially larger) road network G. Note that this allows for the query point Q to be a vertex in G that does not belong to the site network $\mathcal{G}$ proper.

Definition 3.3 (Network Range Query). Let $\mathcal{DB}$ be a collection of sites, $\mathcal{G} = (V, E, W, S)$ be a site network, Q be a location on an edge from E, and ϵ be a positive real value. A network range query $RQ(\mathcal{DB}, \mathcal{G}, Q, \epsilon)$ returns all the sites in $\mathcal{DB}$ whose locations have their shortest distance to Q no greater than ϵ, that is: $RQ(\mathcal{DB}, \mathcal{G}, Q, \epsilon) = \{s \in \mathcal{DB} \mid dist(Q, s.L) \leq \epsilon\}$, where $dist(Q, s.L)$ is the shortest network distance from Q to s based on $\mathcal{G}.W$.

A network range query allows us to find all sites within a given range from a query location. In this work, our goal is to efficiently reduce this set to a subset with of sites with cardinality $\leq k$, while providing a maximum diversity.

Definition 3.4 (k-Diverse Subset). Let $D \subseteq \mathcal{DB}$ be a set of sites, and $div : D \mapsto \mathbb{R}^+$ be a function that maps such set to a positive value (diversity score). The k-diverse subset of $\mathcal{DB}$, $kDS_{div}(\mathcal{DB}, k)$, is defined as the subset of $\mathcal{DB}$ with cardinality at most k, maximizing the diversity score, i.e., $kDS_{div}(\mathcal{DB}, k) = \arg\max_{D \subseteq \mathcal{DB}, |D| \leq k} div(D)$.

Based on Definition 3.3 and Definition 3.4, we can finally define a k-diversified range query as follows:

Definition 3.5. Let $\mathcal{DB}$ be a collection of sites, $\mathcal{G} = (V, E, W, S)$ be a site network, let Q be a vertex in the embedding road network G and let ϵ be a positive real value. Further, let $div : D \mapsto \mathbb{R}^+$ be a function that maps a set of sites onto a positive diversity score and let k be a positive integer. The k-diversified range query $kDRQ(\mathcal{DB}, \mathcal{G}, Q, \epsilon, k)$ is defined as:

$$kDRQ(\mathcal{DB}, \mathcal{G}, Q, \epsilon, k) = kDS_{div}(RQ(\mathcal{DB}, \mathcal{G}, Q, \epsilon), k)$$

In a nutshell, a k-diversified range query returns the k-most diverse subset from among all the sites that are within distance at most ϵ from Q. The choice of diversity measure div is an essential aspect left abstract above. Next, we discuss two choices for this function. A straightforward manner to compute the diversity of sets of items is to simply count their number of unique items.

Definition 3.6 (Set-Union-Based Diversity). Let $D = \{s_1, ..., s_{|D|}\}$ be a set of $|D|$ sites. Then we define set-union-based diversity as $SUBD(D) = |\bigcup_{s \in D} s.I|$, i.e., the number of unique items in D.

Going back to the example shown in Fig. 1, the set-union-based diversity of the set of restaurants $D = \{s_1, s_2, s_3\}$ is $|\bigcup_{s \in D} s.I| = |\{\text{Sushi, Sashimi, Sake, Udon, Tempura, Fish, Salmon, Wine}\}| = 8$.

The most diverse set of size k among $\mathcal{DB}$ using set-union-based diversity is therefore given by $\arg\max_{D \subseteq \mathcal{DB}, |D| \leq k} SUBD(D)$

As simple as this definition appears, finding an optimal k-subset from a set of candidate sites that maximizes set-union-based diversity is NP-hard. It is an instance of the optimization problem of the *set cover problem*, which is at least as hard as the decision problem (deciding if there exists any k-subset) which is one of Karp's 21 NP-complete problems [8, 14].

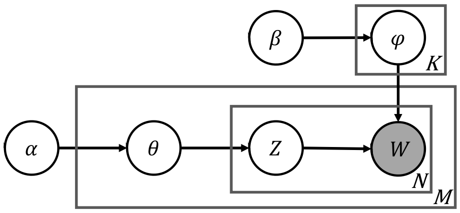


Figure 2: Graphical Model in *plate notation* of LDA-based topic modeling. Boxes represent entities (M sites, N keywords within a site, K latent topics). Nodes correspond to random variables, shaded nodes are observable random variables, and arrows indicate stochastic dependencies.

The second measure (initially proposed in [6]) calculates the diversity of a set by the minimum pair-wise diversity of its elements. This definition has the advantage that it can be used for any type of sites for which pair-wise diversity is defined. In our case, where a site is represented by a set of items, we can use Jaccard similarity index as a measure of diversity.

Definition 3.7 (Pairwise Diversity). Let $D = \{s_1, ..., s_{|D|}\}$ be a set of $|D| > 1$ sites. Then we define pairwise diversity $PD(D)$ as $PD(D) = \min_{s_i, s_j \in D, s_i \neq s_j}(1 - J(s_i, s_j))$, where $J(s_i, s_j) = \frac{|s_i \cap s_j|}{|s_i \cup s_j|}$, i.e., the Jaccard index between two sets.

For example, to get the pairwise diversity of subset $D = \{s_1, s_2, s_3\}$, shown in Fig. 1, this algorithm would compute the Jaccard index for all three pairs from D, $J(s_1, s_2) = \frac{1}{5}$, $J(s_1, s_3) = \frac{1}{5}$ and $J(s_2, s_3) = 0$, and then obtain the pairwise diversity of D as $PD(D) = \frac{4}{5}$ (yielded equally by $\{s_1, s_2\}$ and $\{s_1, s_3\}$).

The most diverse set of size k among $\mathcal{DB}$ using pairwise diversity is thus the set $\arg\max_{D \subseteq \mathcal{DB}, |D| \leq k} PD(D)$

Despite only considering pairwise diversities, the selection of a k-subset that maximizes the pairwise diversity is an NP-hard problem. Even if we could guess the value of the maximum pairwise diversity x, finding a set of k sites that all have a pairwise diversity of x or greater is an instance of the clique problem, another one of Karp's 21 NP-complete problems [14]. A detailed proof for the NP-hardness of maximizing pairwise diversity can be found in [10].

4 TOPIC-BASED DIVERSITY

To reduce the potentially large and redundant space of items, we next propose to model the latent topics of items at each site. For that, we employ Latent Dirichlet Allocation (LDA) [3] – a generative probabilistic model which assumes that each site is a mixture of underlying (latent) topics, and each topic has a (latent) distribution of more and less likely keywords; and we present its use on an empirical case study.

