Asymmetric Distance Estimation with Sketches for Similarity Search in High-Dimensional Spaces

Wei Dong
wdong@cs.princeton.edu

Moses Charikar
moses@cs.princeton.edu

Kai Li
li@cs.princeton.edu

Department of Computer Science, Princeton University
35 Olden Street, Princeton, NJ 08540, USA

ABSTRACT

Efficient similarity search in high-dimensional spaces is important to content-based retrieval systems. Recent studies have shown that sketches can effectively approximate L_1 distance in high-dimensional spaces, and that filtering with sketches can speed up similarity search by an order of magnitude. It is a challenge to further reduce the size of sketches, which are already compact, without compromising accuracy of distance estimation.

This paper presents an efficient sketch algorithm for similarity search with L_2 distances and a novel asymmetric distance estimation technique. Our new asymmetric estimator takes advantage of the original feature vector of the query to boost the distance estimation accuracy. We also apply this asymmetric method to existing sketches for cosine similarity and L_1 distance. Evaluations with datasets extracted from images and telephone records show that our L_2 sketch outperforms existing methods, and the asymmetric estimators consistently improve the accuracy of different sketch methods. To achieve the same search quality, asymmetric estimators can reduce the sketch size by 10% to 40%.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms, Performance

Keywords

similarity search, sketch, asymmetric distance estimation

1. INTRODUCTION

In this paper, we consider similarity search in high-dimensional spaces, a central problem in content-based retrieval systems: given a collection of data objects represented by high-dimensional feature vectors, the objective is to preprocess them so that we can quickly find data objects similar to queries issued at run time. A query is also represented by a high-dimensional feature vector and the problem is to find the k nearest neighbors to the query point.

Sketch construction is an effective way to approximate feature vectors for similarity search. Sketches are compact bit vectors that can be used instead of the original feature vectors to estimate distances. In a search system, sketches are scanned upon a query to quickly generate a small set of candidates which can be ranked later with original feature vectors to obtain the search result. Such a process is often called filtering. As reported by previous work [10, 11, 15], sketches are typically an order of magnitude smaller than their feature vectors, and can significantly improve the space requirement and search speed. A key challenge for sketch construction is to achieve a high ratio of distance estimation accuracy to sketch size.

The traditional method of using sketches to estimate distances, like L_1 distance [10] and cosine similarity [2], is to construct a bit vector for each feature vector where each bit is determined by the position of the feature vector with respect to a random hyperplane. The Hamming distance of such sketches will then be used to approximate the original distance measure. Since computing Hamming distance (counting the number of bits that are different) is simple, the filtering process can be an order of magnitude faster than scanning the original feature vectors. However, the distance estimation with such sketches is a crude approximation; accuracy can be low when sketches are very compact. A challenging question is whether the accuracy of distance estimation can be substantially improved using the same sketches.

Distance estimation using sketches is typically *symmetric* in that two sketches are compared to produce an estimate of the distance. In this paper, we propose a novel *asymmetric* approach to estimate the distance between the feature vector of a query and the sketch of a data object. As the distance estimation is done in two different spaces, we call such algorithms *asymmetric estimators*. Asymmetric estimators do not impose additional space requirements because they use the same sketches for all data points and there is only one query point. These methods can achieve higher accuracy because they take advantage of the position information of the query point in the feature vector space. Another way to look at the effect of the proposed approach is that it allows the sketches to be more compact to achieve the same accuracy. Asymmetric estimators achieve these improvements at the cost of more computation than computing Hamming distance (by a constant factor). In other words, they provide

SIGIR'08, July 20–24, 2008, Singapore.

a way to trade CPU time for accuracy or sketch compaction. However, computation is less of an issue today given the expanding gaps between CPU processing power, memory size and disk bandwidth.

As a specific example, our paper first presents a new L_2 sketch construction and its asymmetric estimator. Our new method has an interesting property particularly desirable for similarity search: it is only sensitive to a small distance range covering most k-nearest neighbors. We then provide general guidelines on designing asymmetric estimators and show how to design asymmetric estimators for two existing sketch constructions for L_1 distance [10] and cosine similarity [7, 2].

We evaluate all three sketch constructions with symmetric and asymmetric estimators using real-life datasets extracted from images and telephone records, and demonstrate the significant space saving (10% to 40%) of the asymmetric estimators, as well as the superiority of our new L_2 sketch over the existing methods.

2. RELATED WORK

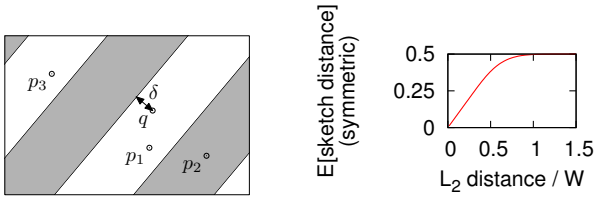
Similarity search in high-dimensional spaces is a long-studied problem so far without a general solution. It is known that when dimensionality is high, existing tree-based index structures degenerate to brute-force scan [16]. The recently developed methods, like VA-file [16] and LSH [6], usually involve scanning a certain portion of the whole dataset. Filtering with sketches can potentially be used to reduce the candidates to be scanned.

Sketches were originally used to answer aggregate queries over data streams [1], and were recently used as a filter to accelerate similarity search [10, 11, 15]. In existing methods for all these applications, however, the estimating operation is carried out completely in the sketch space. Similarity search, as studied in this paper, is a typical scenario where additional information other than sketches can be exploited to improve estimation accuracy. We believe our idea of asymmetric distance estimation will find other applications as well.

