λ-Diverse Nearest Neighbors Browsing for Multidimensional Data

Onur Kucuktunc and Hakan Ferhatosmanoglu

Abstract—Traditional search methods try to obtain the most relevant information and rank it according to the degree of similarity to the queries. Diversity in query results is also preferred by a variety of applications since results very similar to each other cannot capture all aspects of the queried topic. In this paper, we focus on the **λ-diverse k-nearest neighbor search** problem on spatial and multidimensional data. Unlike the approach of diversifying query results in a postprocessing step, we naturally obtain diverse results with the proposed geometric and index-based methods. We first make an analogy with the concept of Natural Neighbors (NatN) and propose a natural neighbor-based method for 2D and 3D data and an incremental browsing algorithm based on Gabriel graphs for higher dimensional spaces. We then introduce a diverse browsing method based on the distance browsing feature of spatial index structures, such as R-trees. The algorithm maintains a Priority Queue with *mindivdist* of the objects depending on both relevancy and angular diversity and efficiently prunes nondiverse items and nodes. We experiment with a number of spatial and high-dimensional data sets, including Factual's (http://www.factual.com/) US points-of-interest data set of 13M entries. On the experimental setup, the diverse browsing method is shown to be more efficient (regarding disk accesses) than k-NN search on R-trees, and more effective (regarding Maximal Marginal Relevance (MMR)) than the diverse nearest neighbor search techniques found in the literature.

Index Terms—Diversity, diverse nearest neighbor search, angular similarity, natural neighbors, Gabriel graph

1 Introduction

Most similarity search methods in the literature produce results based on the ranked degree of similarity to the query. However, the results could be unsatisfactory, especially when there is an ambiguity in the query or when the search results include redundantly similar data. It is typically better to answer the query with diverse search results instead of homogeneous results representing similar cases. A reasonable strategy for responding to an ambiguous query is to return a mixture of results covering all aspects of the query. Different users may have different intentions while searching, and at least the initial result should include a diverse set of results to better match a variety of users' expectations. Redundantly repeating search results is another problem of conventional similarity search techniques, particularly for search spaces that include many duplicate data. In this case, similar but homogeneous information will fill up the top results. This situation has been discussed in several application areas, such as recommender systems [33], online shopping [29], and web search [8].

Similar problems also exist when querying and browsing spatial data. In some applications, diversity is preferred over similarity due to the information overload (see Fig. 1). Suppose a criminal is spotted in NYC by a camera at sixth Ave and 33rd St (**C0**). Back in police department, the police have access to a number of cameras in the city, but have limited screens (say $k = 5$) to display the view of different cameras. The cameras are labeled in order by their distance from **C0**. Instead of returning the closest point-of-interests (POIs), a result set containing close yet diverse results is preferred for such an application.

Diversity in k-NN search is not limited to the spatial domain. A diverse k-NN classifier can be potentially useful for medical diagnosis since it is more likely to unveil minority reports by grouping and eliminating similar cases. Suppose that a number of medical records are labeled with "$+$" and "$-$" labels depending on whether a patient has disease D or not, respectively. Given a patient's medical records q, the aim of k-NN classifier is to classify the patient as D^+ or D^- by finding the majority class label for the closest k records. Fig. 2 depicts the classification results obtained from the k-NN classifier and diverse k-NN classifier.

The relation between diversity and relevance was investigated before, especially in text retrieval and summarization. Researchers have proposed linear combinations of diversity and relevance [4]. However, maximizing diversity of a result set is known to be NP-hard [5], [6]. Hence, some studies develop heuristic techniques [18], [31] to optimize the results.

Considering the diversification problem in the spatial domain, it is possible to present an intuitive solution based on clustering. Data can be initially clustered and then representatives of clusters around the query point can be returned as search results. Although clustering can be computationally expensive, there are methods to generate those representatives with tree-based approaches [21]. The problem of clustering-based methods is that initial clusters may be unsatisfactory depending on the settings of the query. Furthermore, if data needs to be clustered for each query, the method is obviously not scalable. Today, most database management systems support a spatial index

- *O. Kucuktunc is with the Department of Computer Science and Engineering, The Ohio State University, 2015 Neil Ave. DL 395, Columbus, OH 43210. E-mail: kucuktunc.1@osu.edu.*
- *H. Ferhatosmanoglu is with the Department of Computer Engineering, Bilkent University, Ankara 06800, Turkey. E-mail: hakan@cs.bilkent.edu.tr.*

Manuscript received 11 Apr. 2011; revised 10 Oct. 2011; accepted 21 Nov. 2011; published online 5 Dec. 2011.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2011-04-0195. Digital Object Identifier no. 10.1109/TKDE.2011.251.

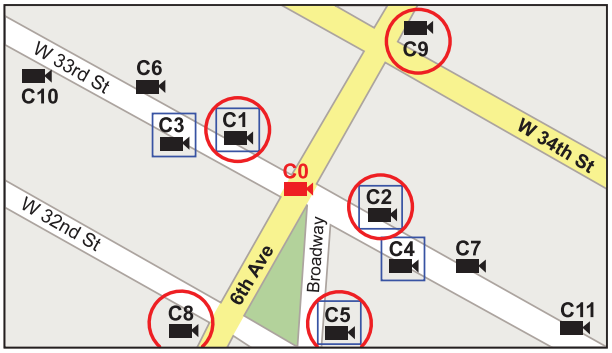

Fig. 1. A location-based service application can be adversely affected with the information overload. A conventional similarity search (e.g., k-nearest neighbor search) returns four cameras in W 33rd street and one from Broadway (blue squares); whereas, diverse browsing can capture spatial distribution around the first camera and provide superior results (red circles).

(e.g., R-tree or one of its variants). Therefore, diversity can be obtained by taking advantage of the spatial index without the extra cost of clustering.

In this paper, we first give a geometric definition of *diversity* by making an analogy with the concept of Natural Neighbors (NatN). We propose a natural neighbor-based method and an incremental browsing algorithm based on Gabriel Graph (GG) for diverse nearest neighbor search problem. We also introduce a diverse browsing method based on the popular distance browsing feature of R-tree index structures. The method maintains a Priority-Queue (PQ) with *mindivdist* of the objects depending on both relevancy and diversity, and efficiently prunes nondiverse items and nodes. Providing a measure that captures both relevancy and diversity, we show that pruning internal nodes with respect to their diversity from the items in the result set helps us to achieve more diverse results. The contributions of this paper can be summarized as follows:

- We formalize the **λ-diverse k-nearest neighbor search problem** based on angular similarity and develop measures to evaluate the relevancy and diversity of the retrieved results.
- We propose two geometric diverse browsing approaches for static databases, each of which effectively captures the spatial distribution around a query point and hence gives a diverse set of results.
- Extending the distance browsing feature, we introduce an efficient λ-diverse k-nearest neighbor search algorithm on R-trees, namely diverse browsing, and prove its correctness.
- We conduct experiments on 2D and multidimensional data sets to evaluate the performance of the proposed methods.

Key advantages of the proposed geometric and index-based methods are as follow:

- Geometric methods are appropriate for static databases and perform very efficiently once the graphs are built.
- Index-based diverse browsing does not require any change in the index structure, therefore, it can easily be integrated into various databases.

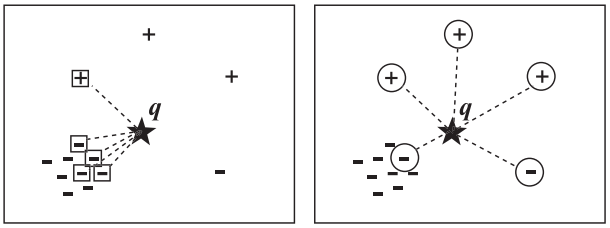

Fig. 2. A classification example with k-NN (*left*) and diverse k-NN (*right*) classifiers for $k = 5$. Although four out of five closest medical data points to q are D^-, a diverse perspective unveils minority reports and classifies the patient as D^+ with 60 percent confidence.

- With effective pruning, diverse browsing performs more efficiently than k-NN search on R-trees and also performs better than the state-of-the-art techniques regarding MMR metric.

The rest of the paper is organized as follows: related work is discussed in Section 2. Problem formulation is given in Section 3. Geometric approaches are defined in Section 4, and index-based diverse browsing is proposed in Section 5. Experiments are reported in Section 6. Section 7 gives conclusions and future work.

2 Related Work

There are notable works on diverse ranking in the literature. Carbonell and Goldstein [4] describe the Maximal Marginal Relevance (MMR) method for text retrieval and summarization. MMR attempts to find a result set by maximizing the query relevance and also minimizing the similarity between documents in the result set. The proposed method combines relevancy and novelty with a user-defined parameter (λ), which affects the importance of relevancy and diversity of the results.

Since the problem of finding diverse results is known to be NP-hard, Jain et al. [15], [18] investigate the k-nearest diverse neighbor search problem and develop two greedy approaches to optimize the results in terms of both relevancy and diversity. Both proposed methods employ the advantages of an available R-tree index. *Immediate Greedy* (IG) incrementally grows the result set R by including nearest points only if they are diverse enough from the data points already in R. *Buffered Greedy* (BG) tries to overcome some deficiencies of IG. They use the R-tree index only for getting the query's nearest neighbors in the data set. Yu et al. [30], [31] address the issue of diversification in recommendation systems and introduce two heuristic algorithms to maximize the diversity under different relevance constraints. They state that maximizing diversity is about finding a balance between relevance and diversity. The proposed *Swap algorithm* basically tries to swap elements which are less likely to contribute to the set diversity with diverse ones. *Greedy algorithm*, similar to IG, includes the next most relevant item to the result set only if that item is diverse with respect to the items already in the result set.

