Efficient Nearest-Neighbor Search in the Probability Simplex

Kriste Krstovski
School of Computer Science
University of Massachusetts
Amherst, MA, 01003, U.S.A.
kriste@cs.umass.edu

David A. Smith
College of Computer and Information Science
Northeastern University
Boston, MA, 02115, U.S.A.
dasmith@ccs.neu.edu

Hanna M. Wallach
School of Computer Science
University of Massachusetts
Amherst, MA, 01003, U.S.A.
wallach@cs.umass.edu

Andrew McGregor
School of Computer Science
University of Massachusetts
Amherst, MA, 01003, U.S.A.
mcgregor@cs.umass.edu

ABSTRACT

Document similarity tasks arise in many areas of information retrieval and natural language processing. A fundamental question when comparing documents is which representation to use. Topic models, which have served as versatile tools for exploratory data analysis and visualization, represent documents as probability distributions over latent topics. Systems comparing topic distributions thus use measures of probability divergence such as Kullback-Leibler, Jensen-Shannon, or Hellinger. This paper presents novel analysis and applications of the reduction of Hellinger divergence to Euclidean distance computations. This reduction allows us to exploit fast approximate nearest-neighbor (NN) techniques, such as locality-sensitive hashing (LSH) and approximate search in k-d trees, for search in the probability simplex. We demonstrate the effectiveness and efficiency of this approach on two tasks using latent Dirichlet allocation (LDA) document representations: discovering relationships between National Institutes of Health (NIH) grants and prior-art retrieval for patents. Evaluation on these tasks and on synthetic data shows that both Euclidean LSH and approximate k-d tree search perform well when a single nearest neighbor must be found. When a larger set of similar documents is to be retrieved, the k-d tree approach is more effective and efficient.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; I.5.3 [**Pattern Recognition**]: Clustering—*Similarity measures*

General Terms

Experimentation, Performance

ICTIR '13 September 29 – October 2, 2013, Copenhagen, Denmark

Keywords

document similarity, topic models, latent Dirichlet allocation, approximate nearest neighbors

1. INTRODUCTION

Many tasks in information retrieval (IR) and natural language processing (NLP) involve performing document similarity comparisons. These tasks include document clustering, retrieving the most relevant documents for a given query, and finding document translation pairs in a large multilingual collection.

Most practical applications of document similarity represent documents in a common *feature space*. For many tasks, such as IR with bag-of-words models, this shared space is sparse: if our document features are the counts of single words, only a few hundred unique words will have non-zero counts in document of a few thousand words. For such representations, inverted indexes of features provide efficient performance. Although boolean, vector-space, and probabilistic methods all compute different similarity functions, their inner loop is a dot product that inspects the sparse set of overlapping features.

A natural question, therefore, is what feature space to select. Representing documents in a shared feature space abstracts away from the specific sequence of words used in each document and, with appropriate representations, can also facilitate the analysis of relationships between documents written using different vocabularies.[1] As a concrete example, identifying academic communities working on related scientific topics can involve comparing the divergent terminology in different subfields [36]. A more extreme form of vocabulary mismatch occurs when documents are written in different languages. Mapping documents in different languages into a common shared space can therefore be an effective method of detecting documents or passages that are translations of each other [27, 31, 23].

Although a sparse word or n-gram vector is a popular representational choice, some researchers have explored "deeper" representations, such as Latent Semantic Indexing (LSI) [14]. LSI has been recast as a generative model of text with Probabilistic Latent Se-

[1]Feature vectors can, of course, represent some sequence information with, e.g., n-grams of terms, but the degenerate case of an indicator function that matched entire documents would be ineffective for similarity comparisons.

mantic Indexing (PLSI) [20]. More recently, statistical topic models, such as latent Dirichlet allocation (LDA) [8], have proven to be highly effective at discovering hidden structure in document collections [e.g., 18].

One of the greatest advantages in using topic models to analyze large document collections is their ability to represent documents as probability distributions over a small number of topics, thereby mapping documents into a low-dimensional latent space—the T-dimensional probability simplex, where T is the number of topics. A document, represented by some point in this simplex, is said to have a particular "topic distribution". This type of document representation makes it appealing for the IR community. As such, feasibility and effectiveness of the topic models have previously been explored in the IR community. For example, Wei and Croft [39] inferred topics over words using LDA to improve document smoothing and ad-hoc retrieval. More recently, Andrzejewski and Buttler [4] showed that LDA has the potential to improve on the task of query expansion for specialized domain collections with a small user base.

As a result of the broad applications of topic modeling, it is appealing and natural to ask whether representing documents in this low-dimensional topic space would yield advantages to various document similarity tasks; however, to date, this question has not really been explored, especially on big, real-world data sets. Although there has been some work on sparse priors for topic models [38], topic distributions are not as sparse as discrete term feature vectors; moreover, they are continuous. Exact similarity computations for most topic distributions therefore require $O(N^2)$ comparisons for near-neighbor detection tasks or $O(kN)$ computations when k queries are compared against a data set of N documents.

To perform similarity search on large collections efficiently, we frame the computation as an approximate nearest neighbor (NN) search problem. NN search is an optimization problem that deals with the task of finding nearest neighbors of a given query q in a metric space of N points. We are consequently able to use different data structures and approximation algorithms to trade off speed and accuracy. In the past, this type of formulation for the document similarity comparison problem has been proven to yield good results in the metric space due to the fact that NN search problem has been designed to handle distance metrics (e.g. cosine, Euclidean, Manhattan, etc.) and therefore could be applied directly (LSH, k-d trees, etc.). For example, locality sensitive hashing (LSH) approaches to approximate NN search using cosine distance have been previously used for tasks such as noun clustering [33], first-story detection on Twitter [30] and ranking document pairs by overlapping words [22].

