Similar Document Detection and Electronic Discovery: So Many Documents, So Little Time

Michael Sperling†, Rong Jin*, Illya Rayvych†, Jianghong Li†, and Jinfeng Yi*
†Stroz Friedberg, NY 11581, USA
{msperling,irayvych,jli}@strozfriedberg.com
*Department of Computer Science and Engineering, Michigan State University, MI 48824, USA
{rongjin, yijinfen}@cse.msu.edu

ABSTRACT

The process of *Electronic Discovery* includes collecting, processing, and classifying large corpora of electronically stored information. It has spawned an entire support industry with annual revenue estimated at 40 billion dollars [1]. Although various automated classification methods, such as predictive coding, have been developed to reduce the cost of the classification process, it still heavily relies on manual review due to concern about acceptance by judges and regulators. In this work, we develop an efficient approach for *similar document detection* that aims to increase the efficiency of manual review, which has been estimated at 73% of total expenditures in the overall process [2]. Effective similar document detection can dramatically decrease these costs, as the number of similar documents in a typical electronic discovery corpus ranges between 25% and 50% [14]. We will also discuss how similar document detection can be used to complement and improve predictive coding methodologies. Given a query document, the proposed approach will efficiently detect the subset of documents within a large corpus whose text is similar to the query document's text. Compared to the existing techniques for similar document detection, the proposed approach is advantageous in two aspects: first, it is based on a lightweight representation of documents that can be efficiently extracted from the document texts; second, it casts the problem of similar document detection into a sequence of one-dimensional range searches that can be efficiently implemented using bi-section search. Our empirical study with a collection of 13 million documents verifies the effectiveness of the lightweight representation and the proposed range search algorithm for similar document detection.

[1] http://www.nelsonmullins.com/DocumentDepot/StrikingtheRightBalancetoControlLitigationSpend.pdf
[2] http://www.rand.org/content/dam/rand/pubs/monographs/2012/RAND_MG1208.pdf

1. INTRODUCTION

We first discuss the application of similar document detection in electronic discovery and its relationship to machine learning techniques such as predictive coding. We then discuss the existing techniques for similar document detection and their limitations, which motivates this work.

1.1 Electronic Discovery

The high cost of document review has led to the application of various machine learning based classification methods to the electronic discovery review process. One challenge to the use of machine learning in this domain is the nature of the document populations which are typically involved in an electronic discovery matter. These populations are characterized as follows:

- *Size*: The corpus may range from hundreds of thousands to hundreds of millions of documents.
- *Schedule*: The documents are very rarely made available as a single corpus. Typically, the documents arrive in waves over a period of months or years.
- *Diversity*: The documents in any single matter can range from email advertisements to highly technical manuals. They also often span a number of languages. Document collections arriving at different times may contain completely different subject matter.

Another challenge to the use of machine learning techniques has been concern about acceptance by judges and regulators. Although recent judicial decisions concerning predictive coding indicate a growing acceptance, and even a preference, amongst judges for technology assisted review, many lawyers are still loath to abandon manual review of documents for automated techniques. Lack of understanding, fear of sanctions due to incomplete productions, waiver of privilege, and even the potential loss of revenue to law firms all contribute to this reluctance to embrace predictive coding.

Due to the continued predominance of manual review, there is continued interest in utilizing tools such as clustering based on semantic similarity, identification of email threads, and identification of textually similar documents to improve the efficiency of manual review. This last approach of identifying textually similar documents (sometimes referred to as near duplicate detection) can have a huge impact on the cost of review. According to [14], $25 \sim 50\%$ of documents

reviewed in the process of electronic discovery are near duplicates.

It is also important to note that near duplicate detection can play a major role in the document review process even when predictive coding is employed. Predictive coding detects semantic similarity, while the goal of near duplicate detection is to find documents that are syntactically similar. Two examples of how duplicate detection can be employed in a predictive coding setting are:

- *Quality control of the training set*: Near duplicate detection can be employed to check the consistency of the training set and ensure that documents designated as responsive and not responsive are not near duplicates of each other. This is especially important when multiple reviewers are used to create the training set. When Stroz Friedberg applied this check to a training set created by a law firm in a recent predictive coding case, a significant number of the documents in the training set were found to violate this constraint.
- *Detection of significant small edits*: There are situations when small differences between documents can be highly significant. For instance, differences between drafts of a contract can be informative even if only a small number of words were changed and successive versions of an earnings report in which earnings are altered could indicate an attempt to manipulate financial results. Predictive coding will generally not differentiate between these highly similar versions of a document. The ability to retrieve all versions of the current document and highlight their differences using near duplicate detection can greatly simplify the process of finding these important variations.

This paper describes an approach to textually similar document detection (henceforth, similar document detection) which is both effective and novel in the industry, and which has been deployed in the Stroz Friedberg review application, resulting in significant improvement in reviewer productivity, improved consistency in predictive coding training sets, and detection of highly significant differences between similar documents.

1.2 Similar Document Detection

The objective of similar document detection is to efficiently find a subset of documents within a large collection that are textually similar to a given query document. Besides electronic discovery, it has found many applications in information retrieval [12], including web crawlers [6], data extraction [22], plagiarism [4], and spam detection [23, 10]. Many algorithms have been developed for similar document detection ([26] and references therein). They are usually divided into two stages. The first stage is to represent the content of documents by a vector. Given the vector representation of documents, the second stage is to map the vector representation to a low dimensional space to perform efficient search.

One shortcoming shared by most algorithms for similar document detection is that they require a "heavy" representation of documents, leading to high cost in both computation and storage space. The most popular representation for similar document detection is n-grams (i.e., n consecutive words, which is also referred to as shringles) [30, 1, 7, 6]. In this representation, the content of a document is represented by a binary vector. The size of the binary vector is the number of unique n-grams, with each entry of the vector indicating if a given n-gram appears in the document. Besides the n consecutive words, both n consecutive characters and sentences have also been used for similar document detection [29, 34]. In order to differentiate dissimilar documents, n has to be sufficiently large, making it computationally expensive to extract the n-gram features. The next popular representation for similar document detection is based on the classical vector space model [12, 10, 19, 23]. In this representation, each document is represented by a vector of word histograms weighted by a tf.idf scheme [12]. In [18, 11], the authors extended the vector space model from words to phrases in order to improve the detection accuracy. Both n-gram and vector space models represent documents by long vectors, requiring a high level of computation and storage space. Although hashing methods [7, 9] are applied to reduce the size of document representation and thus improve detection efficiency, extracting both vector representations for a large collection of documents is still computationally expensive.