A sketch algorithm for L_1 distance was proposed in [10], and later applied to other datasets [11]. The random hyperplane technique, first proposed in [7] for solving the max-cut problem, and later used in [2] for locality sensitive hashing, implicitly gives a sketch algorithm for cosine similarity, which is essentially equivalent to L_2 distance. However, its performance in similarity search has not been experimentally studied. In this paper, we propose asymmetric estimators for both methods, and experimentally evaluate all these methods as well as our newly proposed sketch for L_2 distance.

An analytical performance model of similarity search using sketches was recently given in [15]. We use the same query-processing algorithm, so the performance model can be easily adapted to work with our method.

Dimension reduction is an active research area with many applications, and various methods exist. [5] is a comprehensive survey of popular methods. Though related in concept, sketches are not merely new dimension reduction methods. They are task-specific (similarity search in our case), and the emphasis is more on reducing the size rather than keeping high precision. PCA and random projection are used in this paper as performance baselines.



(a) Partitioning scheme (b) Sketch vs. L_2 distance

Figure 1: Illustration of L_2 Sketch. (a) The space is partitioned randomly into gray and white stripes, inducing a 0/1 sketch function. (b) The left half of the curve shows a near-linear relationship, meeting the requirement of similarity search.

3. SKETCH AND ASYMMETRIC ESTIMATOR FOR L_2 DISTANCE

In this section we present our new sketch algorithm for L_2 distance, which we call L_2 Sketch, and show how to exploit the raw feature vectors for asymmetric distance estimation.

3.1 L_2 Sketch Construction

The goal of sketches for similarity search is different from that for simple distance estimation. First, we only need to know the distance relationships (smaller or larger), rather than exact values. Second, we only need high accuracy for small distances instead of uniform accuracy/inaccuracy across the whole range of distances. These two relaxations potentially allow data reduction more aggressive than traditional dimension reduction methods.

We achieve both requirements by a striping technique. Figure 1(a) gives an illustration of the scheme in R^2 space, and the idea is exactly the same for high-dimensional spaces. We randomly partition the space into interleaving white and gray stripes of equal width, and use stripe colors to determine the sketch values of the points: gray for 0 and white for 1. Points closer to each other are more likely to fall into the same stripe, and subsequently, the XOR of their sketch values has a smaller expectation. We do the random striping multiple times to produce a sequence of bits for each point, and use the Hamming distance of the bit vectors as a proximity estimator. Specifically,

L_2 SKETCH 1. *For a point $p \in R^n$, its L_2 sketch is a bit vector $\sigma(p) \in \{0,1\}^m$, with each bit $\sigma_i(p)$ produced by*

$$\sigma_i(p) = \lfloor h_i(p) \rfloor \mod 2,$$
$$h_i(p) = \frac{A_i \cdot p + b_i}{W}, \qquad \forall i = 1, 2, ..., m$$

where $A_i \in R^n$ is a random vector with each dimension sampled independently from the standard Gaussian distribution $N(0,1)$, and $b_i \in R$ is sampled from the uniform distribution $U[0,W)$. W is called the window size. *For two points $p, q \in R^n$, their sketch distance is defined as*

$$d_\sigma(p,q) = \frac{1}{m}\sum_{i=1}^{m} \sigma_i(p) \oplus \sigma_i(q) = \frac{1}{m} d_H[\sigma(p), \sigma(q)].$$

where d_H is Hamming distance.

The above algorithm specifies how to produce a random partitioning: A_i determines the direction of the stripes, $\|A_i\|_2$

and W together determine the width of the stripes, and the random shift b_i determines the 'phase'. Figure 1(b) shows the relationship between the expectation of symmetric sketch distance and L_2 distance (see Section 3.3 for formulas). The monotonicity of the curve satisfies our ranking purpose. For small L_2 distances, we can see a near-linear relationship, and when L_2 distance grows large, sketch distance quickly converges to the limit 0.5. The window size W determines the sensitive distance range.

3.2 Asymmetric Estimator for L_2 Sketch

Figure 1(a) explains how the original query point is useful for asymmetric distance estimation. Assume p_1, p_2 and p_3 are data points, with only sketches available, and q is the query point, with both sketch and feature vector available. We know the precise location of q, but only know that p_1 and p_3 are in white stripes and p_2 is in a gray stripe. Since p_2 and q are in stripes of different colors, the distance between p_2 and q is lower-bounded by the minimum distance from q to a stripe boundary. This lower bound (marked δ in the figure) replaces the Hamming distance 1 in asymmetric distance estimation. On the other hand, p_1 and p_3 are in stripes of the same color as q, and the value 0 is used. The asymmetric distance estimator is defined below.

L_2 SKETCH 2. *Under the setting of L_2 Sketch 1, for data point p and query point q, both in R^n, the asymmetric sketch distance is defined as*

$$d^*_\sigma(p,q) = \frac{1}{m}\sum_{i=0}^{m} \delta_i(q)[\sigma_i(p) \oplus \sigma_i(q)]$$

where $\delta_i(q) = \min\{\lceil h_i(q)\rceil - h_i(q),\, h_i(q) - \lfloor h_i(q)\rfloor\}$.

The asymmetric sketch distance is a weighted Hamming distance of the sketches. The weight $\delta_i(q)$ is the distance from q to the closest stripe boundary.

3.3 Statistical Analysis

The following lemma gives the relationship between symmetric and asymmetric sketch distances and original L_2 distance.