For points in the probability simplex, similarity comparison is performed by measuring the difference between two probability distributions. As such, distance metrics are not appropriate in the probability simplex, and divergence based measurements are used instead—in particular, information-theoretic measurements of similarity such as Kullback-Leibler and Jensen-Shannon divergence and Hellinger distance.

While LSH schemes exist for both cosine and Euclidean (L_2) distances [10, 3] and k-d trees and their variants work with the Minkowski metrics (L_1, L_2, etc.) [16, 5], these cannot be directly applied to measuring distances in the probability simplex. Therefore, performing document similarity in large datasets where documents are represented as points in the simplex cannot be addressed with the same NN search algorithmic instances as described in the previous paragraph. The inability to perform fast document similarity computations when documents are represented in the simplex has thus limited researchers' ability to explore the potential of these representations on large scales.

This paper introduces a technique for performing nearest neighbor search in the probability simplex, thereby facilitating efficient document similarity computations when documents are represented as (continuous) probability distributions. Our approach works with both LSH and k-d tree methods. We give speed and accuracy results on our approach on two representative document similarity tasks—scientific community discovery and prior-art retrieval for patents—and two approximate NN techniques (Euclidean LSH and k-d trees). Despite our emphasis on document similarity tasks, the techniques presented in this paper could be applied to other probability distribution comparisons tasks that deal with large data sets. These methods should be generally useful for the growing community working on representing the latent structure in documents using probability distributions such as those induced by LDA.

2. FAST NN SEARCH

Comparing all pairs of N documents implies computational complexity of $O(N^2)$. Even if a single query is known a priori, a linear scan through the corpus is in general required. Since this asymptotic growth rate is impractical for large datasets, sub-linear approximate solutions for retrieving the nearest neighbors of a query point have been proposed. In this paper, we concentrate on **LSH** and **k-d trees**. In an early paper on LSH, Indyk and Motwani [21] defined the problem as follows. Given a query point q, return point p that is an ϵ-approximate nearest neighbor of q such that $\forall p'$ nearest neighbors that satisfy the inequality: $Distance(q,p) \leq (1+\epsilon)Distance(q,p')$, or more succinctly $(1+\epsilon)r$ nearest neighbors, where r is the radius of data set points considered for each query point. This approach hashes the original data points into separate buckets using a family of hash functions such that the probability of collision between query point q and points p in the dataset increases with the similarity between them. More formally, the function $p(t) = \Pr[h(q) = h(p) :\parallel q - p \parallel = t]$ is strictly decreasing in t. Once points are hashed into a bucket, the closest point(s) out of those already in the bucket is returned. Utilizing multiple hash functions improves the accuracy. Varying the radius r changes the number of data set points considered for each query point and therefore directly affects the accuracy of the results as well as the running time of the algorithm. Charikar [10] expanded this approach and showed that it could be used to perform approximate cosine distance computation.Other sub-linear solutions organizes points in space by partitioning with optimized data structures. Bentley [6] introduced the multidimensional binary search tree or k-d tree. With k-d trees, points in metric space are stored in a partitioning data structure where data points are represented with nodes along with two pointers and a discriminator variable whose range of values is the dimensionality of the space. The two data pointers point to subtrees or to null based on whether the value of the chosen point dimension (based on the discriminator value) is greater or smaller then the split value for that dimension. In the next section we describe how common probability divergence measurements could be used with these sub-linear nearest neighbor search approaches.

2.1 Transforming Divergences

The original LSH and k-d trees approaches to the nearest neighbor problem were introduced in the Euclidean space where the fol-

lowing distance metric is used:

$$Eu(p,q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} \quad (1)$$

In the probability simplex, distributions are compared using information-theoretic measurements such as Jensen-Shannon and Hellinger divergence. Jensen-Shannon divergence and Hellinger distance are f-divergences [13] as they both measure the similarity between two probability distributions. Jensen-Shannon (JS) divergence was originally derived from Kullback-Leibler (KL) divergence (also known as relative entropy) as its symmetric version [32, 25]:

$$\mathrm{KL}(p,q) = \sum_{i=1}^{n} p(x_i) \log \frac{p(x_i)}{q(x_i)} \quad (2)$$

$$\mathrm{JS}(p,q) = \frac{1}{2}\mathrm{KL}\left(p, \frac{p+q}{2}\right) + \frac{1}{2}\mathrm{KL}\left(q, \frac{p+q}{2}\right) \quad (3)$$

Hellinger (He) distance is also symmetric and is used along with JS divergence in various fields where a comparison between two probability distributions is required [7, 18, 9]:

$$\mathrm{He}(p,q) = \sum_{i=1}^{n}\left(\sqrt{p(x_i)} - \sqrt{q(x_i)}\right)^2 \quad (4)$$

These three information-theoretic measurements are not considered metrics since they do not satisfy the triangle inequality. KL divergence is also not symmetric. While being non-negative, this measurement could have a value of infinity [11]. JS divergence on the other hand is bound by one [25] and it could easily be shown that its unnormalized version is bounded by 2. It is worth noting that there are several variations of the actual Hellinger measurement formula and usually the difference is the constant of $\frac{1}{2}$ placed in front of the sum as well as the square root of the whole equation. Here, we use the most straightforward version. Hellinger distance is also non-negative and bounded by 1, 2, or $\sqrt{2}$ depending on the version of the formula. It can easily be shown that the upper bound for the He version in (4) is 2.