4.1 LDA Based Diversity

A graphical representation of our LDA model is shown in Fig. 2[1]. A vector α of length K is used to parameterize the *a priori* distribution of topics. The parameter K corresponds to the number of latent topics used to model our sites. When a site is created, we assume

[1] Source: https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation

that its topics are chosen following a *Dirichlet distribution* having parameter α which we use to obtain a topic distribution θ for each of our M sites. Thus, the large plate in Fig. 2 corresponds to a set of M sites, each having a topic distribution θ drawn randomly (and Dirichlet distributed) from α.

For each topic, the prior parameter β is used to generate the distribution of words within a topic. Thus, we assume that a topic generates words following a Dirichlet distribution having a vector β of length $|\mathcal{I}|$ as parameter, where (cf. Definition 3.1) $|\mathcal{I}|$ is the number of words we consider in our dictionary. For each of our K topics, a resulting vector φ stores, for each word $i_j \in \mathcal{I}$, the probability of i_j appearing in this topic. Therefore, the smallest plate in Fig. 2 denotes a set of K vectors φ_i of length $|\mathcal{I}|$, mapping each keyword to the probability of appearing within topic i.

To generate the words within a site s_i, a topic is chosen randomly from the topic distribution θ and, given this topic, a number of N_i words are generated randomly from the word distribution φ – where N_i is assumed to be independent from the chosen topic and uniformly distributed. This results, for each site, in a set of N_i words (z, w), where w is a word, and z is the topic of w. In Fig. 2, the node W denotes the set of all $N = \sum_i N_i$ words, and Z is a function that maps each word to the topic that generated it. Node W is shaded, as it is the only variables that can be observed, while all other variables are latent. The reason for choosing a Dirichlet distribution rather than a more straightforward multinomial distribution for the topic and word priors is inspired by research showing that the distribution of words can be better approximated using a Dirichlet distribution [18].

To infer the topics of our site database, we employ a generative process for obtaining labels for the site. Given the observed keywords of sites in our database, LDA optimizes the latent variables so as to maximize the likelihood matching our observed sites and their keywords. This generative process works as follows. Sites are represented as random mixtures over latent topics, where each topic is characterized by a distribution over all $\mathcal{W}$ keywords from a chosen dictionary of most common keywords. LDA assumes the following generative process for database $\mathcal{DB}$ consisting of M sites, each having a number of N_i keywords.

- For each site s_i choose a topic distribution $\theta_i \sim Dir(\alpha)$, where $1 \leq i \leq |\mathcal{DB}|$, and $Dir(\alpha)$ is a Dirichlet distribution with prior α. In our experiments, we initially assume each topic to have uniform prior probabilities, having $\alpha_i = \alpha_j$ for $1 \leq i, j \leq K$. This apriori distribution is adapted using Bayesian inference [3] to maximize the likelihood of generating the observed keywords.
- For each topic, choose $\varphi_i \sim Dir(\beta)$, where $1 \leq i \leq K$. For our experiments, we assume each word to have the same low prior probability, having $\beta_i = 0.001$ for $1 \leq i \leq \mathcal{W}$. These low prior probabilities are desirable for fast convergence, as most keywords are very rare.
- For each word w in site j:
  1. Choose a topic $z \sim Multinomial(\theta_j)$ from the topic distribution of j, and
  2. Choose a word $w \sim Multinomial(\varphi_z)$ from the word distribution φ_z of topic z.

  Here, $Multinomial(x)$ corresponds to a multinomial distribution drawing from a stochastic vector x.

Next, we propose how to define the diversity of a set of sites based on their coverage of latent topics. The idea is to describe a set of sites by the expected number of distinct topics by this set.

Definition 4.1 (Topic-Based Diversity). Let $D = \{s_1, ..., s_{|D|}\}$ be a set of sites and let θ_i denote the latent topic distribution of site s_i and consequently, let $\theta_{i,j}$ denotes the probability of site s_i to belong to topic j ($1 \leq j \leq K$). Topic-based diversity $TBD(D)$ is defined as the *expected* number of unique topics among sites in D. Formally:

$$TBD(D) = \sum_{j=1}^{K} 1 - \prod_{s_i \in D} (1 - \theta_{i,j}). \tag{1}$$

The idea of Equation 1 is to compute, for each topic j, the probability that at least one site in D covers topic j. This probability is equal to the counter-probability of having no site in D cover topic j which is computed as: $P(\text{no topic } j \text{ in } D) := \prod_{s_i \in D}(1 - \theta_{i,j})$. Thus, $1 - P(\text{no topic } j \text{ in } D)$ is the probability of having topic j appear at least once (i.e., topic j being covered), and $TBD(D)$ is the expectation of the number of topics covered.

Example 4.2. Consider the set of $D = \{s_1, s_3, s_5\}$ sites shown in Fig. 1, and assume that LDA returns the following distributions among $K = 3$ latent topics: $\theta_1 = (0.8, 0.1, 0.1)$, $\theta_3 = (0.6, 0.1, 0.3)$, and $\theta_5 = (0.0, 1.0, 0.0)$. This means that site s_1 has a high chance (80%) to belong to Topic 1 (which may correspond to a latent topic "Japanese Food"), s_3 is also likely to belong to the same topic, but also has a higher chance to belong to Topic 3, whereas s_5 is certain to belong to Topic 2 (which could correspond to "Italian Food").

To compute the topic-based diversity of D, we employ Equation 1. For the first topic, we obtain $1 - \prod_{s_i \in D}(1 - \theta_{i,1}) = 1 - (1 - 0.8) * (1 - 0.6) * (1 - 0) = 0.92$. Thus, we have a 92% likelihood that at least one of the three sites has Topic 1. For Topic 2, we see that s_5 is guaranteed to have this topic. Consequently we get $1 - \prod_{s_i \in S}(1 - \theta_{i,2}) = 1 - (1 - 0.1) * (1 - 0.1) * (1 - 1) = 1$. For Topic 3, we obtain a probability of $1 - (1 - 0.1) * (1 - 0.3) * (1 - 0) = 0.37$ of being covered. Summation of these three values yields $TBD(D) = 0.92 + 1.00 + 0.37 = 2.29$. Thus, we expect 2.29 out of these three topics to be covered.

Intuitively, our notion of topic-based diversity provides a more practical definition of diversity that assigns a lower diversity to (different) keywords having a high probability to belong to the same topic. Yet, in terms of computational complexity, the problem of maximizing topic-based diversity remains NP-hard, as shown in the following.