Some other studies attack the diversity problem in various ways. Liu and Jagadish [21] employ the idea of clustering to find a solution to the Many-Answers Problem. They suggest that taking one representative from each cluster results in more diverse results. They propose a tree-based approach for efficiently finding the representatives, even if the search space is constrained at runtime. Halvey

et al. [14] compare dissimilarity and clustering-based diverse reranking methods to introduce diversity in video retrieval results.

The notions of diversity and novelty are generally discussed in the context of information retrieval and recommendation systems. Clarke et al. [8] investigate the problems of ambiguity in queries and redundancy in results and propose an evaluation framework. Chen and Karger [7] describe a retrieval method which assigns negative feedback to the documents that are included in the result list for maximizing diversity. Vee et al. [29] present inverted list algorithms for computing diverse query results in online shopping applications. Ziegler et al. [32], [33] present an algorithmic framework to increase the diversity of a top-k list of recommended products. In order to show its efficiency, they also introduce a new intralist similarity metric.

There are also works on content diversity over continuous data of publish/subscribe systems, such as news alerts, RSS feeds, social network notifications [9], and the diverse skyline [28]. Greedy heuristics were proposed for the problems although the discussions on how relevance and diversity should be combined, and how well greedy approaches approximate the optimal solution is fairly useful. Interested readers may refer to [22] for further information on diversity.

3 Preliminaries

Before we state our diversity and λ-diverse k-nearest neighbor search definitions, let us first analyze the approach used by KNDN-IG and KNDN-BG [15], [18]. For KNDN-IG the objective is to find a fully diverse set of results R close to the query point q. This means that for all $r_1, r_2 \in R$,

$$\text{divdist}(r_1, r_2, V(q)) = \sum_{j=1}^{L} (W_j \times \delta_j) > MinDiv,$$

where $V(q)$ is the diversity attributes, L is the number of dimensions to be diversified (in our case $L = d$), W is the set of weighting factors for the differences $(\delta_1, \ldots, \delta_L)$ sorted in decreasing order, and $MinDiv$ is the minimum diversity distance. The diversity computation in KNDN-IG is simply based on the Gower coefficient [13] with monotonically decaying weights W calculated with

$$W_j = \frac{a^{j-1} \times (1-a)}{1-a^L}, \qquad \text{for } 1 \leq \text{j} \leq \text{L},$$

where a is the rate of decay.

KNDN-BG applies the same technique; however, it stores the eliminated points. Then the method checks if two of those points (say p' and p'') are pairwise-diverse, and also eliminated because of the same resulting point $r_i \in R$. If it finds such a pair, R is updated as $(R \setminus \{r_i\}) \cup \{p', p''\}$.

$MinDiv$ setting assumes that data is in $[0,1]^d$ space; otherwise, it must be set with a knowledge of density and the range of data. When a is selected around 0.1, the dimension with the highest difference is overrated. This suggests that KNDN favors the points along the axes. Hence, it tries to find 2^d diverse points in a d-dimensional space. This behavior is problematic since k could be any number. Consequently, the results do not guarantee that KNDN accurately captures the distribution around the query point. Finally, these methods do not allow to select how diverse/relevant the results will be, unless the importance of diversity is embedded into either $MinDiv$ or a parameters.

Our diversity definition employs the angular similarity between two points regarding the query point. In other words, a diverse k-nearest neighbor search method should maximize the pairwise angular similarity while minimizing the overall distance. Angular similarity and diverse k-nearest neighbor search is defined in Definitions 3.1 and 3.2, respectively.

Definition 3.1 (Angular Similarity). *Given a query point q, two points p_1 and p_2, and an angle θ,* angular similarity *(sim_θ) of p_1 with respect to q and another point p_2 is*

$$\text{sim}_\theta(p_1, q, p_2) = \begin{cases} 1 - \widehat{p_1 q p_2}/\theta, & \text{if } \widehat{p_1 q p_2} < \theta, \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

sim_θ results in 0 if the angle $\widehat{p_1 q p_2}$ is greater than θ. It becomes 1 if both of them point at the same direction.

Definition 3.2 (λ-Diverse k-Nearest Neighbor Search). *Given a set of points S, a query point q, and a diversity ratio λ, the* λ-diverse k-nearest neighbor search *on q retrieves a set of k resulting points $R = \{p_1, \ldots, p_k\}$, such that*

$$R = \underset{\substack{R \subseteq S \\ |R|=k}}{\text{argmin}} \left[\alpha \sum_{i=1}^{k} dist(q, p_i) + \frac{2\beta}{k(k-1)} \sum_{i=1}^{k} \sum_{j=i+1}^{k} \text{sim}_\theta(p_i, q, p_j) \right], \quad (2)$$

where $\alpha = 1 - \lambda$ and $\beta = \lambda$. This function minimizes pairwise angular similarity (depending on λ) and maximizes relevancy (depending on $1 - \lambda$) of the results.

Note that the relative importance of diversity versus relevance is adjusted with the λ parameter. When $\lambda = 0$, the method reduces to k-NN search. For $0.5 \leq \lambda \leq 0.9$, the results are expected to be diverse enough without sacrificing relevancy. As a result, Definition 3.2 provides a more flexible and user-tunable setting for diversification of k-NN queries.

The rest of the paper includes proposed geometric and index-based methods for efficiently solving λ-diverse k-nearest neighbor search problem. The notation used in the paper is given in Table 1.

4 Geometric Diverse Browsing

In the spatial domain, diverse k-nearest neighbor search is conceptually similar to the idea of natural neighbors, which is calculated with Voronoi Diagrams (VD) and Delaunay Triangulation (DT) [2], [19]. In this section, we first present an analogy of diversity with natural neighbors. Based on the analogy, we propose a natural neighbor-based method along with the techniques that are used to efficiently retrieve natural neighbors of a query point. Although the discussions are mostly on the 2D space, the method can be extended to work on the 3D space as well. Observing the limitations of this method in higher dimensional spaces, we present another geometric approach, namely the Gabriel graph-based diverse browsing method. More information on the computation geometry concepts is given below.

TABLE 1
Notation

Symbol	Description
d	# dimensions
S	dataset composed of d-dimensional points
n	$\|S\|$, size of the dataset
k	# search results
q	d-dimensional query point
λ	importance of diversity over relevance
R	set of result points
K	set of result points for k-NN search
sim_θ	angular similarity of two points regarding q
DT(S)	Delaunay triangulation of points in S
GG(S)	Gabriel graph of points in S
k'	# natural neighbors for q in $S \cup \{q\}$
W	weights w_i of each natural neighbor
adj[p]	adjacent nodes/points of p in a graph
$l_{GG}(k)$	# layers required to obtain k Gabriel neighbors
$\circledast_p^q$	pruning sector from q towards p
$\overrightarrow{qp}$	vector from query point q in the direction of p
ϵ	small fraction to relax $\circledast_p^q$
$\theta_\circledast$	central angle of the pruning sector
$r_\circledast$	radius of the pruning sector
p_i	a point in S
p_{nn}	the nearest neighbor point of q
B_i	an index node in R-tree
$mindist$	minimum distance measure [25]
$mindivdist$	measure that combines $mindist$ and sim_θ
PQ	priority queue
cts	current timestamp
ts[$Node$]	timestamp of a node in PQ

4.1 Voronoi Diagrams, Delaunay Triangulations, and Gabriel Graphs

Voronoi diagram, Delaunay triangulation, and Gabriel graph are popular concepts in computational geometry. For the readers who are unfamiliar with these concepts, the definitions and their relationships with each other and with other popular subgraphs are given in this section.

Definition 4.1 (Voronoi Cell). *Given a finite set $S = \{p_1, p_2, \ldots, p_n\}$ of d-dimensional points, the* Voronoi cell *$V(p)$ of a point $p \in S$ is the set of all points of R^d (defining a nonempty, open, and convex region) closer to p than to any other point in S:*

$$V(p) = \{x \in R^d | \forall q \in S, dist(x,p) \leq dist(x,q)\}. \quad (3)$$

Definition 4.2 (Voronoi Diagram). *The* Voronoi diagram *VD(S) is the collection of all Voronoi cells of the points in S, forming a cell complex partitioning R^d.*

Definition 4.3 (Delaunay Triangulation). *Given a finite set $S = \{p_1, p_2, \ldots, p_n\}$ of d-dimensional points, the* Delaunay triangulation *of S is a triangulation DT(S) such that no point in S is inside the circumcircle (circum-hypersphere for $d > 2$) of any simplex in DT(S).*

Delaunay triangulation is a dual graph of Voronoi diagram for the same set of points S.

Definition 4.4 (Gabriel Graph). *The Gabriel graph [12] is the set of edges e_{ij} that is a subset of DT(S), for which the circle (hypersphere for $d > 2$) with diameter $[p_i p_j]$ contains no other points from S:*

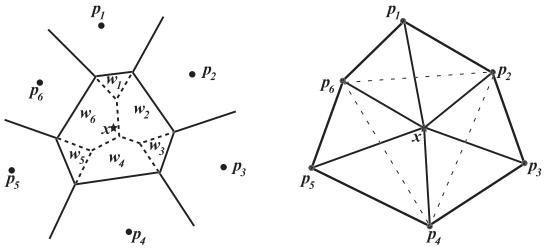

Fig. 3. Natural neighbor weights w_i of x in 2D (left), DT with and without x (right).

$$\text{GG}(S) = \{e_{ij} \subseteq \text{DT}(S) | \forall p_k \in S, |p_k p_i|^2 + |p_k p_j|^2 \geq |p_i p_j|^2\}. \quad (4)$$

Given a finite set $S = \{p_1, p_2, \ldots, p_n\}$ of d-dimensional points, the following subgraph relationships hold

$$\text{MST}(S) \subseteq \text{RNG}(S) \subseteq \text{GG}(S) \subseteq \text{DT}(S), \quad (5)$$

where MST is the minimum spanning tree, and RNG is the relative neighborhood graph [11].