LEMMA 1. *Under the setting of L_2 Sketch, for any $p, q \in R^n$, let $d = \|p-q\|_2$ be the L_2 distance, $d_\sigma = d_\sigma(p,q)$ be the symmetric sketch distance and $d^*_\sigma = d^*_\sigma(p,q)$ be the asymmetric sketch distance, then for any $k \in R$,*

$$E[d_\sigma] = f_0(d/W) \qquad E[d^*_\sigma] = f_1(d/W)$$

where

$$f_k(t) = \int_0^1\int_0^1 \frac{\min\{x,1-x\}^k}{t} \sum_{j\in\mathbb{Z}} \phi\left[\frac{2j+x+y}{t}\right] dxdy$$

and $\phi(\cdot)$ is the probability density function of the standard Gaussian distribution $N(0,1)$.

PROOF. The m bits in the sketch are identically distributed, so we only consider the case when $m = 1$ and omit subscript i as in A_i, b_i and h_i. For $m \neq 1$, the expectation remains the same, and variance is scaled by $1/m$.

We shift the real line by $\lfloor h(p)\rfloor$, so that $h(p) = y \in [0, W)$, following uniform distribution. Let $j \in \mathbb{Z}$ be an arbitrary integer. When $h(q) \in [2jW, (2j+1)W)$, $d_\sigma = d^*_\sigma = 0$. Thus only the case when $h(q) \in [(2j+1)W, (2j+2)W)$ is interesting. In this case, we have have $d_\sigma = 1$ and $d^*_\sigma = \min\{x, W-x\}$, where $x \in [0, W)$ is the distance from $h(q)$ to the left window boundary $(2j+1)W$. The probability for this to happen is

$$\begin{aligned}
&\sum_{j\in\mathbb{Z}} \Pr[h(p) = y \wedge h(q) = (2j+1)W + x] \\
=&\sum_{j\in\mathbb{Z}} \frac{1}{W}\Pr[h(q) - h(p) = (2j+1)W + x - y] \\
=&\sum_{j\in\mathbb{Z}} \frac{1}{W}\Pr[A\cdot(q-p) = (2j+1)W + x - y] \\
=&\sum_{j\in\mathbb{Z}} \frac{1}{Wd}\phi\left[\frac{(2j+1)W + x - y}{d}\right]
\end{aligned}$$

The last step is due to the 2-stability of Gaussian distribution [3], which states that for any $p \in R^n$, $A \cdot p \sim \|p\|_2 \times N(0,1)$.

The expectations of d_σ and d^*_σ are obtained by averaging the corresponding values weighted by the above probability across all $x, y \in [0, W)$. □

The variances of the estimators can be obtained in a similar way, and we do not show them here due to the page limit. Instead, we use experimental evaluation in Section 5 to compare the performance of these estimators.

Choosing a proper window size W is a practical issue. For k-nearest neighbor search, twice the typical distance of the kth nearest neighbor is a good starting point for tuning.

4. ASYMMETRIC ESTIMATORS FOR OTHER SKETCH ALGORITHMS

In this section we describe a general framework for designing asymmetric estimators and use this to devise asymmetric estimators for two existing sketch methods for *cosine* similarity and L_1 distance.

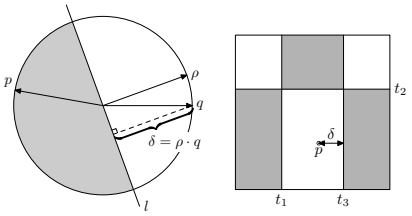
4.1 The Abstract Sketch Algorithm and Asymmetric Estimator

The asymmetric estimator proposed in the previous section can be generalized into the following framework

First, a sketch algorithm σ for a metric space $\langle X, d\rangle$ specifies a family $\sigma = \{\sigma_i \mid i \in I\}$ of bipartitions of the space, mapping an arbitrary point $p \in X$ into a bit sequence $\sigma(p) = \langle\sigma_i(p)\rangle_{i\in I}$. The sketch distance between two points $p, q \in X$ is the expectation of XOR of the corresponding bits in sketches: $d_\sigma(p,q) = E_{i\in I}[\sigma_i(p) \oplus \sigma_i(q)]$. A well-designed sketch algorithm should establish a predictable relationship between the sketch distance d_σ and original distance d. A closely related concept is locality sensitive hashing (LSH) [8]. Actually, a bipartition σ_i is a 0/1 valued LSH function.

Second, an asymmetric distance estimator specifies a distance function d^* (not necessarily the original d) between a point and a partition. In the case of L_2 Sketch, d^* is the L_2 distance measure under a random projection. If we denote by $[0]_{\sigma_i}$and $[1]_{\sigma_i}$ the two parts of σ_i, then the asymmetric sketch distance is simply the average distance from query point q to the part that the data point p is in: $d^*_\sigma(p,q) = E_{i\in I}\{d^*([\sigma_i(p)]_{\sigma_i}, q)\}$. Because d^* gives a refined lower bound of the crude XOR, the asymmetric estimator potentially provides higher accuracy.

Given a sketch algorithm, the challenge then lies in finding a meaningful asymmetric distance function d^*. Though the sketch algorithm and the original distance measure usually



(a) Cosine Sketch (b) L_1 Sketch

Figure 2: The space partitioning schemes of Cosine Sketch and L_1 Sketch, both based on random hyperplane method.

give useful clues, such an estimator is not always obvious and does not always exist. This is one limitation of the asymmetric distance estimation method. It is possible that an asymmetric estimator results in a distance measure different from the original one, but is still helpful in improving the search quality. We later discuss such an estimator for the L_1 case. It is also possible that more than one asymmetric estimator might exist for a single sketch algorithm, each having its own advantage.