Lemma 4.3. *Given a set of $|D|$ sites. The problem* TOPIC-kDIV *of finding the k-Diverse Subset (c.f. Definition 3.4) using topic-based diversity is NP-hard.*

Proof. Let K be the number of latent topics used in Definition 4.1 and assume a special case where each site covers exactly m ($m < k$) topics with uniform probability. Further, assume that $K > mk$. In this case, topic-based diversity is achieved by selecting sites that maximize the number of topics covered with non-zero probability. This problem of finding a set of sites, each having m topics, that maximizes the cover of topics, is an instance of MAX-COVER, another one of Karp's 21 NP-complete problems [14]. Since the constructed case, which is NP-complete, is a special case of *TOPIC-kDIV*, we conclude that *TOPIC-kDIV* is NP-hard. □

Table 1: Top-10 most probably keywords for $K = 10$ latent topics (from Yelp, with Natural Language Tookit).

Topic	Keywords (Probabilities in %)
1	'chicken'(9.2), 'curri'(8.9), 'indian'(6.4), 'masala'(2.7), 'spice'(1.9), 'lamb'(1.9), 'biryani'(1.7), 'tandoori'(1.7), 'rice'(1.7), 'lentil'(1.5)
2	'chicken'(6.5), 'enchilada'(5.5), 'mexican'(5.0), 'taco'(4.0), 'bean'(2.8), 'salsa'(2.6), 'black'(1.8), 'soup'(1.7), 'casserol'(1.6), 'chipotl'(1.5)
3	'chines'(9.2), 'chicken'(7.7), 'fri'(4.3), 'pork'(3.5), 'rice'(2.7), 'noodl'(2.6), 'beef'(2.4), 'stir'(2.4), 'soup'(2.3), 'sauc'(1.6)
4	'thai'(19.8), 'chicken'(7.9), 'curri'(6.8), 'soup'(3.8), 'coconut'(3.5), 'salad'(3.0), 'shrimp'(2.7), 'noodl'(2.6), 'green'(2.0), 'sauc'(1.8)
5	'chicken'(5.4), 'chocol'(3.7), 'cooki'(3.6), 'butter'(3.3), 'peanut'(3.0), 'bake'(2.5), 'chees'(1.9), 'burger'(1.9), 'chip'(1.6), 'casserol'(1.5)
6	'french'(6.9), 'soup'(4.4), 'onion'(3.9), 'chocol'(2.6), 'creme'(2.5), 'chicken'(2.1),'bread'(2.0), 'sauc'(1.5), 'clafouti'(1.5), 'toast'(1.5)
7	'grill'(6.6), 'chicken'(5.4), 'shrimp'(2.6), 'fri'(1.7), 'steak'(1.6), 'southern'(1.6), 'cajun'(1.5), 'grit'(1.5),'pork'(1.4), 'sauc'(1.4)
8	'italian'(6.7), 'lasagna'(3.3), 'pasta'(2.6), 'chicken'(2.5), 'sauc'(2.3), 'tomato'(2.2), 'pizza'(2.1), 'spaghetti'(1.9),'sausag'(1.9), 'soup'(1.8)
9	'miso'(5.8), 'japanes'(4.7), 'teriyaki'(2.7), 'bowl'(2.6), 'salmon'(2.6), 'glaze'(2.4), 'scallop'(1.2), 'eggplant'(1.0), 'crispi'(0.8), 'appet'(0.7)
10	'salad'(4.5), 'potato'(2.0), 'roast'(2.0), 'chicken'(1.9), 'sauc'(1.5), 'bean'(1.4), 'grill'(1.3), 'green'(1.2), 'cake'(1.1), 'pie'(1.1)

Table 2: Most diverse set of sites with $k = 3$.

Measure	Site Information
LDA-based	Oregano's Pizza Bistro: (Italian, Restaurants, Pizza) Wienerschnitzel: (Sandwiches, Fast Food, Food, Hot Dogs, Ice Cream & Frozen Yogurt, Restaurants, Desserts) Umami: (Restaurants, Asian Fusion, Japanese, Soup, Ramen)
Set-Union-Based	Genghis Grill: (Restaurants, Chinese, Vegan, Buffets, Mongolian, Vegetarian, Thai, Korean, Asian Fusion) Noodles & Company: (Specialty Food, Food, Chinese, Noodles, Soup, Asian Fusion, Italian, Salad, Comfort Food, Restaurants, Japanese, Sandwiches, Fast Food, Pasta Shop) Pier 54: (Mediterranean, Lounges, Beer, Wine & Spirits, American (New), Breakfast & Brunch, Italian, Food, Restaurants, Nightlife, Arts & Entertainment, Music Venues, Bars, Cocktail Bars, Wine Bars, Burgers)
Pair-wise-Based	Final Round Sports Bar & Grill: (Sports Bars, Restaurants, Pizza, Bars, American (Traditional), Nightlife) Sweet Dessert Cafe: (Creperies, Cafes, Restaurants, Coffee & Tea, Breakfast & Brunch, Sandwiches, Desserts, Food) McDonald's: (Fast Food, Restaurants, Burgers, Food)

To provide an intuition of our algorithms to efficiently find a set of sites that have a high topic-based diversity score, reconsider the example above. If we were to add an additional site to this set, what type of site be most beneficial to increase the diversity? Clearly, Topic 2 is already fully covered, such that adding more sites that have a high probability of having Topic 2 is futile. At the same time, adding more of Topic 1 has low *utility*, as this topic is already covered with a probability of 92%. However, adding another site having a high probability of having Topic 3 would boost the topic-based diversity score close to 3.0 in this example.

Before we show how this observation can be exploited into a locally optimizing heuristic to efficiently find a set of high topic-based diversity, we will first show a qualitative evaluation that shows that our latent topics are indeed able to describe real-world data in a meaningful way, using cooking recipes and restaurant menus as a sample (yet representative) scenario.

4.2 Case Study: LDA for Restaurant Sites

We now present an empirical evaluation of our LDA-based diversity measure. The semantically useful and humanly intuitive outcomes are shown using a dataset crawled from Yummly, a recipe recommendation website. We extracted the $K = 10$ latent topics from a set of $M = 27,638$ recipes, considering the set of $V = 1,000$ most frequent keywords. Table 1 shows the result of LDA to model this dataset. For each i-th topic, this table shows the vector φ_i, which corresponds to a multinomial distribution over all words in our dictionary. Thus, for each topic i, the probabilities of a word w correspond to the probability that word w will be generated by topic i. For each topic, the ten largest probability values are shown in this table (although many more words may have a non-zero probability to be produced by this topic).