Comparison of the Euclidean distance metric and the Hellinger divergence measurement shows that both measurements have similar algebraic expressions which differs in how the square root function is applied. If we discard the square root used in the Euclidean distance, Hellinger distance (4) becomes equivalent to the Euclidean distance metric (1) between $\sqrt{p_i}$ and $\sqrt{q_i}$. For tasks that involve creating ranked lists, such as the NN-search task, the square root of the Euclidean distance can be discarded since this function is computed across all data points and as such doesn't affect the overall similarity ranking for a given query point. Furthermore, the same function is not consistent across all variations of the Hellinger distance.

Hellinger distance can therefore be computed by first computing the square root of the distributions to be compared and then computing the Euclidean distance between these transformed distributions. Mapping each probability distribution of interest p_i to $\sqrt{p_i}$ therefore allows us to utilize an already established approximation approach for computing Euclidean distance metric such as LSH and k-d trees. Here, we use LSH approach for Euclidean spaces developed by Andoni et al. [3], which is implemented in the exact Euclidean LSH (E2LSH) package [2], and the k-d tree implementation in the ANN library [28].

Aside from the Hellinger distance, another widely used divergence measurement is the Jensen-Shannon divergence. Compared to the Hellinger distance, the algebraic form of this measurement doesn't resemble any of the distance measurements used in the Euclidean space. As such, simple transformations as in the case of the Hellinger distance can't be applied that would allow us to utilize approximate NN-search methods in the Euclidean space to compute it. If we simply view Jensen-Shannon divergence as a function, we could utilize approximation theory to explore ways to approximate this divergence measurement using the Hellinger distance. One way to proceed in this direction is to empirically show a constant factor relationship between these two measurements. In an earlier work by Topsøe [37] it was shown that Jensen-Shannon divergence (referred to as capacitory discrimination) behaves similarly with the triangle divergence (triangular discrimination):

$$\frac{1}{2} \triangle (p,q) \quad \leq JS(p,q) \quad \leq \ln(2) \triangle (p,q) \quad (5)$$

Topsøe [37] has also pointed out that a close relationship between Hellinger distance and triangle divergence exists:

$$He(p,q) \quad \leq \triangle(p,q) \quad \leq 2He(p,q) \quad (6)$$

As in the case with [17] we represent the relationship between Jensen-Shannon and Hellinger in a more concise form:

$$\begin{aligned} \frac{1}{2}He(p,q) &\leq \frac{1}{2} \triangle (p,q) \\ &\leq JS(p,q) \\ &\leq \ln(2) \triangle (p,q) \\ &\leq 2\ln(2)He(p,q) \end{aligned}$$

From the above bounds it is clear there exists an explicit constant factor relationship between the Hellinger distance and Jensen-Shannon divergence:

$$\frac{1}{2}He(p,q) \quad \leq JS(p,q) \quad \leq 2\ln(2)He(p,q) \quad (7)$$

This allows us to approximate Jensen-Shannon divergence with Hellinger distance but to continue further we need to empirically prove and conclude how tightly the above theoretical bounds hold. For this reason, we ran experiments on synthetic data that was generated by drawing samples from a Dirichlet distribution. (In LDA, the posterior distribution of topics for documents follows a Dirichlet distribution.) We generated 100 samples from Dirichlet distributions that varied across different order i.e. dimension values D and values of a symmetric hyperparameter α. For each generated sample set of 100 probability distributions we performed all pairs JS divergence and He measurement. Figure 1 shows the relationship between JS and Hellinger across values of D = 50, 100, 200 and 500 and values of α = 0.001, 0.01, 0.1, 1, 10, 100. From the plots, it is evident that we could approximate Jensen-Shannon divergence with Hellinger distance. Varying the hyperparameter α we vary the sparsity of the distribution across the dimensions. This sparsity is reflected on the range of values that both measurements take. In case of large values of α the divergence range is very small while with very small values (i.e. α<1) the bulk of the divergence mass has tendency to reside in the area close to the upper bound. As we increase the dimension value this mass shifts towards the upper limit of both measurements. Varying the dimension values we also confirm that the constant factor relationship is not affected by the dimensionality of the probability simplex.

We used the same synthetic dataset to explore the performance of the two approximate NN methods considered here: Euclidean LSH (in the E2LSH package) and approximate search in k-d trees (in the ANN package). Figure 2 shows that when the α parameter of the Dirichlet distribution is high, the synthetic topic distributions are uniform and both methods perform quite well at retrieving the

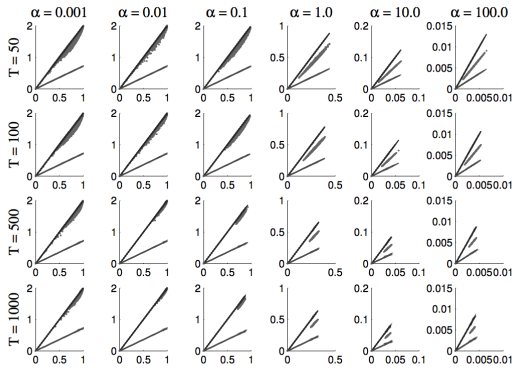


Figure 1: Empirical evidence of the bounds presented in Eq. 7 on 10k document pairs across discrete probability distributions of different lengths and value of the hyperparameter. The lower bound is $He(p,q) = \frac{1}{2\ln(2)} JS(p,q)$ while the upper bound is $He(p,q) = 2JS(p,q)$.

points at ranks 1 through 10. When the α of the Dirichlet prior is low and the topic distributions are sparser, approximate search in k-d trees is much more robust than E2LSH. It is also worth noting that, if all we want is the *nearest* neighbor, which appears on these graphs at rank 2, E2LSH often achieves acceptable performance, though it falls off steeply at higher ranks.