Another limitation of the asymmetric method is the accuracy improvement. The error in distance estimation comes from uncertainty in the positions of the data point and the query point. Even if the asymmetric method fully eliminates the error introduced by the query point, it eliminates only half the source of error. In practice (mainly due to the randomization in sketch algorithms), this 50% upper bound is usually far from tight, even though the asymmetric estimators do make a significant difference in many cases.

In the following two subsections, we restate two existing sketch algorithms in our framework, and devise asymmetric estimators for them. We call these algorithms Cosine Sketch and L_1 Sketch based on the distance measure they approximate.

4.2 Cosine Sketch

Cosine similarity is not a strict distance measure, but is still important to many applications like text document retrieval. For any $p, q \in R^n$, the cosine similarity is defined by $d_{cos}(p,q) = \frac{p \cdot q}{\|p\|_2 \|q\|_2}$, and is related to L_2 distance by

$$[d_{L_2}(p,q)]^2 = \|p\|_2^2 + \|q\|_2^2 - 2\, d_{cos}(p,q)\, \|p\|_2\, \|q\|_2.$$

Thus, L_2 distance and cosine similarity have the same ranking effect for normalized datasets, and for unnormalized datasets, given the 2-norms of the points, a sketch algorithm for cosine similarity automatically induces one for L_2 distance.

The following sketch algorithm for cosine distance is implicitly given in [7, 2]. For simplicity of presentation, we assume all the points are normalized to unit vectors.

Cosine Sketch 1. *For a point $p \in S^{n-1} = \{r \in R^n \mid \|r\|_2 = 1\}$, its cosine sketch is a bit vector $\sigma(p) \in \{0,1\}^m$, with each bit $\sigma_i(p)$ produced by*

$$\sigma_i(p) = \begin{cases} 0 & \text{if } \rho_i \cdot p < 0 \\ 1 & \text{if } \rho_i \cdot p \geq 0 \end{cases} \qquad \forall i = 1, 2, \ldots, m$$

where ρ_i for each i is sampled uniformly at random from the unit hypersphere S^{n-1}. The symmetric sketch distance between $p, q \in S^{n-1}$ is defined as $d_\sigma(p,q) = \frac{1}{m} d_H[\sigma(p), \sigma(q)]$.

The idea of the random hyperplane method is illustrated in Figure 2(a). The random vector $\rho \in S^{n-1}$ determines a hyperplane l (the orthogonal complement of ρ) which bipartitions the sphere. A bit in the sketch records whether or not the point falls on the same side of the hyperplane as ρ. To design an asymmetric distance estimator, we need to assess the distance from the query point q to its neighboring half space (the gray one). We find the distance from q to the hyperplane l, indicated by δ in the figure, a natural choice. Simple enough, δ is exactly $|\rho \cdot q|$. Thus, we have the following asymmetric estimator.

Cosine Sketch 2. *Under the setting of Cosine Sketch 1, for data point p and query point q, both in S^{n-1}, the asymmetric sketch distance is defined as*

$$\begin{aligned} d^*_\sigma(p,q) &= \frac{1}{m} \sum_{i=0}^{m} |\rho_i \cdot q| \times [\sigma_i(p) \oplus \sigma_i(q)] \\ &= \frac{1}{m} \sum_{i=0}^{m} -\mathrm{sgn}(\rho_i \cdot p) \times (\rho_i \cdot q) \times [\sigma_i(p) \oplus \sigma_i(q)] \end{aligned}$$

The following lemma gives the relationship between symmetric and asymmetric sketch distances and the original cosine similarity.

Lemma 2. *Under the setting of Cosine Sketch, for any $p, q \in S^{n-1}$, let $\theta = \widehat{pq}$ be the angle between p and q, $d = \cos(\theta)$ be the cosine similarity, $d_\sigma = d_\sigma(p,q)$ be the symmetric sketch distance and $d^*_\sigma = d^*_\sigma(p,q)$ be the asymmetric sketch distance, then the following relations hold:*

$$E[d_\sigma] = \frac{\theta}{\pi} \tag{a}$$

$$E[\cos(\pi d_\sigma)] = \sum_{k=0}^{m} \binom{m}{k} \frac{\theta^k (\pi - \theta)^{m-k}}{\pi^m} \cos\left(\frac{\pi}{m} k\right) \tag{b}$$

$$E[d^*_\sigma] = \frac{B(\frac{n}{2}, \frac{1}{2})}{2\pi}(1 - d) \tag{c}$$

$$var[d^*_\sigma] = \frac{1}{m}\{\frac{\theta - \frac{1}{2}\sin 2\theta}{n\pi} - E[d^*_\sigma]^2\} \tag{d}$$

where $B(\cdot,\cdot)$ is the Beta function.

Proof. (a) See [7, 2].

(b) Given the angle θ between p and q, the probability that the sketches have different values of the ith bit is θ/π. Thus, the probability that the sketches have Hamming distance k is given by the binomial distribution $B(m, \theta/\pi)$. The expectation of $\cos \pi d_\sigma$ can be obtained by taking the average for k from 0 to m.