4.2 Analogy with Natural Neighbors

The *natural neighbors* of a point $p \in S$ are the points in S sharing an edge with p in DT(S). They also correspond to Voronoi cells that are neighbors of V_p. In case of a point $x \notin S$, its natural neighbors are the points in S whose Voronoi cells would be modified if x is inserted in VD(S). The insertion of x creates a new Voronoi cell V_x^+ that steals volume from the Voronoi cells of its potential natural neighbors (see Fig. 3).

To capture the influence of each NatN, we use natural neighbor weights in natural neighbor interpolation [27]. Let D be the VD(S), and $D^+ = D \cup \{x\}$. The Voronoi cell of a point p in D is defined by V_p, and V_p^+ is its cell in D^+. The natural neighbor weight of x with respect to a point p_i is

$$w_i(x) = \frac{\text{Vol}(V_{p_i} \cap V_x^+)}{\text{Vol}(V_x^+)}, \quad (6)$$

where $\text{Vol}(V_{p_i})$ represents the volume of V_{p_i}, and $0 \leq w_i(x) \leq 1$. The natural neighbor weights are affected by both the distance from x to p_i and the spatial distribution of p_i around x.

4.3 Natural Neighbor-Based Method

Based on the property of the natural neighbor concept which captures both the distance to a query point and also the spatial distribution around it, we claim that the natural neighbors of a query point q give a diverse set of similarity search results, if the natural neighbor weights W are used as ranking measures. The method works as follows: 1) simulate the insertion of q into DT(S), 2) find the natural neighbors of q: $\{p_1, \ldots, p_{k'}\}$ along with the weights $W = \{w_1, \ldots, w_{k'}\}$, and 3) report results according to the weights in descending order. Details are provided in the following sections.

4.3.1 Offline Generation of Delaunay Triangulation

In the preprocessing stage, DT(S) is calculated for all the data points in S. Although the weights are calculated with the overlapping areas of these cells, and natural neighbors

are defined (and also easy to understand) in terms of Voronoi cells, performing operations on DT is computationally more efficient.

There are I/O- and memory-efficient methods for building Delaunay triangulation in 2D and 3D [1], [17]. These methods can generate DTs for billions of points. There are also publicly available implementations for 2D [26] and for higher dimensions [3]. We use Qhull implementation [3] to generate DT(S).

4.3.2 Step 1—Flip-Based Incremental Insertion

We simulate insertion of q into DT(S) with a flip-based insertion algorithm. It is easy to determine the simplex of DT(S) containing q in linear time by inspecting all triangles.

Let τ be the simplex in DT(S) containing q. All vertices of τ automatically become natural neighbors of q once it is inserted to DT(S). Then, the necessary edge flips are carried out until no further edge needs to be flipped, revealing even more natural neighbors of q.

The number of flips needed to insert q is proportional to the degree of q (the number of incident edges) after its insertion. The average degree of a vertex in a 2D DT is six. This number is proportional to the number of dimensions.

4.3.3 Step 2—Find NatNs and Weights

Vertices $\{p_1, \ldots, p_{k'}\}$ adjacent to q in DT($S \cup q$) are the natural neighbors of q. The volume of a d-dimensional Voronoi cell is computed by decomposing it into d-simplices and summing their volumes. The volume of a d-simplex τ with vertices $(v_0, \ldots, v_n)$ [19] is computed as

$$\mathrm{Vol}(\tau) = \frac{1}{d!}|\det(v_1 - v_0 \ \cdots \ v_n - v_0)|, \tag{7}$$

where each column of the $n \times n$ determinant is the difference between the vectors representing two vertices. Weights W are then calculated with (6).

4.3.4 Step 3—Report for Diverse kNN

NatN-based method naturally returns k' results as an answer to query point q. The result set R is ranked according to the weights of each neighbor. If $k' \geq k$, we report the top-k ranked results. Otherwise, k' points are returned.

Overview of the method is given in Algorithm 1. Note that if the number of natural neighbors is greater than k, the points with smaller weights are eliminated. Otherwise, the method may return less than k results.

Algorithm 1. Algorithm NatNDiversitySearch

```
1: procedure NATNDIVERSITYSEARCH(q, k, DT)
2:     DT' ← INSERT(DT, q)
3:     W ← {}
4:     for each point p_i in adj[q] do
5:         w_i ← CALCULATEWEIGHT(DT', p_i, q)
6:         W ← W ∪ {w_i}
7:     end for
8:     W' ← SORT(W)
9:     if adj[q] > k then
10:        W' ← W'[1 : k]
11:    end if
12:    return W'.i
13: end procedure
```

4.4 Limitations

The drawback of using natural neighbors in diverse k-nearest neighbor search is that there is always a fixed number of natural neighbors of a point, and the number is proportional to the number of dimensions. This can be seen as an advantage as well, since parameter k is inherently captured by the process, not specified by the user. However, for browsing purposes, one cannot restrict the search with only natural neighbor results as the user may demand more search results. The search needs to continue incrementally through the neighbors of neighbors with Voronoi cells. Without any assumptions on the distribution of the data, the average degree of a vertex in a 2D DT is 6 [19]. In this case, diverse k-nearest neighbor search with the NatN-based method for 2D space may not return a result set with k items. As a result the method is forced to investigate the neighbors of neighbors with Voronoi cells which were not modified with the insertion of q.

For higher dimensional spaces, the average degree of a point in DT grows quickly with d (approximately d^d) [10]. The problem of selecting a subset of elements in this set to obtain a diverse set of k items cannot be trivially solved with a NatN-based approach. Because of the disadvantages, NatN-based method is more appropriate for low-dimensional data and small k values.

4.5 Gabriel Neighbor-Based Method

In high dimensions, DT is intractable in terms of both construction complexity $O(n^{\lceil d/2 \rceil})$ and browsing efficiency. For better scalability and browsing capability in high dimensional spaces, we use Gabriel graphs instead of DT.

Gabriel graph contains those edges of DT that intersect their Voronoi faces [23]. Hence, GG can be constructed in $O(n \log n)$ time by first constructing DT and VD, and then adding each edge in DT to GG if it intersects its Voronoi face. Without DT and VD, GG can always be constructed by brute-force in $O(n^3)$ time.

The advantage of working with GG is that both nearest neighbor graph (NNG) and minimum spanning tree are subgraphs of it; therefore, GG still captures proximity relationships among data points. Furthermore, GG is reasonably sparse and simple: for planar graphs $|GG(S)| \leq 3n - 8$ [23]. For this reason, the Gabriel graph is found to be effective in constructing power-efficient topology for wireless and sensor networks [20].

Our solution for diverse k-nearest neighbor search problem is to browse GG layer-by-layer, starting from the nearest point p_{nn} to the query point q. Fig. 4 shows an example of GG layers connected with B-spline. For efficiency, the query point is not inserted into GG(S), but rather the spatial location of q is imitated with its nearest neighbor p_{nn}.

After finding p_{nn}, the algorithm iteratively searches the n-degree neighbors of p_{nn} in GG(S), starting with $n = 1$. GGDIVERSITYSEARCH stops when k or more points are included in R. Note that the resulting points are added layer-by-layer; therefore, there is a ranking among layers. However, they are not sorted within layers, since there is no concept similar to natural neighbor weights in Gabriel graphs. In addition, $|R| \geq k$, meaning that the algorithm may return more than k results. The method is given in Algorithm 2.

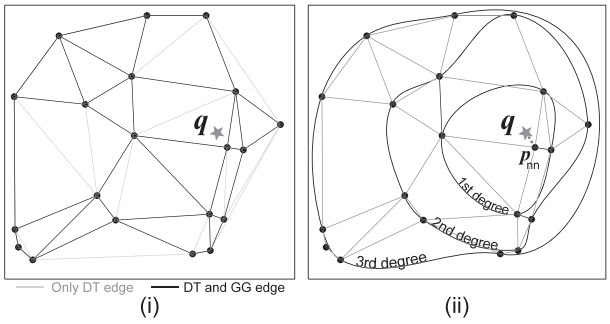

Fig. 4. Incremental browsing of a Gabriel graph. (i) Delaunay triangulation of the points with Gabriel edges highlighted. (ii) For a query point q, diverse results are gathered layer-by-layer starting with p_{nn}.

Algorithm 2. Algorithm GGDiversitySearch

procedure GGDIVERSITYSEARCH (q, k, GG, S)
  $p_{\text{nn}} \leftarrow$ NEARESTNEIGHBOR(S, q)
  $R \leftarrow \{\}$
  $R' \leftarrow \{p_{\text{nn}}\}$
  while $|R| < k$ **do**
    $R \leftarrow R \cup R'$
    $R'' \leftarrow \{\}$
    for each point p in R' **do**
      $R'' \leftarrow R' \cup$ adj$[p]$
    end for
    $R' \leftarrow (R'' \setminus R)$
  end while
  return R
end procedure

It is possible to return exactly k results by examining the last layer of points added into R. One method is to choose a subset of points from the last layer, which optimizes the overall diversity of R. We are using a similar approach in the experiments. The problem can be defined as follows.