In this work, we address this challenge by considering a lightweight representation of documents, in which the content of each document is based simply on the counts of characters and numbers. More specifically, we propose to represent each document by a vector of 62 dimensions, including 52 dimensions for both lower and upper case Latin characters and 10 dimensions for digits. To detect the documents similar to a given query document d_q, we apply a range search algorithm to efficiently identify the subset of documents whose vector representations are within a given range of the vector representation of d_q. Compared to the existing approaches for similar document detection, the key advantage of the proposed approach is its light vector representation, making it attractive both computationally and storage-wise. Despite the simplicity, we verify empirically that this light vector representation is sufficient for similar document detection when the difference between the query document and the matched ones is small. In addition, this approach allows the user to specify the degree of allowed dissimilarity in similar document detection by varying the threshold of range search. This feature is particularly important for electronic discovery as the criterion for documents to be similar varies from one situation to another.

Although the proposed vector representation is already in a significantly lower dimension than the existing approaches, efficiently performing a range search in a 62 dimensional space is still a challenging problem. Although many algorithms have been developed for search, none of them is designed for high dimensional range search. For instance, tree based approaches (e.g., KDtree [5]) are effective for range search in low dimensional space, but fail when dimensionality is high (over 20) [36]; many hashing based methods (e.g., Locality Sensitive Hashing [13]) are mostly designed for finding nearest neighbors, not for range search. The main technical contribution of this work is to develop and evaluate an

efficient algorithm for high dimensional range search. We prove the theoretical guarantee for the proposed algorithm and evaluate its empirical performance on a dataset of 13 million documents.

The rest of the paper is arranged as follows. Section 2 discusses the related work on similar document detection in electronic discovery and high dimensional search. Section 3 describes the proposed approach and its theoretical guarantee. Section 4 presents our empirical study for the proposed algorithm and compares it to the state-of-the-art approaches. Section 5 concludes with suggestions for further work.

2. RELATED WORK

We review the related work of similar document detection in electronic discovery, high dimensional search, and random projection.

Similar document detection in electronic discovery As the similar document population must be presented to reviewers in real time, retrieval of this population must be performed quickly. In order to satisfy this speed requirement, the most common technique for similar document detection in electronic discovery uses pre-built clusters to group similar documents around a centroid. This approach has a number of drawbacks:

- The threshold for "similarity" cannot be changed dynamically when similarity clusters are pre-built using a similarity threshold. Depending on their requirements, users may want to dynamically relax or tighten the criteria for document similarity.
- Due to the large number and variety of documents in a typical document population, it is usually impossible to construct well-separated clusters. Thus it is possible that documents in adjacent clusters are more similar to each other than to their respective centroids.
- The challenge to creating well-separated clusters is exacerbated by the fact that the entire document corpus does not arrive at once, but usually arrives in multiple waves. There are two ways to approach this complication. One way is to cluster each wave of documents separately. This approach has the drawback that similar documents are not identified across waves. The other approach is to merge newly arrived documents into an existing cluster structure by scanning existing centroids for an eligible cluster. This approach leads to degradation of cluster quality and more situations of documents in adjacent clusters being more similar to each other than to their respective centroids.

In contrast, the proposed method quickly identifies similar documents dynamically without pre-built clusters, even when the corpus contains millions of documents. Reviewers can vary the similarity threshold at will, thus retrieving tighter or looser similar document populations.

High dimensional search Given a query $\mathbf{q}$ and a distance threshold r, range search aims to efficiently identify the subset of data points from a database that are within a distance r from $\mathbf{q}$. When data points are represented by low dimensional vectors, a number of efficient solutions, based on pre-built index structures, have been proposed (e.g., KD-tree [5] or R-tree [3]). However, when the dimensionality is high, none of these approaches is more efficient than a simple brute-force search [36]. Several randomized approaches (e.g., Locality Sensitive Hashing (LSH) [9, 13, 2] and its variants [31, 28, 25, 32]) have been developed for high dimensional range search. The main limitation of these approaches is that they are mostly designed for a fixed range. In contrast, this work addresses the general problem of range search where the threshold r is a variable that will be determined by users. Although randomized KD-tree [35] shows encouraging empirical performance for range search, its theoretical guarantee is unknown, making it an unsatisfactory approach. In addition, the size of KD tree grows linear in the number of data points, and a significant amount of memory is needed to hold KD tree in the main memory in order to perform efficient search, making it potentially inefficient for very large data sets.

Besides range search, another important search problem is nearest neighbor (NN) search. Many randomized algorithms have been developed for high dimensional NN search. Among them, the hashing methods have shown promising performance. The most notable hashing method for high dimensional NN search is based on Locality Sensitive Hashing [9, 13, 2]. Many variants of LSH have been developed to speed up the NN search, including entropy-based LSH [31], Multi-Probe LSH [28], and kernelized LSH [25, 32]. One drawback of LSH and its variants is that in order to achieve both high precision and recall, they often require many hashing tables and long codewords, leading to high computational cost in indexing and significant overhead in searching. The data dependent hashing algorithms address this limitation by significantly reducing code length and the number of hashing tables. Example algorithms for data dependent hashing include spectral hashing [37], self-taught hashing [39], semantic hashing [33], Laplacian co-hashing [38], anchor graph hashing [27] and random maximum margin hashing [21]. Since these approaches are developed for NN search, they are in general not suitable for range search.

Random projection The proposed work is also related to the random projection based approaches [20]. The key idea of these approaches is to convert a high dimensional range search problem into a low dimensional one. It first randomly projects data points into a low dimensional space and then performs range search over the projected space using the conventional approaches (e.g., KD-tree). The theoretical foundation of these works is based on the Johnson Lindenstrauss Theorem [20], which claims that the pairwise distance is well preserved through random projection. Random projection has been applied to several applications, including anomaly detection [16], classification [17] and clustering [15]. The main drawback of these approaches is that a large number of random projections is needed to preserve the pairwise distance with high accuracy, and as a consequence, the resulting search problem is no longer low dimensional.

3. EFFICIENT ALGORITHMS FOR HIGH DIMENSIONAL RANGE SEARCH

We first describe the problem of range search, and then present our algorithm and its theoretical guarantee.

Algorithm 1 Efficient Range Search using Gaussian Random Variables

Input:
- $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N)$: the database
- $r > 0$: specified range
- $\tau \geq 1$: threshold factor
- m: the number of one dimension range searches
- $\mathbf{q}$: the query point

//*Offline processing*
Random sample $U = (\mathbf{u}_1, \ldots, \mathbf{u}_m)$, where $\mathbf{u}_k \sim \mathcal{N}(0, I/d), k \in [m]$.
for $i = 1, \ldots, N$ **do**
  Compute $\mathbf{z}_i = \mathbf{x}_i^\top U$
end for
//*Online processing*
Compute the projection $\mathbf{z}^q = (z_1^q, \ldots, z_m^q)^\top = \mathbf{q}^\top U$ for query $\mathbf{q}$
for $k = 1, 2, \ldots, m$ **do**
  if $k = 1$ **then**
    Compute the set $\mathcal{D}_1(r, \mathbf{q})$ as $\mathcal{D}_1(r, \mathbf{q}) = \left\{ i \in [N] : |z_{i,k} - z_k^q| \leq \tau \frac{r}{\sqrt{d}} \right\}$
  else
    Update the set $\mathcal{D}_k(r, \mathbf{q})$ as $\mathcal{D}_k(r, \mathbf{q}) = \left\{ i \in \mathcal{D}_{k-1}(r, \mathbf{q}) : |z_{i,k} - z_k^q| \leq \tau \frac{r}{\sqrt{d}} \right\}$
  end if
end for
Output the set $\mathcal{D}_m(r, \mathbf{q})$.