(c) By linearity of expectation, we only need to consider the case $m = 1$, and we drop the subscript i. We use hyperspherical coordinate system, where each point is represented with one radial coordinate r and $n-1$ angular coordinates $\phi_1, \ldots, \phi_{n-1}$. Because the radius is always 1 in S^{n-1}, we simply omit this coordinate. Moreover, for particular p and q, we rotate the coordinate system so that

$$\begin{aligned} &\phi_i(p) = \phi_i(q) = \frac{\pi}{2} \quad \forall i = 1, \ldots, n-2 \\ &\phi_{D-1}(p) = 0, \quad \phi_{D-1}(q) = \theta. \end{aligned}$$

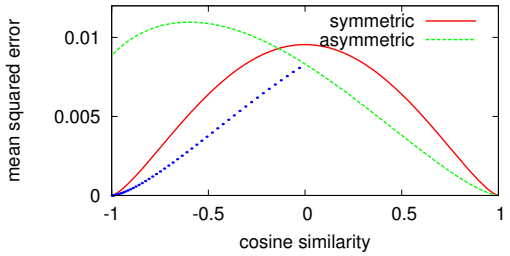


Figure 3: Mean squared error of Cosine Sketch estimators vs. cosine similarity ($n = 128, m = 256$). The asymmetric curve has its left half higher than the symmetric one. We use a trick (described in Section 4.2) to lower it down to the dotted curve, which mirrors the right half of the asymmetric curve.

The corresponding Cartesian coordinates are $p = \langle 0, \ldots, 1, 0\rangle$ and $q = \langle 0, \ldots, \cos(\theta), \sin(\theta)\rangle$. Let the hyper-spherical coordinates of the random vector $\rho \in S^{n-1}$ be $\langle \phi_1, \ldots, \phi_{D-1}\rangle$. We thus have

$$\rho \cdot q \;=\; \sin(\phi_1) \cdots \sin(\phi_{n-2}) \cos(\phi_{n-1} - \theta)$$

Only the hyperplanes that separate p and q make a non-zero contribution to the asymmetric distance. These hyperplanes correspond to $\rho \in \Omega_1 \cup \Omega_2$, where

$$\Omega_1 = \{\rho \,|\, \rho \cdot p < 0, \rho \cdot q > 0\} = \{\rho \,|\, \frac{1}{2}\pi < \phi_{n-1}(\rho) < \frac{1}{2}\pi + \theta\},$$

$$\Omega_2 = \{\rho \,|\, \rho \cdot p > 0, \rho \cdot q < 0\} = \{\rho \,|\, \frac{3}{2}\pi < \phi_{n-1}(\rho) < \frac{3}{2}\pi + \theta)\}.$$

Thus

$$E[d^*_\sigma(p,q)] = \frac{\int_{\Omega_1} \rho \cdot q \, d\rho + \int_{\Omega_2} -\rho \cdot q \, d\rho}{\int_{S^{n-1}} d\rho} = \frac{B(\frac{n}{2}, \frac{1}{2})}{2\pi}(1 - \cos\theta)$$

(d) Similar to (c). □

The symmetric estimator actually estimates the angle θ instead of the cosine similarity. In practice we use $\cos(\pi d_\sigma)$ as a biased estimator of cosine similarity.

Figure 3 shows the mean squared error of both symmetric and asymmetric estimators when $n = 128$ and $m = 256$ based on Lemma 2. We can see that the asymmetric estimator works well when cosine similarity is close to 1, but degrades badly when smaller than 0. For similarity search under cosine similarity, this works well, for only similarity close to 1 is interesting. For estimating L_2 distance, however, we actually want low error across the whole range. We use the following trick to circumvent this flaw.

For arbitrary $p, q \in S^{n-1}$, we have $d_{cos}(-p, q) = -d_{cos}(p, q)$; and at any time, either $d_{cos}(p, q)$ or $d_{cos}(-p, q)$ is non-negative, and is estimated more accurately. Furthermore, the sketch $\sigma(-p)$ of $-p$ is exactly the binary complement of that of p, and is thus available at query time. As a result, we always compute an estimation for both p and $-p$, and use the better one to produce the final estimation.

4.3 L_1 Sketch

[10] introduced a sketch algorithm for weighted L_1 distance. To simplify the presentation, we only consider the unweighted case here. Our asymmetric estimator can be easily adapted to work with the weighted case. Following is the unweighted version of the L_1 sketch algorithm proposed in [10]. Note that the algorithm works only for limited space, and requires the range of each dimension as input.

L_1 SKETCH 1. *Given a closed space $X = [l_1, u_1] \times \cdots \times [l_n, u_n] \subset R^n$, for a point $p \in X$, its L_1 sketch is a bit vector $\sigma(p) \in \{0, 1\}^m$, each bit $\sigma_i(p)$ produced by*

$$\sigma_i(p) = \sigma_{i,1}(p) \oplus \sigma_{i,2}(p) \oplus \cdots \oplus \sigma_{i,b}(p)$$

$$\sigma_{i,j}(p) = \begin{cases} 0 & \text{if} \quad p_{s_{i,j}} < t_{i,j} \\ 1 & \text{else} \end{cases}$$

$$\forall i = 1, 2, ..., m, \quad j = 1, 2, ..., b.$$

Each bit in the sketch is the XOR of b independently generated bits, and b is a parameter specified by the user. For each $\langle i, j\rangle$pair, the index $s_{i,j}$ and threshold $t_{i,j}$ are generated in the following two steps:

1. *Pick $s_{i,j}$ such that $\Pr[s_{i,j} = k] \propto (u_k - l_k), k = 1, 2, ...n;$*
2. *Pick $t_{i,j}$ uniformly at random from $[l_{s_{i,j}}, u_{s_{i,j}}]$.*

The symmetric sketch distance between $p, q \in S^{n-1}$ is defined as $d_\sigma(p, q) = \frac{1}{m} d_H[\sigma(p), \sigma(q)]$.

It can be proved that if $b = 1$, the expectation of sketch distance is proportional to L_1 distance. XORing b bits has the effect of increasing the sensitivity to small distance range, similar to what is shown in Figure 1(b). Though the theoretical properties of this XOR idea are interesting, no intuitive explanation is given in [10, 11, 15]. Here we give a geometrical view of the XOR method, which is rather straightforward.