Let q be a query point, and k be the number of results for a diverse k-nearest neighbor search. Suppose R is the Gabriel neighbors of q, inserted layer-by-layer up to the layer $l_{GG}(k) - 1$, satisfying $|R| < k$. The problem is to select a set of $k - |R|$ objects from the layer $l_{GG}(k)$ so that the diversity of R' will be maximum. L refers to the last layer to be investigated. Algorithm 3 finds a local maximum for diversity of the results.

Algorithm 3. Algorithm GGOptimizeLastLayer

Procedure GGOPTIMIZELASTLAYER(q, k, R, L)
  $R' \leftarrow R$
  $L' \leftarrow L$
  while $|R'| < k$ **do**
    $o \leftarrow argmin_{obj \in L'}$ DIV$(R' \cup obj)$
    $R' \leftarrow R' \cup \{o\}$
    $L' \leftarrow L' \setminus \{o\}$
  end while
  return R'
end procedure

5 Index-Based Diverse Browsing

Spatial databases mostly come with an index structure, such as the widely used R-tree [25]. A popular method called *distance browsing* [16] tries to visit the k-nearest neighbors (k-NN) of a point in a spatial database that uses the R-tree index. We introduce *diverse browsing* for the diverse k-nearest neighbor search over an R-tree index.

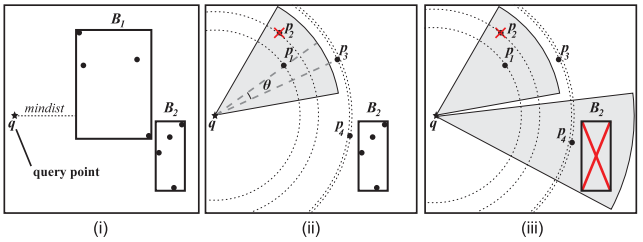

Fig. 5. Example for diverse browsing. (i) Suppose B_1 and B_2 are two internal nodes of the R-tree index. (ii) When the closest MBR is investigated and the closest point p_1 is added to the result set, p_2 is pruned because of high angular and distance similarity to p_1. $mindivdist[p_3]$ also increases here due to its high angular similarity to p_1. (iii) Next, p_4 is added to the result set and causes B_2 to be pruned since none of the items in B_2 can be diverse.

The principal idea of diverse browsing is to use the distance browsing method with a pruning mechanism that omits non-diverse data points and minimum bounding rectangles (MBRs). A priority queue is maintained with respect to *mindivdist* measure, which is a combination of the $mindist$ [16] and the angular similarity (1) for the object obj (either a data point or an R-tree index node). In each iteration, the closest object is investigated (see Fig. 5).

In the following sections, *mindivdist* measure and pruning mechanisms (5.1), correctness of the algorithm (5.2), and their integration into our incremental diverse browsing method (5.3) are explained. Note that the algorithm is given in two parts (Algorithms 4 and 5).

Algorithm 4. Algorithm MinDivDist

procedure MINDIVDIST(q, obj, R, λ)
  $\theta_s \leftarrow \frac{2\pi}{k+\epsilon}$
  $r_s \leftarrow 1 + \lambda$
  $\delta \leftarrow mindist(q, obj)$
  if obj is a point **then**
    for each point p in R **do**
      $s[p] \leftarrow \text{sim}_\theta(obj, q, p)$
      if $s[p] > 0$ and $\delta < |\overrightarrow{qp}| \times r_s$ **then**
        return PRUNE(obj)
      end if
    end for
    $mindivdist \leftarrow \lambda \times \max(s) + (1 - \lambda) \times \delta$
  else if obj is a rectangle **then**
    for each point p in R **do**
      $s \leftarrow \min_{y \in obj.corners}(\text{sim}_\theta(y, q, p))$
      if $\forall y \in obj.corners$ in $\circledast_p^q$ **then**
        return PRUNE(obj)
      else if $\exists y \in obj.corners$ in $\circledast_p^q$ **then**
        $\delta \leftarrow min(|\overrightarrow{qp}| \times r_s, |\overrightarrow{qy}|)$
      end if
      $mdds[p] \leftarrow \lambda \times s + (1 - \lambda) \times \delta$
    end for
    $mindivdist \leftarrow \max(mdds)$
  end if
  return $mindivdist$
end procedure

Algorithm 5. Algorithm DiverseKNNSearch

```
1: procedure DIVERSEKNNSEARCH(q, k, λ, R-tree)
2:     R ← {}
3:     cts ← 0
4:     PQ ← MINPRIORITYQUEUE
5:     ENQUEUE(PQ, ⟨R − tree.root, cts, 0⟩)
6:     while not ISEMPTY(PQ) and |R| < k do
7:         while ts[TOP(PQ)] < cts do
8:             e ← DEQUEUE(PQ)
9:             ENQUEUE(PQ, ⟨e, cts, MINDIVDIST(e)⟩)
10:        end while
11:        e ← DEQUEUE(PQ)
12:        if e is a point then
13:            R ← R ∪ {e}
14:            cts ← cts + 1
15:        else
16:            for each obj in node e do
17:              ENQUEUE(PQ, ⟨obj, cts, MINDIVDIST(obj)⟩)
18:            end for
19:        end if
20:    end while
21:    return R
22: end procedure
```

5.1 MinDivDist Measure and Pruning

When a point p is added to the result set R, we draw an imaginary sector $\circledast_p^q$ from the query point q in the direction of p with $\theta_{\circledast} = 2 \times \theta_s$ angle and $r_{\circledast} = r_s \times |\overrightarrow{qp}|$. Every point in this sector will eventually be pruned. By default, $r_s = 1 + \lambda$ and $\theta_s = \frac{2\pi}{k+\epsilon}$, unless specified otherwise.

We use the term *mindivdist* as an alternative to *mindist* in distance browsing. *mindivdist* of points and MBRs in the priority queue are calculated according to the angular similarity and distance with respect to the elements in R (see Algorithm 4). Note that the points with *mindivdist* closer to 0 are more likely to be included in the result set.

Points without enough angular diversity and distance from another point in R are pruned. Similarly, the algorithm also prunes MBRs only if none of the corners of the object are diverse enough to be in the result set. The advantage here is that all the pruned data can be displayed as *similar results* of each resulting point with a small modification since we have the information why a point is pruned.

5.2 Correctness of the Algorithm

The efficiency of the proposed method comes from the *diverse browsing* of the R-tree structure. As in incremental nearest neighbor search algorithms and distance browsing [16], a min-priority queue is maintained after each operation. However, instead of *mindist* metric we use the result of the MINDIVDIST function as the value of each object in PQ. MINDIVDIST gives a non-negative value of a point or an MBR depending on its angular diversity and distance to the query point depending on both R and λ. In each iteration, the object with the lowest *mindivdist* (top of PQ) is investigated.

As *mindivdist* of each object in PQ depends on the current state of R, some of *mindivdist* values will be obsolete after another point is inserted into R. But, instead of updating all the objects in PQ (which would be inefficient), we argue to update only the top of PQ with a *timestamp-based* approach until an up-to-date object is acquired. Current timestamp (cts) is incremented every time a point is included in the result set. The proposed method based on timestamp-based update of *mindivdist* in PQ is proved by Lemma 5.1 and Theorem 5.2.

Lemma 5.1. *Update operation on an object, which is on top of PQ and has an earlier timestamp, can either increase* mindivdist *of the object or does not affect it at all.*

Proof. An object obj is updated only when $\text{ts}[obj] < cts$. Since cts increases when a new item is added to R, there are exactly $(cts - ts[obj])$ new items in the result set R^+ compared to the time when $mindivdist[obj]$ was calculated.

Suppose that the new *mindivdist* of obj at the current timestamp is $mindivdist^+[obj]$. If obj is a point, three possible outcomes of the update are as follow:

1. $\exists p \in R^+,\ obj$ resides in $\circledast_p^q$
   $\Rightarrow$ PRUNE(obj).
2. $\exists p \in R^+,\ \text{sim}_\theta(obj, q, p) > \max_{r \in R}(\text{sim}_\theta(obj, q, r))$
   $\Rightarrow mindivdist^+[obj] > mindivdist[obj]$.
3. $\forall p \in R^+,\ \text{sim}_\theta(obj, q, p) \leq \max_{r \in R}(\text{sim}_\theta(obj, q, r))$
   $\Rightarrow mindivdist^+[obj] = mindivdist[obj]$.

On the other hand, if obj is a leaf or internal node, the *mindivdist* depends on the corners of the MBR:

1. $\exists p \in R^+,\ \forall y \in obj.corners,\ y$ resides in $\circledast_p^q$
   $\Rightarrow$ PRUNE(obj).
2. $\exists p \in R^+,\ \exists y \in obj.corners,$
   $\text{sim}_\theta(y, q, p) > \max_{r \in R}(\text{sim}_\theta(obj, q, r))$
   $\Rightarrow mindivdist^+[obj] > mindivdist[obj]$.
3. $\forall p \in R^+,\ \exists y \in obj.corners,\ y$ resides in $\circledast_p^q$
   $\text{sim}_\theta(y, q, p) \leq \max_{r \in R}(\text{sim}_\theta(obj, q, r))$
   $\Rightarrow mindivdist^+[obj] = mindivdist[obj]$.