Let $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ be a collection of vectors (i.e., the database), where $\mathbf{x}_i \in \mathbb{R}^d$ and $d \gg 1$ is the dimension of the space. Let $\mathbf{q} \in \mathbb{R}^d$ be a query point. The goal of range search is to find a subset of data points in $\mathcal{D}$ that are within distance r from $\mathbf{q}$, where r is the range specified by the user. Define $\mathcal{D}(r, \mathbf{q})$ as the subset of data points in $\mathcal{D}$ that are within distance r from the query $\mathbf{q}$, i.e.,

$$\mathcal{D}(r, \mathbf{q}) = \{\mathbf{x} \in \mathcal{D} : |\mathbf{x} - \mathbf{q}|_2 \leq r\}$$

Let $m(r, \mathbf{q}) = |\mathcal{D}(r, \mathbf{q})|$ be the number of data points within the given range, and $A(r, \mathbf{q}) = \max_{\mathbf{x} \in \mathcal{D}(r, \mathbf{q})} |\mathbf{x} - \mathbf{q}|_\infty$ be the maximum difference in any attributes between the query point and the data points within the given range. Evidently, we have $m(r, \mathbf{q}) \leq N$ and $A(r, \mathbf{q}) \leq r$. In general, we assume r is sufficiently small such that $m(r, \mathbf{q})$ has a weak dependence on N. For the simplicity of our analysis, we assume both the data points in $\mathcal{D}$ and the query $\mathbf{q}$ have bounded length, $|\mathbf{x}|_2 \leq 1, \forall \mathbf{x} \in \mathcal{D}$ and $|\mathbf{q}| \leq 1$.

To speed up the search, we propose to convert high dimensional range search into a sequence of one-dimensional range searches. More specifically, we randomly sample multiple vectors from a Gaussian distribution, denoted by $\mathbf{u}_1, \ldots, \mathbf{u}_m$. For each randomly sampled vector $\mathbf{u}_i$, we project both the query $\mathbf{q}$ and the data points in $\mathcal{D}$ along the direction of $\mathbf{u}_i$, and find the subset of data points in $\mathcal{D}$ whose projections are within a certain threshold ρ (not r, but dependent on r) of the query $\mathbf{q}$, denoted by $\mathcal{D}_i$. To implement efficient one dimensional range search, we rank the projection of data points in $\mathcal{D}$ along the direction of $\mathbf{u}_i$ in a descending order and perform a bi-section search to find the subset of data points whose projections are within a given range. The intersection of the data points returned by all of the one dimensional range searches is used to form the final result, i.e., $\mathcal{D}(r, \mathbf{q}) = \cap_{i=1}^m \mathcal{D}_i$. Algorithm 1 gives the detailed steps for the proposed algorithm. We note that although the proposed algorithm is also based on random projection, it is different from the existing approaches in that it does not try to approximate the pairwise distance by random projection. Instead, it tries to approximate the binary decision, i.e. whether a data point is within a certain range of a query, by a sequence of binary decisions based on one dimensional projections of both the data point and the query.

As the first step of our analysis, we show that a single one dimension range search with appropriately chosen threshold ρ is able to ensure a high recall, namely that with a high probability, a data point within the specified range will be returned by the one dimension range search. The theorem below gives the performance guarantee for one-dimensional range search.

THEOREM 1. *Let $\mathbf{u}$ be a vector randomly sampled from $\mathcal{N}(0, I/d)$. With a probability $1 - \delta - \frac{c \ln d}{d^3}$, we have*

$$\sup_{\mathbf{x} \in \mathcal{D}(r, \mathbf{q})} |(\mathbf{x} - \mathbf{q})^\top \mathbf{u}| \leq \frac{r}{\sqrt{d}} \left(C_1 \ln \frac{2m(r, \mathbf{q})}{\delta} + C_2 \sqrt{\ln \frac{2m(r, \mathbf{q})}{\delta}} \right)$$

where

$$C_1 = 6K_2, \quad C_2 = \sqrt{6K_2 + \frac{c \ln d}{d^2}} \tag{1}$$

The proof can be found in Appendix A.

Remark. As indicated by Theorem 7, when the dimensionality d is very high, with a high probability, $|(\mathbf{x} - \mathbf{q})^\top \mathbf{u}|$ is upper bounded by $O(|\mathbf{x} - \mathbf{q}|/\sqrt{d})$, implying that most of the data points within the given range of a query will be returned by one-dimensional range search (i.e. high recall) provided a sufficiently large threshold. The problem with Theorem 1 is that it does not provide a lower bound $|(\mathbf{x} - \mathbf{q})^\top \mathbf{u}|$. Without a lower bound, the one dimensional range search may result in a high recall but a very poor precision as many of the returned data points can be far away from the query $\mathbf{q}$. We address this problem by performing multiple one dimensional range searches, and only the data points found by all one dimensional range searches will be returned.

First, we extend the result from Theorem 1 to multiple one-dimension range searches by using the union bound.

COROLLARY 2. *Let $\mathbf{u}_1, \ldots, \mathbf{u}_m$ be m vectors randomly sampled from $\mathcal{N}(0, I/d)$. With a probability $1 - m\delta - \frac{c \ln d}{d^3}$, we have*

$$\sup_{\mathbf{x} \in \mathcal{D}(r, \mathbf{q})} \max_{1 \leq k \leq m} |(\mathbf{x} - \mathbf{q})^\top \mathbf{u}_k| \leq \frac{r}{\sqrt{d}} \left(C_1 \ln \frac{2m(r, \mathbf{q})}{\delta} + C_2 \sqrt{\ln \frac{2m(r, \mathbf{q})}{\delta}} \right)$$

where C_1 and C_2 are defined (1).

As indicated by the above corollary, if we set $\tau = \tau_0$ in Algorithm 1, where τ_0 is given by

$$\tau_0 = C_1 \ln \frac{2m(r,\mathbf{q})}{\delta} + C_2\sqrt{\ln \frac{2m(r,\mathbf{q})}{\delta}} \quad (2)$$

then, with a high probability, *all* documents within distance r from the query document will be returned by Algorithm 1, i.e., we have a high recall for the returned documents.

THEOREM 3. *Assume m is sufficiently large, i.e.,*

$$m \geq 64K_1\left(C_1 \ln \frac{2}{\delta} + C_2\sqrt{\ln \frac{2}{\delta}}\right)$$

where C_1 and C_2 are defined in (1). Then, with a probability $1-(m+1)\delta - \frac{mc\ln d}{d^3}$, we have

$$\max_{1\leq k\leq m} |(\mathbf{x}-\mathbf{q})^\top \mathbf{u}_k| \geq \frac{|\mathbf{x}-\mathbf{q}|}{2\sqrt{d}}$$

The proof can be found in Appendix B.