Figure 2(b) illustrates a bipartition of a 2D space according to the sketch algorithm, with $b = 3$. The three random dimension-threshold pairs partition the space into six rectangular regions, and the XOR scheme induces a gray-white coloring of the regions such that touching regions are always colored differently. Note that regions of the same color, as in L_2 Sketch, are considered as one partition.

Given the geometrical view of the bipartition scheme, one asymmetric estimator is straightforward. We take the distance from the query point q to the closest boundary of the region, which is t_3 in the figure, as the asymmetric distance function d^*. This distance is indicated in the figure by δ. Following is the algorithm of the asymmetric estimator.

L_1 SKETCH 2. *Under the setting of L_1 Sketch 1, for data point p and query point q, both in X, the asymmetric sketch distance is defined as*

$$d^*_\sigma(p, q) = \frac{1}{m} \sum_{i=0}^{m} \delta_i(q)[\sigma_i(p) \oplus \sigma_i(q)]$$

where $\delta_i(q) = \min\{|q_{s_{i,j}} - t_{i,j}| \,|\, j = 1, 2, \ldots, b\}$.

The drawback of the above asymmetric estimator is that it is not a proper estimator of L_1 distance. It can be shown that when $b = 1$, the asymmetric distance is in fact proportional to the L_2 distance. Nevertheless, as the experimental results in Section 5 show, it does improve search quality in practice.

Dataset	# Points	Dimension	Total size
image	4,459,549	128	2.28GB
audio	2,663,040	192	2.04GB

Table 1: Summary of datasets.

5. EVALUATION

In this section, we conduct experiments with two real-life datasets to study the performance of the methods described in the previous two sections. We first measure the accuracy of different methods to demonstrate the advantage of sketch methods over traditional dimension reduction methods and the improvement of asymmetric estimators over the basic symmetric ones. We then plug our methods into a similarity search algorithm and evaluate the space reduction achievable by the new asymmetric estimators.

We have shown in Section 4.2 that L_2 distance and cosine similarity are essentially equivalent. Due to the page limit, we evaluate both L_2 Sketch and cosine sketch under L_2 measure by using the Cosine Sketch to estimate L_2 distance.

5.1 Datasets

We use two high-dimensional datasets, *i.e.* images and audio, in our evaluation. These datasets are reasonably large and reflect real-life usage. Table 1 is a summary of these datasets.

Image: The image dataset is extracted from the Caltech 101 [4] object recognition benchmark. This dataset contains 101 object categories and a background clutter category, with 9144 images in total. We convert the images into grayscale PGM format and use the SIFT [9, 12] algorithm with default parameters to extract feature vectors, obtaining nearly 4.5 million 128-dimensional feature vectors. We treat feature vectors extracted from the same image as independent objects, as we only consider similarity search on the feature vector level.

Audio: The audio dataset is the LDC-SWITCHBOARD-1 [13] collection, which is a collection of about 2,400 two-side telephone conversations among 543 speakers from all areas of the United States. The conversations are segmented into individual words based on human transcription. We use the Marsyas library [14] to extract feature vectors from the word segments. For each word segment, we take a 512-sample sliding window with variable stride to obtain 32 windows, and from each window extract the first six MFCC parameters, resulting in a 192-dimensional feature vector. About 2.5 million feature vectors are extracted.

5.2 More on the Methods

For L_2 distance, we compare the following six methods.

- L_2 Sketch with symmetric and asymmetric estimators. We use window size $W = 900$ for the image dataset, and $W = 23$ for the audio dataset. These parameters are tuned to be optimal for 100-nearest neighbors search. We build lookup tables based on Lemma 1 to map the sketch distances to L_2 distance for explicit L_2 distance estimation. For similarity search, the sketch distances are directly used.

- Cosine Sketch with symmetric and asymmetric estimators. We use Lemma 2 to directly convert the sketch distances to cosine similarity, and further convert cosine similarity to L_2 distance with saved 2-norms.

- PCA (Principal Component Analysis) and RP (Random Projection), also considered as sketches to serve as performance baselines.

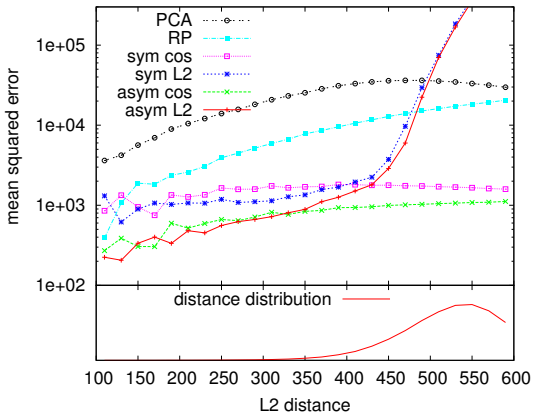


Figure 4: Accuracy of various L_2 distance estimation methods on different distance values of the image dataset. Note that only the small distances are interesting to similarity search.

For L_1 distance, we simply compare the symmetric and asymmetric estimators of L_1 Sketch.

We always allocate sketches in full bytes to simplify implementation. Actually, varying sketch size by less than one byte does not make a practically significant difference in performance. For Cosine Sketch, we need the 2-norms of the vectors to estimate L_2 distance, and count 4 extra bytes into sketch size. For PCA and RP, we use 32-bit floating-point representation, and thus count each dimension as four bytes.