We have shown that some updated objects are pruned. If not, its *mindivdist* either increases or stays the same. □

Theorem 5.2. *An object on top of PQ with the current timestamp provides a lower bound for the* mindivdist *of all objects in PQ, even if there are other objects in PQ with earlier timestamps.*

Proof. Suppose obj is on top of PQ with the current timestamp, and let obj' be another object in PQ with an older timestamp ($ts[obj'] < cts$). Following Lemma 5.1, even if the *mindivdist* of obj' is updated, it is either pruned or $mindivdist^+[obj'] \geq mindivdist[obj']$. Since obj' is not on top of PQ, $mindivdist[obj'] \geq mindivdist[obj]$. Hence $mindivdist^+[obj'] \geq mindivdist[obj]$. Therefore, $mindivdist[obj]$ is still a lower bound for the *mindivdist* of the objects in PQ. □

5.3 Incremental Diverse Browsing

After extending the distance browsing feature of R-trees with diverse choices, incremental browsing of an R-tree gives diverse results depending on λ. Details of the method are given in Algorithm 5, excluding certain boundary conditions, i.e., when $\lambda = 1$, or PQ becomes empty. Note that q is

the query point in a d-dimensional search space where other points are already partitioned with an R-tree index.

The proposed algorithm has the following properties:

Property 5.3. *Diverse k-nearest neighbor results obtained by the diverse browsing method always contain p_{nn}, the nearest neighbor of the query point q.*

Proof. Initially $R = \emptyset$. Therefore *mindivdist* of every object $obj \in$ PQ is calculated solely depending on the $mindist$ to the query point (see Algorithm 4). The algorithm's behavior is similar to that of the distance browsing method at this stage. When the first point p is dequeued from PQ, it is included to R. p is also the point with the minimum distance to q. Hence, $p_{\mathrm{nn}} \in R$. □

Property 5.4. *Diverse browsing can capture the set P_c which comprises k points uniformly distributed around q with the same distances as p_{nn}.*

Proof. The method selects the points in P_c without pruning any of them. We guarantee that the points in P_c are retrieved without any assumptions about the order. In addition, $\min \widehat{p_i q p_j} = 2\pi/k$ where $p_i, p_j \in P_c$. Therefore, $[2\pi/k] \geq [2\pi/(k+\epsilon)] = \theta_s$. Hence, no point in P_c is pruned. □

Dissimilarity-based diversification methods (e.g., [18]) do not support this property since they are likely to prune some points in P_c, especially when $k > 6$.

6 Experiments

We define the evaluation measures in Section 6.1. Real and synthetic data sets used in the experiments are summarized in Section 6.2. Evaluation and discussion of the methods for spatial and high-dimensional data sets are given in Sections 6.3 and 6.4.

6.1 Evaluation Measures

In order to measure how well the methods capture the relevancy and the spatial distribution around the query point, we use the evaluation measures given in Definitions 6.1, 6.2, and 6.3.

Definition 6.1 (Angular Diversity). *Given a query point q and a set of results R, angular diversity measures the spatial diversity around the query point*

$$\mathrm{DIV}(q, R) = 1 - \frac{\left\| \sum_{p_i \in R} \frac{\overrightarrow{qp_i}}{\|\overrightarrow{qp_i}\|} \right\|}{|R|}. \tag{8}$$

The intuition behind this measure is that each of the points in R tries to influence the overall result in the direction of the point itself. If the result set is fully diverse, sum of these "forces" will be closer to the center; therefore, the average influence on the query point gives an idea of how diverse the result set is. This measure can easily be applied to higher dimensions since it consists of simple vector additions and normalization. See Fig. 6 for the angular diversity of the points in Fig. 3.

However, the angular diversity measure is not adequate to evaluate the diversity of a result set since an algorithm can always return a better set of items more distant than the nearest neighbors if the distance factor is omitted.

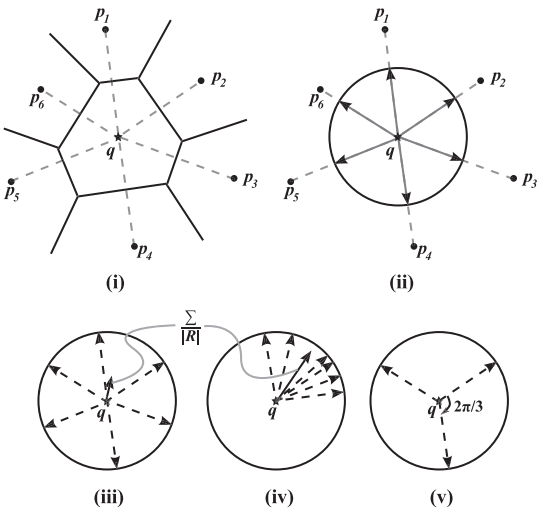

Fig. 6. Angular diversity measure. (i) Suppose diverse 6-nearest neighbor search for q retrieves $\{p_1, \ldots, p_6\}$. (ii) Angular diversity of the result set is calculated by the sum of vectors $q\vec{p}_i$ on a unit circle/sphere. (iii) When the result set is diverse, the average of these vectors will be close to the center; otherwise, (iv) the average will be close to the circle. (v) For example, maximum angular diversity (DIV=1) in 2D for $k = 3$ can be achieved with points which have an angle of $2\pi/3$ pairwise.

Definition 6.2 (Relevance). *Given a query point q, its k-nearest neighbors K, and a set of results R, relevance measure calculates the normalized average distance of the points in R with respect to the k-nearest neighbors:*

$$\mathrm{REL}(q, K, R) = \frac{\sum_{p_j \in K} \left\| \overrightarrow{qp_j} \right\|}{\sum_{p_i \in R} \left\| \overrightarrow{qp_i} \right\|}. \tag{9}$$

The result of REL is in the interval [0,1]. Magnitude of each vector is calculated in the euclidean space although any metric distance measure can be applied to the function, as long as the measure is consistent with the one used in calculation of k-NN.

Definition 6.3 (Diverse-Relevance). *Given a query point q, its k-nearest neighbors K, a set of results R, and a parameter λ, diverse-relevance measures both relevancy and angular diversity of the results:*

$$\mathrm{DIVREL}(q, K, R) = \lambda \times \mathrm{DIV}(q, R) + (1 - \lambda) \times \mathrm{REL}(q, R). \tag{10}$$

DIVREL is based on the Maximal Marginal Relevance method [4]. When $\lambda = 0$, the measure evaluates the relevance of the results excluding the diversity. The aim of our methods is to maximize the diverse-relevance of a result set depending on the value of λ.

6.2 Data Sets

We conduct our experiments on both real data sets of POIs in 2D and synthetic high-dimensional data sets to evaluate the efficiency and effectiveness of the proposed methods. The properties of our data sets are summarized in Table 2.

Real data sets. We use four real-life data sets in our experiments. ROAD is the latitudes and longitudes of road crossings in Montgomery County MD [24]. 10 percent of the ROAD data set is randomly selected as query points. The North East NE data set contains postal addresses from three

TABLE 2
Properties of Data Sets

Dataset	d	Card.	Description
ROAD	2	64K	road crossings, Montgomery ct
NE	2	124K	postal addresses, northeast of US
CAL	2	105K	points of interest in California
USPOI	2	5,8M	POIs in the US, Factual data
NORM	6	500K	normal distribution ($\mu = 0, \sigma^2 = 1$)
UNI	6	500K	uniform distribution
SKEW	6	500K	skew normal dist. (μ=0,σ^2=1,α=1)
HDIM	10	1M	uniform distribution

metropolitan areas (New York, Philadelphia, and Boston).[1] The CAL data set consists of points of interest in California.[2] To avoid querying outside of the data region, 500 points from both data sets are randomly selected as queries.

Finally, the USPOI data set is extracted from more than 13M points-of-interest in the US, gathered by Factual.[3] Among those, 8.8M have the location information and 5.8M are unique. We have randomly chosen 1,000 POIs as queries.

Synthetic data sets. We generate four synthetic high-dimensional data sets. NORM is a 6D data set generated with normal distribution ($\mu = 0, \sigma^2 = 1$). UNI and HDIM are 6D and 10D data sets generated with uniform distribution. SKEW is generated with skew normal distribution ($\mu = 0$, $\sigma^2 = 1, \alpha = 1$). For each of these synthetic data sets, 200 query points are produced with the original distribution and parameters.

6.3 Evaluations

Real data sets. We compare the results of geometric approaches (NatN-based and GG-based) with diverse browsing of R-tree and KNDN-IG and KNDN-BG [18] on ROAD, NE, CAL, and USPOI data sets. In order to be consistent, the results of the NatN-based method is obtained first. Depending on the number of natural neighbors of each query k', we run other algorithms with $k = k'$. R-trees are built with a page size of 512 bytes (which holds 64 data points) and a fill factor of 0.5. Immediate greedy and buffered greedy approaches of the KNDN method are adopted for the spatial domain: the threshold parameter *MinDiv* is set to 0.1, which is also modified according to the value of λ. Note that when $\lambda = 0$, both KNDN and diverse browsing on R-tree methods reduce to k-NN.

Similar results for all real data sets (see Fig. 7) indicate that geometric methods naturally produce diverse results in terms of the DIV measure. Since the natural and Gabriel neighbors of a point are fixed in a data set, these methods are the most efficient ones, only if 1) the purpose of the search is angular diversity, and 2) the spatial database is stable. The GG-based method has an advantage over the NatN-based method while it enables for incremental diverse browsing.