Remark. As indicated by Theorem 3, if we set $\tau = 1/\sqrt{2}$ in Algorithm 2, then, given a sufficiently large number of random projections, there is a high probability that none of the documents returned by Algorithm 1 will have a distance larger than or equal to r, implying that we will have a high precision for the returned documents. By combining the results from Corollary 2 and Theorem 3, we have, with a high probability,

$$\frac{|\mathbf{x}-\mathbf{q}|}{2\sqrt{d}} \leq \max_{1\leq k\leq m} |(\mathbf{x}-\mathbf{q})^\top \mathbf{u}_k| \leq \frac{\tau_0|\mathbf{x}-\mathbf{q}|}{\sqrt{d}}$$

provided m is sufficiently large.. Hence, by varying τ in Algorithm 2 between $1/\sqrt{2}$ and τ_0, we will be able to make an appropriate tradeoff between high precision and high recall.

In cases when the dimensionality is high, sampling $\mathbf{u}_1,\ldots,\mathbf{u}_m$ from a Gaussian distribution can be computationally expensive. We address this challenge by replacing Gaussian random variables in Algorithm 1 with Rademacher random variables [3]. More specifically, to construct each random vector $\mathbf{u}_i$, we first sample d Randemacher variables $\sigma_i^1,\ldots,\sigma_i^d$, with $\Pr(\sigma_i^k=-1)=\Pr(\sigma_i^k=+1)=1/2$, and form $\mathbf{u}_i$ as $\mathbf{u}_i = (\sigma_i^1,\ldots,\sigma_i^d)$. The details are given in Algorithm 2. Below we will show that the Rademacher variable based approach yields similar performance as the one based on the Gaussian variables. Similar to the analysis for Algorithm 1, we first analyze the performance of one dimensional range search based on the Rademacher random variable.

THEOREM 4. *Let $\mathbf{u} = \frac{1}{\sqrt{d}}(u_1,\ldots,u_d)$ be a random vector with u_i drawn from a Bernoulli distribution $\Pr(u_i=1) = \Pr(u_i=-1)=1/2$. Then, with a probability $1-\delta$, for a fixed data point $\mathbf{x}$, we have*

$$\sup_{\mathbf{x}\in\mathcal{D}(r,\mathbf{q})} |(\mathbf{x}-\mathbf{q})^\top \mathbf{u}| \leq \frac{r}{\sqrt{d}}\left(2\ln\frac{2m(r,\mathbf{q})}{\delta} + \sqrt{2\ln\frac{2m(r,\mathbf{q})}{\delta}}\right)$$

The proof can be found in Appendix C. Compared to Theorem 1, we found that Theorem 4 is slightly stronger with a larger probability guarantee.

We then show the performance guarantee for multiple one-dimensional searches based on Rademacher random variables.

THEOREM 5. *Let $U = \frac{1}{\sqrt{d}}(\mathbf{u}_1,\ldots,\mathbf{u}_m)$ be random variables with $U_{i,j}$ having equal probability to be $+1$ and -1. With a probability at least $1-2m/[d^3]$, we have*

$$\sup_{\mathbf{x}\in\mathcal{D}(r,\mathbf{q})} \max_{1\leq k\leq m} |(\mathbf{x}-\mathbf{q})^\top \mathbf{u}_k| \leq \frac{r}{\sqrt{d}}\left(2\ln\frac{2m(r,\mathbf{q})}{\delta} + \sqrt{2\ln\frac{2m(r,\mathbf{q})}{\delta}}\right)$$

When m is sufficiently large, i.e.,

$$m \geq 64K_1\left(2\ln\frac{2}{\delta} + \sqrt{2\ln\frac{2}{\delta}}\right)$$

Algorithm 2 Efficient Range Search using Rademacher Random Variables

Input:
- $\mathcal{D} = \{\mathbf{x}_1,\ldots,\mathbf{x}_N\}$: the database
- $r > 0$: specified range
- $\tau \geq 1$: threshold factor
- m: the number of one dimension range searches
- $\mathbf{q}$: the query point

//*Offline processing*
Random sample $U = \frac{1}{\sqrt{d}}(\mathbf{u}_1,\ldots,\mathbf{u}_m)$, where $U_{i,j}$ is a Rademacher variable with $\Pr(U_{i,j}=-1) = \Pr(U=+1) = 1/2$.
for $i = 1,\ldots,N$ **do**
  Compute $\mathbf{z}_i = \mathbf{x}_i^\top U$
end for
//*Online processing*
Compute the projection $\mathbf{z}^q = (z_1^q,\ldots,z_m^q)^\top = \mathbf{q}^\top U$ for query $\mathbf{q}$
for $k = 1,2,\ldots,m$ **do**
  if $k = 1$ **then**
    Compute the set $\mathcal{D}_1(r,\mathbf{q})$ as $\mathcal{D}_1(r,\mathbf{q}) = \left\{i\in[N] : |z_{i,k}-z_k^q| \leq \tau\frac{r}{\sqrt{d}}\right\}$
  else
    Update the set $\mathcal{D}_k(r,\mathbf{q})$ as $\mathcal{D}_k(r,\mathbf{q}) = \left\{i\in\mathcal{D}_{k-1}(r,\mathbf{q}) : |z_{i,k}-z_k^q| \leq \tau\frac{r}{\sqrt{d}}\right\}$
  end if
end for
Output the set $\mathcal{D}(r,\mathbf{q})$.

[3] A Rademacher random variable has equal probability to be -1 and $+1$

Then, with a probability $1-(m+1)\delta$*, we have*

$$\max_{1\le k\le m} |(\mathbf{x}-\mathbf{q})^\top \mathbf{u}_k| \ge \frac{|\mathbf{x}-\mathbf{q}|}{2\sqrt{d}}$$

The proof of Theorem 5 is similar to that of Theorem 3. We skip the proof due to space constraints. Compared to Theorem 3, the theoretical guarantee provided by Theorem 5 is almost identical except for the constant C_1 and C_2.

4. EXPERIMENT

In this experiment, we first demonstrate the effectiveness of the proposed light vector representation for similar document detection. We then evaluate both the efficiency and effectiveness of the proposed algorithm for similar document detection. The successful outcome of this experiment validates the use of this approach in the Stroz Friedberg review application.

4.1 Dataset

The collection used in our study consists of $13,228,105$ documents drawn from an actual, and consequently not publically available, e-discovery project. The size of documents in this collection varies from 1 character to $51,034,295$ characters, and the average document length is $12,521$ characters. The documents included in this collection are very diverse, including an English dictionary, customer lists, recipes, parent teacher association meeting minutes, project management reports, contracts, and descriptions of clinical drug trials. We utilized this collection rather than a publically available collection as we do not believe there exists a publically available collection of the size and diversity necessary to realistically test performance of our proposed algorithm. To evaluate the performance of the proposed approach, we select the query documents by randomly sampling 0.01% of documents in the collection that have more than 20 characters, which leads to $1,283$ query documents.