3. FAST NN SEARCH RESULTS

Document similarity occurs in wide variety of areas in IR and NLP. Furthermore, tasks in other fields, such as computer vision, also utilize similarity comparisons across vector representations. In order to demonstrate the effectiveness and efficiency of the approximation approach for retrieving similar documents in large data sets represented in the probability simplex, we explore two different tasks. For our first task, we will use document-topics representations of National Institutes of Health (NIH) grants and perform comparison between speed and accuracy of using regular all-pairs comparison and approximate comparison in the probability simplex. Next, we apply approximate nearest neighbor computations in topic space to the task of prior-art retrieval for patents.

Both tasks involve comparison of documents in a mutual, shared space and as such could be formulated as NN search problem. They are diverse as they offer us a variety of objectives for the NN search problem ranging from 1-NN to k-NN. We use the probability simplex as the shared space. For each task we consider as a baseline the probability distribution similarity of the document representations. The reader should note that our goal is not the performance comparisons across different document representations in a shared space—metric space vs. probability simplex nor it is to compare probability simplexes of different natures—LDA generated probability distributions vs. other type of probability distributions. Rather, we are going to compare LSH and k-d tree approximation of the Hellinger distance to regular Hellinger distance and therefore showing the loss in performance when using the approximate approach. Furthermore, we are going to compare approximation of the Hellinger distance to Jensen-Shannon divergence and therefore showing the performance difference between using Hellinger distance over JS divergence and using approximate Hellinger over JS divergence. We will demonstrate that for two diverse document similarity tasks, we are able to use our fast approximate NN search to perform the necessary similarity comparisons between documents represented as points in the probability simplex. This results in significant speed increase while maintaining small loss in accuracy.

3.1 Mapping NIH Funding

In previous work, Talley et al. [36] showed that LDA could be useful in facilitating categorization and relationship discovery in large document collections. In particular, this work explores the usefulness of LDA in mapping documents into a common vector space in order to perform similarity discovery in the probability simplex. The United States NIH, which consists of 27 Centers and Institutes, funds between 70-80K grants. These organizational entities, while having independent missions, often, as research goals expand, fund grants in areas that overlap one another's missions. Exploring relationships between grants is hard given the continuously increasing number of funded grants. Talley et al. [36] constructed a graph-based layout of grants, where grants were arranged based on a weighted sum of KL divergence computed on word probability distributions and topic distributions.

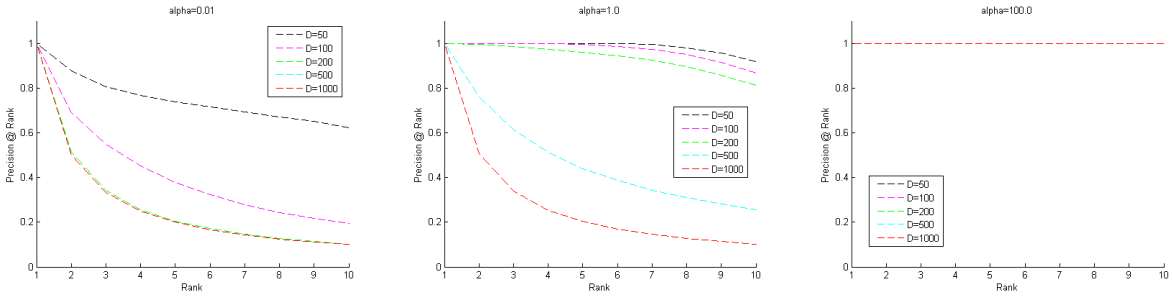
In our task the objective is to showcase the speed benefits and quantify the accuracy tradeoff of using approximate, LSH and k-d trees based, version of the Hellinger distance rather than JS for a real-world task. As such, we use the pre-computed topics distribution representations of the documents used by Talley et al. [36]. Since the goal of Talley et al. [36] is to construct a graph-based layout of grants there is no absolute best approach in terms of the type of similarity measurements to be used and therefore there are no base results that could be used to show difference in performance. We show that clustering approaches as in Talley et al. [36] could be practical to use and used on larger collections. In addition they could also be used in combination with online learning algorithms as in Langford et al. [24].

3.1.1 Data Set and Results

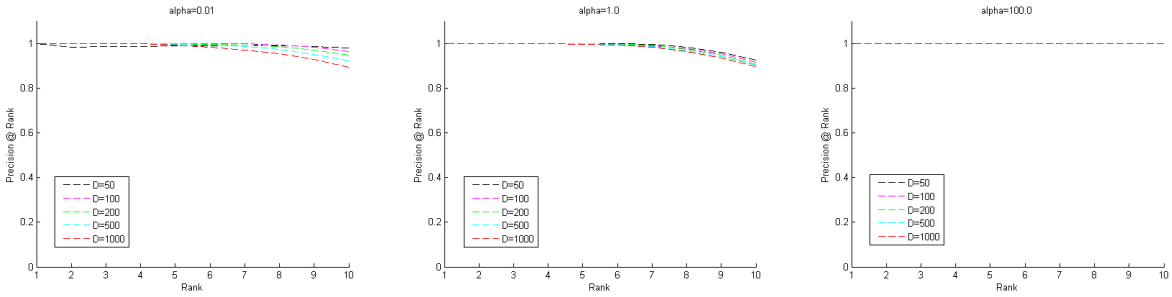
The NIH data consists of abstracts and titles of grants from 2007–2010 as well as MEDLINE journal articles published between 2007–2010 that cite NIH grants and intramural and sub-awards. We use the topics distribution representations of these documents as in Talley et al. [36]. In other words, the topic modeling setup is identical in terms of the number of topics, hyperparameter values and number of Gibbs sampling iterations.