In most cases a spatial index is used to represent certain points-of-interests, and those databases are updated constantly. Roads are built, new restaurants are opened, old buildings are replaced by new ones. Because geometric methods require a preprocessing time for building the entire DT or GG, they are not appropriate for dynamic databases. In addition, users may want to adjust how diverse versus relevant the search results should be. Index-based methods provide such flexibility. We will discuss the advantages and disadvantages of each method in Section 6.4.

If we focus on index-based search methods, both diverse browsing and KNDN start with DIVREL ≈ 1 when $\lambda = 0$, and they try to adjust their results as the user asks for more diversity. It is seen that diverse browsing performs better in producing a more diverse result set for the spatial domain compared to KNDN-IG and KNDN-BG (about 20 percent improvement for $\lambda = 1$, 10 percent improvement overall). The diverse browsing method also gives a high diverse-relevant set of results (see Table 3) as the user seeks diversity in the results (about 15 to 25 percent improvement).

Synthetic data sets. The purpose of experimenting in multidimensional space is to show the efficiency of each algorithm and the diverse-relevance of the results. Fig. 8 shows the comparison of diverse browsing, GG-based and KNDN methods on synthetic 6D data sets. The NatN-based method is omitted, because it is not scalable to high dimensions due to its high average degree (see Section 4.4). In order to measure the efficiency of each index-based method, we spot the page accesses when $\lambda = 1$, for which the algorithms investigate the highest number of internal nodes.

For the queries where relevance is preferred over diversity (i.e., $\lambda < 0.5$), diverse browsing and KNDN-BG perform better than the GG-based method since they are both based on the distance browsing feature of R-trees. On the other hand, the Gabriel graph-based method is extremely powerful for diversity-dominant queries (i.e., $\lambda \geq 0.5$) in terms of both computational efficiency and the diverse-relevance of the results. After retrieving the nearest neighbor p_{nn} in the database, it only takes page accesses equal to the number of layers $l_{GG}(k)$ required to obtain k Gabriel neighbors. From our observations, $l_{GG}(k) \leq 2$ for $k = O(d^2)$. The GG-based method also improves the diverse-relevance of the results up to 25 percent when $\lambda \geq 0.5$.

The most challenging data set we experiment on is HDIM data set. Because generating the GG efficiently in high-dimensions is not the concern of this paper, we decided to extract only the necessary Gabriel-edges for this experiment. Figs. 8d and 8h suggest that the GG-based method is highly effective for diversity-intended queries, where index-based methods return similar results for different λ values. This is obviously because the euclidean distance in higher dimensions may not accurately measure the similarity. But still, diverse browsing method is able to produce the same results for $\lambda \leq 0.5$ as KNDN-IG and KNDN-BG with 36 percent less page accesses. For $\lambda > 0.5$, diverse browsing is more effective and efficient.

Diverse browsing makes less disk accesses as it successfully prunes out the index nodes that are not diverse with respect to the results found. Hence, it does not make any unnecessary disk accesses. However, both KNDN methods iteratively investigate all nearest neighbors to find the next diverse element.

Effects of the parameters. So far we investigate the effects of parameter λ on the diversity of the results and

1. http://www.rtreeportal.org.
2. http://www.cs.fsu.edu/~lifeifei/SpatialDataset.htm.
3. http://www.factual.com.

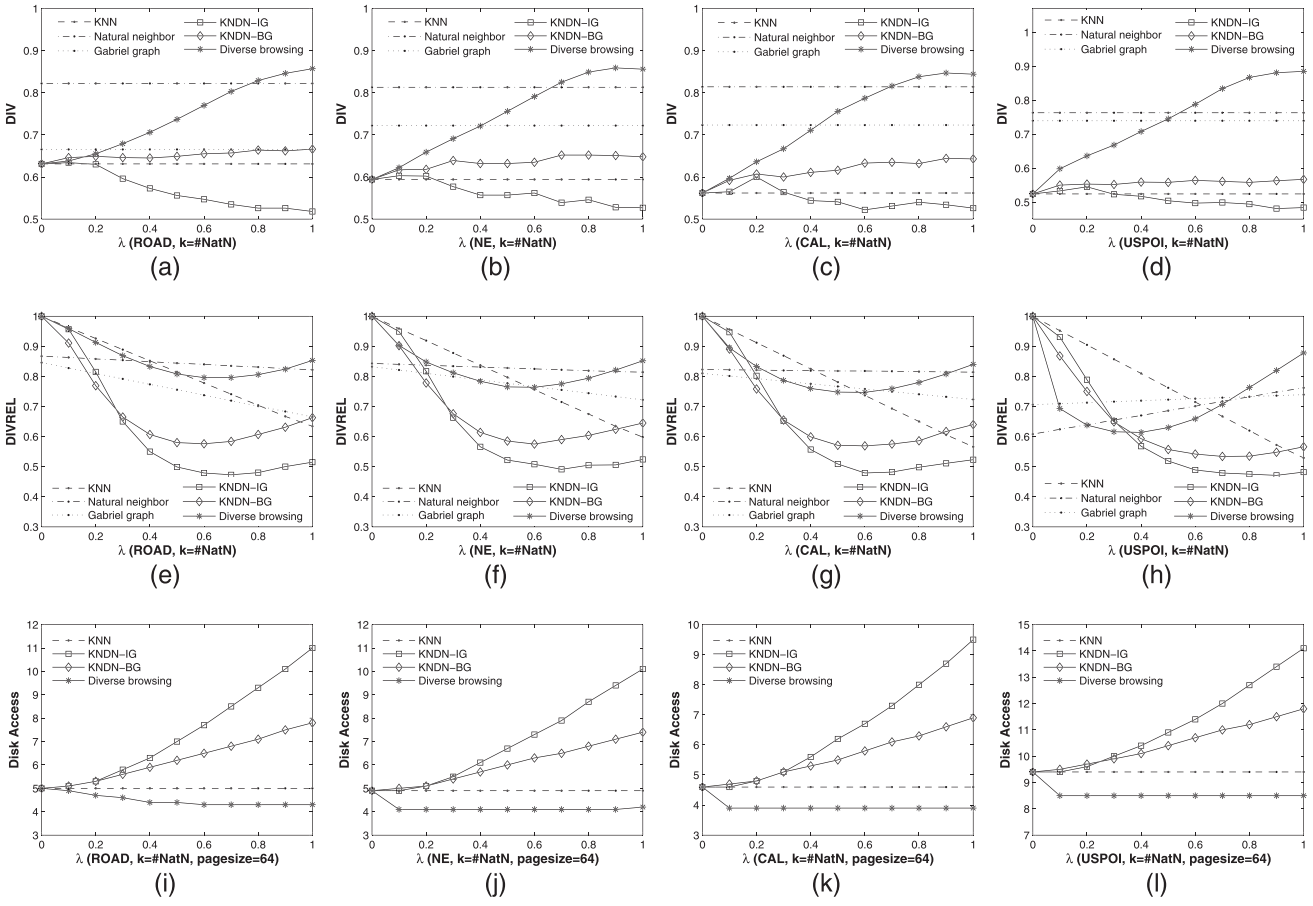

Fig. 7. Comparison of the algorithms on ROAD (a,e,i), NE (b,f,j), CAL (c,g,k), and USPOI (d,h,l) data sets. Aim of the methods is to maximize the diverse-relevance (DIVREL) of the results. Angular diversity (DIV) of the geometric approaches are stable because the natural and Gabriel neighbors of a point are fixed.

disk accesses. However, there are two other parameters that can affect the results and may be preferred to be specified manually: pruning angle θ_s and radius r_s.

We previously mentioned that the pruning angle parameter θ_s is empirically set to $2\pi/(k+\epsilon)$. To see how different θ_s values change the diversity and efficiency of diverse browsing, we experiment on ROAD data set for $\lambda = \{0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$. Fig. 9a shows that the maximum DIV is achieved around $\theta_s = 60 \sim 2\pi/(k+\epsilon)$ for $k = 6$ on 2D space. On the other hand, increasing this parameter will cause more pruning; as a result, the algorithm will have to investigate more MBRs. In order to keep the page accesses low, a smaller θ_s can be prefered (see Fig. 9b).

r_s parameter directly affects the pruning area. The points included in the result set become less relevant; although, the diversity may increase for some cases, especially when more diverse results are included instead of more relevant ones for small λ values (see Fig. 9c). Because the algorithm prunes more MBRs with increasing r_s, it will access more pages to find the result. The results show that our preference for this parameter as $r_s = 1 + \lambda$ is an optimal choice between the diversity and efficiency tradeoff (see Fig. 9d).

6.4 Discussions

Proposed geometric and an index-based diverse browsing methods have their own advantages in terms of preprocessing, querying, flexibility, and scalability. A summary of the proposed methods are given in Table 4.