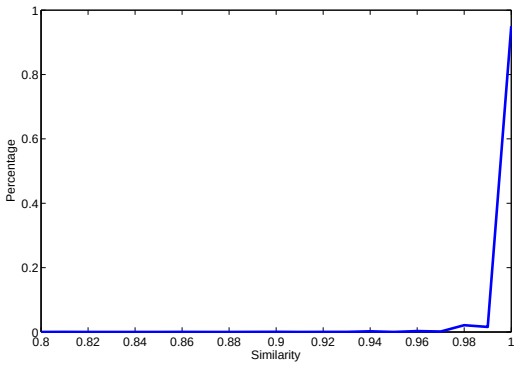
4.2 Experiment 1: validating the lightweight vector representation

To evaluate the effectiveness of the proposed vector representation for similar document detection, for each query document $\mathbf{q}$, we first find the matched documents $\{\mathbf{x}_i\}$ that satisfy the condition

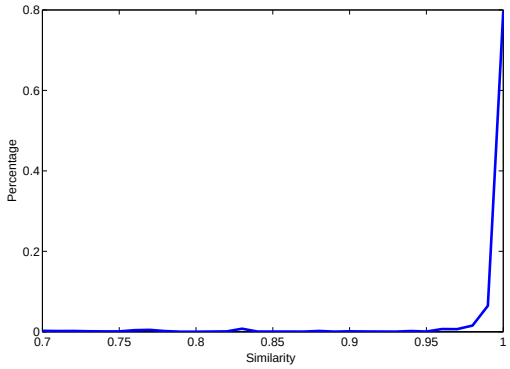
$$|\mathbf{x}-\mathbf{q}|_2 \le \gamma |\mathbf{q}|_2, \tag{3}$$

where $\{\mathbf{x}_i\}$ and $\mathbf{q}$ are the lightweight vector representations of documents and the query, respectively, and γ is set to 0.025 which is a reasonable coefficient based on our experience. Note that the distance threshold in (3) is set to $\gamma|\mathbf{q}|_2$, thus dependent on the length of the query document. This is more appropriate than a constant threshold because the allowed difference between two similar documents should depend on the size of the documents. Given the matched documents found by the criterion in (3), we then measure the similarity between the query document d_q and each matched document d, based on the edit distance $\text{dist}(d_q, d)$ between their texts, i.e.,

$$\text{sim}(d_q, d) = 1 - \frac{\text{dist}(d_q, d)}{\max(|d_q|, |d|)}$$



(a) $\gamma = 0.025$



(b) $\gamma = 0.05$

Figure 1: The distribution of similarity between query documents and the matched documents found by the proposed lightweight representation

where $|d_q|$ and $|d|$ represents the number of characters in d_q and d, respectively. Our hypothesis is that if the lightweight vector representation is sufficient for similar document detection, we would expect to observe on average a high similarity between query documents and matched ones.

Figure 1(a) shows the distribution of similarity averaged over $1,283$ query documents. We observe that around 99% of matched documents found by the proposed vector representation have similarity $\ge 95\%$. To further validate the proposed vector representation for similar document detection, we relax coefficient γ in (3) to 0.05 and show the similarity distribution in Figure 1(b). We again observe that close to 90% of the matched documents found have more than 90% similarity. Based on these results, we conclude that the proposed lightweight vector representation is sufficient for similar document detection when the difference between similar documents is small.

4.3 Experiment 2: evaluating the proposed algorithm for similar document detection

As Algorithms 1 and 2 share the same idea and have similar theoretical guarantees, we only evaluate the performance of Algorithm 2 due to its simplicity. Below, we first discuss the implementation of our algorithm, and then present the

experimental results.

Implementation Similar to Experiment 1, we set the threshold r in Algorithm 2 to be $r = \gamma|\mathbf{q}|_2$. For parameter τ in Algorithm 2, we set it to be $\tau = \tau_0$ as defined in (2), with $m(r, \mathbf{q}) = 10$ (best guess based on experience [4]), $\delta = 0.1$, and $C_1 = C_2 = 1$. Since our data is stored in an Oracle database, we implement our algorithm using the PL/SQL language. We pre-compute the random projections for all the documents in the collection. Our timing experiments were run on an Oracle server (version 11g) with 4 cores (2 GHz per core) and a total of 24G memory. In order to eliminate the effects of other users and database caching during the timing experiments, the procedure running the timing experiments had exclusive use of the server and the database global cache was cleared after each document in the test set was processed.

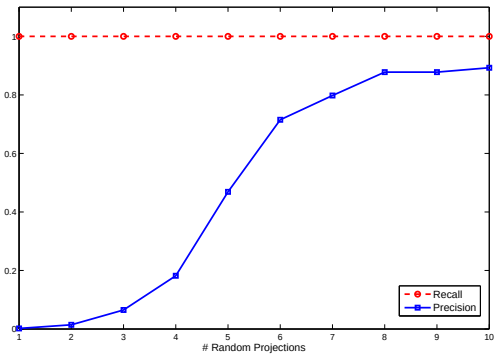
Results for searching accuracy We first evaluate the performance of the proposed approach by precision and recall. Given a query document $\mathbf{q}$, let $\mathcal{D}(r, \mathbf{q})$ be the subset of the documents within the distance r from $\mathbf{q}$, and $\widehat{\mathcal{D}}(r, \mathbf{q})$ be the subset of documents returned by Algorithm 2. The precision and recall is defined as

$$\text{Prec} = \frac{|\mathcal{D}(r, \mathbf{q}) \cap \widehat{\mathcal{D}}(r, \mathbf{q})|}{|\widehat{\mathcal{D}}(r, \mathbf{q})|}, \quad \text{Recall} = \frac{|\mathcal{D}(r, \mathbf{q}) \cap \widehat{\mathcal{D}}(r, \mathbf{q})|}{|\mathcal{D}(r, \mathbf{q})|}$$

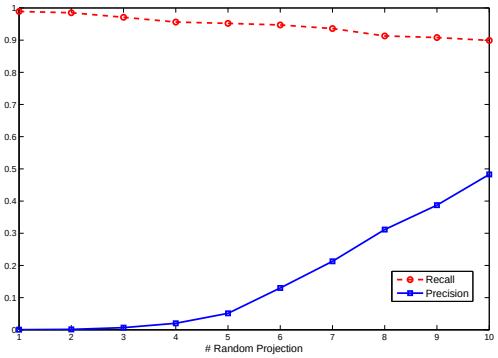
Figure 2(a) shows the precision and recall curves as we increase the number of random projections. We observe that as we increase the number of random projections, the recall remains almost unchanged at 1, while the precision improves from less than 0.2% to almost 90%. To further validate the proposed approach, we relax γ to 0.05 and show the precision and recall curves in Figure 2(b). We observe a small decrease in recall and a significant improvement in precision as we increase the number of random projections. Both results verify that the proposed algorithm is effective for high dimensional range search provided that the difference between similar documents is specified to be small.