Intuitively, we see that the topics found by LDA make sense. We see that the ten topics corresponds to Indian, Mexican, Chinese, Thai, American, French, Cajun, Italian, Japanese and Healthy cuisines. LDA understands that some keywords appear in most topics at different frequencies. For example, the term "Chicken" appears as one of the Top-10 keywords in all topics except Japanese. We also see that keywords such as "Curry" (tokenized to "Curri" in Table 1) appears with high probability in Topic 2 (Indian) and Topic 5 (Thai). We also see that words such as "Thai", "Lasagna" and "Miso" are very discriminative, appearing with very high probability in one topic only.

To show that our LDA-based approach to define diversity (Def. 4.1) yields intuitive sets of diverse sites, we compare our approach to traditional diversity measures in Table 2 by performing trained LDA model on test dataset from Yelp.

For each of the three diversity measures proposed in Section 3, this table shows the set of $k = 3$ most diverse sites among 150 randomly selected candidate sites. To find this set, we employ the *Swap Algorithm* that has been proposed in [21] to heuristically find *k subsets* having high diversity. This algorithm is shown in Algorithm 1.

We see that the Set-Union-Based approach (Def. 3.6) yields semantically non-diverse results. The problem of this approach, which maximizes the *number* of unique keywords among the result sites, is that sites having a large number of keywords get an unfair advantage. In addition, the semantic of words is not considered, as all words are treated as equally in-equal categories.

Algorithm 1: The Swap Algorithm [21]

```
Input: Set of sites $D$, Integer $k$
ResultSet ← ∅
foreach $s \in D$ do
    if |ResultSet| < $k$ then
        ResultSet ← ResultSet ∪ $s$
    else
        C ← ResultSet ∪ $s$
        worstSite ← $\arg\max_{s' \in C} Div(C \setminus s')$
        ResultSet ← $C \setminus$ worstSite
return ResultSet
```

In contrast, the pair-wise diversity approach (c.f. Definition 3.7), which maximizes the pairwise diversity between sites, suffers from similar problems, as many pairs of sites have a Jaccard similarity of zero (no overlapping keywords). But without the ability to find the semantic topics that connect keywords, this approach also falls into the trap of returning sites that have different, but semantically similar keywords.

To summarize, we observe that our LDA-based definition of diversity is capable to understand which keywords correspond to the same topic, thus maximizing the semantic overlap between returned sites. We also note the advantage of an LDA-based approach to probabilistically map between topics and keywords. This is an example, for example, comparing to a purely ontology-based approach, where each keyword would belong to exactly one topic. Such an ontology approach would be forced to map common terms such as "chicken" deterministcally to one topic.

Finally, we note that the *Swap* algorithm (Algorithm 1) only yields a heuristic approximation of the optimal set that optimizes topic-based diversity. We emphasize that the methodologies that we elaborate upon in the remainder of this work do not propose any new heuristics to select k diverse sites from a set of n candidates. For solutions to this problem we refer the interested reader to algorithms surveyed and proposed by Vieira et. al. [21]. Instead, our goal is to efficiently find high-diversity candidates by traversing the spatial network in an effective manner. Thus, we want to quickly lead our algorithm to sites that are likely to contribute to the final result without having to explore the entire collection of sites along the network within the query range.

5 PROCESSING KDRQ QUERIES

We now present our query processing approach, starting with the novel index structure, followed by the algorithmic processing.

5.1 DivMap: A Topic-Based Diversity Maximizing Index Structure

To efficiently support k-diversified range queries on spatial networks, we propose a specialized index structure, which is inspired by the concept of distance signatures [11].

5.1.1 General Idea. At each node of the network, a distance signature stores approximate distance information of nodes in the spatial neighborhood. In detail, for different distance ranges (such as $[0m, 500m]$,$[500m, 1000m]$, ...), the set of other nodes having a shortest-path distance within each range is memorized. For example, one node v may contain the information that the distance to node u is bounded by 1500 – 2000 units. Having such information at each node in the network, allows to find a shortest paths between v and u more efficiently than a blind search such as Dijkstra's algorithm or an A^* search would provide.

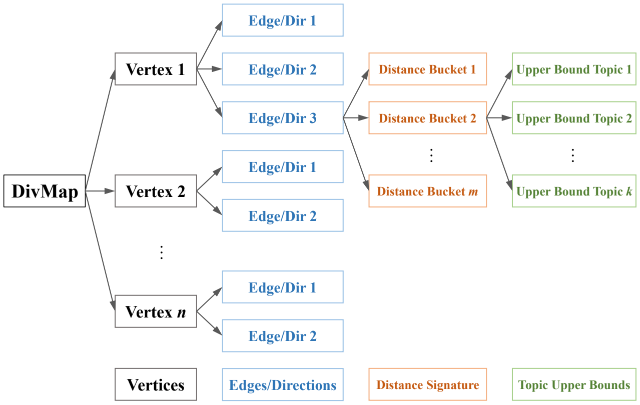


Figure 3: Schematic Sketch of our Diversity Map Index.

In our setting, we are not interested in finding shortest paths, but we are interested in finding high-diversity sites within a given range to return to our user. Thus, we propose to store at each node and for each adjacent edge, an approximation of *topics* that can be found by following the corresponding direction. Since our intent is to maximize diversity, it suffices to store upper-bounds to topic diversity for each direction and each distance range.

An overview of our proposed Diversity Map Index (*DivMap*) is depicted in Fig. 3. At each node v of the network, the algorithm places a virtual "signpost". Pointing in each direction that can be taken from v, this signpost gives a distance-approximated summary of the topics can be found by following this direction. In particular, this information includes the maximum topic values of sites found in each direction, and for each distance bucket defined by the distance signature. The following describes our algorithm to build this index structure.