TABLE 3
Diversity (DIV), Diverse-Relevance (DIVREL) and the Number of Disk Accesses (DA) for the USPOI Data Set for $\lambda = \{0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$

	KNN			Natural Neigh.		Gabriel Graph		KNDN-IG			KNDN-BG			Diverse Browsing		
λ	DIV	DIVREL	DA	DIV	DIVREL	DIV	DIVREL	DIV	DIVREL	DA	DIV	DIVREL	DA	DIV	DIVREL	DA
0.5	0.524	0.762	9.4	0.763	0.686	0.739	0.723	0.504	0.519	10.9	0.558	0.557	10.4	0.745	0.630	8.5
0.6	0.524	0.715	9.4	0.763	0.701	0.739	0.726	0.497	0.489	11.5	0.564	0.542	10.7	0.788	0.659	8.5
0.7	0.524	0.667	9.4	0.763	0.717	0.739	0.730	0.499	0.479	12.2	0.561	0.534	11.0	0.834	0.707	8.5
0.8	0.524	0.620	9.4	0.763	0.732	0.739	0.733	0.494	0.475	12.9	0.558	0.535	11.3	0.867	0.763	8.5
0.9	0.524	0.572	9.4	0.763	0.748	0.739	0.736	0.481	0.471	13.6	0.563	0.548	11.6	0.881	0.820	8.5
1.0	0.524	0.529	9.4	0.763	0.762	0.739	0.739	0.484	0.482	14.3	0.567	0.566	11.9	0.885	0.878	8.5
AVG	0.524	0.644	9.4	0.763	0.724	0.739	0.731	0.493	0.486	12.6	0.562	0.547	11.1	0.833	0.742	8.5

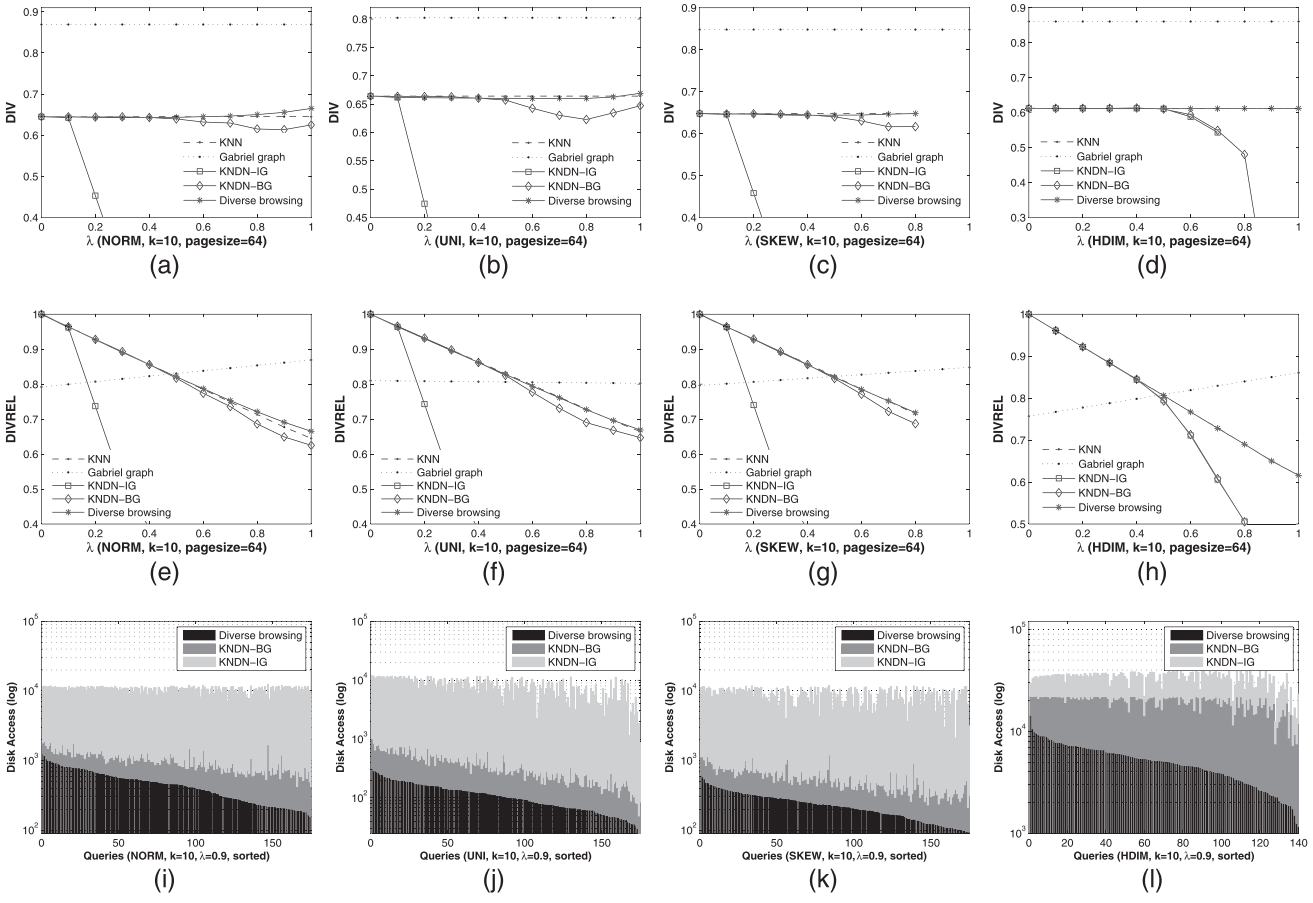

Fig. 8. Comparison of the algorithms on NORM (a,e,i), UNI (b,f,j), SKEW (c,g,k), and HDIM (d,h,l) data sets.

Preprocessing. The advantage of the index-based diverse browsing method is that it does not require any preprocessing and is ready to execute on any spatial database that use data partitioning, such as R-tree, R*-tree. On the other hand, geometric methods require to build DT or GG, which can be very complex depending on dimensionality and cardinality. As a result, we suggest index-based diverse browsing for a dynamic database, which is more likely to be based an index that handles insert, delete and update operations efficiently; whereas geometric methods for static databases, which would not cause DT and GG to be frequently calculated.

Querying. As mentioned before, the NatN-based method naturally returns a result set with a fixed number of points. If the user does not specify k and the purpose is to find a perfectly balanced diverse and relevant set of results (see DIVREL graphs at $\lambda \approx 0.5$), the NatN-based method is appropriate. However, diverse browsing is more suitable for diverse k-NN search, which requires exactly k results to be returned. If the query asks for at least k results, the GG-based method can be used as well.

Flexibility. We can investigate this property in two different ways. The first is the flexibility of setting the importance of diversity over relevance. Only diverse browsing method adjusts itself for various λ values, since the natural and Gabriel neighbors are fixed in a graph. Also note that among the index-based methods, only diverse browsing can provide diverse results when the user is willing to sacrifice relevancy (see Fig. 10). The second is the flexibility of incremental diverse browsing, where the user demands more search results. Both index-based diverse browsing and GG-based methods enable the retrieval of additional diverse results.

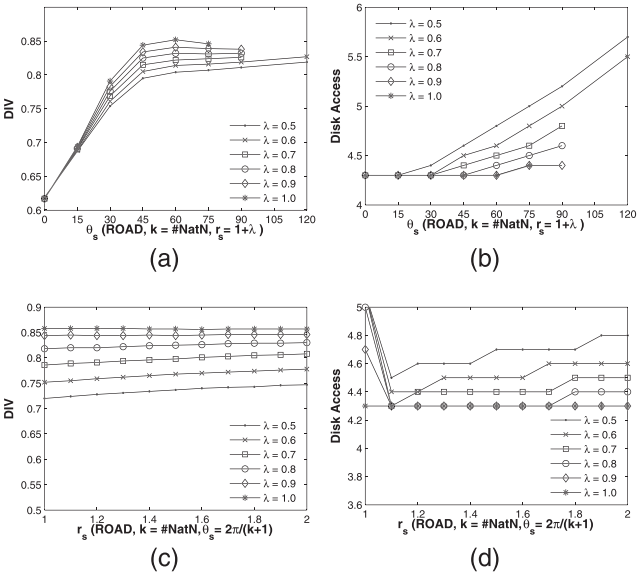

Fig. 9. Effects of the parameters θ_s and r_s on ROAD data set. (a) Maximum diversity is achieved when $\lambda = 1$ and θ_s is set to $2\pi/(k+\epsilon)$. (b) θ_s also increases the number of disk accesses. (c) Diversity increases with r_s for small λ values. (d) $r_s = 1 + \lambda$ is shown to be an optimal point between diversity and efficiency.

TABLE 4
Comparison of the Methods

Method	Results	Ordered	Prep.	Incr.
NatN	k'	by w_i	build DT	NO
GG	$\geq k$	by layer	build GG	YES
Diverse Browsing	k	by *mindivdist*	NO	YES

Scalability. For multidimensional spaces, the NatN-based method is intractable (see Section 4.4). Since data partitioning methods are shown to be inefficient for high dimensional data, the GG-based method can be preferred over index-based diverse browsing. Experiments (see Figs. 8g and 8h) show that the GG-based method is in fact very efficient ($\sim$10K versus $l_{GG}(k)$ page accesses) and effective (0.6 versus 0.85 DIVREL for $\lambda = 1$) in multidimensional data sets.

7 Conclusions

In this work, we investigate the diversification problem in multidimensional nearest neighbor search. Because diverse k-nearest neighbor search is conceptually similar to the idea of natural neighbors, we give a definition of *diversity* by making an analogy with the concept of natural neighbors and propose a natural neighbor-based method. Observing the limitations of NatN-based method in higher dimensional spaces, we present a Gabriel graph-based method that scales well with dimensionality. We also introduce an index-based diverse browsing method, which maintains a priority queue with *mindivdist* of the objects depending on both relevancy and diversity, and efficiently prunes non-diverse items and nodes in order to efficiently get the diverse nearest neighbors. To evaluate the diversity of a given result set to a query point, a measure that captures both the relevancy and angular diversity is presented. We experiment on spatial and multidimensional, real and synthetic data sets to observe the efficiency and effectiveness of proposed methods, and compare with index-based techniques found in the literature.