Results for searching efficiency In order to present reviewers with only truly similar documents, we add to Algorithm 2 a post procedure that removes any returned document if its distance to the query document is larger than the given threshold. As a result, the runtime includes two components: the time to perform the range search using Algorithm 2, and the time used to check if each returned document is within distance $\gamma|\mathbf{q}|_2$ from the query document $\mathbf{q}$. We note that by increasing the number of random projections, we can significantly improve the precision and thus reduce the time spent checking if the returned documents are within the given range of the query, but at the price of increasing the time for performing the range search. Based on our experience, we found that setting the number of random projections to 8 seems to be a good tradeoff between the two components of runtime. The results for using 8 random projections are given in Table 1. We observed that compared to the exhaustive search (the last column in Table 1), the time used to find the matched documents is reduced dramatically

[4]Although the number may be very different from the true number of matched documents, it will have little impact on our algorithm since it appears in the form of $\ln m(r, \mathbf{q})$.



(a) $\gamma = 0.025$



(b) $\gamma = 0.05$

Figure 2: Precision (solid curve) and recall (dotted curve) for Algorithm 2 with varied numbers of random projections.

by the proposed algorithm.

While Algorithm 2 provides excellent precision and recall, the average runtime to find similar documents is still too long for real time response to reviewers, due to the algorithm's implementation in our application. The document vector representations and random projections are stored in an Oracle database, and the sequential range searches on the random projections are accomplished via a SQL statement with a WHERE clause of the form $\bigcap_{1 \le i \le m} |(\mathbf{x}-\mathbf{q})^\top \mathbf{u}_i| \le \rho$. Due to Oracle's indexing structure, the speed of this statement is heavily dependent on the number of documents that satisfy the first projection range. For our test set, the average number of documents that satisfied the first projection range was $263,883$ when $\gamma = 0.025$ and $525,264$ when $\gamma = 0.05$, which caused a significant delay in obtaining the similar document set. We therefore introduced a heuristic to reduce the number of documents in the first projection range by first filtering on 2 additional one dimensional ranges. The first one-dimensional filter returns the documents satisfying the condition $||\mathbf{x}|_2 - |\mathbf{q}|_2| \le \gamma|\mathbf{q}|_2$ and the second filter returns the documents satisfying the condition $||\mathbf{x}|_1 - |\mathbf{q}|_1| \le \gamma|\mathbf{q}|_2$. Introducing these filters reduces the average number of documents satisfying the new first range search to $56,591$ when $\gamma = 0.025$ and to $113,739$ when $\gamma = 0.05$. While these filters have poor precision on their

Table 1: Running time (seconds) for Algorithm 2 (using 8 random projections) and exhaustive search

Time (second)	Alg. 2	Alg. 2 + two filters	Two filters	Exhaustive search
$\gamma = 0.025$	2.57	0.48	2.93	5452.80
$\gamma = 0.05$	4.00	0.95	11.43	5452.80

Table 2: Precision and recall for Algorithm 2 (using 8 random projections) with and without additional two one-dimensional filters

γ		Alg. 2	Alg. 2 + two filters	Two filters	KD-tree
0.025	Recall	0.999	0.992	0.992	0.960
	Prec	0.912	0.956	0.021	N/A
0.05	Recall	0.981	0.949	0.964	0.940
	Prec	0.312	0.542	0.006	N/A

own (Table 2), using them in conjunction with Algorithm 2 reduces the average runtime to less than 1 second (Table 1) with a small degradation in recall (Table 2). For completeness, we also include in Table 1 and 2 the results of the two additional filters by themselves. We conclude that the additional two filters are effective in improving both efficiency and search accuracy.

We finally compare the proposed approach to the randomized KD-tree [35], a state-of-the-art approach for high dimensional range search. We apply the FLANN library [5] to construct a randomized KD-tree for the entire document collection where each document is represented by its 62 tuple vector. It takes over ten hours to construct the KD-tree, and the resulting index structure consumes roughly twice the storage space as the original data. The recall values of KD-tree are given in Table 2. We did not include the precision of KD-tree in this comparison because the program has a post procedure that removes any returned document if its distance to the query document is larger than the given threshold. As a result, its precision is always 1. It is clear that the proposed approach, despite its simplicity, performs slightly better than KD-tree in recall without incurring the additional storage and computational costs of KD-tree.

5. CONCLUSION

In this work, we study the problem of similar document detection in the domain of electronic discovery. We develop a lightweight representation and an algorithm for similar document detection based on efficient high dimensional range search. Our empirical study with a collection of over 13 million documents shows encouraging results of the proposed algorithm in both searching accuracy and searching efficiency. In the future, we plan to improve the efficiency and the effectiveness of the proposed approach by exploring data dependent sampling approaches, such as sampling the random vectors based on the covariance structure of the data. We also plan to expand the lightweight document representation to further improve its efficiency and accuracy for similar document detection.

[5] `http://www.cs.ubc.ca/~mariusm/index.php/FLANN/FLANN`

6. REFERENCES

[1] Evaluating topic-driven web crawlers. In *Proceedings of the 24th Annual International ACM SIGIR Conference On Research and Development in Information Retrieval*, pages 241–249, 2001.

[2] A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Commun. ACM*, 51(1):117–122, 2008.

[3] L. Arge, M. Berg, H. Haverkort, and K. Yi. The Priority R-tree: a practically efficient and worst-case optimal R-tree. In *SIGMOD*, 2004.

[4] B. S. Baker. On finding duplication and near-duplication in large software systems. In *Proceedings of the Second Working Conference on Reverse Engineering*, WCRE '95, 1995.

[5] J. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18:509–517, 1975.

[6] K. Bharat, A. Z. Broder, J. Dean, and M. R. Henzinger. A comparison of techniques to find mirrored hosts of the WWW. *J American Society for Information Science (JASIS)*, 51(12):1114–1122, 2000.

[7] A. Z. Broder, M. Charikar, A. M. Frieze, and M. Mitzenmacher. Min-wise independent permutations. *Journal of Computer and System Sciences*, 60:327–336, 1998.

[8] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *CoRR*, abs/0805.4471, 2008.

[9] M. Charikar. Similarity estimation techniques from rounding algorithms. In *STOC*, pages 380–388, 2002.

[10] A. Chowdhury, O. Frieder, D. A. Grossman, and M. C. McCabe. Collection statistics for fast duplicate document detection. *ACM Trans. Inf. Syst.*, 20(2):171–191, 2002.

[11] J. W. Cooper, A. R. Coden, and E. W. Brown. Detecting similar documents using salient terms. In *Proceedings of the eleventh international conference on Information and knowledge management*, CIKM '02, pages 245–251, 2002.

[12] W. Croft, D. Metzler, and T. Strohmann. *Search Engines: Information Retrieval in Practice*. Pearson, London, England, 2010.

[13] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Symposium on Computational Geometry 2004*, pages 253–262, 2004.

[14] Equivio, 2006.