We implement asymmetric sketch distances in the following way. Assume an m-bit sketch length, we pre-calculate for each query point an $m \times 2$ matrix M, $M[i][j]$ being the value to be added if bit-i of the data sketch is j. Therefore, evaluating the asymmetric sketch distance for each data point involves m floating-point additions.

5.3 Evaluation of Distance Estimation

Because our asymmetric estimator of L_1 Sketch does not produce a proper estimation of L_1 distance, we do not consider L_1 distance here. We only evaluate the L_2 distance related methods. Also, we only use the image dataset, which is sufficient to demonstrate the behavior of different methods.

First, we measure the mean squared error of various methods at different distance values. We randomly sample 100,000 pairs of points from the image dataset, create sketches for them, estimate the distance for each pair with sketches, and then compare the estimations with real distances. We then bin the squared error values according to the corresponding real distances and take the average of each bin. For asymmetric estimators, we use the raw feature vector of one arbitrary point from each pair. The results are plotted in Figure 4. We also attach the distribution of real L_2 distance in the bottom of the figure to show the relative importance of different distance values.

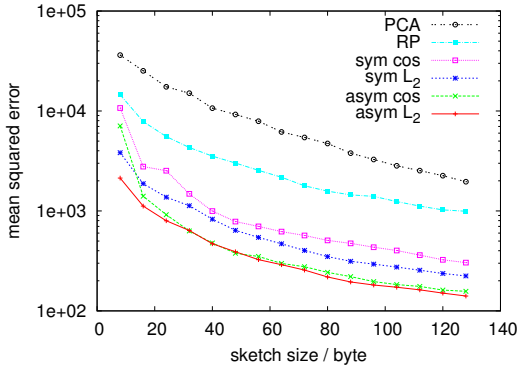


Figure 5: Mean squared error of different methods vs. sketch size, image dataset.

Figure 4 gives a clear view of how different methods behave for different distance values. The L_2 Sketch curves are most interesting, for the errors are low when distance is small, but beyond 400, the errors increase dramatically as distance grows. Note that the turning point is tunable in practice via the window size W. The curves of the other methods are relatively flat. Especially, the Cosine Sketch estimators are pretty consistent across the whole distance range, and are the first choice if one is interested in estimating distances for arbitrary points rather than for the nearest neighbors only.

We then consider the overall accuracy vs. sketch size. For the purpose of nearest neighbor search, it does not make sense to cover the whole distance range from zero to infinite. Instead, we only consider the distance range from 0 to 300, which is around one standard deviation (~ 60) beyond the average distance of the 100th nearest neighbor (~ 250). We sample from the image dataset 1000 pairs of points that are within this distance range, and take the mean squared error of different methods at different sketch sizes. The results are shown in Figure 5.

The curves in Figure 5 can be grouped into three categories, from low to high in accuracy: dimension reduction methods, sketches with symmetric estimators and sketches with asymmetric estimators. The relative performances of these methods are mostly consistent across the whole figure. The more than 50% error reduction of the sketch methods over PCA and RP clearly shows their superiority. The figure also shows that the improvement of an asymmetric estimator (from "sym cos" to "asym cos") is larger than using a better sketch scheme (from "sym cos" to "sym L_2").

In the above two figures, RP consistently out-performs PCA in terms of mean squared error. This is mainly because PCA has a systematic bias that makes it always lower-estimate actual distance. This bias does not affect similarity search and we will see that PCA is actually better at similarity search.

5.4 Evaluation of Similarity Search

With the accuracy improvements of our new methods confirmed, in this subsection we evaluate the methods in the scenario of similarity search, and see how they improve search quality, and equivalently, reduce the space requirement to achieve a fixed quality.

Here is how we create the evaluation benchmark: for each

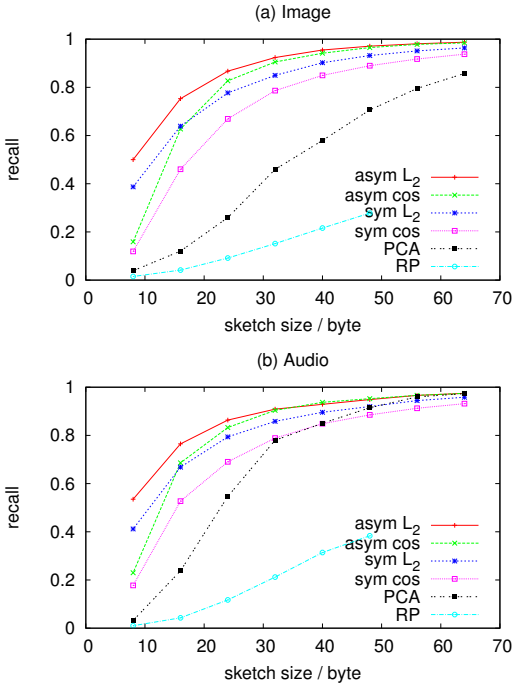


Figure 6: Recall vs. sketch size, L_2 distance.

dataset, we pick 100 points at random as query points, and use the rest of the points as the data points indexed. For each query point, we sequentially scan all the data points for k nearest neighbors under each distance measure concerned, and use these results as the ground truth. We use $k = 100$ in our experiments.

We measure search quality by *recall*, which is the percentage of the true nearest neighbors found by the query algorithm. We do not consider *precision* here, but use the equivalent *filter ratios* t and t' as input parameters to the query algorithm explained below.

We use the same query algorithm as modeled in [15]. The query algorithm with symmetric estimators proceeds in two steps: first, scan the sketches for a candidate set of $t \times k$ points with smallest sketch distances to the query point; second, scan the raw feature vectors of the candidate set, and find the top k within them as the final result. We use $t = 20$ in all our experiments.