We computed JS divergence across all pairs of ∼350K documents where each document was represented as a distribution over 550 topics. We use the top 10 ranked document pairs for each query grant as a set of relevant grants. As pointed out earlier, for a clustering task of this nature, where the objective is to create graph-based layout, there is no absolute best approach and as such we did not try to compare this clustering approach with existing IR approaches whose document representations are in the metric space and use different similarity score functions, such as BM25.

We create a query set of 10,000 randomly chosen grants and we use the E2LSH algorithm to compute approximate LSH Hellinger similarity across the remaining database of ∼343K grants. We also computed the k-d trees based approximate Hellinger similarity using the ANN implementation of k-d trees configured with default parameters. Performance is evaluated by computing the precision of the top 5 (P@5), recall of the top 5 (R@5), and mean average precision (MAP) over each query result given the top 10 closest grants obtained in our exhaustive all-pairs JS similarity comparison. Since the total number of nearest neighbor grants reported by the LSH approach depends on the preset value of the radius R, we evaluated the relative performance of LSH with R set to R=0.4, R=0.6 (the default value), and R=0.8. We also had E2LSH and



(a) E2LSH: Euclidean LSH



(b) ANN: Approximate search in k-d trees

Figure 2: Comparison of approximate nearest-neighbor search techniques on synthetic data. The nearest neighbor at rank 1 is the original point. When the Dirichlet α parameter used to generate the synthetic data is low and the data are drawn from sparse multinomial distributions, E2LSH with $R = 0.6$ (2a) performs poorly. In some cases, however, it still achieves good recall at rank 2, which may be acceptable if there is only a single true neighbor. Also, neighbors are easier to find in lower-dimensional (D) spaces. In later experiments, these dimensions correspond to LDA topics. Approximate search in k-d trees (2b) performs much better across all dimensionalities and degrees of uniformity.

Divergence Type	MAP	P@5	R@5	Speedup
He LSH R=0.4	0.14	0.26	0.13	983.88
He LSH R=0.6	0.53	0.70	0.35	654.22
He LSH R=0.8	0.92	0.99	0.49	336.27
He k-d trees	0.92	0.99	0.49	1425.23

Table 1: Finding similar NIH grants: Performance comparison between the all pairs JS divergence, the LSH, and approximate k-d trees using Hellinger distance. Grants are represented by distributions over 550 topics.

ANN configured to return only the top 10 nearest neighbors discovered in the given radius. Table 1 shows the results obtained along with the relative difference in time between all pairs JS divergence, the approximate LSH based Hellinger distance with different value of R and the approximate k-d trees based Hellinger distance. When running all-pairs JS based similarity computation, the code implementation could significantly affect the processing time. Due to the size of the test collection, in our implementation, for each query document k we go over the list of n documents in the test collection. When R=0.6, we are able to retrieve only half of the 10 relevant documents for each query; for a radius of R=0.8, we end up obtaining MAP of 0.92 and since we retrieve only the top 10 nearest neighbors we could interpret this value as recall at ten which means that on average 9.2 out of the 10 relevant documents were retrieved. We obtain the same performance with the ANN implementation of k-d trees. When R=0.8, we achieve almost perfect P@5. From this relative comparison between the three all-pairs comparison approaches one could infer that the approximate methods manages to retrieve almost the whole mass of the documents discovered by the JS divergence without a significant loss in precision and recall. Overall, the difference between the E2LSH based divergence measurements is heavily influenced by the settings of the Euclidean LSH algorithm.

3.2 Retrieving Related Patents

In the process of reviewing patent applications, an important step is the search for prior art. In that step, patent examiners compare the patent application with previously granted patents in order to evaluate its novelty. Patent applications, from document structure perspective, are very complex as they contain several different sections written in different styles and language. Since their goal is to protect their inventions, patent authors intentionally use vague vocabulary and come up with new terminology in order to extend the patent coverage. In addition, authors also try to use esoteric language in order to make the patent application appear different from previously granted applications. Also, since patents deal with inventions in different domains, they tend to be written with different vocabularies while conveying the same idea. All these issues cause significant effort for patent examiners in reviewing applications.

Prior-art search involves composing a proper query or a set of queries and evaluating the relevance of the obtained search results. Unlike typical retrieval systems, the goal is to be able to retrieve all relevant patents and thus it is mostly recall oriented. Some of these challenges are discussed as part of the TREC Chemistry Track [26] and other conferences and workshops [35]. While topic models have been previously used to augment the traditional task of information retrieval [41], topic model representations for the prior-art patent search, to the best of our knowledge have not been explored before. We believe that it might be beneficial to explore this type of

representation especially due to the fact that patents have definite vocabulary differences because they are often written with obfuscatory goals and as such, representing patents in low-dimensional latent space, such as the topic space, abstracts beyond the specific words used in each document. The latter may be very beneficial when performing prior-art patents search in different languages. Exploring topic representations for this task may be useful either as a standalone representation and/or as part of other representations or a combination of them. Using representations in the probability simplex allows us to formulate the patent retrieval task as NN search problem in the probability simplex but in order to be used in real-world scenarios we need to first empirically confirm a fast NN approach. In this section, we explore using a topic representation of patent applications inferred by LDA to perform prior-art search. We use the same training and test set as in [40].