[15] X. Z. Fern and C. E. Brodley. Random projection for high dimensional data clustering: A cluster ensemble approach. In *Proceedings of the Twentieth International Conference (ICML 2003)*, pages 186–193, 2003.

[16] J. E. Fowler and Q. Du. Anomaly detection and reconstruction from random projections. *IEEE Transactions on Image Processing*, 21(1):184–195, 2012.

[17] J. E. Fowler, Q. Du, W. Zhu, and N. H. Younan. Classification performance of random-projection-based dimensionality reduction of hyperspectral imagery. In *IEEE International Geoscience and Remote Sensing Symposium, IGARSS 2009*, pages 76–79, 2009.

[18] K. M. Hammouda and M. S. Kamel. Efficient phrase-based document indexing for web document clustering. *IEEE Trans. Knowl. Data Eng.*, 16(10):1279–1296, 2004.

[19] T. C. Hoad and J. Zobel. Methods for identifying versioned and plagiarized documents. *J. Am. Soc. Inf. Sci. Technol.*, 54:203–215, February 2003.

[20] W. Johnson and J. Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary Mathematics*, 26:189ÍC206, 1984.

[21] A. Joly and O. Buisson. Random maximum margin hashing. In *CVPR*, pages 873–880, 2011.

[22] S. Joshi, N. Agrawal, R. Krishnapuram, and S. Negi. A bag of paths model for measuring structural similarity in web documents. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '03, 2003.

[23] A. Kolcz, A. Chowdhury, and J. Alspector. Improved robustness of signature-based near-replica detection via lexicon randomization. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, pages 605–610, 2004.

[24] V. Koltchinskii. *Oracle Inequalities in Empirical Risk Minimization and Sparse Recovery Problems*. Springer, 2011.

[25] B. Kulis and K. Grauman. Kernelized locality-sensitive hashing for scalable image search. In *IEEE International Conference on Computer Vision (ICCV)*, 2009.

[26] J. P. Kumar and P. Govindarajulu. Duplicate and near duplicate documents detection: A review. *European Journal of Scientific Research*, 32:514–527, 2009.

[27] W. Liu, J. Wang, S. Kumar, and S.-F. Chang. Hashing with graphs. In *International Conference on Machine Learning (ICML 2011)*, 2011.

[28] Q. Lv, W. Josephson, Z. Wang, M. Charikar, and K. Li. Multi-probe lsh: Efficient indexing for high-dimensional similarity search. In *VLDB*, pages 950–961, 2007.

[29] U. Manber. Finding similar files in a large file system. In *Proceedings of the USENIX Winter 1994 Technical Conference on USENIX Winter 1994 Technical Conference*, pages 2–2, 1994.

[30] G. S. Manku, A. Jain, and A. Das Sarma. Detecting near-duplicates for web crawling. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 141–150, 2007.

[31] R. Panigrahy. Entropy based nearest neighbor search in high dimensions. In *SODA*, pages 1186–1195, 2006.

[32] M. Raginsky and S. Lazebnik. Locality-sensitive binary codes from shift-invariant kernels. In *The Neural Information Processing Systems (NIPS 2009)*, 2009.

[33] R. Salakhutdinov and G. E. Hinton. Semantic hashing. *International Journal of Approximate Reasoning*, 50(7):969–978, 2009.

[34] N. Shivakumar and H. Garcia-molina. Scam: A copy detection mechanism for digital documents. In *In Proceedings of the Second Annual Conference on the Theory and Practice of Digital Libraries*, 1995.

[35] C. Silpa-Anan and R. Hartley. Optimised kd-trees for fast image descriptor matching. In *Proceedings of 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, 2008.

[36] R. Weber, H.-J. Schek, and S. Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *Proceedings of the 24rd International Conference on Very Large Data Bases (VLDB 98)*, pages 194–205, 1998.

[37] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *NIPS 2008*, pages 1753–1760, 2008.

[38] D. Zhang, J. Wang, D. Cai, and J. Lu. Laplacian co-hashing of terms and documents. In *ECIR*, pages 577–580, 2010.

[39] D. Zhang, J. Wang, D. Cai, and J. Lu. Self-taught hashing for fast similarity search. In *SIGIR*, pages 18–25, 2010.

APPENDIX

A: Proof of Theorem 1

PROOF. Central to our analysis is Talagrand's inequality [24] given in the following theorem.

THEOREM 6. *(Talagrand's inequality) Let $X_1, \ldots, X_m$ be independent random variables in $\mathcal{X}$. For any class of functions $\mathcal{F}$ on $\mathcal{X}$ that is uniformly bounded by a constant $U > 0$ and for all $\delta > 0$, with a probability $1 - \delta$, we have*

$$\left| \sup_{f \in \mathcal{F}} \left| \sum_{i=1}^{n} f(X_i) \right| - \mathrm{E} \sup_{f \in \mathcal{F}} \left| \sum_{i=1}^{n} f(X_i) \right| \right| \leq K_1 U \ln \frac{K_1}{\delta} + \sqrt{K_1 \sigma^2 \ln \frac{K_1}{\delta}}$$

where K_1 is an universal constant and σ^2 is defined as

$$\sigma^2 = \mathrm{E} \sup_{f \in \mathcal{F}} \sum_{i=1}^{n} f^2(X_i)$$

We also need the following theorem to bound $|\mathbf{u}|_\infty$.

THEOREM 7. *(Lemma 2.2 [8]) Let $\mathbf{u}$ be a vector randomly sampled from $\mathcal{N}(0, I/d)$. Then, for any $\beta \geq 3$, we have, with a probability $1 - cd^{-\beta} \ln d$, such that*

$$\max_{1 \leq k \leq d} |u_k| \leq \frac{K_2 \beta}{\sqrt{d}}$$

where K_2 and c are constants.

We now proceed to our proof. First, according to Theorem 7, with a probability $1 - cd^{-3} \ln d$, we have

$$|\mathbf{u}|_\infty = \max_{1 \leq k \leq d} |u_k| \leq \frac{3K_2}{\sqrt{d}}$$

As a result,

$$U = \max_{\mathbf{x} \in \mathcal{D}(r, \mathbf{q})} |(\mathbf{x} - \mathbf{q}) \circ \mathbf{u}|_\infty \leq \frac{3K_2 A(r, \mathbf{q})}{\sqrt{d}}$$

We then prove the result by using the Berstein inequality. For any fixed $\mathbf{x} \in \mathcal{D}(r, \mathbf{q})$, according to the Berstein inequality, with a probability $1-\delta$, we have

$$\left|(\mathbf{x}-\mathbf{q})^\top \mathbf{u}\right| \le 2U \ln\frac{2}{\delta} + \sqrt{2\sigma^2 \ln\frac{2}{\delta}}$$

where

$$\begin{aligned}\sigma^2 &= \mathrm{E}|(\mathbf{x}-\mathbf{q}) \circ \mathbf{u}|_2^2 \\ &\le \Pr(|\mathbf{u}|_\infty \le 3K_2/\sqrt{d})|\mathbf{u}|_\infty^2 r^2 + \\ &\quad \left(1 - \Pr(|\mathbf{u}|_\infty > 3K_2/\sqrt{d})\right) r^2 \\ &\le \frac{3K_2 r^2}{d} + \frac{c\ln d}{d^3} r^2 = \frac{r^2}{d}\left(3K_2 + \frac{c\ln d}{d^2}\right)\end{aligned}$$