Results suggest that geometric approaches are suitable for static data, and index-based diverse browsing is for dynamic databases. Our index-based diverse browsing method performed more efficient than k-NN search with distance browsing on R-tree (in terms of the number of disk accesses) and more effective than other methods found in the literature (in terms of MMR). In addition, Gabriel graph-based method performed well in high dimensions, which can be investigated more and applied to other research fields where search in multidimensional space is required. Since there are numerous application areas of diverse k-nearest neighbor search, we plan to extend our method to work with different types of data and distance metrics.

Acknowledgments

The authors would like to thank Factual for sharing USPOI data, and the reviewers for their constructive comments. This research has been supported in part by US National Science Foundation (NSF) IIS-0546713.

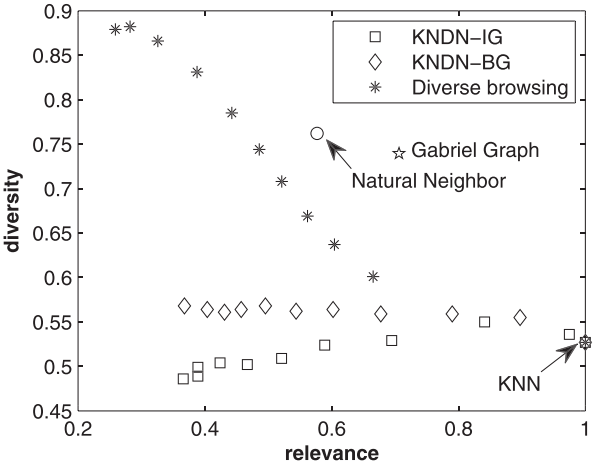

Fig. 10. Relevance versus diversity for USPOI data set. Each point corresponds to the average values of DIV and REL metrics for a run with a different λ ($\lim_{\lambda \to 0} REL \to 1$ for all index-based methods).

References

[1] P.K. Agarwal, L. Arge, and K. Yi, "I/O-Efficient Construction of Constrained Delaunay Triangulations," *Proc. 13th European Symp. Algorithms (ESA '05)*, pp. 355-366, 2005.

[2] F. Aurenhammer and R. Klein, "Voronoi Diagrams," *Handbook of Computational Geometry*, J. Sack and G. Urrutia, eds., chapter 5, pp. 201-290, Elsevier Science Publising, 2000.

[3] C.B. Barber, D.P. Dobkin, and H. Huhdanpaa, "The Quickhull Algorithm for Convex Hulls," *ACM Trans. Math. Software*, vol. 22, no. 4, pp. 469-483, 1996.

[4] J. Carbonell and J. Goldstein, "The use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries," *Proc. 21st Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '98)*, pp. 335-336, 1998.

[5] B. Carterette, "An Analysis of NP-Completeness in Novelty and Diversity Ranking," *Proc. Second Int'l Conf. Theory of Information Retrieval (ICTIR '09)*, pp. 200-211, 2009.

[6] B. Carterette, "An Analysis of NP-Completeness in Novelty and Diversity Ranking," *Information Retrieval*, vol. 14, no. 1, pp. 89-106, 2011.

[7] H. Chen and D.R. Karger, "Less is More: Probabilistic Models for Retrieving Fewer Relevant Documents," *Proc. 29th Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '06)*, pp. 429-436, 2006.

[8] C.L. Clarke, M. Kolla, G.V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon, "Novelty and Diversity in Information Retrieval Evaluation," *Proc. 31st Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '08)*, pp. 659-666, 2008.

[9] M. Drosou and E. Pitoura, "Diversity Over Continuous Data," *IEEE Data Eng. Bull.*, vol. 32, no. 4, pp. 49-56, Dec. 2009.

[10] R.A. Dwyer, "Higher-Dimensional Voronoi Diagrams in Linear Expected Time," *Proc. Fifth Ann. Symp. Computational Geometry (SCG '89)*, pp. 326-333, 1989.

[11] S. Fortune, *Voronoi Diagrams and Delaunay Triangulations*. 1992.

[12] K.R. Gabriel and R.R. Sokal, "A New Statistical Approach to Geographic Variation Analysis," *Systematic Zoology*, vol. 18, no. 3, pp. 259-278, 1969.

[13] J.C. Gower, "A General Coefficient of Similarity and Some of Its Properties," *Biometrics*, vol. 27, no. 4, pp. 857-871, 1971.

[14] M. Halvey, P. Punitha, D. Hannah, R. Villa, F. Hopfgartner, A. Goyal, and J.M. Jose, "Diversity, Assortment, Dissimilarity, Variety: A Study of Diversity Measures Using Low Level Features for Video Retrieval," *Proc. 31th European Conf. IR Research on Advances in Information Retrieval (ECIR '09)*, pp. 126-137, 2009.

[15] J.R. Haritsa, "The KNDN Problem: A Quest for Unity in Diversity," *IEEE Data Eng. Bull.*, vol. 32, no. 4, pp. 15-22, Dec. 2009.

[16] G.R. Hjaltason and H. Samet, "Distance Browsing in Spatial Databases," *ACM Trans. Database Systems*, vol. 24, no. 2, pp. 265-318, 1999.

[17] M. Isenburg, Y. Liu, J. Shewchuk, and J. Snoeyink, "Streaming Computation of Delaunay Triangulations," *ACM Trans. Graphics,* vol. 25, pp. 1049-1056, July 2006.
[18] A. Jain, P. Sarda, and J.R. Haritsa, "Providing Diversity in K-Nearest Neighbor Query Results," *Proc. Eighth Pacific-Asia Conf. Advances in Knowledge Discovery and Data Mining (PAKDD '04),* pp. 404-413, 2003.
[19] H. Ledoux and C. Gold, "An Efficient Natural Neighbour Interpolation Algorithm for Geoscientific Modelling," *Proc. 11th Int'l Symp. Spatial Data Handling (SDH '04),* pp. 23-25, 2004.
[20] X.-Y. Li, P.-J. Wan, Y. Wang, and O. Frieder, "Sparse Power Efficient Topology for Wireless Networks," *Proc. 35th Ann. Hawaii Int'l Conf. System Sciences (HICSS '02),* pp. 3839-3848, 2002.
[21] B. Liu and H.V. Jagadish, "Using Trees to Depict a Forest," *Proc. VLDB Endowment,* vol. 2, no. 1, pp. 133-144, 2009.
[22] D.B. Lomet, "Letter from the Editor-in-Chief," *IEEE Data Eng. Bull.,* vol. 32, no. 4, p. 1, Dec. 2009.
[23] D. Matula and R. Sokal, "Properties of Gabriel Graphs Relevant to Geographical Variation Research and the Clustering of Points on the Plane," *Geographical Analysis,* vol. 12, pp. 205-222, 1980.
[24] B.-U. Pagel, F. Korn, and C. Faloutsos, "Deflating the Dimensionality Curse Using Multiple Fractal Dimensions," *Proc. 16th Int'l Conf. Data Eng. (ICDE '00),* pp. 589-598, 2000.
[25] N. Roussopoulos, S. Kelley, and F. Vincent, "Nearest Neighbor Queries," *Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '95),* pp. 71-79, 1995.
[26] J.R. Shewchuk, "Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator," *Selected papers from the Workshop on Applied Computational Geometry, Towards Geometric Engineering,* pp. 203-222, 1996.
[27] R. Sibson, "A Brief Description of Natural Neighbor Interpolation," *Interpolating Multivariate Data,* vol. 21, pp. 21-36, 1981.
[28] Y. Tao, "Diversity in Skylines," *IEEE Data Eng. Bull.,* vol. 32, no. 4, pp. 65-72, Dec. 2009.
[29] E. Vee, U. Srivastava, J. Shanmugasundaram, P. Bhat, and S. Amer-Yahia, "Efficient Computation of Diverse Query Results," *Proc. 24th Int'l Conf. Data Eng. (ICDE '08),* pp. 228-236, 2008.
[30] C. Yu, L. Lakshmanan, and S. Amer-Yahia, "It Takes Variety to Make a World: Diversification in Recommender Systems," *Proc. 12th Int'l Conf. Extending Database Technology (EDBT '09),* pp. 368-378, 2009.
[31] C. Yu, L. Lakshmanan, and S. Amer-Yahia, "Recommendation Diversification Using Explanations," *Proc. 25th Int'l Conf. Data Eng. (ICDE '09),* pp. 1299-1302, 2009.
[32] C.-N. Ziegler and G. Lausen, "Making Product Recommendations More Diverse," *IEEE Data Eng. Bull.,* vol. 32, no. 4, pp. 23-32, Dec. 2009.
[33] C.-N. Ziegler, S.M. McNee, J.A. Konstan, and G. Lausen, "Improving Recommendation Lists through Topic Diversification," *Proc. 14th Int'l Conf. World Wide Web (WWW '05),* pp. 22-32, 2005.

Onur Kucuktunc received the BS and MSc degrees from the Department of Computer Engineering, Bilkent University, Turkey, in 2007 and 2009, respectively. Currently, he is working toward the PhD degree in the Department of Computer Science and Engineering at The Ohio State University (OSU). His research interests include similarity and diversity search, sentiment analysis and opinion retrieval, and bibliographic recommendation.

Hakan Ferhatosmanoglu received the PhD degree in computer science from the University of California, Santa Barbara, in 2001. He is an associate professor at Bilkent University, Turkey. He was with The Ohio State University (OSU) before joining Bilkent. His research interests include high-performance management and mining of multidimensional data, scientific databases, and bioinformatics. He received Career Awards from the US Department of Energy and National Science Foundation; IBM Faculty, OSU Lumley Research and OSU Large Interdisciplinary Research Awards.