5.1.2 Index Construction. Our index construction algorithm shown in Algorithm 2 requires a site network $\mathcal{G}$ as defined in Definition 3.2 and a boundary set B to discretize the distance space, for example, $B = \{[0m, 500m], [500m, 1000m], \ldots\}$. The algorithm iteratively processes each vertex and adjacent node independently, which allows for great parallelizability. For each vertex v and adjacent edge e, a breadth-first search is used to find build the virtual signpost. For this purpose, Line 4 removes other adjacent edges of v to ensure that only paths crossing e are explored. Forcing the algorithm to use edge e, we invoke Dijkstra's single-source shortest-path algorithm [7] in Line 5, to explore all sites reachable by using edge e from vertex v. Whenever Dijkstra's algorithm completes a node having a site s, we checks if this site is useful to change the virtual signpost of node v. For this purpose, we use the shortest path distance between Q and s (returned from Dijkstra's algorithm), to find distance bucket the current site falls into to in Lines 7-8. The $index\_of$ function of Line 8 simply returns the index of the bucket that contains distance d. For example, if the distance signature

Algorithm 2: *DivMap* Index Construction

```
   Input: Site network $\mathcal{G} = (V, E, W, S)$, boundary set $B$
1  foreach node $v \in V$ do
2      foreach edge $e$ adjacent to $v$ do
3          $DivMap_e \leftarrow$ matrix($|B|, K$)
                                          //All zeroes $|B| \times K$ matrix
4          Remove all edges adjacent to $v$ other than $e$
5          Invoke Dijkstra algorithm starting at $v$
6          foreach site $s$ found do
7              $d \leftarrow dist(v, s)$
                          //shortest path distance between $v$ and $s$
8              $index \leftarrow index\_of(d, B)$
                                       //Distance "Bucket" $s$ falls into
9              for topic in 1:K do
10                 $DivMap_e[index, topic] \leftarrow$
                     $max(DivMap_e[index, topic], \theta_{s,topic})$
11             Restore all edges adjacent to $v$
12 return DivMap
```

buckets are $[0, 500m]$, $[500m, 1000m]$, $[1000m, 1500m]$, ..., then this function would return index 3 for $d = 1200m$. Next, we process the topic vector θ of site s. Our goal is to see if there is any topic i such that s has a higher topic value $\theta_{s,i}$ than all the sites seen in the same distance bucket so far. This check is performed in Line 10, the heart of the algorithm. Here, the maximum diversity values of current signpost in the corresponding distance signature bucket are increased, if $\theta_{s,i}$ is larger than the currently largest value.

Finally, the edges that were ignored in this round are restored in Line 11. Once all nodes and adjacent nodes are processed, the complete index is returned.

5.2 Efficient *k*DRQ Processing

This section describes how our index structure proposed in Section 5.1 can be used to efficiently answer k-diversified range queries.

The general idea of this algorithm is as follows. Instead of using a naïve Dijkstra search to find all the sites inside the query range, we exploit the *DivMap* index to greedily direct the search to sites that locally complement the diversity of the k-most diverse sites that we have already found so far. For example, if the most diverse set of k sites found so far is completely covering Topic 1 and 2, but is only partially covering Topic 3, and is not covering Topic 4 at all, then the algorithm will will be guided to take directions that are extremely likely to lead to sites having Topic 4, and somewhat likely to lead to sites having Topic 3. However, our algorithm has to find a balance between two aspect to optimize: distance and diversity. Following a purely "distance-first" approach, the algorithm would degenerate to Dijkstra algorithm. The advantage is that this algorithm may quickly find some sites, which may already yield a high diversity value "by chance". Following a purely "diversity-first" approach, our algorithm would head straight for the best sites (in terms of diversity given the current set of sites seen so far), while ignoring sites that are close to already explored parts of the network, and which could be added to the result at little cost.

Algorithm 3: *DivMap* Based Diverse Range Query

```
Input: Site network $\mathcal{G} = (V, E, W, S)$, Query $Q \in V$, integer $k$,
       range $\epsilon$, Diversity Index $DivMap$
initialization: D $\leftarrow \emptyset$, MaxHeap $H \leftarrow \{((Q, Q), \infty)\}$,
  $distList \leftarrow \{(Q, 0)\}$
                      // $H$ is a heap sorted by Utility (Definition 5.1)
while $H \neq \emptyset$ do
    $e \leftarrow H$.extractMax            //Remove (pop) the max-Utility
    $e^{tail}.label \leftarrow$ "Green"  //Mark tail node of $e$ visited
    D $\leftarrow Swap(\mathrm{D} \cup S(e^{tail}), k)$
                 //Swap sites $S(e^{tail})$ at $e^{tail}$ into D (See Alg. 1)
    if D has changed then
        Update utility $u_{(\mathrm{D}, \epsilon - distList[e_i^{tail}], DivMap)}(e_i)$ for
          each $e_i$ in H
    foreach edge $e_{adj}$ adjacent to $e^{tail}$ do
        if $e_{adj}^{tail}.label =$ "Green" then
            continue
        else if $e_{adj}^{tail}$ in distList then
            if $distList[e_{adj}^{tail}] \leq distList[e^{tail}] + w(e_{adj})$
              then
                continue
            else
                $H.remove(e_{drop})$ if $e_{drop}^{tail} = e_{adj}^{tail}$ for $e_{drop}$ in
                  $H$
        $distList[e_{adj}^{tail}] \leftarrow distList[e^{tail}] + w(e_{adj})$
        H.insert($e_{adj}, u_{(\mathrm{D}, \epsilon - distList[e_{adj}^{tail}], DivMap)}(e_{adj})$)
return D
```

Intuitively, a balanced algorithm should prefer directions that lead to interesting sites (in terms of diversity given the current set of sites) which are not too far away. Our *DivMap* index proposed in Section 5.1 allows to do that: Not only does it tell the algorithm with direction to follow to find interesting sites, but it also gives the algorithm an approximation of how far it will have to travel to find these sites.

Similar to Dijkstra's algorithm, we maintain a priority queue of "active" edges adjacent to vertices that have already been explored and processed. In each iteration, we greedily select an active edge which maximizes the utility given the currently best k-set of sites. Whenever a site s is found, we use a greedy *swap* algorithm [21], to see if swapping s with any of the up to k currently selected sites improves the topic-based diversity (c.f. Definition 4.1). Note that this swap operation requires $O(k^2)$-time, as k topic diversity values need to be computed, each for a set of no more than k sites.