We complete the proof by taking the union of the probability over all $\mathbf{x} \in \mathcal{D}(r, \mathbf{q})$. □

B: Proof of Theorem 3

Proof. Since

$$\max_{1\le k\le m} |(\mathbf{x}-\mathbf{q})^\top \mathbf{u}_k| \ge \sqrt{\frac{1}{m}\sum_{k=1}^m |(\mathbf{x}-\mathbf{q})^\top \mathbf{u}_k|^2}$$

it is sufficient to bound $\frac{1}{m}\sum_{k=1}^m |(\mathbf{x}-\mathbf{q})^\top \mathbf{u}_k|^2$. Using the Telegrand inequality, we have

$$\begin{aligned}&\sup_{\mathbf{x}\in\mathcal{D}} \left| \frac{1}{m}\sum_{k=1}^m |(\mathbf{x}-\mathbf{q})^\top \mathbf{u}_k|^2 - \mathrm{E}\left[\frac{1}{m}\sum_{k=1}^m |(\mathbf{x}-\mathbf{q})^\top \mathbf{u}_k|^2\right]\right| \\ &\le K_1 U \ln\frac{K_1}{\delta} + \sqrt{K_1\sigma^2 \ln\frac{K_1}{\delta}}\end{aligned}$$

where

$$\begin{aligned}U &= \sup_{\mathbf{x}\in\mathcal{D}} \sup_{1\le k\le m} |(\mathbf{x}-\mathbf{q})^\top \mathbf{u}_k|^2 \\ \sigma^2 &= \mathrm{E}\sup_{\mathbf{x}\in\mathcal{D}} \sum_{k=1}^m \frac{|(\mathbf{x}-\mathbf{q})^\top \mathbf{u}_k|^4}{|\mathbf{x}-\mathbf{q}|^4}\end{aligned}$$

According to Theorem 1, we have, with a probability $1 - m\delta - \frac{cm\ln d}{d^3}$, for any $\mathbf{x} \in \mathcal{D}$,

$$|(\mathbf{x}-\mathbf{q})^\top \mathbf{u}_k| \le \frac{4|\mathbf{x}-\mathbf{q}|}{\sqrt{d}}\left(C_1 \ln\frac{2}{\delta} + C_2\sqrt{\ln\frac{2}{\delta}}\right), k = 1, \ldots, m$$

and therefore

$$U = \sup_{\mathbf{x}\in\mathcal{D}} \max_{1\le k\le m} \frac{|(\mathbf{x}-\mathbf{q})^\top \mathbf{u}_k|^2}{|\mathbf{x}-\mathbf{q}|^2} \le \frac{16}{d}\left(C_1 \ln\frac{2}{\delta} + C_2\sqrt{\ln\frac{2}{\delta}}\right)^2$$

To bound the variance σ^2, we have

$$\sigma^2 \le \frac{U}{|\mathbf{x}-\mathbf{q}|^2}\mathrm{E}\sum_{k=1}^m |(\mathbf{x}-\mathbf{q})^\top \mathbf{u}_k|^2$$

Using the bound of U, we have, with a probability $1 - m\delta - \frac{cm\ln d}{d^3}$,

$$\sigma^2 \le \frac{16m}{d^2}\left(C_1 \ln\frac{2}{\delta} + C_2\sqrt{\ln\frac{2}{\delta}}\right)^2$$

Finally, using the fact

$$\mathrm{E}\left[\frac{1}{m}\sum_{k=1}^m |(\mathbf{x}-\mathbf{q})^\top \mathbf{u}_k|^2\right] = \frac{|\mathbf{x}-\mathbf{q}|^2}{d},$$

and the upper bounds for U and σ^2, we have, with a probability $1 - m\delta - mcd^{-3}\ln d$,

$$\begin{aligned}&\sup_{\mathbf{x}\in\mathcal{D}} \frac{1}{m}\sum_{k=1}^m \frac{|(\mathbf{x}-\mathbf{q})^\top \mathbf{u}_k|^2}{|\mathbf{x}-\mathbf{q}|^2} \ge \\ &\quad \frac{1}{d}\left(1 - \frac{16K_1C_\delta^2}{m}\ln\frac{K_1}{\delta} - 4C_\delta\sqrt{\frac{K_1}{m}\ln\frac{K_1}{\delta}}\right)\end{aligned}$$

where

$$C_\delta = C_1\frac{2}{\delta} + C_2\sqrt{\frac{2}{\delta}}$$

By choosing m that satisfies the following condition

$$\frac{16K_1C_\delta^2}{m}\ln\frac{K_1}{\delta} + 4C_\delta\sqrt{\frac{K_1}{m}\ln\frac{K_1}{\delta}} \le \frac{3}{4} \tag{4}$$

we have

$$\max_{1\le k\le m} |(\mathbf{x}-\mathbf{q})^\top \mathbf{u}_k| \ge \frac{|\mathbf{x}-\mathbf{q}|_2}{2\sqrt{d}}$$

The condition

$$m \ge 64K_1\left(C_1\ln\frac{2}{\delta} + C_2\sqrt{\ln\frac{2}{\delta}}\right)$$

follows directly from the condition in (4). □

C: Proof of Theorem 4

Proof. We first fix $\mathbf{x} \in \mathcal{D}(r, \mathbf{q})$. We write $(\mathbf{x}-\mathbf{q})^\top \mathbf{u}$ as $\sum_{i=1}^d (x_i - q_i)u_i$. Since $\mathrm{E}[(\mathbf{x}-\mathbf{q})^\top \mathbf{u}] = 0$, using Berstein inequality, we have, with a probability $1-\delta$,

$$|(\mathbf{x}-\mathbf{q})^\top \mathbf{u}| \le 2V\ln(2/\delta) + \sqrt{2\sigma^2\ln(2/\delta)}$$

where $V = \max_i (x_i - q_i)u_i$ and $\sigma^2 = \mathrm{E}\left[(\mathbf{x}_i - q_i)^2 u_i^2\right]$. Using the fact $V \le |\mathbf{x}-\mathbf{q}|_\infty \le |\mathbf{x}-\mathbf{q}|$, $\sigma^2 = |\mathbf{x}-\mathbf{q}|^2$, we have

$$|(\mathbf{x}-\mathbf{q})^\top \mathbf{u}| \le \frac{r}{\sqrt{d}}\left(2\ln\frac{2m(r,\mathbf{q})}{\delta} + \sqrt{2\ln\frac{2m(r,\mathbf{q})}{\delta}}\right)$$

We complete the proof by taking the union bound over the set $\mathcal{D}(r, \mathbf{q})$. □