While our goal is not to show absolute performance results over the task we do make a comparison of the performance of our approach with the approach explained in [40]. We do however focus on three objectives—to show that our approximate technique massively speeds up the NN search task—to compare whether JS divergence or Hellinger distance is most suited for this task—and to provide new, first exploratory results on using topic space for patents representation. As part of the latter task, we show improvements over the approach explained in [40] by performing rank aggregation over the ranked lists obtained from our LDA based representation using fast NN search and ranked lists obtained from the representation in [40].Results from rank aggregation are presented in Section 3.2.3 .

3.2.1 Experimental Setup

For our experiments we use the USPTO collections which consists of ∼1.6M patents published between 1980 and 1997 [29]. We first represent each patent by extracting the text found in the following six fields: title of invention (TTL), abstract field (ABST), primary claim (PCLM), drawing description (DRWD), detail description (DETD) and background summary (BSUM). This same set of fields was previously used by Xue and Croft [40] in their exploratory analysis of the impact of each field on retrieval performance. We then map each patent into a latent topic space using LDA. Due to the size of the collection and in order to perform efficient per document topics distribution inference we use an on-line variational Bayes (VB) algorithm developed by Hoffman et al. [19]. To further speed up estimating the posterior per-document topics distributions, we utilize the Vowpal Wabbit [24] implementation of this algorithm.

Out of the original vocabulary of ∼4.5M tokens found in the collection, we use a small subset of 32,609 to represent patents. We derived this vocabulary by filtering out all tokens whose frequency of occurrence is less than 1K and more than 350K. We further filtered out numeric tokens and tokens with fewer than four characters.

3.2.2 Evaluation Task and Results

Evaluating prior-art search requires relevance judgments which, due to the nature of the problem, are infeasible to obtain and therefore previous work on this topic has used the patent's citation fields (UREF) entries as relevance judgments. As in the previous work by Xue and Croft [40], we use patents published in the time period between 1980–1996 as test collection. We filter out patents that do not contain the following five fields: TTL, ABST, PCLM, DRWD and DETS. The query data consists of patents published in 1997 whose total number of citation fields is more than 20 and contain all five previously mentioned fields. We evaluate the performance of exhaustively computing JS divergence, Hellinger distance, the approximate k-d trees and LSH Hellinger distance with values of R=0.4, R=0.6 and R=0.8. Accuracy was evaluated using MAP, precision of the top 10 (P@10) and recall of the top 10 (R@10) while speed performance was evaluated by measuring the relative difference in time between all pairs JS divergence, approximate k-d trees based Hellinger and different variations of the approximate LSH Hellinger for different value of R across different topics dimensionality. As with the NIH grants, we computed the Hellinger distance approximation using E2LSH [2]. E2LSH is configured to run with probability of success set to default value of $(1 - \delta) = 0.9$. We also use the same k-d trees implementation in the ANN package [28] configured with default parameters.

Method type	MAP	P@10	R@10
Xue and Croft	0.204	0.416	0.138
JS	0.172	0.343	0.111
He	0.178	0.345	0.112
He LSH R=0.4	0.056	0.161	0.051
He LSH R=0.6	0.091	0.248	0.078
He LSH R=0.8	0.161	0.344	0.111
He k-d trees	0.159	0.345	0.112
Agg. rank	0.232	0.442	0.145

Table 2: Prior-art patent search performance comparison using MAP, P@10 and R@10 between all pairs JS divergence, all pairs Hellinger distance, the approximate LSH based Hellinger distance and k-d trees using LDA with T=500.

Shown in Table 2 are results obtained when using the same query set as in [40] and 70k patents chosen from the complete test collection as in [40] that also contain the relevant patents for the query set. The goal of this test was to compare the relative performance of the three similarity measurements on the task of performing patent retrieval and therefore we experimented with only one set of topics T=500. We ran Vowpal Wabbit [24] implementation of LDA with default values of the hyper-parameters α = 0.1 and β = 0.1 and number of training passes set to five. Along with our results we also show results of re-running the approach of Xue and Croft [40] with all five patent fields (field="all") and weight set to tf on the same test collection of 70k patents. We decided to utilize this particular configuration of their approach as it uses all application fields for constructing the query (as it is in our case). In addition, we show results of using the tf weight as it yields best P@10 over the other two weight types as reported by Xue and Croft [40].

While using LDA representation we don't achieve better performance compared to our baseline results derived using the approach by Xue and Croft [40] we do show that using approximate LSH based Hellinger distance with R=0.8 and approximate k-d trees based Hellinger we achieve almost the same P@10 and R@10 results as in the case of regular JS divergence and Hellinger distance. We don't achieve the same MAP value since both approximate approaches are configured to return the top 200 nearest neighbor points compared to the regular JS divergence and Hellinger distance where we evaluate across all points.

Shown in Table 3 are P@10 results obtained using LDA with number of topics set to T=50, T=100, T=200, T=500 across results obtained when using all-pairs JS divergence, Hellinger distance and the approximate LSH Hellinger distance with values of R=0.4, R=0.6 and R=0.8. The goal is to show how sensitive the retrieval is to the topic dimensionality.