Starting at the query point Q, all adjacent unvisited edges are stored in a candidate list, sorted in ascending order by their utility. To balance between spatial proximity to unvisited direction and diversity, we define a distance-weighted utility $u_{D,\epsilon,DivMap}(e = (v^{head}, v^{tail}))$, where v^{head} and v^{tail} are the head and tail end of e. This function estimates the utility of v exploring an direction through e as the expected gain of diversity, given the current set of selected sites D, the range query parameter ϵ, and exploiting information stored in our index *DivMap*, formally:

Definition 5.1 (Utility). Let $\mathcal{G}$ be a site network and let $DivMap$ be a diversity map index as described in Section 5.1.2. Further, let D be the current set of at most k sites selected as result, v be a visited node and v' be one of the unvisited adjacent nodes to v through e. $\lambda \in [0, 1]$ and ϵ be real values. The utility $u_{(D,\epsilon,DivMap)}(e)$ of exploring vertex v via edge e is defined as:

$$u_{(D,\epsilon,DivMap)}(e) = \tag{2}$$

$$\sum_{sig=1}^{m} \lambda^{sig-1} \sum_{j=1}^{K} DivMap_e([sig, j]) \cdot \prod_{s_i \in D} (1 - \theta_{i,j})$$

where m is the index of distance buckets that ϵ falls into

In a nutshell, Equation 2 uses, for each topic j, the probability $\prod_{s_i \in D}(1 - \theta_{i,j})$ that topic j is not covered by the current site set D. The remaining utility of topic j is multiplied with the upper bound value of topic j in the first bucket of distance signature bucket $DivMap_e([1, j])$, i.e., the maximum value that we can reach for topic j in the first bucket. This procedure is repeated for each distance signature bucket, but the utility of each bucket after the first is penalized by a cumulative factor of λ. For example, for $\lambda = 0.5$, the diversity of the fourth bucket will be reduced by a factor of $0.5^3 = 0.125$.

The factor λ allows to select the trade-off between distance-greed and diversity-greed. In the extreme case where $\lambda = 1$, all buckets will be weighted equally, allowing the algorithm to chase a site in the outer buckets before finding any other sites. On the contrary, the other extreme of $\lambda = 0$ will completely ignore any bucket beyond the first, thus allowing the algorithm only to consider sites within the first bucket. Depending on the size of the buckets in the distance signature, this setting will force the algorithm to explore parts of the network close to Q first. The choice of λ is not trivial, and as our experimental evaluation in Section 6 shows, a good trade-off requires $0 < \lambda < 1$.

Once a node is visited (initially, the query node Q), all adjacent edges are added to the priority queue if the distance between Q and tail of each edge does not exceed the range query parameter ϵ.

A formal algorithm for this index-supported search is found in Algorithm 3. This algorithm maintains a max-heap that stores all active network edges sorted by their utility. In each iteration, the edge e having the highest utility is processed as follows: The tail node of e, denoted as e^{tail}, is marked as visited (Line 4). Then, all sites located at e^{tail} are processed using the Swap algorithm (c.f. Algorithm 1) in Line 5. If any site at e^{tail} is added to the current result set D in this way, all utility values have to be recomputed in Line 7, as the utility (c.f. Definition 5.1) depends on the current set of sites. Computing the utility of a edge requires $O(k \cdot K \cdot m)$, where k is the number of results, K is the number of topics, and m is the number of distance signature buckets. Doing this for each of the C currently active candidates nodes yields a total time complexity of $O(C \cdot log(C) \cdot k \cdot K \cdot m)$, as the insertion into a heap of size C requires $O(log(C))$ time.

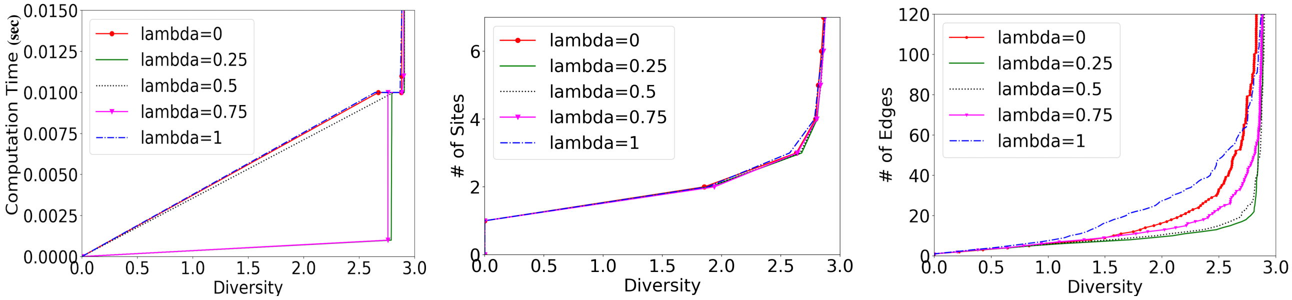
Once e^{tail} has been processed, all neighbors edges, denoted as e_{adj}, of e^{tail} are handled in three cases. In Case I (Lines 9 - 10), the adjacent edge whose tail node have already been visited are ignored, as site at this node have already been processed. Case II considers vertices e_{adj}^{tail} which have not been processed, but which are already in the candidate heap H. For these scenario, we check the distance between Q and e_{adj}^{tail} (Lines 11-15). Note that according to Definition 2, higher ϵ means higher utility. Since our algorithm does not necessarily process vertices by their distance (like Dijkstra algorithm would) and may visit a node before it's shortest path has been found, this step ensures that we always store the shortest distance between Q and each visited node in $distList$, which as well guarantees that the highest utility of each visited node is kept in H. Thus, if previous path to e_{adj}^{tail} is shorter than current found path, then we can simply skip e_{adj}(Line 13). Otherwise, we need to remove all the previous path to e_{adj}^{tail}(Line 15). And the algorithm in Lines 16 - 17 will be executed to store/update the distance information of e_{adj}^{tail} in $distList$, and insert e_{adj} and corresponding utility into H if e_{adj}^{tail} has never been met or already met but not processed yet while keeping the shortest path to Q, which we treat as Case III.

The algorithm terminates when H is empty, or when an maximum budget of iterations (not denoted in Algorithm 3 for brevity) is reached. The later termination criterion is to ensure that high-utility results can be returned to the user, without the requiring all sites to be explored first.

6 EXPERIMENTS

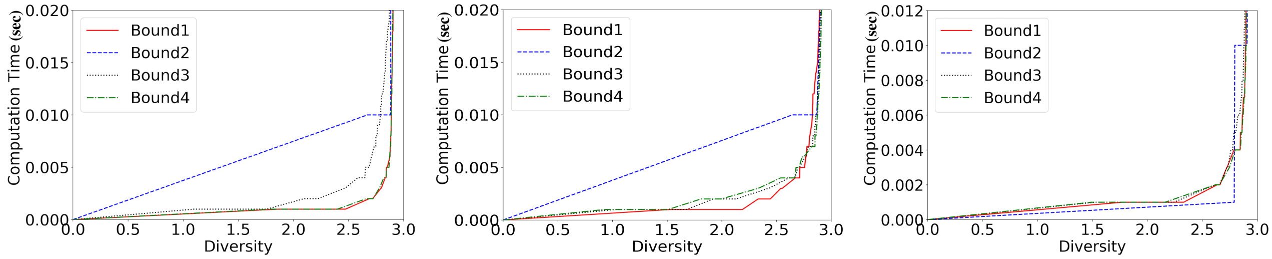
The datasets used for constructing site network consist of two main components: (1) the walkway network from part of Arizona, U.S., obtained from OpenStreetMap, is used as road network; it includes 18,773 nodes and 48,548 edges. (2) to construct the site database, sites are obtained from Yelp in the same area, containing both spatial locations and an average of 4.067 keywords textual information for 882 restaurants in this region. The experiments are conducted on a PC with Intel(R) Xeon(R) CPU E3-1240 v6 @3.70GHz, 32 GB RAM and 512 GB disk storage. Windows 10 Enterprise 64-bit is the operating system, and all algorithms are implemented by Python 2.7. The source code can be found at https://github.com/XTRunner/KDRQ_2019. The distance range, ϵ, is by default set to $2000m$ for all the following experiments, assuming which is the maximum distance a user would walk. For each experiment, 200 vertices are randomly selected from the site network as query points and the average results are presented.