Asymmetric estimators take more computation than the symmetric ones, and filtering all the sketches with asymmetric estimators is not always affordable. As a work-around, we extend the query algorithm with an extra filtering step. We first scan the sketches with a symmetric estimator for $t' \times t \times k$ candidate points, and then use an asymmetric estimator to rank them and choose the top $t \times k$ candidates for final re-ranking with raw feature vectors. Our experience shows that $t' = 10$, which we use throughout the following experiments, provides nearly identical recall as using asymmetric estimators to scan all the sketches.

The relationships between t, k and recall are thoroughly studied in [15] and are not our focus in this paper. Instead, we focus on the recall/size performance of different sketch algorithms. Figure 6 shows the experimental results of L_2 distance, and Figure 7 shows those for L_1 distance.

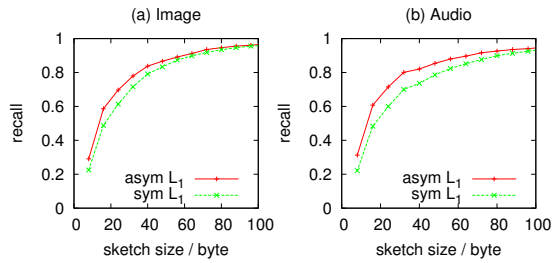


Figure 7: Recall vs. sketch size, L_1 distance.

		Image			Audio		
Recall		0.85	0.9	0.95	0.85	0.90	0.95
L_2	sym	32	40	56	32	42	60
	asym	22	29	39	23	32	48
Reduction		31%	28%	30%	28%	24%	20%
Cos	sym	40	54	73	41	54	73
	asym	26	32	42	25	32	46
Reduction		35%	41%	43%	30%	41%	37%
L_1	sym	51	66	92	64	80	120
	asym	43	59	83	47	64	106
Reduction		16%	11%	10%	27%	20%	12%

Table 2: Space requirement (in bytes) of various methods to achieve specific recalls. Here L_2 sketch and Cosine sketch are evaluated with L_2 distance, and L_1 sketch is evaluated with L_1 distance.

Figure 6 shows that for each sketch method, the asymmetric estimator always out-performs the symmetric one. The improvement is especially significant when the sketch is small. But even at a baseline recall of 0.90, the asymmetric methods still achieve a 0.05 improvement in most cases. The results for L_1 distance, as shown in Figure 7, are not as good; however, we do see a consistent improvement.

The performance of PCA on the audio dataset is pretty interesting, as the recall grows very quickly when the size is smaller than 32, or 8 dimensions, reaching one of the sketch methods at 8 dimensions, and then slows down significantly. This suggests that the intrinsic dimension of the audio dataset might be around 10.

Table 2 shows the minimal sketch sizes of symmetric and asymmetric methods to reach various recall values. For L_2 distance, the asymmetric estimators achieve sketch size reduction from 20% to 43%. For L_1 distance, the reduction is smaller, from 10% to 27%.

6. CONCLUSION

In this paper, we proposed the idea of asymmetric distance estimators to exploit the raw query points, which are not used by traditional methods when estimating distances with sketches. We apply the idea to three sketch algorithms, one of them newly proposed in this paper. Our experimental results confirm the precision improvement of the asymmetric estimators, and show that to achieve the same search quality, the asymmetric estimators can reduce the space requirement by 10% to 40%.

The asymmetric estimators designed in this paper are a proof of concept, and the potential performance limit of asymmetric estimators is an open question. Designing better asymmetric estimators, especially for L_1 distance, is an interesting direction for future research.

Acknowledgments

This work is supported in part by NSF grants EIA-0101247, CCR-0205594, CCR-0237113, CNS-0509447, DMS-0528414 and by research grants from Google, Intel, Microsoft, and Yahoo!. Wei Dong is supported by Gordon Wu Fellowship.

7. REFERENCES

[1] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. In *STOC '96: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 20–29, 1996.

[2] M. S. Charikar. Similarity estimation techniques from rounding algorithms. In *STOC '02: Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 380–388, 2002.

[3] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *SCG '04: Proceedings of the twentieth annual symposium on Computational geometry*, pages 253–262, 2004.

[4] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In *IEEE. CVPR 2004, Workshop on Generative-Model Based Vision*, 2004.

[5] I. Fodor. A survey of dimension reduction techniques.

[6] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In *Proc. of 25th Intl. Conf. on Very Large Data Bases(VLDB)*, pages 518–529, 1999.

[7] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, 1995.

[8] P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *STOC '98: Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613, 1998.

[9] D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. of the International Conference on Computer Vision ICCV, Corfu*, pages 1150–1157, 1999.

[10] Q. Lv, M. Charikar, and K. Li. Image similarity search with compact data structures. In *CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 208–217, 2004.

[11] Q. Lv, W. Josephson, Z. Wang, M. Charikar, and K. Li. Efficient filtering with sketches in the ferret toolkit. In *MIR '06: Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, pages 279–288, 2006.

[12] SIFT demo. `http://www.cs.ubc.ca/~lowe/keypoints/`.

[13] SWITCHBOARD-1 Release 2. `http://www.ldc.upenn.edu/Catalog/docs/switchboard/`.

[14] G. Tzanetakis and P. Cook. *MARSYAS: A Framework for Audio Analysis*. Cambridge University Press, 2000.

[15] Z. Wang, W. Dong, W. Josephson, Q. Lv, M. Charikar, and K. Li. Sizing sketches: a rank-based analysis for similarity search. In *SIGMETRICS '07: Proceedings of the 2007 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 157–168, 2007.

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