Table 4 shows the relative differences in time between all pairs JS divergence and approximate LSH based Hellinger distance with

Metric Type	T=50	T=100	T=200	T=500
JS	0.212	0.266	0.306	0.343
He	0.222	0.273	0.320	0.345
He LSH R=0.4	0.168	0.170	0.181	0.161
He LSH R=0.6	0.219	0.252	0.276	0.248
He LSH R=0.8	0.223	0.274	0.319	0.344
He k-d trees	0.223	0.274	0.320	0.345

Table 3: Prior-art patent search performance comparison between the all pairs JS divergence, all pairs Hellinger distance and the approximate LSH based Hellinger distance over P@10 for LDA models with topic dimensionality T=50, T=100, T=200 and T=500.

Div.	T=50	T=100	T=200	T=500
JS	6.4	4	2.3	1
He LSH R=0.4	85.6	75.3	57.8	35.6
He LSH R=0.6	59.2	53.4	38.1	27.9
He LSH R=0.8	46.1	33.0	29.0	16.7
He k-d trees	793.6	410.4	224.6	98.4

Table 4: Relative speed improvement on prior-art patent search between all-pairs JS divergence and approximate He divergence via k-d trees and LSH across different values of radius R.

different value of R. Results shown are based on comparing the running time of E2LSH and ANN against the all-pairs similarity comparison using JS divergence. As in the case of the NIH grants, due to the size of the test collection, we compute JS divergence using an implementation where for each query document k we go over the list of n documents in the test collection. Compared to computing Hellinger distance with LSH, approximating Hellinger with k-d trees gives us a significant improvement in speed across all values of T while maintaining the same performance across all three evaluation metrics (MAP, P@10 and R@10).

3.2.3 Rank Aggregation

Using the ranked lists obtained through the approach by Xue and Croft [40] we perform rank aggregation with the ranked lists obtained by our LDA representation. The goal of our task was to show whether using our LDA approach as a standalone similarity retrieval approach we could improve an existing IR system. Since both types of ranked lists were generated using different retrieval models and using different scoring functions we first normalize both types of ranked lists. We explored two approaches for score and rank normalization as in [12]. With the first approach, scores are normalized using the maximum and the minimum score values in the original ranked list:

$$norm\_similarity = \frac{unnorm\_similarity - min\_similarity}{max\_similarity - min\_similarity} \quad (8)$$

With the rank based normalization approach actual rank values are used to generate normalized similarity scores using the following formula:

$$rank\_similarity = 1 - \frac{rank-1}{num\_of\_retrieved\_docs} \quad (9)$$

Rank aggregation is a topic which has been extensively explored and various different approaches for combining ranked lists have been proposed. An overview of these approaches is given in [12]. Examples of rank aggregation approaches are given in [15], [1] and [34], to name a few. In our case we explored the approaches proposed by Shaw and Fox [34]. In particular we explore and experimented with the following five approaches CombMIN, CombMAX, CombSUM, CombANZ and CombMNZ.

Aggr. type & baseline	norm. score	norm. rank
Xue and Croft	0.173	0.173
JS	0.158	0.158
CombMIN	0.077	0.143
CombMAX	0.174	0.178
CombSUM	0.174	0.187
CombANZ	0.079	0.151
CombMNZ	0.174	0.187

Table 5: Prior-art patent search: P@10 on held-out development set across various rank aggregation approaches using normalized scoring and rank function values.

For two given ranked lists and for a given ranked document that exists in both lists, CombMIN and CombMAX use the minimum and the maximum score of the two respectfully. CombSUM as the name implies, for a given ranked document that exists in both lists assign the sum of the two individual scores. CombANZ combines the two scores of the given document that exists in both lists by computing its average while CombMNZ sums the two scores and multiplies them by the number of ranked lists where that document exists which in our case is two. For the remaining documents in both ranked lists than exist in only one of them all of the five approaches retain the original score. To determine the most suitable rank aggregation approach from the five mentioned before, we ran experiments using the same patents training set as in [40].

Table 5 shows that the best improvement over the Xue and Croft baseline uses rank-based normalization. Since we deal with only two sets of ranked lists, both the CombSUM and CombMNZ aggregation approaches yield the same performance. Table 2 shows the rank aggregation result when we applied the rank based normalization along with the CombMNZ rank aggregation approach on our test collection.

4. CONCLUSIONS

Approximate nearest-neighbor techniques have been effectively applied in many areas of IR and NLP. We have shown that these methods can be extended to comparing distributions in the probability simplex. Empirical results in searching topic distributions to find similar NIH grants and prior art for patents show that accurate results can be obtained while achieving significant improvements in runtime.

5. ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval, in part by NSF grant #IIS-0910884 , in part by NSF grant #SBE-0965436, and in part by NSF CCF 0953754. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

References

[1] D. R. Alexander Klementiev and K. Small. An unsupervised learning algorithm for rank aggregation. In *ECML*, 2007.

[2] A. Andoni and P. Indyk. *LSH Algorithm and Implementation(E2LSH)*, 2005. `http://www.mit.edu/~andoni/LSH/`.

[3] A. Andoni, M. Datar, N. Immorlica, P. Indyk, and V. Mirrokni. Locality-sensitive hashing using stable distributions.

In G. Shakhnarovich, T. Darrell, and P. Indyk, editors, *Nearest Neighbor Methods in Learning and Vision: Theory and Practice*, pages 61–72. MIT Press, 2005.