Evaluation of Parameter λ: In order to maximize the efficiency of the *DivMap* index, the parameter λ, which balances between "spatial greed" and "diversity greed", must be chosen wisely. Recall (cf. Alg. 3 and Def. 2) that we choose the direction with highest utility and λ controls the weight of each bucket while calculating it. The boundary set is set to be $[0, 500m]$, $[500m, 1000m]$, $[1000m, 1500m]$ and $[1500m, 2000m]$ for this part. Our experimental results, shown in Fig. 4, evaluates the impacts of λ in terms of three different measurements – time efficiency (computation time to approach certain diversity), site efficiency (visited sites), and edge efficiency (visited edge). Specifically, Figure 4(a) shows that the algorithms run consistently fast (less than 0.2 seconds run-time) for all settings of λ. In addition to run-times, we also measure platform independent



(a) Time efficiency (b) Site efficiency (c) Edge efficiency

Figure 4: Efficiency Evaluation for different values of λ



(a) $\lambda = 0$ (b) $\lambda = 1$ (c) $\lambda = 0.25$

Figure 5: Effect of different boundary sets

measure to understand how different values of λ explore the network differently. In Figure 4(b), we see that, for all λ values, the gain in diversity per site is approximately the same. This may seem counter-intuitive at first, as $\lambda = 1$ may walk far away to visit the (diversity-wise) best sites first, while $\lambda = 1$ has to first explore the sites in the first distance signature bucket. Yet, in order to achieve diversity close to the maximum of 3 (for $k = 3$), the sites in the first bucket seem to be sufficiently diverse. Furthermore, an important observation of Figure 4(c) is that extreme λ values, i.e., $\lambda = 0$ or $\lambda = 1$, yield worse results in terms of edge efficiency. Thus, a pure "spatial first" or "diversity first" approach is not recommended, either exploring the network too locally, or chasing too far away sites. In our dataset, it appears that $\lambda = 0.25$ provides a good trade-off, but this choice depends on characteristics of the dataset, such as the density of sites, the number of latent features, and the number of distance buckets. We note that finding heuristics to quickly estimate this hyper-parameter for a new dataset is part of our future work. The following evaluation experiments utilize the extreme cases (i.e., $\lambda = 0$ and 1) to compare with the result of the trade-off scenario (i.e., $\lambda = 0.25$).

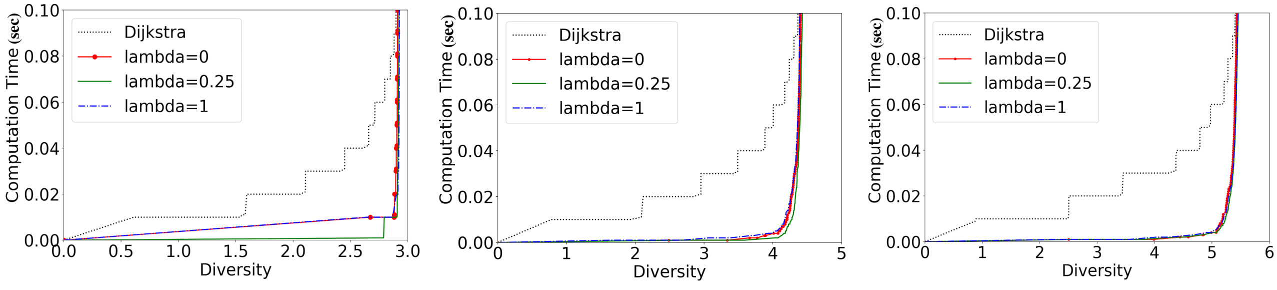
Evaluation of the boundary sets B: Besides of λ, boundary set (parameter B in Algorithm 2) is as well an influential factor of efficiency. In our experiments, four different boundary sets are considered:

- Boundary 1: [0-1000, 1000-2000]
- Boundary 2: [0-500, 500-1000, 1000-1500, 1500-2000]
- Boundary 3: [0-250, 250-500, 500-750, ..., 1750-2000]
- Boundary 4: [0-1000, 1000-1414, 1414-1732, 1732-2000]

Boundaries 1, 2 and 3 have equally sized distance intervals but with different number of buckets, while Boundary 4 has the length of the intervals implying an equal area allocated for each bucket, e.g., the disk with radius 1000m and the annulus between it and the concentric disk with radius 1414m have equal areas. Moreover, without loss of generality, besides of two extreme λ values (i.e., $\lambda = 0$ and $\lambda = 1$), we also present the result when $\lambda = 0.25$ in Fig. 5(c) which is shown to the best λ option for our dataset from above.

We can observe from Fig. 5(a) ($\lambda = 0$) that the result using Boundary 4 is identical to using Boundary 1, since only the first bucket influences the result, allowing the algorithm to quickly find high-diversity results. The diversity increases slowly for Boundary 2, but eventually catch up with Boundary 1 and 4. Thus, it appears that Boundary 2 groups sites inefficiently into distance signature buckets for this study region. Using Boundary 3 (the black dot line), we obtain significantly less diversity in general. The reason is that in the case of $\lambda = 0$, the algorithm is forced to stay very close to explored areas, being constrained in its freedom to chase high-diversity sites, and thus becoming more of a distance-first algorithm.

For $\lambda = 1$ (i.e., all buckets within given distance range are considered equally at each step), shown in the right of Fig. 5(b), bucket range influences the results in a different way. First, we see that the differences between Boundary 3 and Boundary 1 and 4 are smaller, comparing to when $\lambda = 0$. This is because the case of $\lambda = 1$ ignores the distance of buckets, going purely "diversity-first". Thus, having more buckets simply gives this algorithm more details where to find the currently highest utility sites. The reason for the separation of Boundary 1 and 4 is that the outer rings are now further away, and the $\lambda = 1$ algorithm chases sites in these rings oblivious of the distance requires to get there, thus incurring a potentially large detour. Having these additional details allows it to chase the best sites, thus finding higher diversity sites, but taking more time (number of edges explored) to find them. Finally, the case $\lambda = 0.25$ (Fig. 5(c))



(a) $k = 3$

(b) $k = 5$

(c) $k = 7$

Figure 6: ***DivMap*** **index evaluation**

is considered as a trade-off decision between distance-first ($\lambda = 0$) and diversity-first ($\lambda = 1$), showing much different trends for all four boundary sets. The smaller Boundary 2 surpassed Boundary 1 and 4, allowing the algorithm to more quickly detect high-diversity results. The result for Boundary 3, compares to the previous, starting to merge into the trend of Boundary 1 and 4. We conclude that a balanced value $0 < \lambda < 1$ is preferable, to avoid inefficiencies by either ignoring spatial distance or diversity.

Evaluation of parameter k. In the next experiment, we evaluate the parameter k, while also comparing our approach to distance-breath first Dijkstra search through the network. The resulting diversities for k values of 3, 5 and 7 are very similar to each other, as shown in Fig. 6 – demonstrating that (the performance of) *DivMap*-based kDRQ dominates Dijkstra algorithm in terms of diversity and computation time. We observe that $\lambda = 0.25$ offers slightly better result when $k = 3$, and an additional observation from Fig. 6 is that $\lambda = 0$ initially has a better performance, but is eventually surpassed by $\lambda = 1$. This experiment supports the intuition that our *DivMap* index is able to obtain high-diversity much quicker especially when $\lambda = 0.25$ (in terms of computation time) than a traditional Dijkstra search which ignores topic information. We also conclude that the overall run-times using different settings for parameter λ do not drastically affect run-times, but as we see in Figure 4(c), the network space explored is tremendously different, suggesting that a balanced value of λ, which neglects neither the spatial nor the diversity dimension, should be preferred.

7 CONCLUSIONS AND FUTURE WORK

We introduced the kDRQ – a novel query aiming at determining the set of k objects on a road network with highest diversity in terms of the set of their descriptive keywords, and are within a given distance from the user's location. We proposed an efficient processing approach for kDRQ, relying on a novel index structure and an LDA-based heuristic for diversity measure. Our experiments demonstrated that the proposed approach can adjust to find the right balance between diversity-first and distance-first to determine high-diversity results.

As part of our future work, we plan to investigate: (a) heuristics to automatically choose the search parameter λ sensitive to both the terms dataset and the area around the query point, (b) extend kDRQ to incorporate *continuous* variants that consider the changes of the answer-set both due to objects (i.e., users) motion, as well as due to changes in the description items (e.g., a restaurant updates part of its menu at certain hour) and (c) the broader impact of the properties of the diversification function [10] on the quality of the results of the heuristics.

ACKNOWLEDGEMENTS

This research has been supported by NSF grants CCF 1637541, III-1823279, CNS-1823267, ONR grant N00014-14-1-0215, and by NSERC Canada.

REFERENCES

[1] S. Abbar et al. Diverse near neighbor problem. In *SOCG*, pages 207–214, 2013.
[2] D. Amagata and T. Hara. Diversified set monitoring over distributed data streams. In *DEBS*, pages 1–12. ACM, 2016.
[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *JMLR*, 3(Jan):993–1022, 2003.
[4] S. Börzsönyi et al. The skyline operator. In *ICDE*, pages 421–430, 2001.
[5] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*, pages 335–336, 1998.
[6] C. F. Costa and M. A. Nascimento. Towards spatially-and category-wise k-diverse nearest neighbors queries. In *SSTD*, pages 163–181. Springer, 2017.
[7] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.
[8] M. R. Garey and D. S. Johnson. *Computers and intractability*, volume 29. wh freeman New York, 2002.
[9] J. Gittelsohn and S. Sharma. Physical, consumer, and social aspects of measuring the food environment among diverse low-income populations. *American Journal of Preventive Medicine*, 36(4), 2009.
[10] S. Gollapudi and A. Sharma. An axiomatic framework for result diversification. *IEEE Data Eng. Bull.*, 32(4):7–14, 2009.
[11] H. Hu, D. L. Lee, and V. Lee. Distance indexing on road networks. In *VLDB*, pages 894–905. VLDB Endowment, 2006.
[12] S. Ilarri, A. Illarramendi, E. Mena, and A. P. Sheth. Semantics in location-based services. *IEEE Internet Computing*, 15(6):10–14, 2011.
[13] A. Jain, P. Sarda, and J. R. Haritsa. Providing diversity in k-nearest neighbor query results. In *PAKDD*, pages 404–413, 2004.
[14] R. M. Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.
[15] M. Koubarakis et al., editors. *Spatio-Temporal Databases: The CHOROCHRONOS Approach*, volume 2520 of *LNCS*. Springer, 2003.
[16] O. Kucuktunc and H. Ferhatosmanoglu. λ-diverse nearest neighbors browsing for multidimensional data. *TKDE*, pages 481–493, 2013.
[17] K. C. K. Lee, W.-C. Lee, and H. V. Leong. Nearest surrounder queries. In *ICDE*, pages 85–85, 2006.
[18] R. E. Madsen, D. Kauchak, and C. Elkan. Modeling word burstiness using the dirichlet distribution. In *ICML*, pages 545–552. ACM, 2005.
[19] J. H. Schiller and A. Voisard, editors. *Location-Based Services*. Morgan Kaufmann, 2004.
[20] T. Seidl and H.-P. Kriegel. Optimal multi-step k-nearest neighbor search. In *Acm Sigmod Record*, volume 27, pages 154–165. Acm, 1998.
[21] M. R. Vieira, H. L. Razente, M. C. Barioni, M. Hadjieleftheriou, D. Srivastava, C. Traina, and V. J. Tsotras. On query result diversification. In *ICDE*, pages 1163–1174. IEEE, 2011.
[22] C. Zhang et al. Diversified spatial keyword search on road networks. In *EDBT*, pages 367–378, 2014.
[23] B. Zheng, K. Zheng, X. Xiao, H. Su, H. Yin, X. Zhou, and G. Li. Keyword-aware continuous knn query on road networks. In *ICDE*, pages 871–882, 2016.
[24] K. Zheng, S. Shang, N. J. Yuan, and Y. Yang. Towards efficient search for activity trajectories. In *ICDE*, pages 230–241, 2013.