[4] D. Andrzejewski and D. Buttler. Latent topic feedback for information retrieval. In *KDD*, pages 600–608, 2011.

[5] S. Arya and D. M. Mount. Approximate nearest neighbor queries in fixed dimensions. In *Proc. ACM-SIAM Sympos. Discrete Algorithms*, pages 271–280, 1993.

[6] J. L. Bentley. Multidimensional binary search trees used for associative searching. *CACM*, 18(9):509–517, Sept. 1975.

[7] D. M. Blei and J. D. Lafferty. A correlated topic model of Science. *AAS*, 1(1):17–35, 2007.

[8] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *JMLR*, 3:993–1022, 2003.

[9] J. Boyd-Graber and P. Resnik. Holistic sentiment analysis across languages: Multilingual supervised latent Dirichlet allocation. In *EMNLP*, 2010.

[10] M. S. Charikar. Similarity estimation techniques from rounding algorithms. In *STOC*, pages 308–388, 2002.

[11] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley, 1991.

[12] W. B. Croft. Combining approaches to information retrieval. In *Advances Information Retrieval: Recent Research from the CIIR*, chapter 1, pages 1–36. Kluwer, 2000.

[13] I. Csiszár. Why least squares and maximum entropy? An axiomatic approach to inference for linear inverse problems. *Annals of Statistics*, 19(4):2032–2066, 1991.

[14] S. Deerwester, S. T. Dumais, T. K. Landauer, O. W. Furnas, and R. A. Harshinan. Indexing by latent semantic analysis. *JASIS*, 41(6):391–407, 1990.

[15] M. Fernández, D. Vallet, and P. Castells. Probabilistic score normalization for rank aggregation. In *ECIR*, pages 553–556, 2006.

[16] J. H. Friedman, J. L. Bentley, and R. A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software*, 3(3):209–226, 1977.

[17] S. Guha, A. McGregor, and S. Venkatasubramanian. Streaming and sublinear approximation of entropy and information distances. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 733–742, 2006.

[18] D. L. W. Hall, D. Jurafsky, and C. D. Manning. Studying the history of ideas using topic models. In *EMNLP*, 2008.

[19] M. Hoffman, D. Blei, and F. Bach. Online learning for latent Dirichlet allocation. In *NIPS*, pages 856–864, 2010.

[20] T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR*, pages 50–57, 1999.

[21] P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *STOC*, 1998.

[22] K. Krstovski and D. A. Smith. A minimally supervised approach for detecting and ranking document translation pairs. In *Proc. Workshop on Statistical MT*, pages 207–216, 2011.

[23] K. Krstovski and D. A. Smith. Online polylingual topic models for fast document translation detection. In *Proc. Workshop on Statistical MT*, 2013.

[24] J. Langford, L. Li, and T. Zhang. Sparse online learning via truncated gradient. *JMLR*, 10:777–801, June 2009.

[25] J. Lin. Divergence measures based on Shannon entropy. *IEEE Trans. Information Theory*, 37(1):145–151, 1991.

[26] M. Lupu, J. Huang, J. Zhu, and J. Tait. TREC-CHEM: Large scale chemical information retrieval evaluation at TREC. *SIGIR Forum*, 43:63–70, 12 2009.

[27] D. Mimno, H. Wallach, J. Naradowsky, D. A. Smith, and A. McCallum. Polylingual topic models. In *EMNLP*, pages 880–889, 2009.

[28] D. M. Mount and S. Arya. *ANN: A Library for Approximate Nearest Neighbor Searching*, 2010. `http://www.cs.umd.edu/~mount/ANN/`.

[29] U. S. Patent and T. Office. Patent full-text databases, January 2012. `http://patft.uspto.gov`.

[30] S. Petrovic, M. Osborne, and V. Lavrenko. Streaming first story detection with application to Twitter. In *NAACL*, 2010.

[31] J. Platt, K. Toutanova, and W. tau Yih. Translingual document representations from discriminative projections. In *EMNLP*, pages 251–261, 2010.

[32] C. R. Rao. Diversity: Its measurement, decomposition, apportionment and analysis. *Sankhyā: The Indian Journal of Statistics*, 44(A1):1–22, 1982.

[33] D. Ravichandran, P. Pantel, and E. Hovy. Randomized algorithms and NLP: Using locality sensitive hash functions for high speed noun clustering. In *ACL*, 2005.

[34] J. A. Shaw and E. A. Fox. Combination of multiple searches. In *TREC-2*, pages 243–252, 1994.

[35] J. Tait, C. Harris, and M. Lupu, editors. *PaIR '10: Proceedings of the 3rd international workshop on Patent information retrieval*, 2010.

[36] E. Talley, D. Newman, D. Mimno, B. Herr, H. Wallach, G. Burns, M. Leenders, and A. McCallum. Database of NIH grants using machine-learned categories and graphical clustering. *Nature Methods*, 8:443–444, 2011.

[37] F. Topsøe. Some inequalities for information divergence and related measures of discrimination. *IEEE Trans. Information Theory*, 44(4):1602–1609, 2000.

[38] H. M. Wallach, D. Mimno, and A. McCallum. Rethinking LDA: Why priors matter. In *NIPS*, 2009.

[39] X. Wei and W. B. Croft. LDA-based document models for ad-hoc retrieval. In *SIGIR*, pages 178–185, 2006.

[40] X. Xue and W. B. Croft. Automatic query generation for patent search. In *CIKM*, 2009.

[41] X. Yi and J. Allan. Evaluating topic models for information retrieval. In *CIKM*, pages 1431–1432, 2008.